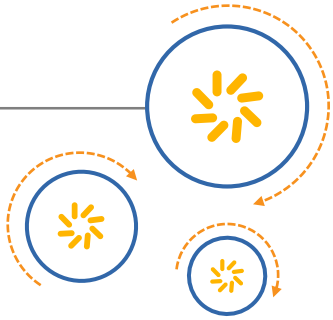
NOTICE REGARDING QUALCOMM ATHEROS, INC.

Effective June 2016, Qualcomm Atheros, Inc. (QCA) transferred certain of its assets, including substantially all of its products and services, to its parent corporation, Qualcomm Technologies, Inc. Qualcomm Technologies, Inc. is a wholly-owned subsidiary of Qualcomm Incorporated. Accordingly, references in this document to Qualcomm Atheros, Inc., Qualcomm Atheros, Atheros, QCA or similar references, should properly reference, and shall be read to reference, Qualcomm Technologies, Inc.

Qualcomm Atheros, Inc.



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

Reference Manual

80-Y9571-8 Rev. B

November 18, 2015

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

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

Revision history

Revision	Date	Description
A	June 2015	Initial release
B	November 2015	Add 2.14.1.5 to 2.14.1.9. Add 2.14.2.46 to 2.14.2.70.

Contents

1 Overview

This document provides detailed API function calls and related data structures used in QCA SOHO Switch SDK (SSDK) package. For the usage of the SSDK, please refer to the document "QCA SOHO switch software development kit user manual".

2 SSDK API Module Documentation

In SSDK, enum sw_error_t is used to return API executing result, detailed definitions are as below:

```
typedef enum {
        SW_OK              = 0,       /* Operation succeeded                */
        SW_FAIL            = -1,      /* Operation failed                   */
        SW_BAD_VALUE       = -2,      /* Illegal value                      */
        SW_OUT_OF_RANGE    = -3,      /* Value is out of range              */
        SW_BAD_PARAM       = -4,      /* Illegal parameter(s)               */
        SW_BAD_PTR         = -5,      /* Illegal pointer value              */
        SW_BAD_LEN         = -6,      /* Wrong length                       */
        SW_BAD_STATE       = -7,      /* Wrong state of state machine       */
        SW_READ_ERROR      = -8,      /* Read operation failed              */
        SW_WRITE_ERROR     = -9,      /* Write operation failed             */
        SW_CREATE_ERROR    = -10,     /* Fail in creating an entry          */
        SW_DELETE_ERROR    = -11,     /* Fail in deleteing an entry         */
        SW_NOT_FOUND       = -12,     /* Entry not found                    */
        SW_NO_CHANGE       = -13,     /* The parameter(s) is the same       */
        SW_NO_MORE         = -14,     /* No more entry found                */
        SW_NO_SUCH         = -15,     /* No such entry                      */
        SW_ALREADY_EXIST   = -16,     /* Tried to create existing entry     */
        SW_FULL            = -17,     /* Table is full                      */
        SW_EMPTY           = -18,     /* Table is empty                     */
        SW_NOT_SUPPORTED   = -19,     /* This request is not support        */
        SW_NOT_IMPLEMENTED = -20,     /* This request is not implemented    */
        SW_NOT_INITIALIZED = -21,     /* The item is not initialized        */
        SW_BUSY            = -22,     /* Operation is still running         */
        SW_TIMEOUT         = -23,     /* Operation Time Out                 */
        SW_DISABLE         = -24,     /* Operation is disabled              */
        SW_NO_RESOURCE     = -25,     /* Resource not available (memory ...) */
        SW_INIT_ERROR      = -26,     /* Error occured while INIT process   */
        SW_NOT_READY       = -27,     /* The other side is not ready yet    */
        SW_OUT_OF_MEM      = -28,     /* Cpu memory allocation failed.      */
        SW_ABORTED         = -29      /* Operation has been aborted.        */
} sw_error_t;
```

2.1 FAL_ACL

2.1.1 Typedef documentation

2.1.1.1 typedef a_uint32_t fal_acl_action_map_t

This type defines the action in ACL rule. Every bit stands for one action item:

```
#define    FAL_ACL_ACTION_PERMIT        0     /*permit forwarding of matched
packet*/
#define    FAL_ACL_ACTION_DENY          1     /*drop matched packet*/
#define    FAL_ACL_ACTION_REDPT         2     /*redirect packet to target port*/
#define    FAL_ACL_ACTION_RDTCPU        3     /*redirect packet to CPU*/
#define    FAL_ACL_ACTION_CPYCPU        4     /*copy packet to CPU*/
#define    FAL_ACL_ACTION_MIRROR        5     /*enable mirror action*/
#define    FAL_ACL_ACTION_MODIFY_VLAN   6     /*enable vlan modification action*/
#define    FAL_ACL_ACTION_NEST_VLAN     7     /*enable nest vlan modification
action*/
#define    FAL_ACL_ACTION_REMARK_UP     8     /*enable vlan priority modification
action*/
#define    FAL_ACL_ACTION_REMARK_QUEUE  9     /*enable egress queue action*/
#define    FAL_ACL_ACTION_REMARK_STAG_VID     10 /*enable service tag modification
action*/
#define    FAL_ACL_ACTION_REMARK_STAG_PRI     11 /*enable service priority
modification action*/
#define    FAL_ACL_ACTION_REMARK_STAG_DEI     12 /*enable service dei modification
action*/
#define    FAL_ACL_ACTION_REMARK_CTAG_VID     13 /*enable customer tag
modification action*/
#define    FAL_ACL_ACTION_REMARK_CTAG_PRI     14 /*enable customer priority
modification action*/
#define    FAL_ACL_ACTION_REMARK_CTAG_CFI     15 /*enable customer cfi
modification action*/
#define    FAL_ACL_ACTION_REMARK_LOOKUP_VID   16 /*enable lookup vlan modification
action*/
#define    FAL_ACL_ACTION_REMARK_DSCP         17 /*enable dscp modification
action*/
#define    FAL_ACL_ACTION_POLICER_EN          18 /*enable policer action*/
#define    FAL_ACL_ACTION_WCMP_EN             19 /*enable wcmp action*/
#define    FAL_ACL_ACTION_ARP_EN              20 /*enable arp action*/
#define    FAL_ACL_ACTION_POLICY_FORWARD_EN   21 /*enable policy forward action*/
#define    FAL_ACL_ACTION_BYPASS_EGRESS_TRANS 22 /*enable bypass egress check
action*/
#define    FAL_ACL_ACTION_MATCH_TRIGGER_INTR  23 /*enable trigger interrupt
action*/
```

Comments: It's a bit map type; we can access it through macro.

- FAL_ACTION_FLG_SET
- FAL_ACTION_FLG_CLR
- FAL_ACTION_FLG_TST

2.1.1.2 typedef a_uint32_t fal_acl_field_map_t[2]

This type defines the field in ACL rule. Every bit stands for one field item:

```
#define    FAL_ACL_FIELD_MAC_DA          0   /*flag for destination MAC address*/
#define    FAL_ACL_FIELD_MAC_SA          1   /*flag for source MAC address*/
#define    FAL_ACL_FIELD_MAC_ETHTYPE     2   /*flag for Ethernet Type*/
#define    FAL_ACL_FIELD_MAC_TAGGED      3   /*flag for tagged*/
#define    FAL_ACL_FIELD_MAC_UP          4   /*flag for priority of VLAN*/
#define    FAL_ACL_FIELD_MAC_VID         5   /*flag for VLAN ID*/
#define    FAL_ACL_FIELD_IP4_SIP         6   /*flag for ipv4 source IP address*/
#define    FAL_ACL_FIELD_IP4_DIP         7   /*flag for ipv4 destination IP
address*/
#define    FAL_ACL_FIELD_IP6_LABEL       8   /*flag for IPv6 label*/
#define    FAL_ACL_FIELD_IP6_SIP         9   /*flag for ipv6 source IP address*/
#define    FAL_ACL_FIELD_IP6_DIP         10  /*flag for ipv6 destination IP
address*/
#define    FAL_ACL_FIELD_IP_PROTO        11   /*flag for IP protocol type*/
#define    FAL_ACL_FIELD_IP_DSCP         12   /*flag for IP DSCP*/
#define    FAL_ACL_FIELD_L4_SPORT        13   /*flag for IP source port*/
#define    FAL_ACL_FIELD_L4_DPORT        14   /*flag for IP destination port*/
#define    FAL_ACL_FIELD_UDF             15   /*flag for user defined field*/
#define    FAL_ACL_FIELD_MAC_CFI         16   /*flag for VLAN cfi*/
#define    FAL_ACL_FIELD_ICMP_TYPE       17   /*flag for IP ICMP type*/
#define    FAL_ACL_FIELD_ICMP_CODE       18   /*flag for IP ICMP code*/
#define    FAL_ACL_FIELD_TCP_FLAG        19   /*flag for TCP flag*/
#define    FAL_ACL_FIELD_RIPV1           20   /*flag for IP RIPv1*/
#define    FAL_ACL_FIELD_DHCPV4          21   /*flag for IP DHCPv4*/
#define    FAL_ACL_FIELD_DHCPV6          22   /*flag for IP DHCPv6*/
#define    FAL_ACL_FIELD_MAC_STAG_VID    23   /*flag for service VLAN ID*/
#define    FAL_ACL_FIELD_MAC_STAG_PRI    24   /*flag for service VLAN priority*/
#define    FAL_ACL_FIELD_MAC_STAG_DEI    25   /*flag for service VLAN dei*/
#define    FAL_ACL_FIELD_MAC_STAGGED     26   /*flag for service tagged*/
#define    FAL_ACL_FIELD_MAC_CTAG_VID    27   /*flag for customer VLAN ID*/
#define    FAL_ACL_FIELD_MAC_CTAG_PRI    28   /*flag for customer VLAN priority*/
#define    FAL_ACL_FIELD_MAC_CTAG_CFI    29   /*flag for customer VLAN cfi*/
#define    FAL_ACL_FIELD_MAC_CTAGGED     30   /*flag for customer tagged*/
#define    FAL_ACL_FIELD_INVERSE_ALL     31   /*flag for inverse all field*/
```

Comments: It's a bit map type; we can access it through macro.

- FAL_FIELD_FLG_SET
- FAL_FIELD_FLG_CLR
- FAL_FIELD_FLG_TST

2.1.2 Enumeration Type Documentation

2.1.2.1 enum fal_acl_bind_obj_t

This enum defines the ACL will work on which particular object.

Enumeration values:

FAL_ACL_BIND_PORT Acl will work on particular port.

2.1.2.2 enum fal_acl_direc_t

This enum defines the ACL will work on which direction traffic.

Enumeration values:

FAL_ACL_DIREC_IN ACL will work on ingressive traffic.

FAL_ACL_DIREC_EG ACL will work on regressive traffic.

FAL_ACL_DIREC_BOTH ACL will work on both ingressive and regressive traffic.

2.1.2.3 enum fal_acl_field_op_t

This enum defines the ACL field operation type.

Enumeration values:

FAL_ACL_FIELD_MASK match operation is mask.

FAL_ACL_FIELD_RANGE match operation is range.

FAL_ACL_FIELD_LE match operation is less and equal.

FAL_ACL_FIELD_GE match operation is great and equal.

FAL_ACL_FIELD_NE match operation is not equal.

2.1.2.4 enum fal_acl_rule_type_t

This enum defines the ACL rule type.

Enumeration values:

FAL_ACL_RULE_MAC include MAC, UDF fields.

FAL_ACL_RULE_IP4 include MAC, IP4 and TCP/UDP UDF fields.

FAL_ACL_RULE_IP6 include MAC, IP6 and TCP/UDP UDF fields.

FAL_ACL_RULE_UDF only include user defined fields.

