BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617
Phone: 949-926-5000
Fax: 949-926-5203

BCM6XXX Switch Additional API

Version 0.5

Date 04/30/2009

BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203

Revision History

Revision	Date	Description
0.1	11/03/08	Initial document release.
0.2	11/10/08	1. Section 2.2.7: Correct the syntax 2. Added the switch buffer management API (section 3)
0.3	11/13/08	3. Added more details of switch buffer management in the Overview of section-3.
0.4	01/16/09	1. Updated section 2.2.7 2. Added sections 2.2.8 and 2.2.9 3. Updated section 3.2 4. Added section-4
0.5	04/30/09	1. Updated sections 3.1.1, 3.1.2 and 3.1.3

BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203

Page intentionally blank

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203

Table of Contents

BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203

List of Tables

BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203

1 Purpose

Document additional switch API supported for BCM6368 and BCM6816 products. Please note that this document is a supplement to the BCM API document (56XX-PG616-R).

2 Port Configuration API

2.1 Overview

The following table lists all port configuration APIs described in this document along with the information of which BCM6XXX devices support these API.

API Name	BCM6816 Support	Remarks
bcm_port_replace_tag_set bcm_port_replace_tag_get	Yes	
bcm_port_replace_tpid_set bcm_port_replace_tpid_get	Yes	
bcm_port_replace_tci_set bcm_port_replace_tci_get	Yes	
bcm_port_replace_vid_set bcm_port_replace_vid_get	Yes	
bcm_port_replace_8021p_set bcm_port_replace_8021p_get	Yes	
bcm_port_replace_cfi_set bcm_port_replace_cfi_get	Yes	
bcm_port_tag_mangle_set bcm_port_tag_mangle_get	Yes	

Table 1: Port Configuration API support matrix

2.2 Port Configuration Functions

2.2.1 Configure Replacement VLAN Tag for Egress Packets with VLAN ID = 0xFFF

bcm_port_replace_tag_set
bcm_port_replace_tag_get

Get or Set the replacement VLAN tag (32-bit tag including TPID).

Syntax

#include <bcm/port.h>
int bcm_port_replace_tag_get(IN int unit, IN bcm_port_t port, OUT uint32_t *vlan_tag);
int bcm_port_replace_tag_set(IN int unit, IN bcm_port_t port, IN uint32_t vlan_tag);

Parameters

unit	BCM device number
port	Device or logical port number.
vlan_tag	The 32-bit VLAN tag

Description
The VLAN tag of a tagged egress packet on a given port with VLAN ID = 0xFFF is replaced with the configured replacement VLAN tag of the given port when requested to do so using bcm_port_tag_actions_set API. These API allow configuring and reading the replacement VLAN tag of the given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.2 Configure Replacement TPID for Egress Packets with VLAN ID = 0xFFF

bcm_port_replace_tpid_set
bcm_port_replace_tpid_get

Get or Set the replacement TPID.

Syntax
#include <bcm/port.h>
int bcm_port_replace_tpid_get(IN int unit, IN bcm_port_t port, OUT uint16_t *tpid);
int bcm_port_replace_tpid_set(IN int unit, IN bcm_port_t port, IN uint16_t tpid);

Parameters

unit	BCM device number
port	Device or logical port number.
tpid	The 16-bit TPID value.

Description
The TPID of a tagged egress packet on a given port with VLAN ID = 0xFFF is replaced with the replacement TPID of the given port when requested to do so using bcm_port_tag_actions_set API. These API allow configuring and reading the replacement TPID of the given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.3 Configure Replacement TCI of VLAN Tag for Egress Packets with VLAN ID = 0xFFF

bcm_port_replace_tci_set
bcm_port_replace_tci_get

Get or Set the replacement TCI (Tag Control Identifier: The 16-bit TCI field of VLAN tag which includes 12-bit VID, 3-bit 802.1p Priority and 1-bit CFI).

Syntax
#include <bcm/port.h>
int bcm_port_replace_tci_get(IN int unit, IN bcm_port_t port, OUT bcm_vlan_t *tci);
int bcm_port_replace_tci_set(IN int unit, IN bcm_port_t port, IN bcm_vlan_t tci);

Parameters

unit	BCM device number
port	Device or logical port number.
vlan_tag	The 16-bit TCI field of VLAN tag.