2.1.3 Structure documentation

2.1.3.1 struct fal_acl_rule_t

This struct defines the ACL rule.

```
typedef struct
{
        fal_acl_rule_type_t rule_type;  /*The rule type*/
        fal_acl_field_map_t field_flg;  /*The rule field map flag*/

        /* fields of mac rule */
        fal_mac_addr_t     src_mac_val;  /*Value of source MAC address*/
        fal_mac_addr_t     src_mac_mask; /*Mask of source MAC address*/
        fal_mac_addr_t     dest_mac_val; /*Value of destination MAC address*/
        fal_mac_addr_t     dest_mac_mask;/*Mask of destination MAC address*/
        a_uint16_t         ethtype_val;  /*Value of Ethernet type*/
        a_uint16_t         ethtype_mask; /*Value of Ethernet type */
        a_uint16_t         vid_val; /*Value of VLAN id*/
        a_uint16_t         vid_mask; /*Mask of VLAN id*/
        fal_acl_field_op_t vid_op; /*Operation type of VLAN id*/
        a_uint8_t          tagged_val; /*Value of tagged*/
        a_uint8_t          tagged_mask; /*Mask of tagged*/
        a_uint8_t          up_val; /*Value of VLAN priority*/
        a_uint8_t          up_mask; /*Mask of VLAN priority*/
        a_uint8_t          cfi_val; /*Value of VLAN cfi*/
        a_uint8_t          cfi_mask; /*Mask of VLAN cfi*/
        a_uint16_t         resv0; /*For reserved use*/

        /* fields of enhanced mac rule*/
        a_uint8_t          stagged_val; /*Value of service tagged*/
        a_uint8_t          stagged_mask;/*Mask of service tagged*/
        a_uint8_t          ctagged_val; /*Value of customer tagged*/
        a_uint8_t          ctagged_mask;/*Mask of customer tagged*/
        a_uint16_t         stag_vid_val;/*Value of service VLAN id*/
        a_uint16_t         stag_vid_mask; /*Mask of service VLAN id*/
        fal_acl_field_op_t stag_vid_op;   /*Operation type of service VLAN id*/
        a_uint16_t         ctag_vid_val;  /*Value of customer VLAN id*/
        a_uint16_t         ctag_vid_mask; /*Mask of customer VLAN id*/
        fal_acl_field_op_t ctag_vid_op;   /*Operation type of customer VLAN id*/
```

```
        a_uint8_t          stag_pri_val;  /*Value of service VLAN priority*/
        a_uint8_t          stag_pri_mask; /*Mask of service VLAN priority*/
        a_uint8_t          ctag_pri_val;  /*Value of customer VLAN priority*/
        a_uint8_t          ctag_pri_mask; /*Mask of customer VLAN priority*/
        a_uint8_t          stag_dei_val;  /*Value of service VLAN dei*/
        a_uint8_t          stag_dei_mask; /*Mask of service VLAN dei*/
        a_uint8_t          ctag_cfi_val;  /*Value of customer VLAN cfi*/
        a_uint8_t          ctag_cfi_mask; /*Mask of customer VLAN cfi*/

        /* fields of ip4 rule */
        fal_ip4_addr_t      src_ip4_val;  /*Value of ipv4 source IP address*/
        fal_ip4_addr_t      src_ip4_mask; /*Mask of ipv4 source IP address*/
        fal_ip4_addr_t      dest_ip4_val; /*Value of ipv4 destination IP address*/
        fal_ip4_addr_t      dest_ip4_mask;/*Mask of ipv4 destination IP address*/

        /* fields of ip6 rule */
        a_uint32_t         ip6_lable_val; /*Value of ipv6 IP label*/
        a_uint32_t         ip6_lable_mask;/*Mask of ipv6 IP label*/
        fal_ip6_addr_t      src_ip6_val;  /*Value of ipv6 source IP address*/
        fal_ip6_addr_t      src_ip6_mask; /*Mask of ipv6 source IP address*/
        fal_ip6_addr_t      dest_ip6_val; /*Value of ipv6 destination IP address*/
        fal_ip6_addr_t      dest_ip6_mask;/*Mask of ipv6 destination IP address*/

        /* fields of ip rule */
        a_uint8_t          ip_proto_val;  /*Value of IP protocol*/
        a_uint8_t          ip_proto_mask; /*Mask of IP protocol*/
        a_uint8_t          ip_dscp_val;   /*Value of IP dscp*/
        a_uint8_t          ip_dscp_mask;  /*Mask of IP dscp*/

        /* fields of layer four */
        a_uint16_t         src_l4port_val;  /*Value of IP source port*/
        a_uint16_t         src_l4port_mask; /*Mask of IP source port*/
        fal_acl_field_op_t src_l4port_op;   /*Operation type of IP source port*/
        a_uint16_t         dest_l4port_val; /*Value of IP destination port*/
        a_uint16_t         dest_l4port_mask;/*Mask of IP destination port*/
        fal_acl_field_op_t dest_l4port_op;  /*Operation type of IP destination
port*/
        a_uint8_t          icmp_type_val;  /*Value of IP icmp type*/
        a_uint8_t          icmp_type_mask; /*Mask of IP icmp type*/
        a_uint8_t          icmp_code_val;  /*Value of IP icmp code*/
        a_uint8_t          icmp_code_mask; /*Mask of IP icmp code*/
        a_uint8_t          tcp_flag_val;   /*Value of IP tcp flag*/
        a_uint8_t          tcp_flag_mask;  /*Mask of IP tcp flag*/
        a_uint8_t          ripv1_val;  /*Value of IP ripv1*/
```

```
        a_uint8_t          ripv1_mask; /*Mask of IP ripv1*/
        a_uint8_t          dhcpv4_val; /*Value of IP dhcpv4*/
        a_uint8_t          dhcpv4_mask; /*Mask of IP dhcpv4*/
        a_uint8_t          dhcpv6_val;  /*Value of IP dhcpv6*/
        a_uint8_t          dhcpv6_mask; /*Mask of IP dhcpv6*/

        /* user defined fields */
        fal_acl_udf_type_t udf_type;  /*user define field type*/
        a_uint8_t udf_offset;  /*user define field offset*/
        a_uint8_t udf_len;  /*user define field length*/
        a_uint8_t udf_val[FAL_ACL_UDF_MAX_LENGTH];  /*value of user define field*/
        a_uint8_t udf_mask[FAL_ACL_UDF_MAX_LENGTH]; /*Mask of user define field*/

        /* fields of action */
        fal_acl_action_map_t  action_flg;  /*The rule field map flag*/
        fal_pbmp_t            ports;  /*The destination port*/
        a_uint32_t            match_cnt;  /*Matched packet counter*/
        a_uint16_t            vid;  /*The VLAN id of action*/
        a_uint8_t             up;  /*The VLAN priority of action*/
        a_uint8_t             queue;  /*The queue id of action*/
        a_uint16_t            stag_vid; /*The service VLAN id of action*/
        a_uint8_t             stag_pri;  /*The service VLAN priority of action*/
        a_uint8_t             stag_dei;  /*The service VLAN dei of action*/
        a_uint16_t            ctag_vid;  /*The customer VLAN id of action*/
        a_uint8_t             ctag_pri;  /*The customer VLAN priority of action*/
        a_uint8_t             ctag_cfi;  /*The customer VLAN cfi of action*/
        a_uint16_t            policer_ptr;  /*The policer id of action*/
        a_uint16_t            arp_ptr;   /*The arp index of action*/
        a_uint16_t            wcmp_ptr;  /*The wcmp index of action*/
        a_uint8_t             dscp; /*The dscp of action*/
        a_uint8_t             rsv;  /*The reserved use of action*/
        fal_policy_forward_t  policy_fwd;  /*The policy forward of action*/
        fal_combined_t    combined;  /*The rule position in extension rule*/
} fal_acl_rule_t;
```

2.1.4 Function documentation

2.1.4.1 fal_acl_list_bind

Definition	Bind an ACL list to a particular object.	
Prototype	sw_error_t fal_acl_list_bind(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	fal_acl_direct_t direc,	Direction of this binding operation

	fal_acl_bind_obj_t obj_t,	Object type of this binding operation
	a_uint32_t obj_idx	Object index of this binding operation
	)	
Return Value	SW_OK or error code	

2.1.4.2 fal_acl_list_creat

Definition	Create an ACL list.	
Prototype	sw_error_t fal_acl_list_creat(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	a_uint32_t prio	ACL list priority
	)	
Description	The smaller the priority of a list is, the higher it is, that means the list could be first matched.	
Return Value	SW_OK or error code	

2.1.4.3 fal_acl_list_destroy

Definition	Destroy an ACL list.	
Prototype	sw_error_t fal_acl_list_destroy(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	)	
Description	Before destroying an ACL list, this ACL list must be unbounded, or this ACL, list can't be destroyed.	
Return Value	SW_OK or error code	

2.1.4.4 fal_acl_list_unbind

Definition	Unbind an ACL list from a particular object.	
Prototype	sw_error_t fal_acl_list_unbind(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	fal_acl_direct_t direc,	Direction of this binding operation
	fal_acl_bind_obj_t obj_t,	Object type of this binding operation
	a_uint32_t obj_idx	Object index of this binding operation
	)	
Return Value	SW_OK or error code	

2.1.4.5 fal_acl_port_udf_profile_get

Definition	Get user define fields profile on a particular port.	
Prototype	sw_error_t fal_acl_port_udf_profile_get(	
	a_uint32_t device_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_acl_udf_type_t direc,	UDF type
	a_uint32_t * offset,	UDF offset
	a_uint32_t * length	UDF length
	)	
Return Value	SW_OK or error code	

2.1.4.6 fal_acl_port_udf_profile_set

Definition	Set user defined fields profile on a particular port.	
Prototype	sw_error_t fal_acl_port_udf_profile_set(	
	a_uint32_t device_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_acl_udf_type_t direc,	UDF type
	a_uint32_t offset,	UDF offset
	a_uint32_t length	UDF length
	)	
Description	In every port, up to 5 types of user define field can be supported and setup. In order to use these user defined field, user defined field data and mask must be setup during ACL rule adding.	
Return Value	SW_OK or error code	

2.1.4.7 fal_acl_rule_add

Definition	Add one or more rule to an existing ACL list.	
Prototype	sw_error_t fal_acl_rule_add(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	a_uint32_t rule_id,	First rule ID of this adding operation in list
	a_uint32_t rule_nr,	Rule number of this adding operation
	fal_acl_rule_t * rule	Rules content of this adding operation
	)	
Description	Adding one or more ACL rule to an existing ACL list in hardware, it will be affective after binding this ACL list to particular port.	
Return Value	SW_OK or error code	

2.1.4.8 fal_acl_rule_delete

Definition	Delete one or more rule from an existing ACL list.	
Prototype	sw_error_t fal_acl_rule_delete(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	a_uint32_t rule_id,	First rule ID of this adding operation in list
	a_uint32_t rule_nr,	Rule number of this adding operation
	)	
Description	After ACL list unbounded from particular port, rules of this list can be deleted from hardware, or can't be permitted.	
Return Value	SW_OK or error code	

2.1.4.9 fal_acl_rule_query

Definition	Query one particular ACL rule in a particular ACL list.	
Prototype	sw_error_t fal_acl_rule_query(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	a_uint32_t rule_id,	First rule ID of this adding operation in list
	fal_acl_rule_t * rule	Rules content of this adding operation
	)	
Return Value	SW_OK or error code	

2.1.4.10 fal_acl_status_get

Definition	Get working status of ACL engine on a particular device.	
Prototype	sw_error_t fal_acl_status_get(	
	a_uint32_t device_id,	Device ID
	a_bool_t * enable	A_TRUE or A_FALSE
	)	
Return Value	SW_OK or error code	

2.1.4.11 fal_acl_status_set

Definition	Set working status of ACL engine on a particular device.	
Prototype	sw_error_t fal_acl_status_set(	
	a_uint32_t device_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	

Description	Only ACL engine is in working status, ACL function can be effective.
Return Value	SW_OK or error code

2.1.4.12 fal_acl_rule_active

Definition	Active one or more rule in an existing ACL list.	
Prototype	sw_error_t fal_acl_rule_active(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	a_uint32_t rule_id,	First rule ID of this active operation in list
	a_uint32_t rule_nr,	Rule number of this active operation
	)	
Description	If rule is on inactive status, use this API to active rule.	
Return Value	SW_OK or error code	

2.1.4.13 fal_acl_rule_deactive

Definition	De-active one or more rule in an existing ACL list.	
Prototype	sw_error_t fal_acl_rule_deactive(	
	a_uint32_t device_id,	Device ID
	a_uint32_t list_id,	ACL list ID
	a_uint32_t rule_id,	First rule ID of this active operation in list
	a_uint32_t rule_nr,	Rule number of this active operation
	)	
Description	To make one or more ACL rule is ineffective, call this API.	
Return Value	SW_OK or error code	

2.1.4.14 fal_acl_rule_src_filter_sts_set

Definition	Set status of one ACL rule source filter.	
Prototype	sw_error_t fal_acl_rule_src_filter_sts_set(	
	a_uint32_t device_id,	Device ID
	a_uint32_t rule_id,	Rule ID of this operation in list
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	This rule ID means hardware rule ID instead of software list ID and rule ID. By this way, we can active or de-active any ACL rule from hardware level.	
Return Value	SW_OK or error code	

2.1.4.15 fal_acl_rule_src_filter_sts_get

Definition	Get status of one ACL rule source filter.	
Prototype	sw_error_t fal_acl_rule_src_filter_sts_get(	
	a_uint32_t device_id,	Device ID
	a_uint32_t rule_id,	Rule ID of this operation in list
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	This rule ID means hardware rule ID instead of software list ID and rule ID. By this way, we can active or de-active any ACL rule from hardware level.	
Return Value	SW_OK or error code	

2.2 FAL_COSMAP

2.2.1 Structure documentation

2.2.1.1 fal_egress_remark_table_t

```
typedef struct
{
    a_bool_t remark_dscp;  // remark DSCP or not
    a_bool_t remark_up;    // remark 802.1P priority or not
    a_bool_t remark_dei;    // remark dei or not
    a_uint8_t g_dscp;          // green packet DSCP
    a_uint8_t y_dscp;          // yellow packet DSCP
    a_uint8_t g_up;            // green packet 802.1P priority
    a_uint8_t y_up;            // yellow packet 802.1P priority
    a_uint8_t g_dei;            // green packet dei
    a_uint8_t y_dei;            // yellow packet dei
} fal_egress_remark_table_t;
```

2.2.2 Function documentation

2.2.2.1 fal_cosmap_dscp_to_pri_set()

Definition	Set DSCP to internal priority mapping on one particular device.	
Prototype	sw_error_t fal_cosmap_dscp_to_pri_set(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t dscp,	DSCP
	a_uint32_t pri	Internal priority
	);	
Return Value	SW_OK or error code	

2.2.2.2 fal_cosmap_dscp_to_pri_get()

Definition	Get DSCP to internal priority mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_dscp_to_pri_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t * pri`	Internal priority
	`);`	
Return Value	SW_OK or error code	

2.2.2.3 fal_cosmap_dscp_to_dp_set()

Definition	Set DSCP to internal drop precedence mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_dscp_to_dp_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t dp`	Internal drop precedence
	`);`	
Return Value	SW_OK or error code	

2.2.2.4 fal_cosmap_dscp_to_dp_get()

Definition	Get DSCP to internal drop precedence mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_dscp_to_dp_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t * dp`	Internal drop precedence
	`);`	
Return Value	SW_OK or error code	

2.2.2.5 fal_cosmap_up_to_pri_set()

Definition	Set 802.1P to internal priority mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_up_to_pri_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t up,`	802.1P
	`a_uint32_t pri`	Internal priority
	`);`	
Return Value	SW_OK or error code	

2.2.2.6 fal_cosmap_up_to_pri_get()

Definition	Get 802.1P to internal priority mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_up_to_pri_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t up,`	802.1P
	`a_uint32_t * pri`	Internal priority
	`);`	
Return Value	SW_OK or error code	

2.2.2.7 fal_cosmap_up_to_dp_set()

Definition	Set dot1p to internal drop precedence mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_up_to_dp_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t up,`	802.1P
	`a_uint32_t dp`	Internal drop precedence
	`);`	
Return Value	SW_OK or error code	

2.2.2.8 fal_cosmap_up_to_dp_get()

Definition	Get dot1p to internal drop precedence mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_up_to_dp_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t up,`	802.1P
	`a_uint32_t * dp`	Internal drop precedence
	`);`	
Return Value	SW_OK or error code	

2.2.2.9 fal_cosmap_dscp_to_ehpri_set()

Definition	Set DSCP to internal priority mapping on one particular device for 0, 5, 6 ports.	
Prototype	`sw_error_t fal_cosmap_dscp_to_ehpri_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t pri`	Internal priority
	`);`	
Return Value	SW_OK or error code	

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

2.2.2.10 fal_cosmap_dscp_to_ehpri_get()

Definition	Get DSCP to internal priority mapping on one particular device for 0, 5, 6 ports.	
Prototype	`sw_error_t fal_cosmap_dscp_to_pri_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t * pri`	Internal priority
	`);`	
Return Value	SW_OK or error code	

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

2.2.2.11 fal_cosmap_dscp_to_ehdp_set()

Definition	Set DSCP to internal drop precedence mapping on one particular device for 0, 5, 6 ports.	
Prototype	`sw_error_t fal_cosmap_dscp_to_dp_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t dp`	Internal drop precedence
	`);`	
Return Value	SW_OK or error code	

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

2.2.2.12 fal_cosmap_dscp_to_ehdp_get()

Definition	Get DSCP to internal drop precedence mapping on one particular device for 0, 5, 6 ports.	
Prototype	`sw_error_t fal_cosmap_dscp_to_dp_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t dscp,`	DSCP
	`a_uint32_t * dp`	Internal drop precedence
	`);`	
Return Value	SW_OK or error code	

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

2.2.2.13 fal_cosmap_up_to_ehpri_set()

Definition	Set 802.1P to internal priority mapping on one particular device for 0, 5, 6 ports.	
Prototype	`sw_error_t fal_cosmap_up_to_pri_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t up,`	802.1P

	a_uint32_t pri	Internal priority
	);	
Return Value	SW_OK or error code	

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

2.2.2.14 fal_cosmap_up_to_ehpri_get()

Definition	Get 802.1P to internal priority mapping on one particular device for 0, 5, 6 ports.	
Prototype	sw_error_t fal_cosmap_up_to_pri_get(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t up,	802.1P
	a_uint32_t * pri	Internal priority
	);	
Return Value	SW_OK or error code	

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

2.2.2.15 fal_cosmap_up_to_ehdp_set()

Definition	Set dot1p to internal drop precedence mapping on one particular device for 0, 5, 6 ports.	
Prototype	sw_error_t fal_cosmap_up_to_dp_set(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t up,	802.1P
	a_uint32_t dp	Internal drop precedence
	);	
Return Value	SW_OK or error code	

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

2.2.2.16 fal_cosmap_up_to_ehdp_get()

Definition	Get dot1p to internal drop precedence mapping on one particular device for 0, 5, 6 ports.	
Prototype	sw_error_t fal_cosmap_up_to_dp_get(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t up,	802.1P
	a_uint32_t * dp	Internal drop precedence
	);	
Return Value	SW_OK or error code	

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

2.2.2.17 fal_cosmap_pri_to_queue_set()

Definition	Set internal priority to queue mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_pri_to_queue_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t pri,`	Internal priority
	`a_uint32_t queue`	Queue ID
	`);`	
Description	This function is for port 1/2/3/4, which have four egress queues.	
Return Value	SW_OK or error code	

2.2.2.18 fal_cosmap_pri_to_queue_get()

Definition	Get internal priority to queue mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_pri_to_queue_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t pri,`	Internal priority
	`a_uint32_t * queue`	Queue ID
	`);`	
Description	This function is for port 1/2/3/4, which have four egress queues.	
Return Value	SW_OK or error code	

2.2.2.19 fal_cosmap_pri_to_ehqueue_set()

Definition	Set internal priority to queue mapping on one particular device.	
Prototype	`sw_error_t al_cosmap_pri_to_ehqueue_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t pri,`	Internal priority
	`a_uint32_t queue`	Queue ID
	`);`	
Description	This function is for port 0/5/6, which have six egress queues.	
Return Value	SW_OK or error code	

2.2.2.20 fal_cosmap_pri_to_ehqueue_get()

Definition	Get internal priority to queue mapping on one particular device.	
Prototype	`sw_error_t fal_cosmap_pri_to_ehqueue_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t pri,`	Internal priority
	`a_uint32_t * queue`	Queue ID
	`);`	

Description	This function is for port 0/5/6, which have six egress queues
Return Value	SW_OK or error code

2.2.2.21 fal_cosmap_egress_remark_set()

Definition	Set egress queue based CoS remap table on one particular device.	
Prototype	sw_error_t fal_cosmap_egress_remark_set(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t tbl_id,	CoS remark table ID
	fal_egress_remark_table_t * tbl	Remark entry
	);	
Return Value	SW_OK or error code	

2.2.2.22 fal_cosmap_egress_remark_get()

Definition	Get egress queue based CoS remap table on one particular device.	
Prototype	sw_error_t fal_cosmap_egress_remark_get(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t tbl_id,	CoS remark table ID
	fal_egress_remark_table_t * tbl	Remark entry
	);	
Return Value	SW_OK or error code	

2.3 FAL_FDB

2.3.1 Struct documentation

```
/* This structure defines the Fdb entry. */
typedef struct
{
    fal_mac_addr_t addr;
    a_uint16_t    fid;
    fal_fwd_cmd_t dacmd;
    fal_fwd_cmd_t sacmd;
    union
    {
        a_uint32_t id;
        fal_pbmp_t map;
    } port;
    a_bool_t portmap_en; // If portmap_en is A_TRUE, port.map
```

```
                    // is valid, else port.id is valid.
    a_bool_t is_multicast;
    a_bool_t static_en;
    a_bool_t leaky_en;   // if leaky_en is A_TRUE, packets which
                    // DA matches this entry would be leaky.
    a_bool_t mirror_en;  // If mirror_en is A_TRUE, packets which
                    // DA matches this entry would be mirrored.
    a_bool_t clone_en;   // If clone_en is A_TRUE, which means this
                    // address is a mac clone address.
    a_bool_t cross_pt_state;
    a_bool_t da_pri_en;  // if da_pri_en is set to A_TRUE, da_queue
                    // may be used as frames' internal riority.
    a_uint8_t da_queue;
    a_bool_t white_list_en;
    a_bool_t load_balance_en;// only available for Dakota ESS
    a_uint8_t load_balance; // only available for Dakota ESS
} fal_fdb_entry_t;

/* This structure defines the Fdb operation options. */
typedef struct
{
a_bool_t port_en;      // when FDB operation is "GET NEXT",
                       // port is valid.
a_bool_t fid_en;       // when FDB operation is "GET NEXT" or
                       // "TRANSFER", vid is valid.
a_bool_t multicast_en; // when FDB operation is "GET NEXT" or
                       // "TRANSFER", MAC address in the valid
                       // ARL entry must be multicat address.
} fal_fdb_op_t;
```

2.3.2 Function documentation

2.3.2.1 fal_fdb_add

Definition	Add an FDB entry to a particular device.	
Prototype	`sw_error_t fal_fdb_add (`	
	`  a_uint32_t dev_id,`	Device ID
	`  const fal_fdb_entry_t *entry`	FDB entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.2 fal_fdb_del_all

Definition	Delete all FDB entries from a particular device.

Prototype	sw_error_t fal_fdb_del_all (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t flag	Flag
	)	
Description	If FAL_FDB_DEL_STATIC bit is set in flag, delete all FDB entries, else only delete dynamic entries.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.3 fal_fdb_del_by_mac

Definition	Delete an FDB entry by MAC address from a particular device.	
Prototype	sw_error_t fal_fdb_del_by_mac (	
	a_uint32_t dev_id,	Device ID
	const fal_fdb_entry_t *entry	FDB entry
	)	
Description	Accept address and FID field in FDB entry for input. For IVL learning FID field should be set to VLAN ID, while for SVL learning FID field should be set to FAL_SVL_FID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.4 fal_fdb_del_by_port

Definition	Delete FDB entries on a particular port.	
Prototype	sw_error_t fal_fdb_del_by_port (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t flag	Flag
	)	
Description	If FAL_FDB_DEL_STATIC bit is set in flag, delete all FDB entries on the port, else only delete dynamic entries on this port.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.5 fal_fdb_find

Definition	Find a particular FDB entry from a particular device.	
Prototype	sw_error_t fal_fdb_find (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_entry_t *entry	FDB entry
	)	
Description	Accept address and FID field in FDB entry for input. For IVL learning FID field should be set to VLAN ID, while for SVL learning FID field should be set to FAL_SVL_FID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.6 fal_fdb_transfer

Definition	Transfer FDB entries' port information on a particular device.	
Prototype	`sw_error_t fal_fdb_transfer (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t old_port,`	Source port ID
	`fal_port_t new_port,`	Destination port ID
	`a_uint32_t fid,`	Database ID
	`fal_fdb_op_t *option`	Operation options
	`)`	
Description	For IVL learning FID field should be set to VLAN ID, while for SVL learning FID field should be set to FAL_SVL_FID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.7 fal_fdb_port_add

Definition	Add a port to an existing FDB entry on a particular device.	
Prototype	`sw_error_t fal_fdb_port_add (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t fid,`	Filtering database ID
	`fal_mac_addr_t *addr,`	MAC address
	`fal_port_t port_id`	Port ID
	`)`	
Description	For IVL learning FID field should be set to VLAN ID, while for SVL learning FID field should be set to FAL_SVL_FID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.8 fal_fdb_port_del

Definition	Delete a port from an exsiting FDB entry on a particular device.	
Prototype	`sw_error_t fal_fdb_port_del (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t fid,`	Filtering database ID
	`fal_mac_addr_t *addr,`	MAC address
	`fal_port_t port_id`	Port ID
	`)`	
Description	For IVL learning FID field should be set to VLAN ID, while for SVL learning FID field should be set to FAL_SVL_FID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.9 fal_fdb_extend_first

Definition	Get first FDB entry from a particular device.	
Prototype	`sw_error_t fal_fdb_extend_first (`	

	a_uint32_t dev_id,	Device ID
	fal_fdb_op_t *option,	Operation options
	fal_fdb_entry_t *entry	FDB entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.10 fal_fdb_extend_next