Description
The 16-bit TCI of a tagged egress packet with VLAN ID = 0xFFF is replaced with the configured replacement TCI of the given port when requested to do so using bcm_port_tag_actions_set API. These API allow configuring and reading the replacement TCI of the given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.4 Configure Replacement VLAN ID for Egress Packets with VLAN ID = 0xFFF

bcm_port_replace_vid_set
bcm_port_replace_vid_get

Get or Set the replacement VLAN ID.

Syntax
#include <bcm/port.h>
int bcm_port_replace_vid_get(IN int unit, IN bcm_port_t port, OUT bcm_vlan_t *vid);
int bcm_port_replace_vid_set(IN int unit, IN bcm_port_t port, IN bcm_vlan_t vid);

Parameters

unit	BCM device number
port	Device or logical port number.

vid VLAN ID.

Description
The VLAN ID of a tagged egress packet on a given port with VLAN ID = 0xFFF is replaced with the replacement VLAN ID of that port when requested to do so using bcm_port_tag_actions_set API. These API allow configuring and reading the replacement VLAN ID of the given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.5 Configure Replacement 802.1p for Egress Packets with VLAN ID = 0xFFF

bcm_port_replace_8021p_set
bcm_port_replace_8021p_get

Get or Set the replacement 802.1p Priority.

Syntax
#include <bcm/port.h>
int bcm_port_replace_8021p_get(IN int unit, IN bcm_port_t port, OUT uint8_t *priority);
int bcm_port_replace_8021p_set(IN int unit, IN bcm_port_t port, IN uint8_t priority);

Parameters

unit	BCM device number
port	Device or logical port number.
priority	802.1p priority.

Description
The 802.1p priority of a tagged egress packet on a given port with VLAN ID = 0xFFF is replaced with the replacement 802.1p priority of that port when requested to do so using bcm_port_tag_actions_set API. These API allow configuring and reading the replacement 802.1p priority of the given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.6 Configure Replacement CFI for Egress Packets with VLAN ID = 0xFFF

bcm_port_replace_cfi_set
bcm_port_replace_cfi_get

Get or Set the replacement CFI (Canonical Format Indicator) bit.

Syntax
#include <bcm/port.h>
int bcm_port_replace_cfi_get(IN int unit, IN bcm_port_t port, OUT uint8_t *cfi);
int bcm_port_replace_cfi_set(IN int unit, IN bcm_port_t port, IN uint8_t cfi);

Parameters

unit	BCM device number
port	Device or logical port number.
cfi	The CFI bit of 802.1Q tag.

Description
The CFI bit in the 802.1Q tag of a tagged egress packet on a given port with VLAN ID = 0xFFF is replaced with the configured replacement CFI bit of that port when requested to do so using bcm_port_tag_actions_set API. These API allow configuring and reading the replacement 802.1Q CFI bit of the given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.7 VLAN Tag Mangling Operations

bcm_port_tag_mangle_set
bcm_port_tag_mangle_get

Get or Set the VLAN tag mangling operations.

Syntax
#include <bcm/port.h>
int bcm_port_tag_mangle_get(IN int unit, IN bcm_port_t port, OUT uint16_t *op_map);
int bcm_port_tag_mangle_set(IN int unit, IN bcm_port_t port, IN uint16_t op_map);

Parameters

unit	BCM device number
port	Device or logical port number.
op_map	The bit map of tag mangling operations given in the below definitions.

#define BCM_PORT_REPLACE_TPID 0x8000 /* Replace TPID */
#define BCM_PORT_REPLACE_VID 0x4000 /* Replace VLAN ID */

```
#define BCM_PORT_REPLACE_CFI      0x2000  /* Replace CFI bit */
#define BCM_PORT_REPLACE_8021P    0x1000  /* Replace 802.1p bits */
```

Description
These API allow configuring and reading the VLAN tag mangling operations for tagged egress packets with VLAN ID = 0xFFF. The various tag mangling operations supported are:

1. Replace the VID
2. Replace 802.1p
3. Replace CFI
4. Replace TPID

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.8 VLAN Tag Mangling Match VID Configuration

bcm_port_tag_mangle_match_vid_set
bcm_port_tag_mangle_match_vid_get

Get or Set the match VID that is used to trigger the VLAN tag mangling operations.

Syntax

```
#include <bcm/port.h>
int bcm_port_tag_mangle_match_vid_get(IN int unit, IN bcm_port_t port, OUT bcm_vlan_t *vid);
int bcm_port_tag_mangle_match_vid_set(IN int unit, IN bcm_port_t port, IN bcm_vlan_t vid);
```

Parameters

unit	BCM device number
port	Device or logical port number.
vid	The match VLAN ID value.