Definition	Get next FDB entry from a particular device.	
Prototype	sw_error_t fal_fdb_extend_next (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_op_t *option,	Operation options
	fal_fdb_entry_t *entry	FDB entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.11 fal_fdb_first

Definition	Get first FDB entry from particular device.	
Prototype	sw_error_t fal_fdb_first (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_entry_t *entry	FDB entry
	)	
Description	For AR8337N, this API is not supported. Use fal_fdb_extend_first.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.12 fal_fdb_next

Definition	Get next FDB entry from particular device.	
Prototype	sw_error_t fal_fdb_next (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_entry_t *entry	FDB entry
	)	
Description	For AR8337N, this API is not supported. Use fal_fdb_extend_next.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.13 fal_fdb_iterate

Definition	Iterate all FDB entries on a particular device.	
Prototype	sw_error_t fal_fdb_iterate (	
	a_uint32_t dev_id,	Device ID

	a_uint32_t *iterator,	FDB entry index If it's zero means get the first entry.
	fal_fdb_entry_t *entry	FDB entry
	)	
Description	For AR8337N, this API is not supported. Use fal_fdb_extend_first and fal_fdb_extend_next.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.14 fal_fdb_age_ctrl_get

Definition	Get dynamic address aging status on particular device.	
Prototype	sw_error_t fal_fdb_age_ctrl_get (	
	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.15 fal_fdb_age_ctrl_set

Definition	Set dynamic address aging status on particular device.	
Prototype	sw_error_t fal_fdb_age_ctrl_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.16 fal_fdb_age_time_get

Definition	Get dynamic address aging time on a particular device.	
Prototype	sw_error_t fal_fdb_age_time_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t *time	Aging time
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.17 fal_fdb_age_time_set

Definition	Set dynamic address aging time on a particular device.	
Prototype	sw_error_t fal_fdb_age_time_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t *time	Aging time

	)	
Description	This operation will set dynamic address aging time on a particular device. The unit of time is second. Because different device has differnet hardware granularity, function will return actual time in hardware.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.18 fal_fdb_vlan_ivl_svl_get

Definition	Get FDB mode for invalid VLAN on a particular device.	
Prototype	sw_error_t fal_fdb_vlan_ivl_svl_get (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_smode *smode	IVL or SVL
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.19 fal_fdb_vlan_ivl_svl_set

Definition	Set FDB mode for invalid VLAN a particular device.	
Prototype	sw_error_t fal_fdb_vlan_ivl_svl_set (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_smode smode	IVL or SVL
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.20 fal_fdb_port_learn_get

Definition	Get dynamic address learning status on a particular port.	
Prototype	sw_error_t fal_fdb_port_learn_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.21 fal_fdb_port_learn_set

Definition	Set dynamic address learning status on a particular port.	
Prototype	sw_error_t fal_fdb_port_learn_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.3.2.22 fal_fdb_port_learn_static_get

Definition	Get dynamic address learning mode on a particular port.	
Prototype	`sw_error_t fal_fdb_port_learn_static_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.23 fal_fdb_port_learn_static_set

Definition	Set dynamic address learning mode on a particular port.	
Prototype	`sw_error_t fal_fdb_port_learn_static_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Description	If this feature enabled, dynamic address learned in this port will be added to FDB table as static entries.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.24 fal_fdb_learn_limit_get

Definition	Get dynamic address learning count limit on a particular device.	
Prototype	`sw_error_t fal_fdb_learn_limit_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable,`	A_TRUE or A_FALSE
	`a_uint32_t *cnt`	Limit count
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.25 fal_fdb_learn_limit_set

Definition	Set dynamic address learning count limit on a particular device.	
Prototype	`sw_error_t fal_fdb_learn_limit_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable,`	A_TRUE or A_FALSE

	a_uint32_t cnt	Limit count
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.26 fal_port_fdb_learn_limit_get

Definition	Get dynamic address learning count limit on a particular port.	
Prototype	sw_error_t fal_port_fdb_learn_limit_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable,	A_TRUE or A_FALSE
	a_uint32_t *cnt	Limit count
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.27 fal_port_fdb_learn_limit_set

Definition	Set dynamic address learning count limit on a particular port.	
Prototype	sw_error_t fal_port_fdb_learn_limit_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable,	A_TRUE or A_FALSE
	a_uint32_t cnt	Limit count
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.28 fal_fdb_learn_exceed_cmd_get

Definition	Get dynamic address learning count exceed command on a particular device.	
Prototype	sw_error_t fal_fdb_learn_exceed_cmd_get (	
	a_uint32_t dev_id,	Device ID
	fal_fwd_cmd_t *cmd	Forwarding command
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.29 fal_fdb_learn_exceed_cmd_set

Definition	Set dynamic address learning count exceed command on a particular device.	
Prototype	sw_error_t fal_fdb_learn_exceed_cmd_set (	
	a_uint32_t dev_id,	Device ID

	fal_fwd_cmd_t cmd	Forwarding command
	)	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_DROP and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.30 fal_port_fdb_learn_exceed_cmd_get

Definition	Get dynamic address learning count exceed command on a particular port.	
Prototype	sw_error_t fal_port_fdb_learn_exceed_cmd_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_fwd_cmd_t *cmd	Forwarding command
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.31 fal_port_fdb_learn_exceed_cmd_set

Definition	Set dynamic address learning count exceed command on a particular port.	
Prototype	sw_error_t fal_port_fdb_learn_exceed_cmd_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_fwd_cmd_t cmd	Forwarding command
	)	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_DROP and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.32 fal_fdb_resv_add

Definition	Add a particular reserve FDB entry.	
Prototype	sw_error_t fal_fdb_resv_add (	
	a_uint32_t dev_id,	Device ID
	fal_fdb_entry_t *entry	Reserve FDB entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.33 fal_fdb_resv_del

Definition	Delete a particular reserve FDB entry through MAC address.	
Prototype	`sw_error_t fal_fdb_resv_del (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fdb_entry_t *entry`	Reserve FDB entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.34 fal_fdb_resv_find

Definition	Find a particular reserve FDB entry by MAC address.	
Prototype	`sw_error_t fal_fdb_resv_find (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fdb_entry_t *entry`	Reserve FDB entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.3.2.35 fal_fdb_resv_iterate

Definition	Iterate all reserve FDB entries on a particular device.	
Prototype	`sw_error_t fal_fdb_resv_iterate (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t *iterator,`	Reserve FDB entry index If it's zero means get the first entry.
	`fal_fdb_entry_t *entry`	Reserve FDB entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4 FAL_IGMP

2.4.1 Struct documentation

```
/* This structure defines the igmp source group address. */
typedef struct
{
    fal_addr_type_t type;      // address type, IPv4 or IPv6
    union
    {
        fal_ip4_addr_t ip4_addr;
        fal_ip6_addr_t ip6_addr;
    } u;
} fal_igmp_sg_addr_t;
```

```
/* This structure defines the igmp source group entry. */
typedef struct
{
    fal_igmp_sg_addr_t source; // multicast source router address
    fal_igmp_sg_addr_t group;  // multicast group address
    fal_pbmp_t port_map;       // multicast group destination pmap
} fal_igmp_sg_entry_t;
```

2.4.2 Function documentation

2.4.2.1 fal_port_igmps_status_get

Definition	Get IGMP/MLD snooping status on particular port.	
Prototype	`sw_error_t fal_port_igmps_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.2 fal_port_igmps_status_set

Definition	Set IGMP/MLD snooping status on particular port.	
Prototype	`sw_error_t fal_port_igmps_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If IGMP/MLD snooping is enabled, the port will examine all received frames and forward IGMP/MLD frames by fal_igmp_mld_cmd_set.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.3 fal_port_igmp_mld_join_get

Definition	Get IGMP/MLD hardware join status on a particular port.	
Prototype	`sw_error_t fal_port_igmp_mld_join_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.4 fal_port_igmp_mld_join_set

Definition	Set IGMP/MLD hardware join status on a particular port.	
Prototype	`sw_error_t fal_port_igmp_mld_join_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If IGMP/MLD hardware join is enabled, hardware will dynamic add or update multicast entry.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.5 fal_port_igmp_mld_leave_get

Definition	Get IGMP/MLD hardware fast leave on a particular port.	
Prototype	`sw_error_t fal_port_igmp_mld_leave_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.6 fal_port_igmp_mld_leave_set

Definition	Set IGMP/MLD hardware fast leave on a particular port.	
Prototype	`sw_error_t fal_port_igmp_mld_leave_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If IGMP/MLD hardware fast leave is enabled, hardware will dynamic delete or update multicast entry.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.7 fal_igmp_mld_entry_v3_get

Definition	Get IGMPv3/MLDv2 packets hardware acknowledgement status on a particular device.	
Prototype	`sw_error_t fal_igmp_mld_entry_v3_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.4.2.8 fal_igmp_mld_entry_v3_set

Definition	Set IGMPv3/MLDv2 packets hardware acknowledgement status on a particular device.	
Prototype	`sw_error_t fal_igmp_mld_entry_v3_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.9 fal_igmp_mld_cmd_get

Definition	Get IGMP/MLD packets forwarding command on a particular device.	
Prototype	`sw_error_t fal_igmp_mld_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t *cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.10 fal_igmp_mld_cmd_set

Definition	Set IGMP/MLD packets forwarding command on a particular device.	
Prototype	`sw_error_t fal_igmp_mld_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t cmd`	Forwarding command
	`)`	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_CPY_TO_CPU and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.11 fal_igmp_mld_rp_get

Definition	Get IGMP/MLD router ports on a particular device.	
Prototype	`sw_error_t fal_igmp_mld_rp_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_pbmp_t *pts`	Dedicates ports
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.12 fal_igmp_mld_rp_set

Definition	Set IGMP/MLD router ports on a particular device.	
Prototype	`sw_error_t fal_igmp_mld_rp_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_pbmp_t pts`	Dedicates ports
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.13 fal_igmp_mld_entry_creat_get

Definition	Get the status of creating multicast entry during IGMP/MLD join/leave procedure.	
Prototype	`sw_error_t fal_igmp_mld_entry_creat_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.14 fal_igmp_mld_entry_creat_set

Definition	Set the status of creating multicast entry during IGMP/MLD join/leave procedure.	
Prototype	`sw_error_t fal_igmp_mld_entry_creat_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	Enable hardware add new address to multicast table when received IGMP/MLD join frame, and remove address from multicast table when received IGMP/MLD leave frame.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.15 fal_igmp_mld_entry_static_get

Definition	Get the static status of multicast entry which learned by hardware.	
Prototype	`sw_error_t fal_igmp_mld_entry_static_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.16 fal_igmp_mld_entry_static_set

Definition	Set the static status of multicast entry which learned by hardware.	
Prototype	`sw_error_t fal_igmp_mld_entry_static_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If the static status of multicast entry is enabled, multicast entry learned by hardware will not be aged.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.17 fal_igmp_mld_entry_leaky_get

Definition	Get the leaky status of multicast entry which learned by hardware.	
Prototype	`sw_error_t fal_igmp_mld_entry_leaky_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.18 fal_igmp_mld_entry_leaky_set

Definition	Set the leaky status of multicast entry which learned by hardware.	
Prototype	`sw_error_t fal_igmp_mld_entry_leaky_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If the leaky status of multicast entry is enabled, multicast entry learned by hardware will be set with leaky flag.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.19 fal_igmp_mld_entry_queue_get

Definition	Get the queue status of multicast entry which learned by hardware.	
Prototype	`sw_error_t fal_igmp_mld_entry_queue_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable,`	A_TRUE or A_FALSE
	`a_uint32_t *queue`	Queue ID
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.20 fal_igmp_mld_entry_queue_set

Definition	Set the queue status of multicast entry which learned by hardware.	
Prototype	sw_error_t fal_igmp_mld_entry_queue_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable,	A_TRUE or A_FALSE
	a_uint32_t queue	Queue ID
	)	
Description	If the queue status of multicast entry is enabled, multicast entry learned by hardware will be set with queue flag.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.21 fal_port_igmp_mld_learn_limit_get

Definition	Get IGMP hardware learning count limit on a particular port.	
Prototype	sw_error_t fal_port_igmp_mld_learn_limit_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable,	A_TRUE or A_FALSE
	a_uint32_t *cnt	Limit count
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.22 fal_port_igmp_mld_learn_limit_set

Definition	Set IGMP hardware learning count limit on a particular port.	
Prototype	sw_error_t fal_port_igmp_mld_learn_limit_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable,	A_TRUE or A_FALSE
	a_uint32_t cnt	Limit count
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.23 fal_port_igmp_mld_learn_exceed_cmd_get

Definition	Get IGMP hardware learning count exceed command on a particular port.	
Prototype	sw_error_t fal_port_igmp_mld_learn_exceed_cmd_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	fal_fwd_cmd_t *cmd	Forwarding command
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.24 fal_port_igmp_mld_learn_exceed_cmd_set

Definition	Set IGMP hardware learning count exceed command on a particular port.	
Prototype	sw_error_t fal_port_igmp_mld_learn_exceed_cmd_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_fwd_cmd_t cmd	Forwarding command
	)	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_DROP and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.25 fal_igmp_sg_entry_set

Definition	Set IGMP multicast source group entry on a particular device.	
Prototype	sw_error_t fal_igmp_sg_entry_set (	
	a_uint32_t dev_id,	Device ID
	fal_igmp_sg_entry_t *entry	Entry
	)	
Description	For IGMPv3/MLDv2 packets, hardware can recognize these packets, but can't process them, thus copy or redirect these packets to CPU. This API is used to create source group entries based on IGMPv3/MLDv2 packets processed by high layer application.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.26 fal_igmp_sg_entry_clear

Definition	Clear IGMP multicast source group entry on a particular device.	
Prototype	sw_error_t fal_igmp_sg_entry_clear (	
	a_uint32_t dev_id,	Device ID
	fal_igmp_sg_entry_t *entry	Entry
	)	
Description	For IGMPv3/MLDv2 packets, hardware can recognize these packets, but can't process them, thus copy or redirect these packets to CPU. This API is used to clear source group entries based on IGMPv3/MLDv2 packets processed by high layer application.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.4.2.27 fal_igmp_sg_entry_show

Definition	Show IGMP multicast source group entry on a particular device.	
Prototype	sw_error_t fal_igmp_sg_entry_show (	
	a_uint32_t dev_id,	Device ID
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.5 FAL_INIT

2.5.1 Structure documentation

2.5.1.1 struct ssdk_init_cfg

Before SSDK initialization, some configuration such as cpu_mode, reg_func etc should be provided with structure ssdk_init_cfg:

```
typedef struct
{
          a_uint32_t cpu_bmp;
          a_uint32_t lan_bmp;
          a_uint32_t wan_bmp;
} ssdk_port_cfg;
typedef struct
{
        hsl_init_mode   cpu_mode;
        hsl_access_mode reg_mode;
        hsl_reg_func    reg_func;

        ssdk_chip_type  chip_type;

        /* os specific parameter */
        /* when uk_if based on netlink, it's netlink protocol type*/
        /* when uk_if based on ioctl, it's minor device number, major number
           is always 10(misc device) */
        a_uint32_t      nl_prot;

        /* chip specific parameter */
        void *          chip_spec_cfg;
          /* port cfg */
          ssdk_port_cfg   port_cfg;
} ssdk_init_cfg;
```

2.5.1.2 struct ssdk_cfg_t

structure ssdk_cfg_t is used to initialize SSDK with necessary as below definition:

```
#define CFG_STR_SIZE 20
typedef struct
{
        a_uint8_t build_ver[CFG_STR_SIZE];
        a_uint8_t build_date[CFG_STR_SIZE];

        a_uint8_t chip_type[CFG_STR_SIZE]; //GARUDA
        a_uint8_t cpu_type[CFG_STR_SIZE];  //mips
        a_uint8_t os_info[CFG_STR_SIZE];   //OS=linux OS_VER=2_6

        a_bool_t  fal_mod;
        a_bool_t  kernel_mode;
        a_bool_t  uk_if;

        ssdk_features features;
        ssdk_init_cfg init_cfg;
} ssdk_cfg_t;
```

2.5.2 Function documentation

2.5.2.1 fal_init

Definition	Init FAL layer with necessary configuration.	
Prototype	sw_error_t fal_init(	
	a_uint32_t device_id,	Device ID
	ssdk_cfg_t ssdk_cfg,	Configuration for initialization
	)	
Description	The operation will init FAL layer and HSL layer.	
Return Value	SW_OK or error code	

2.5.2.2 fal_reduced_init

Definition	Init FAL layer with necessary configuration.	
Prototype	sw_error_t fal_reduced_init(	
	a_uint32_t device_id,	Device ID
	hsl_init_mode cpu_mode,	CPU port connecting mode, including one CPU, two CPU and no-CPU mode.
	hsl_access_mode reg_mode,	Register access mode, MDIO and header modes can be selected.

	)	
Description	The operation will init FAL layer and HSL layer with reduced mode.	
Return Value	SW_OK or error code	

2.5.2.3 fal_reset (a_uint32_t *dev_id*)

Definition	Reset FAL layer to initialization status.	
Prototype	sw_error_t fal_reset(	
	a_uint32_t device_id,	Device ID
	)	
Description	The operation will reset FAL layer and HSL layer.	
Return Value	SW_OK or error code	

2.5.2.4 fal_ssdk_cfg

Definition	Get SSDK configure information.	
Prototype	sw_error_t fal_ssdk_cfg(	
	a_uint32_t device_id,	Device ID
	ssdk_cfg_t * ssdk_cfg	Configuration for initialization
	)	
Description	The operation will return the SSDK configure information to ssdk_cfg.	
Return Value	SW_OK or error code	

2.6 FAL_INERFACE_CTRL

2.6.1 Struct documentation

```
/* This structure defines the mac configuration. */
typedef struct
{
    fx100_ctrl_link_mode_t link_mode; // Support Fx100BASE_MODE
    a_bool_t               overshoot; // overshoot test mode
    a_bool_t               loopback;  // loopback test mode.
    a_bool_t               fd_mode;   // Duplex mode.
                                      // 0 - Half 1 - Full
    a_bool_t               col_test;
    sgmii_fiber_mode_t     sgmii_fiber_mode; // FX100_SERDS_MODE
    a_bool_t               crs_ctrl;
    a_bool_t               loopback_ctrl;
    a_bool_t               crs_col_100_ctrl;
    a_bool_t               loop_en;
} fal_fx100_ctrl_config_t;
```

```
/* This structure defines the mac configuration. */
typedef struct
{
    fal_interface_mac_mode_t   mac_mode; // MAC mode
    union
    {
        fal_mac_rgmii_config_t rgmii;    // RGMII configuration
        fal_mac_gmii_config_t  gmii;     // GMII configuration
        fal_mac_mii_config_t   mii;      // MII configuration
        fal_mac_sgmii_config_t sgmii;    // SGMII configuration
        fal_mac_rmii_config_t  rmii;     // RMII configuration
        fal_mac_fiber_config_t fiber;    // Fiber configuration
    } config;
} fal_mac_config_t;

/* This structure defines the phy configuration. */
typedef struct
{
    fal_interface_mac_mode_t mac_mode;   // MAC mode
    a_bool_t                 txclk_delay_cmd;
    a_bool_t                 rxclk_delay_cmd;
    a_uint32_t               txclk_delay_sel;
    a_uint32_t               rxclk_delay_sel;
} fal_phy_config_t;
```

2.6.2 Function documentation

2.6.2.1 fal_port_3az_status_get

Definition	Get 802.3az status on a particular port.	
Prototype	`sw_error_t fal_port_3az_status_get (`	
	`  a_uint32_t dev_id,`	Device ID
	`  fal_port_t port_id,`	Port ID
	`  a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.2 fal_port_3az_status_set

Definition	Set 802.3az status on a particular port.	
Prototype	`sw_error_t fal_port_3az_status_set (`	

	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.3 fal_interface_mac_mode_get

Definition	Get interface mode on a particular MAC device.	
Prototype	sw_error_t fal_interface_mac_mode_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_mac_config_t *config	Interface configuration
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.4 fal_interface_mac_mode_set

Definition	Set interface mode on a particular MAC device.	
Prototype	sw_error_t fal_interface_mac_mode_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_mac_config_t *config	Interface configuration
	)	
Description	The supported interface mode on a particular MAC device includes RGMII, GMII, MII, SGMII, FIBER, and RMII.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.5 fal_interface_phy_mode_get

Definition	Get interface PHY mode on a particular PHY device.	
Prototype	sw_error_t fal_interface_phy_mode_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t phy_id,	PHY ID
	fal_phy_config_t *config	Interface configuration
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.6 fal_interface_phy_mode_set

Definition	Set interface PHY mode on a particular PHY device.	
Prototype	`sw_error_t fal_interface_phy_mode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t phy_id,`	PHY ID
	`fal_phy_config_t config`	Interface configuration
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.7 fal_interface_fx100_ctrl_get

Definition	Get fx100 control configuration on a particular device.	
Prototype	`sw_error_t fal_interface_fx100_ctrl_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fx100_ctrl_config_t *config`	fx100 control configuration
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.8 fal_interface_fx100_ctrl_set

Definition	Set fx100 control configuration on a particular device.	
Prototype	`sw_error_t fal_interface_fx100_ctrl_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fx100_ctrl_config_t *config`	fx100 control configuration
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.9 fal_interface_fx100_status_get

Definition	Get fx100 status on a particular device.	
Prototype	`sw_error_t fal_interface_fx100_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t *status`	fx100 status
	`)`	
Description	fx100 status can refer to link_mode field in fal_fx100_ctrl_config_t.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.10 fal_interface_mac06_exch_get

Definition	Get MAC0 and MAC6 exchange status on a particular device.	
Prototype	`sw_error_t fal_interface_mac06_exch_get (`	
	`a_uint32_t dev_id,`	Device ID

	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.6.2.11 fal_interface_mac06_exch_set

Definition	Set MAC0 and MAC6 exchange status on a particular device.	
Prototype	sw_error_t fal_interface_mac06_exch_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7 FAL_IP

```
/* IP entry operation flags. */
#define FAL_IP_ENTRY_ID_EN              0x1
#define FAL_IP_ENTRY_INTF_EN            0x2
#define FAL_IP_ENTRY_PORT_EN            0x4
#define FAL_IP_ENTRY_STATUS_EN          0x8
#define FAL_IP_ENTRY_IPADDR_EN          0x10

/* IP WCMP hash key flags */
#define FAL_WCMP_HASH_KEY_SIP           0x1
#define FAL_WCMP_HASH_KEY_DIP           0x2
#define FAL_WCMP_HASH_KEY_SPORT         0x4
#define FAL_WCMP_HASH_KEY_DPORT         0x8
```

2.7.1 Enumeration type documentation

2.7.1.1 enum fal_source_guard_mode_t

This enum defines source guard mode type.

Enumeration values:

FAL_MAC_IP_GUARD Check SMAC & SIP

FAL_MAC_IP_PORT_GUARD Check SMAC & SIP & SP

FAL_MAC_IP_VLAN_GUARD Check SMAC & SIP & VID

FAL_MAC_IP_PORT_VLAN_GUARD Check SMAC & SIP & SP & VID

FAL_NO_SOURCE_GUARD Disable

2.7.1.2 enum fal_arp_learn_mode_t

This enum defines ARP learn mode type.

Enumeration values:

FAL_ARP_LEARN_LOCAL Only learn ARP to Router.

FAL_ARP_LEARN_ALL Learn All ARP

2.7.2 Struct documentation

```
/* This struct defines the host entry. */
typedef struct
{
    a_uint32_t entry_id;
    a_uint32_t flags;   // FAL_IP_IP4_ADDR: IPv4 entry
                        // FAL_IP_IP6_ADDR: IPv6 entry
                        // FAL_IP_CPU_ADDR: this entry is for router
                        // addr, frame should be redirect to cpu.
    a_uint32_t status;  // 7 - static  1~6 - dynamic  0 - invalid
    fal_ip4_addr_t ip4_addr;
    fal_ip6_addr_t ip6_addr;
    fal_mac_addr_t mac_addr;
    a_uint32_t intf_id;
    a_uint32_t lb_num;  // total 3 bits for load balance and highest
                        // bit for load balance enable or not
    a_uint32_t expect_vid; // change the frame to expect vid
    fal_port_t port_id;    // determin destination port num
    a_bool_t mirror_en;    // frame should be mirrored,
                           // and action must be FAL_MAC_FRWRD.
    a_bool_t counter_en;   // frame should be added to counter.
    a_uint32_t counter_id;
    a_uint32_t packet;     // packet field in counter
    a_uint32_t byte;       // byte field in counter
    a_bool_t pppoe_en;     // add or change pppoe header
    a_uint32_t pppoe_id;
    fal_fwd_cmd_t action;  // packets forwarding command
} fal_host_entry_t;

/* This struct defines the interface mac entry. */
typedef struct
{
    a_uint32_t entry_id;
    a_uint32_t vrf_id;
```

```
    a_uint16_t vid_low;
    a_uint16_t vid_high;      // vid range
    fal_mac_addr_t mac_addr;  // router mac address
    a_bool_t ip4_route;
    a_bool_t ip6_route;
} fal_intf_mac_entry_t;

/* This struct defines the ip wcmp entry. */
typedef struct
{
    a_uint32_t nh_nr;
    a_uint32_t nh_id[16];
} fal_ip_wcmp_t;

/* This struct defines the default route entry. */
typedef struct
{
    a_bool_t valid;
    a_uint32_t vrf_id;
    fal_addr_type_t ip_version; //0 for IPv4 and 1 for IPv6
    a_uint32_t droute_type; //0 for ARP and 1 for WCMP
    a_uint32_t index;  //arp entry index or means wcmp index
} fal_default_route_t;

/* This struct defines the host route entry. */
typedef struct
{
    a_bool_t valid;
    a_uint32_t vrf_id;
    a_uint32_t ip_version; //0 for IPv4 and 1 for IPv6
    union {
    fal_ip4_addr_t ip4_addr;
    fal_ip6_addr_t ip6_addr;
    }route_addr;
    a_uint32_t prefix_length;
} fal_host_route_t;

typedef struct
{
    fal_mac_addr_t mac_addr;
    fal_ip4_addr_t ip4_addr;
    a_uint32_t     vid;
    a_uint8_t      load_balance;
} fal_ip4_rfs_t;
```

```
typedef struct
{
    fal_mac_addr_t mac_addr;
    fal_ip6_addr_t ip6_addr;
    a_uint32_t     vid;
    a_uint8_t      load_balance;
} fal_ip6_rfs_t;

typedef enum
{
    FAL_DEFAULT_FLOW_FORWARD = 0,
    FAL_DEFAULT_FLOW_DROP,
    FAL_DEFAULT_FLOW_RDT_TO_CPU,
    FAL_DEFAULT_FLOW_ADMIT_ALL,
} fal_default_flow_cmd_t;
```