Description
These API allow configuring and retrieving the match VLAN ID that is used to trigger the egress tag mangling operations on a given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

2.2.9 Egress VLAN Tag Stripping

bcm_port_tag_strip_set
bcm_port_tag_strip_get

Enable/disable the egress tag stripping of packets whose VLAN ID matches the match_vid. The get operation retrieves the enable/disable status of the tag stripping on a given port.

Note: The given port has to be a member of the match_vid VLAN for this operation to be successful.

Syntax
#include <bcm/port.h>
int bcm_port_tag_strip_get(IN int unit, IN bcm_port_t port, OUT uint8 *val);
int bcm_port_tag_strip_set(IN int unit, IN bcm_port_t port, IN uint8 val);

Parameters

unit	BCM device number
port	Device or logical port number.
val	Enable/Disable value (Zero means Disable and non-zero means Enable)

Description
These API allow configuring and retrieving the egress tag stripping on a given port for those packets which have the VLAN ID same as the match_vid of that port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Port module is not initialized
BCM_E_XXX	Operation failed

3 Switch Control API: Buffer Management

3.1 Overview

The switch supports buffer management and flow control in order to efficiently use the limited number of packet buffer resources. The switch also provides the flexibility to manage buffers across its queues based on the priority. For this, the switch supports several per-queue and total thresholds per each priority and a global pause/drop control register. The pause/drop control register allows enabling or disabling of TXQ drop, TXQ pause, Total drop, and Total Pause buffer control mechanisms.

3.1.1 TXQ Drop and TXQ Pause Mechanisms

Note: On 6816B0, the TXQ Pause Mechanism outlined below is augmented with the Low Threshold mechanism. See section 3.1.3.

The following thresholds are provided for each priority in order to support the TXQ drop and TXQ pause buffer control mechanisms.

1. Priority Q Hysteresis Threshold
2. Priority Q Pause Threshold
3. Priority Q Drop Threshold

The TXQ Pause and TXQ Drop mechanisms use the above 3 thresholds and the pause capability of ingress port to decide on whether an ingress packet can be buffered or not.

If the ingress port is Pause capable, an ingress packet will be dropped or paused as given below.

1. Pause if TXQ Pause is enabled and the buffers consumed by the egress TX queue is more than the Priority Q Pause Threshold of the priority of ingress packet. The Pause ON message will be sent from the ingress port to its link partner upon this Pause condition.
2. Start dropping if TXQ Drop is enabled and the buffers consumed by the egress TX queue is more than the Priority Q Drop Threshold of the priority of ingress packet. The buffering will resume only when the buffers consumed by the egress TX queue become less than the Priority Q Hysteresis Threshold of the priority of ingress packet. The Pause OFF message will be sent from the ingress port to its link partner upon this Pause condition.

If the ingress port is not Pause capable, the drop decision is made as given below.

1. Start dropping if TXQ Pause is enabled and the buffers consumed by the egress TX queue is more than the Priority Q Pause Threshold of the priority of ingress packet. The buffering will resume only when the buffers consumed by the egress TX queue become less than the Priority Q Hysteresis Threshold of the priority of ingress packet.
2. Drop if TXQ Drop is enabled and the buffers consumed by the egress TX queue is more than the Priority Q Drop Threshold of the priority of ingress packet.

3.1.2 Total Drop and Total Pause Mechanisms

Note: On 6816B0, the Total Pause Mechanism outlined below is augmented with the Low Threshold mechanism. See section 3.1.3.

The following thresholds are provided for each priority in order to support the Total drop and Total Pause buffer control mechanisms.

4. Priority Q Total Hysteresis Threshold
5. Priority Q Total Pause Threshold
6. Priority Q Total Drop Threshold

The Total Pause and Total Drop mechanisms use the above 3 thresholds and the pause capability of ingress port to decide on whether an ingress packet can be buffered or not.

If the ingress port is Pause capable, an ingress packet will be dropped or paused as given below.

3. Pause if Total Pause is enabled and the total buffers consumed is more than the Total Pause Threshold of the priority of ingress packet and the buffers consumed in the egress TXQ is more than the Rx Reserved count (Rx-Base Reserve Register at page 0Ah and offset 52h). The Pause ON message will be sent from the ingress port to its link partner upon this Pause condition.
4. Start dropping (in other words, stop buffering) if Total Drop is enabled and the total buffers consumed is more than the Total Drop Threshold of the priority of ingress packet. The buffering will resume only when the total buffers consumed becomes less than the Total Hysteresis Threshold of the priority of ingress packet. The Pause OFF message will be sent from the ingress port to its link partner upon this Pause condition.