2.7.3 Function documentation

2.7.3.1 fal_ip_host_add

Definition	Add one host entry to one particular device.	
Prototype	`sw_error_t fal_ip_host_add (`	
	`  a_uint32_t dev_id,`	Device ID
	`  fal_host_entry_t *host_entry`	Host entry
	`)`	
Description	For AR8337N, the intf_id field in host_entry means VLAN ID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.2 fal_ip_host_del

Definition	Delete one host entry from one particular device.	
Prototype	`sw_error_t fal_ip_host_del (`	
	`  a_uint32_t dev_id,`	Device ID
	`  a_uint32_t del_mode,`	Delete operation mode
	`  fal_host_entry_t *host_entry`	Host entry
	`)`	

Description	For AR8337N, the intf_id field in host_entry means VLAN ID. For del_mode, below options are supported: #define FAL_IP_ENTRY_INTF_EN 0x2 #define FAL_IP_ENTRY_PORT_EN 0x4 #define FAL_IP_ENTRY_STATUS_EN 0x8 #define FAL_IP_ENTRY_IPADDR_EN 0x10
Return Value	Returns SW_OK on success and sw_error_t on failure.

2.7.3.3 fal_ip_host_get

Definition	Get one host entry from one particular device.	
Prototype	`sw_error_t fal_ip_host_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t get_mode,`	Get operation mode
	`fal_host_entry_t *host_entry`	Host entry
	`)`	
Description	For AR8337N, the intf_id field in host_entry means VLAN ID. For get_mode, only one option is supported: #define FAL_IP_ENTRY_IPADDR_EN 0x10	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.4 fal_ip_host_next

Definition	Next one host entry from one particular device.	
Prototype	sw_error_t fal_ip_host_next (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t next_mode,	Next operation mode
	fal_host_entry_t *host_entry	Host entry
	)	
Description	For AR8337N, the intf_id field in host_entry means VLAN ID. For next_mode, below options are supported: #define FAL_IP_ENTRY_INTF_EN 0x2 #define FAL_IP_ENTRY_PORT_EN 0x4 #define FAL_IP_ENTRY_STATUS_EN 0x8	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.5 fal_ip_host_counter_bind

Definition	Bind one counter entry to one host entry on one particular device.	
Prototype	`sw_error_t fal_ip_host_counter_bind (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t entry_id`	Host entry ID
	`a_uint32_t cnt_id,`	Counter entry ID

	a_bool_t enable	A_TRUE means bind, A_FALSE means unbind
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.6 fal_ip_host_pppoe_bind

Definition	Bind one PPPoE session entry to one host entry on one particular device.	
Prototype	sw_error_t fal_ip_host_pppoe_bind (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t entry_id	Host entry ID
	a_uint32_t cnt_id,	PPPoE session table entry ID
	a_bool_t enable	A_TRUE means bind, A_FALSE means unbind
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.7 fal_ip_pt_arp_learn_get

Definition	Get ARP packets type to learn on one particular port.	
Prototype	sw_error_t fal_ip_pt_arp_learn_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t *flags	ARP type flags
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.8 fal_ip_pt_arp_learn_set

Definition	Set ARP packets type to learn on one particular port.	
Prototype	sw_error_t fal_ip_pt_arp_learn_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t flags	ARP type flags
	)	
Description	Supported ARP type flags are FAL_ARP_LEARN_REQ and FAL_ARP_LEARN_ACK.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.9 fal_ip_arp_learn_get

Definition	Get ARP packets type to learn on one particular device.

Prototype	sw_error_t fal_ip_arp_learn_get (	
	a_uint32_t dev_id,	Device ID
	fal_arp_learn_mode_t *mode	Learning mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.10 fal_ip_arp_learn_set

Definition	Set ARP packets type to learn on one particular device.	
Prototype	sw_error_t fal_ip_arp_learn_set (	
	a_uint32_t dev_id,	Device ID
	fal_arp_learn_mode_t mode	Learning mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.11 fal_ip_source_guard_get

Definition	Get IP packets source guarding mode on one particular port.	
Prototype	sw_error_t fal_ip_source_guard_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_source_guard_mode_t *mode	Source guarding mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.12 fal_ip_source_guard_set

Definition	Set IP packets source guarding mode on one particular port.	
Prototype	sw_error_t fal_ip_source_guard_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_source_guard_mode_t mode	Source guarding mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.13 fal_ip_arp_guard_get

Definition	Get ARP packets source guarding mode on one particular port.	
Prototype	sw_error_t fal_ip_arp_guard_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	fal_source_guard_mode_t *mode	Source guarding mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.14 fal_ip_arp_guard_set

Definition	Set ARP packets source guarding mode on one particular port.	
Prototype	sw_error_t fal_ip_arp_guard_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_source_guard_mode_t mode	Source guarding mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.15 fal_ip_router_status_get

Definition	Get IP unicast routing status on one particular device.	
Prototype	sw_error_t fal_ip_router_status_get (	
	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.16 fal_ip_router_status_set

Definition	Set IP unicast routing status on one particular device.	
Prototype	sw_error_t fal_ip_router_status_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.17 fal_ip_intf_entry_add

Definition	Add one interface entry to one particular device.	
Prototype	sw_error_t fal_ip_intf_entry_add (	
	a_uint32_t dev_id,	Device ID
	fal_intf_mac_entry_t *entry	Interface entry
	)	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.7.3.18 fal_ip_intf_entry_del

Definition	Delete one interface entry from one particular device.	
Prototype	`sw_error_t fal_ip_intf_entry_del (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t del_mode,`	Delete operation mode
	`fal_intf_mac_entry_t *entry`	Interface entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.19 fal_ip_intf_entry_next

Definition	Next one interface entry from one particular device.	
Prototype	`sw_error_t fal_ip_intf_entry_next (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t next_mode,`	Next operation mode
	`fal_intf_mac_entry_t *entry`	Interface entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.20 fal_ip_unk_source_cmd_get

Definition	Get unknown source IP packets forwarding command on one particular device.	
Prototype	`sw_error_t fal_ip_unk_source_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t *cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.21 fal_ip_unk_source_cmd_set

Definition	Set unknown source IP packets forwarding command on one particular device.	
Prototype	`sw_error_t fal_ip_unk_source_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t cmd`	Forwarding command
	`)`	

Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_FRWRD, FAL_MAC_DROP and FAL_MAC_RDT_TO_CPU are supported.
Return Value	Returns SW_OK on success and sw_error_t on failure.

2.7.3.22 fal_arp_unk_source_cmd_get

Definition	Get unknown source ARP packets forwarding command on one particular device.	
Prototype	`sw_error_t fal_arp_unk_source_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t *cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.23 fal_arp_unk_source_cmd_set

Definition	Set unknown source ARP packets forwarding command on one particular device.	
Prototype	`sw_error_t fal_arp_unk_source_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t cmd`	Forwarding command
	`)`	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_FRWRD, FAL_MAC_DROP and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.24 fal_ip_age_time_get

Definition	Get IP host entry aging time on one particular device.	
Prototype	`sw_error_t fal_ip_age_time_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t *time`	Aging time
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.25 fal_ip_age_time_set

Definition	Set IP host entry aging time on one particular device.	
Prototype	`sw_error_t fal_ip_age_time_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t *time`	Aging time
	`)`	

Description	This operation will set dynamic entry aging time on a particular device. The unit of time is second. Real aging time will be returned in time field.
Return Value	Returns SW_OK on success and sw_error_t on failure.

2.7.3.26 fal_ip_wcmp_entry_get

Definition	Get IP WCMP table one particular device	
Prototype	`sw_error_t fal_ip_wcmp_entry_get (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t wcmp_id,`	WCMP entry ID
	`fal_ip_wcmp_t *wcmp`	WCMP entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.27 fal_ip_wcmp_entry_set

Definition	Set IP WCMP table one particular device	
Prototype	`sw_error_t fal_ip_wcmp_entry_set (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t wcmp_id,`	WCMP entry ID
	`fal_ip_wcmp_t *wcmp`	WCMP entry
	`)`	
Description	Hardware supports 0-15 hash values and 4 different host tables.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.28 fal_ip_wcmp_hash_mode_get

Definition	Get IP WCMP hash key mode.	
Prototype	`sw_error_t fal_ip_wcmp_hash_mode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t *hash_mode`	IP WCMP hash key flags
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.29 fal_ip_wcmp_hash_mode_set

Definition	Set IP WCMP hash key mode.	
Prototype	`sw_error_t fal_ip_wcmp_hash_mode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t *hash_mode`	IP WCMP hash key flags
	`)`	

Description	For hash_mode, below options can be supported: #define FAL_WCMP_HASH_KEY_SIP 0x1 #define FAL_WCMP_HASH_KEY_DIP 0x2 #define FAL_WCMP_HASH_KEY_SPORT 0x4 #define FAL_WCMP_HASH_KEY_DPORT 0x8
Return Value	Returns SW_OK on success and sw_error_t on failure.

2.7.3.30 fal_ip_vrf_base_addr_set

Definition	Set IP base address with VRF.	
Prototype	`sw_error_t fal_ip_vrf_base_addr_set (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	VRF ID
	`fal_ip4_addr_t addr`	IP4 address
	`)`	
Description	Hardware supports 8 base address with different VRF.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.31 fal_ip_vrf_base_addr_get

Definition	Get IP base address with VRF.	
Prototype	`sw_error_t fal_ip_vrf_base_addr_get (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	VRF ID
	`fal_ip4_addr_t *addr`	IP4 address
	`)`	
Description	Hardware supports 8 base address with different VRF.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.32 fal_ip_vrf_base_mask_set

Definition	Set IP base address with VRF.	
Prototype	`sw_error_t fal_ip_vrf_base_mask_set (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	VRF ID
	`fal_ip4_addr_t addr`	IP4 mask address
	`)`	
Description	Hardware supports 8 base mask address with different VRF.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.33 fal_ip_vrf_base_mask_get

Definition	Get IP base address with VRF.

Prototype	sw_error_t fal_ip_vrf_base_mask_get (	
	a_uint32_t dev_id,	Device ID
	uint32_t vrf_id,	VRF ID
	fal_ip4_addr_t *addr	IP4 mask address
	)	
Description	Hardware supports 8 base mask address with different VRF.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.34 fal_ip_default_route_set

Definition	Set default routing entry.	
Prototype	sw_error_t fal_ip_default_route_set (	
	a_uint32_t dev_id,	Device ID
	uint32_t droute_id,	Entry ID
	fal_default_route_t *entry	Default route
	)	
Description	Hardware supports each 8 entry for IPv4 and IPv6.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.35 fal_ip_default_route_get

Definition	Get default routing entry.	
Prototype	sw_error_t fal_ip_default_route_get (	
	a_uint32_t dev_id,	Device ID
	uint32_t droute_id,	Entry ID
	fal_default_route_t *entry	Default route
	)	
Description	Hardware supports each 8 entry for IPv4 and IPv6.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.36 fal_ip_host_route_set

Definition	Set host routing entry.	
Prototype	sw_error_t fal_ip_host_route_set (	
	a_uint32_t dev_id,	Device ID
	uint32_t hroute_id,	Entry ID
	fal_host_route_t *entry	Host route
	)	
Description	Hardware supports each 16 entry for IPv4 and IPv6.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.37 fal_ip_host_route_get

Definition	Get host routing entry.	
Prototype	sw_error_t fal_ip_host_route_get (	
	a_uint32_t dev_id,	Device ID
	uint32_t hroute_id,	Entry ID
	fal_host_route_t *entry	Host route
	)	
Description	Hardware supports each 16 entry for IPv4 and IPv6.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.38 fal_ip_wcmp_entry_set

Definition	Set WCMP entry.	
Prototype	sw_error_t fal_ip_wcmp_entry_set (	
	a_uint32_t dev_id,	Device ID
	uint32_t wcmp_id,	WCMP ID
	fal_ip_wcmp_t *wcmp	WCMP
	)	
Description	Hardware supports 4 WCMP.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.39 fal_ip_wcmp_entry_get

Definition	Set WCMP entry.	
Prototype	sw_error_t fal_ip_wcmp_entry_get (	
	a_uint32_t dev_id,	Device ID
	uint32_t wcmp_id,	WCMP ID
	fal_ip_wcmp_t *wcmp	WCMP
	)	
Description	Hardware supports 4 WCMP.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.40 fal_ip_rfs_ip4_rule_set

Definition	Set IP4 RFS.	
Prototype	sw_error_t fal_ip_rfs_ip4_rule_set (	
	a_uint32_t dev_id,	Device ID
	fal_ip4_rfs_t *rfs	RFS
	)	
Description	Set IPv4 load balance.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.41 fal_ip_rfs_ip4_rule_del

Definition	Delete IP4 RFS.	
Prototype	`sw_error_t fal_ip_rfs_ip4_rule_del (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_ip4_rfs_t *rfs`	RFS
	`)`	
Description	Delete IPv4 load balance.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.42 fal_ip_rfs_ip6_rule_set

Definition	Set IP6 RFS.	
Prototype	`sw_error_t fal_ip_rfs_ip6_rule_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_ip6_rfs_t *rfs`	RFS
	`)`	
Description	Set IPv6 load balance.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.43 fal_ip_rfs_ip6_rule_del

Definition	Delete IP6 RFS.	
Prototype	`sw_error_t fal_ip_rfs_ip6_rule_del (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_ip6_rfs_t *rfs`	RFS
	`)`	
Description	Delete IPv6 load balance.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.44 fal_default_flow_cmd_set

Definition	Set flow command.	
Prototype	`sw_error_t fal_default_flow_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	VRF ID
	`fal_flow_type_t type`	Type
	`fal_default_flow_cmd_t cmd`	Command
	`)`	
Description	Below type can be supported: ▪ FAL_FLOW_LAN_TO_LAN = 0 ▪ FAL_FLOW_WAN_TO_LAN ▪ FAL_FLOW_LAN_TO_WAN ▪ FAL_FLOW_WAN_TO_WAN	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.45 fal_default_rt_flow_cmd_get

Definition	Get RT flow command.	
Prototype	`sw_error_t fal_default_rt_flow_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	RFS
	`fal_flow_type_t *type`	Type
	`fal_default_flow_cmd_t *cmd`	Command
Description	)	
Description	Below type can be supported: ▪ FAL_FLOW_LAN_TO_LAN = 0 ▪ FAL_FLOW_WAN_TO_LAN ▪ FAL_FLOW_LAN_TO_WAN ▪ FAL_FLOW_WAN_TO_WAN	

2.7.3.46 fal_default_rt_flow_cmd_set

Definition	Set RT flow command.	
Prototype	`sw_error_t fal_default_rt_flow_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	VRF ID
	`fal_flow_type_t type`	Type
	`fal_default_flow_cmd_t cmd`	Command
	)	
Description	Below type can be supported: ▪ FAL_FLOW_LAN_TO_LAN = 0 ▪ FAL_FLOW_WAN_TO_LAN ▪ FAL_FLOW_LAN_TO_WAN ▪ FAL_FLOW_WAN_TO_WAN	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.7.3.47 fal_default_rt_flow_cmd_get

Definition	Get RT flow command.	
Prototype	`sw_error_t fal_default_rt_flow_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`uint32_t vrf_id,`	RFS
	`fal_flow_type_t *type`	Type
	`fal_default_flow_cmd_t *cmd`	Command
Description	)	
Description	Below type can be supported: ▪ FAL_FLOW_LAN_TO_LAN = 0 ▪ FAL_FLOW_WAN_TO_LAN ▪ FAL_FLOW_LAN_TO_WAN ▪ FAL_FLOW_WAN_TO_WAN	

2.8 FAL_LEAKY

2.8.1 Enumeration type documentation

2.8.1.1 enum fal_leaky_ctrl_mode_t

This enum defines the leaky control mode.

Enumeration values:

FAL_LEAKY_PORT_CTRL control leaky through port which packets received

FAL_LEAKY_FDB_CTRL control leaky through FDB entry

2.8.2 Function documentation

2.8.2.1 fal_mc_leaky_mode_get

Definition	Get multicast packets leaky control mode on a particular device.	
Prototype	`sw_error_t fal_mc_leaky_mode_get(`	
	`  a_uint32_t device_id,`	Device ID
	`Fal_leaky_ctrl_mode_t ctrl_mode,`	Leaky control mode
	`)`	
Return Value	SW_OK or error code	

2.8.2.2 fal_mc_leaky_mode_set

Definition	Set multicast packets leaky control mode on a particular device.	
Prototype	`sw_error_t fal_mc_leaky_mode_set(`	
	`  a_uint32_t device_id,`	Device ID
	`Fal_leaky_ctrl_mode_t ctrl_mode,`	Leaky control mode
	`)`	
Return Value	SW_OK or error code	

2.8.2.3 fal_uc_leaky_mode_get

Definition	Get unicast packets leaky control mode on a particular device.	
Prototype	`sw_error_t fal_uc_leaky_mode_get(`	
	`  a_uint32_t device_id,`	Device ID
	`Fal_leaky_ctrl_mode_t ctrl_mode,`	Leaky control mode
	`)`	
Return Value	SW_OK or error code	

2.8.2.4 fal_uc_leaky_mode_set

Definition	Set unicast packets leaky control mode on a particular device.	
Prototype	`sw_error_t fal_uuc_leaky_mode_set(`	
	`  a_uint32_t device_id,`	Device ID
	`Fal_leaky_ctrl_mode_t ctrl_mode,`	Leaky control mode
	`)`	
Return Value	SW_OK or error code	

2.8.2.5 fal_port_arp_leaky_get

Definition	Get ARP packets leaky control status on a particular port.	
Prototype	`sw_error_t fal_port_arp_leaky_get(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.8.2.6 fal_port_arp_leaky_set

Definition	Set ARP packets leaky control status on a particular port.	
Prototype	`sw_error_t fal_port_arp_leaky_set(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.8.2.7 fal_port_mc_leaky_get

Definition	Get multicast packets leaky control status on a particular port.	
Prototype	`sw_error_t fal_port_mc_leaky_get(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.8.2.8 fal_port_mc_leaky_set

Definition	Set multicast packets leaky control status on a particular port.	
Prototype	`sw_error_t fal_port_mc_leaky_set(`	

	a_uint32_t device_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	SW_OK or error code	

2.8.2.9 fal_port_uc_leaky_get

Definition	Get unicast packets leaky control status on a particular port.	
Prototype	sw_error_t fal_port_uc_leaky_get(	
	a_uint32_t device_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	SW_OK or error code	

2.8.2.10 fal_port_uc_leaky_set

Definition	Set unicast packets leaky control status on a particular port.	
Prototype	sw_error_t fal_port_uc_leaky_set(	
	a_uint32_t device_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	SW_OK or error code	

2.9 FAL_LED

2.9.1 Enumeration type documentation

2.9.1.1 enum led_blink_freq_t

This enum defines the led control blink frequency mode.

Enumeration values:

LED_BLINK_2HZ, Led BLINK frequency is 2HZ

LED_BLINK_4HZ, Led BLINK frequency is 4HZ

LED_BLINK_8HZ, Led BLINK frequency is 8HZ

LED_BLINK_TXRX Frequency relates to speed, 1000M-8HZ,100M->4HZ,10M->2HZ,Others->4HZ

2.9.1.2 enum led_pattern_mode_t

This enum defines the led control pattern mode.

Enumeration values:

LED_ALWAYS_OFF, Led mode is off.

LED_ALWAYS_ON, Led mode is off.

LED_ALWAYS_BLINK, Led mode is blink.

2.9.1.3 enum led_pattern_group_t

This enum defines the led group.

Enumeration values:

LED_LAN_PORT_GROUP control lan ports

LED_WAN_PORT_GROUP control wan ports

LED_MAC_PORT_GROUP control mac ports

2.9.1.4 fal_led_ctrl_pattern_get

Definition	Get LED control pattern on a particular device.	
Prototype	`sw_error_t fal_led_ctrl_pattern_get(`	
	`  a_uint32_t device_id,`	Device ID
	`led_pattern_group_t group,`	Pattern group, LAN or WAN
	`led_pattern_id_t id`	Pattern ID
	`Led_ctrl_pattern_t * pattern`	LED control pattern
	`)`	
Return Value	SW_OK or error code	

2.9.1.5 fal_led_ctrl_pattern_set

Definition	Set LED control pattern on a particular device.	
Prototype	`sw_error_t fal_led_ctrl_pattern_set(`	
	`  a_uint32_t device_id,`	Device ID
	`led_pattern_group_t group,`	Pattern group, LAN or WAN
	`led_pattern_id_t id`	Pattern ID
	`Led_ctrl_pattern_t * pattern`	LED control pattern
	`)`	
Return Value	SW_OK or error code	

2.10 FAL_MIB

2.10.1 Structure documentation

2.10.1.1 struct fal_mib_info_t

This structure defines the MIB information.

```
typedef struct
{
        a_uint32_t RxBroad;
        a_uint32_t RxPause;
        a_uint32_t RxMulti;
        a_uint32_t RxFcsErr;
        a_uint32_t RxAllignErr;
        a_uint32_t RxRunt;
        a_uint32_t RxFragment;
        a_uint32_t Rx64Byte;
        a_uint32_t Rx128Byte;
        a_uint32_t Rx256Byte;
        a_uint32_t Rx512Byte;
        a_uint32_t Rx1024Byte;
        a_uint32_t Rx1518Byte;
        a_uint32_t RxMaxByte;
        a_uint32_t RxTooLong;
        a_uint32_t RxGoodByte_lo; /**<  low 32 bits of RxGoodByte statistc item */
        a_uint32_t RxGoodByte_hi; /**<  high 32 bits of RxGoodByte statistc item*/
        a_uint32_t RxBadByte_lo;  /**<  low 32 bits of RxBadByte statistc item */
        a_uint32_t RxBadByte_hi;  /**<  high 32 bits of RxBadByte statistc item */
        a_uint32_t RxOverFlow;
        a_uint32_t Filtered;
        a_uint32_t TxBroad;
        a_uint32_t TxPause;
        a_uint32_t TxMulti;
        a_uint32_t TxUnderRun;
        a_uint32_t Tx64Byte;
        a_uint32_t Tx128Byte;
        a_uint32_t Tx256Byte;
        a_uint32_t Tx512Byte;
        a_uint32_t Tx1024Byte;
        a_uint32_t Tx1518Byte;
        a_uint32_t TxMaxByte;
        a_uint32_t TxOverSize;
        a_uint32_t TxByte_lo;        /**<  low 32 bits of TxByte statistc item */
```

```
        a_uint32_t TxByte_hi;        /**<  high 32 bits of TxByte statistc item */
        a_uint32_t TxCollision;
        a_uint32_t TxAbortCol;
        a_uint32_t TxMultiCol;
        a_uint32_t TxSingalCol;
        a_uint32_t TxExcDefer;
        a_uint32_t TxDefer;
        a_uint32_t TxLateCol;
        a_uint32_t RxUniCast;
        a_uint32_t TxUniCast;
} fal_mib_info_t;
```

Total 41 MIB counters are supported.