If the ingress port is not Pause capable, the drop decision is made as given below.

3. Start dropping if Total Pause is enabled and the total buffers consumed is more than the Total Pause Threshold of the priority of ingress packet and the buffers consumed in the egress TXQ is more than the Rx Reserved count. The buffering will resume only when the total buffers consumed becomes less than the Total Hysteresis Threshold of the priority of ingress packet.
4. Drop if Total Drop is enabled the total buffers consumed is more than the Total Drop Threshold of the priority of ingress packet.

Note: There are two prefetch buffers per port in the switch. These are taken into account when enforcing the Total Drop/Pause Threshold. As a result, the maximum number of buffers allocated will remain 18 buffers below the configured Total Drop/Pause Threshold.

3.1.3 Per Priority Queue Low Threshold Buffer Control (6816 B0 only)

If the Total Pause Mechanism is enabled, the BCM6816B0 also supports a 'Priority Q Low Drop Threshold' per priority that allows the switch to use a minimum number of buffers per each priority queue, even when the switch is congested (total consumed buffers has exceeded the Total Pause Threshold).
An ingress packet of a given priority will be buffered in a queue, even if the total number of consumed buffers has exceeded Total Pause Threshold of that priority, on the following conditions:

- The queue has not yet consumed the Low Threshold number of buffers (of that priority).
- There are available (unallocated) buffers.

Note that a queue still might not be able to buffer a packet if there are no available buffers.

3.1.4 Buffer Management API List

The following table lists all switch buffer management APIs described in this document along with the information of which BCM6XXX devices support these API. These buffer management APIs allow configuring and retrieving the various buffer management thresholds and enabling and disabling various buffer control mechanisms.

API Name	BCM6816 Support	Remarks
bcm_buffer_threshold_set	Yes	

bcm_buffer_threshold_get		
bcm_buffer_control_set bcm_buffer_control_get	Yes	

Table 2: Switch buffer management API

3.2 Switch Buffer Management Functions

3.2.1 Manage the switch buffer thresholds

bcm_switch_control_priority_get
bcm_switch_control_priority_set

Get or Set

a) Switch buffer per queue Hysteresis and Drop thresholds for each priority level.
b) Global Hysteresis and Drop thresholds for each priority level.

Syntax

#include <bcm/switch.h>
int **bcm_switch_control_priority_get** (IN int unit, IN bcm_cos_t priority, IN bcm_switch_control_t type, OUT int *value);
int **bcm_switch_control_priority_set** (IN int unit, IN bcm_cos_t priority, IN bcm_switch_control_t type, IN int value);

Parameters

unit BCM device number
priority Specifies the priority Q whose threshold is being managed
type Specifies the threshold being configured or retrieved. The following enums in bcm_switch_control_t are used for specifying type.

```
bcmSwitchTotalDropThreshold,       /* Configure Total Drop Threshold */
bcmSwitchTotalPauseThreshold,      /* Configure Total Pause Threshold */
bcmSwitchTotalHysteresisThreshold,/* Configure Total Hysteresis Threshold */
bcmSwitchTxQHiDropThreshold,       /* Configure TxQ Hi Drop Threshold */
bcmSwitchTxQHiPauseThreshold,      /* Configure TxQ Hi Pause Threshold */
bcmSwitchTxQHiHysteresisThreshold,/* Configure TxQ Hi Hysteresis Threshold */
bcmSwitchTxQLowDropThreshold,  /* Configure TxQ LOW Drop Threshold */
```

value The threshold value. The maximum allowed value is 0x400.

Description

These API allow configuring and retrieving the per-queue and global switch buffer thresholds for a given priority level.

3.2.2 Buffer Control

bcm_buffer_control_set
bcm_buffer_control_get

Enable or Disable

a) Tx Queue Drop mechanism
b) Tx Queue Pause mechanism
c) Total Drop mechanism
d) Total Pause mechanism

Syntax

```
#include <bcm/switch.h>
int bcm_switch_control_get(IN int unit, IN bcm_switch_control_t type, OUT int *buf_ctrl_map);
int bcm_switch_control_set(IN int unit, IN bcm_switch_control_t type, IN int buf_ctrl_map);
```

Parameters

unit BCM device number
type Specifies which switch control operation is requested. The following enums in bcm_switch_control_t are used for specifying this type.

```
    bcmSwitchBufferControl,                 /* Enable/Disable Total/TxQ Pause/Drop */
```

buf_ctrl_map The bit map of various buffering controls

```
    #define BCM_SWITCH_TXQ_PAUSE        0x1   /* Enable/Disable TXQ DROP. */
    #define BCM_SWITCH_TXQ_DROP         0x2   /* Enable/Disable TXQ PAUSE. */
    #define BCM_SWITCH_TOTAL_PAUSE      0x4   /* Enable/Disable TOTAL DROP. */
    #define BCM_SWITCH_TOTAL_DROP       0x8   /* Enable/Disable TOTAL PAUSE. */
```

Description

These API allow configuring and retrieving the various switch flow control buffer management mechanisms.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Buffer management module is not initialized
BCM_E_XXX	Operation failed

BROADCOM CORPORATION
5300 California Avenue
Irvine, CA 92617
Phone: 949-926-5000
Fax: 949-926-5203

4 BCM Switch/ENET Driver API

The APIs listed in this section are only available through the ethswctl_api library. Many of these API may also be available in Switch SDK with different names.

4.1 Port Pause Configuration API

4.1.1 Pause Capability Configuration API:

bcm_port_pause_capability_set
bcm_port_pause_capability_get

Configure or Retrieve the pause capability of a given port.

Syntax
#include <bcm/ethswctl_api.h>
int bcm_port_pause_capability_set(IN int unit, IN bcm_port_t port, IN uint8 val);
int bcm_port_pause_capability_get(IN int unit, IN bcm_port_t port, OUT uint8 *val);

Parameters

unit	BCM device number
port	Device or logical port number.
val	Pause capability values.

Description
These API allow configuring and retrieving the pause capability of a given port.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Buffer management module is not initialized
BCM_E_XXX	Operation failed

4.2 ENET Driver Configuration API

4.2.1 Rx Scheduling Configuration

bcm_robo_enet_driver_rx_scheduling_set
bcm_robo_enet_driver_rx_scheduling_get

Configure or Retrieve the ENET driver rx scheduling.

Syntax
#include <bcm/ethswctl_api.h>
int bcm_enet_driver_rx_scheduling_set(IN int unit, IN int scheduling);
int bcm_enet_driver_rx_scheduling_get(IN int unit, OUT int *scheduling);

Parameters

unit	BCM device number
scheduling	The scheduling mechanism

#define SP_SCHEDULING 0
#define WRR_SCHEDULING 1

Note: For RR scheduling, select WRR scheduling and set all the weights as same.

Description
These API allow configuring and retrieving the ENET driver scheduling mechanism across the rx iudma channels.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Buffer management module is not initialized
BCM_E_XXX	Operation failed

4.2.2 WRR Scheduling Configuration

bcm_robo_enet_driver_wrr_config_set
bcm_robo_enet_driver_wrr_config_get

Configure or Retrieve the WRR scheduling parameters.

Syntax
#include <bcm/ethswctl_api.h>
int bcm_enet_driver_wrr_weights_set(IN int unit, IN int max_pkts_per_iter, IN int *weights);
int bcm_enet_driver_wrr_weights_get(IN int unit, OUT int *max_pkts_per_iter, OUT int *weights);

Parameters

unit	BCM device number
max_pkts_per_iter	Max number of packets when the weights will be reloaded.
weights	Integer array of WRR weights of all 4 channels.

Note: For RR scheduling set all the weights as same.

Description
These API allow configuring and retrieving the WRR scheduling parameters.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Buffer management module is not initialized
BCM_E_XXX	Operation failed

4.3 VLAN API

4.3.1 Read or Write a VLAN Table Entry

bcm_robo_vlan_port_set
bcm_robo_vlan_port_get

Write or Read a VLAN table entry.

Syntax
#include <bcm/ethswctl_api.h>
int bcm_vlan_port_set(IN int unit, IN int vid, IN int fwd_map, IN int untag_map);
nt bcm_vlan_port_get(IN int unit, IN int vid, OUT int * fwd_map, OUT int *untag_map);

Parameters

unit	BCM device number
vid	VLAN ID
fwd_map	Members of the VLAN
untag_map	Untagged members of the VLAN

Description

These API allow reading and writing VLAN table entries. The set operation allows configuring a given VLAN entry and the get operation allows reading the fwd_map and untag_map of the given VLAN.

Returns

BCM_E_NONE	Operation completed successfully.
BCM_E_INIT	Buffer management module is not initialized
BCM_E_XXX	Operation failed