2.10.2 Function documentation

2.10.2.1 fal_get_mib_info

Definition	Get MIB information on a particular port.	
Prototype	`sw_error_t fal_get_mib_info(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`Fal_mib_info_t mib_info`	MIB information
	`)`	
Return Value	SW_OK or error code	

2.10.2.2 fal_mib_status_get

Definition	Get MIB status on a particular device.	
Prototype	`sw_error_t fal_mib_status_get(`	
	`  a_uint32_t device_id,`	Device ID
	`a_bool_t * enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.10.2.3 fal_mib_status_set

Definition	Set MIB status on a particular device.	
Prototype	`sw_error_t fal_mib_status_set(`	
	`  a_uint32_t device_id,`	Device ID
	`a_bool_t enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.10.2.4 fal_mib_cpukeep_get

Definition	Get MIB CPU keep status on a particular device.	
Prototype	`sw_error_t fal_mib_cpukeep_get(`	
	`  a_uint32_t device_id,`	Device ID
	`a_bool_t * enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.10.2.5 fal_mib_cpukeep_set

Definition	Set MIB CPU keep status on a particular device.	
Prototype	`sw_error_t fal_mib_cpukeep_set(`	
	`  a_uint32_t device_id,`	Device ID
	`a_bool_t enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.11 FAL_MIRROR

2.11.1 Function documentation

2.11.1.1 fal_mirr_analysis_port_get

Definition	Get the mirror analysis port on a particular device.	
Prototype	`sw_error_t fal_mirr_analysis_port_get(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t * port_id,`	Port ID
	`)`	
Return Value	SW_OK or error code	

2.11.1.2 fal_mirr_analysis_port_set

Definition	Set the mirror analysis port on a particular device.	
Prototype	`sw_error_t fal_mirr_analysis_port_set(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`)`	
Description	This analysis port works for both ingress and egress mirror.	
Return Value	SW_OK or error code	

2.11.1.3 fal_mirr_port_eg_get

Definition	Get egress mirror status on a particular port.	
Prototype	`sw_error_t fal_mirr_port_eg_get(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t * enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.11.1.4 fal_mirr_port_eg_set

Definition	Set egress mirror status on a particular port.	
Prototype	`sw_error_t fal_mirr_port_eg_set(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.11.1.5 fal_mirr_port_in_get

Definition	Get ingress mirror status on a particular port.	
Prototype	`sw_error_t fal_mirr_port_in_get(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t * enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.11.1.6 fal_mirr_port_in_set

Definition	Set ingress mirror status on a particular port	
Prototype	`sw_error_t fal_mirr_port_in_set(`	
	`  a_uint32_t device_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable,`	TRUE or A_FALSE
	`)`	
Return Value	SW_OK or error code	

2.12 FAL_MISC

2.12.1 Struct documentation

```
typedef struct
{
    a_uint32_t entry_id;         // pppoe session table entry index
    a_uint32_t session_id;       // pppoe session id
    a_bool_t   multi_session;    // multicast match
    a_bool_t   uni_session;      // unicast match
} fal_pppoe_session_t;
```

2.12.2 Function documentation

2.12.2.1 fal_arp_status_get

Definition	Get ARP packets hardware acknowledgement status on a particular device.	
Prototype	`sw_error_t fal_arp_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.2 fal_arp_status_set

Definition	Set ARP packets hardware acknowledgement status on a particular device.	
Prototype	`sw_error_t fal_arp_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	For AR8337N, this API is not supported. Use fal_port_arp_req_status_set and fal_port_arp_ack_status_set.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.3 fal_port_arp_req_status_get

Definition	Get ARP REQ packets hardware acknowledgement status on a particular port.	
Prototype	`sw_error_t fal_port_arp_req_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID

	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.4 fal_port_arp_req_status_set

Definition	Set ARP REQ packets hardware acknowledgement status on a particular port.	
Prototype	sw_error_t fal_port_arp_req_status_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.5 fal_port_arp_ack_status_get

Definition	Get ARP ACK packets hardware acknowledgement status on a particular port.	
Prototype	sw_error_t fal_port_arp_ack_status_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.6 fal_port_arp_ack_status_set

Definition	Set ARP ACK packets hardware acknowledgement status on a particular port.	
Prototype	sw_error_t fal_port_arp_ack_status_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.7 fal_arp_cmd_get

Definition	Get ARP packets forwarding command on a particular device.	
Prototype	sw_error_t fal_arp_cmd_get (	
	a_uint32_t dev_id,	Device ID

	fal_fwd_cmd_t *cmd	Forwarding command
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.8 fal_arp_cmd_set

Definition	Set ARP packets forwarding command on a particular device.	
Prototype	sw_error_t fal_arp_cmd_set (	
	a_uint32_t dev_id,	Device ID
	fal_fwd_cmd_t cmd	Forwarding command
	)	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_CPY_TO_CPU, FAL_MAC_RDT_TO_CPU and FAL_MAC_FRWRD are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.9 fal_frame_max_size_get

Definition	Get max frame size which can receive on a particular device.	
Prototype	sw_error_t fal_frame_max_size_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t *size	Packet size
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.10 fal_frame_max_size_set

Definition	Set max frame size which can receive on a particular device.	
Prototype	sw_error_t fal_frame_max_size_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t size	Packet size
	)	
Description	Max frame size can be received and transmitted by MAC. If a packet's size larger than max frame size, it should be dropped by MAC. The value is for normal packet, it should be added 4 by MAC if support VLAN, added 8 for double VLAN.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.11 fal_bc_to_cpu_port_get

Definition	Get status of braodcast packets broadcasting to CPU on a particular device.	
Prototype	sw_error_t fal_bc_to_cpu_port_get (	
	a_uint32_t dev_id,	Device ID

	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.12 fal_bc_to_cpu_port_set

Definition	Set status of broadcast packets broadcasting to CPU on a particular device.	
Prototype	sw_error_t fal_bc_to_cpu_port_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	For AR8337N, this API is not supported. Use fal_port_bc_filter_set.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.13 fal_port_bc_filter_get

Definition	Get flooding status of broadcast packets on a particular port.	
Prototype	sw_error_t fal_port_bc_filter_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.14 fal_port_bc_filter_set

Definition	Set flooding status of broadcast packets on a particular port.	
Prototype	sw_error_t fal_port_bc_filter_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If enable broadcast packets filter on one port, then broadcast packets can't flood out from this port.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.15 fal_port_unk_mc_filter_get

Definition	Get flooding status of unknown multicast packets on a particular port.	
Prototype	sw_error_t fal_port_unk_mc_filter_get (	

	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.16 fal_port_unk_mc_filter_set

Definition	Set flooding status of unknown multicast packets on a particular port.	
Prototype	sw_error_t fal_port_unk_mc_filter_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If enable unknown multicast packets filter on one port, then unknown multicast packets can't flood out from this port.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.17 fal_port_unk_uc_filter_get

Definition	Get flooding status of unknown unicast packets on a particular port.	
Prototype	sw_error_t fal_port_unk_uc_filter_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.18 fal_port_unk_uc_filter_set

Definition	Set flooding status of unknown unicast packets on a particular port.	
Prototype	sw_error_t fal_port_unk_uc_filter_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If enable unknown unicast packets filter on one port, then unknown unicast packets can't flood out from this port.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.19 fal_port_unk_sa_cmd_get

Definition	Get forwarding command for packets which source address is unknown on a particular port.	
Prototype	`sw_error_t fal_port_unk_sa_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID
	`fal_fwd_cmd_t *cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.20 fal_port_unk_sa_cmd_set

Definition	Set forwarding command for packets which source address is unknown on a particular port.	
Prototype	`sw_error_t fal_port_unk_sa_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID
	`fal_fwd_cmd_t cmd`	Forwarding command
	`)`	
Description	Particular device can support parts of forwarding commands. For AR8337N, this API is not supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.21 fal_cpu_port_status_get

Definition	Get CPU port status on a particular device**.**	
Prototype	`sw_error_t fal_cpu_port_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.22 fal_cpu_port_status_set

Definition	Set CPU port status on a particular device.	
Prototype	`sw_error_t fal_cpu_port_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If CPU port status is enabled, CPU is connected to switch.	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.12.2.23 fal_cpu_vid_en_get

Definition	Get to CPU VID enable status on a particular device.	
Prototype	`sw_error_t fal_cpu_vid_en_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.24 fal_cpu_vid_en_set

Definition	Set to CPU VID enable status on a particular device.	
Prototype	`sw_error_t fal_cpu_vid_en_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If CPU VID status is enabled, internal VID will be used to replace the packet original VID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.25 fal_eapol_status_get

Definition	Get EAPOL packets hardware acknowledgement on a particular port.	
Prototype	`sw_error_t fal_eapol_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.26 fal_eapol_status_set

Definition	Set EAPOL packets hardware acknowledgement on a particular port.	
Prototype	`sw_error_t fal_eapol_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.12.2.27 fal_eapol_cmd_get

Definition	Get EAPOL packets forwarding command on a particular device.	
Prototype	`sw_error_t fal_eapol_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fwd_cmd_t *cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.28 fal_eapol_cmd_set

Definition	Set eapol packets forwarding command on a particular device.	
Prototype	`sw_error_t fal_eapol_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fwd_cmd_t cmd`	Forwarding command
	`)`	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_CPY_TO_CPU and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.29 fal_port_dhcp_get

Definition	Get DHCP packets hardware acknowledgement status on a particular port.	
Prototype	`sw_error_t fal_port_dhcp_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.30 fal_port_dhcp_set

Definition	Set DHCP packets hardware acknowledgement status on a particular port.	
Prototype	`sw_error_t fal_port_dhcp_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE

	)	
Description	If port DHCP enabled, DHCP packets on this port shall be redirect to CPU.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.31 fal_ripv1_status_get

Definition	Get RIPV1 packets hardware acknowledgement on a particular device.	
Prototype	sw_error_t fal_ripv1_status_get (	
	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.32 fal_ripv1_status_set

Definition	Set RIPV1 packets hardware acknowledgement on a particular device.	
Prototype	sw_error_t fal_ripv1_status_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If RIPV1 status enabled, RIP packets can copy to CPU.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.33 fal_pppoe_status_get

Definition	Get PPPoE packets hardware acknowledgement status on a particular device.	
Prototype	sw_error_t fal_pppoe_status_get(	
	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.34 fal_pppoe_status_set

Definition	Set PPPoE packets hardware acknowledgement status on a particular device.	
Prototype	sw_error_t fal_pppoe_status_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	PPPoE packets forward ctrl refer to I fal_pppoe_cmd_set.	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.12.2.35 fal_pppoe_cmd_get

Definition	Get PPPoE packets forwarding command on a particular device.	
Prototype	`sw_error_t fal_pppoe_cmd_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t *cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.36 fal_pppoe_cmd_set

Definition	Set PPPoE packets forwarding command on a particular device.	
Prototype	`sw_error_t fal_pppoe_cmd_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t cmd`	Forwarding command
	`)`	
Description	Particular device can support parts of forwarding commands. For AR8337N, only FAL_MAC_FRWRD and FAL_MAC_RDT_TO_CPU are supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.37 fal_pppoe_session_add

Definition	Add a PPPoE session entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_add (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t session_id,`	PPPoE session ID
	`a_bool_t strip_hdr`	Strip or not strip PPPoE header
	`)`	
Description	For AR8337N, this API is not supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.38 fal_pppoe_session_del

Definition	Delete a PPPoE session entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_del (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t session_id`	PPPoE session ID
	`)`	
Description	For AR8337N, this API is not supported.	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.12.2.39 fal_pppoe_session_get

Definition	Get a PPPoE session entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t session_id,`	PPPoE session ID
	`a_bool_t *strip_hdr`	Strip or not strip PPPoE header
	`)`	
Description	For AR8337N, this API is not supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.40 fal_pppoe_session_table_add

Definition	Add a PPPoE session entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_table_add (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_pppoe_session_t session_tbl`	PPPoE session table
	`)`	
Description	For AR8337N, only Multicast packets will match in PPPoE session table. Once matched, PPPoE header of the packets will be removed.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.41 fal_pppoe_session_table_del

Definition	Delete a PPPoE session entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_table_del (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_pppoe_session_t session_tbl`	PPPoE session table
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.42 fal_pppoe_session_table_get

Definition	Get a PPPoE session entry from a particular device.	
Prototype	`sw_error_t fal_pppoe_session_table_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_pppoe_session_t session_tbl`	PPPoE session table
	`)`	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.12.2.43 fal_rtd_pppoe_en_get

Definition	Get routed PPPoE status on a particular device.	
Prototype	`sw_error_t fal_rtd_pppoe_en_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.44 fal_rtd_pppoe_en_set

Definition	Set routed PPPoE status on a particular device.	
Prototype	`sw_error_t fal_rtd_pppoe_en_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	When a packet is routed, and fal_ip_host_pppoe_bind is disabled, if routed PPPoE status is enabled, hardware will remove PPPoe header from the packet.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.45 fal_pppoe_session_id_get

Definition	Get a PPPoE session ID entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_id_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t index,`	PPPoE session ID table index
	`a_uint32_t *id`	PPPoE session ID
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.12.2.46 fal_pppoe_session_id_set

Definition	Set a PPPoE session ID entry to a particular device.	
Prototype	`sw_error_t fal_pppoe_session_id_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t index,`	PPPoE session ID table index
	`a_uint32_t id`	PPPoE session ID
	`)`	

Description	This API configures the PPPoE session ID table, which includes pppoe_index and pppoe_session_id field. fal_ip_host_pppoe_bind binds a ARP entry to one of the entry in pppoe session ID table by pppoe_index. When a packet is routed, and fal_ip_host_pppoe_bind is enabled, hardware will add or change PPPoE header by pppoe_session_id binded with corresponding ARP entry.
Return Value	Returns SW_OK on success and sw_error_t on failure.

2.13 FAL_NAT

2.13.1 Typedef documentation

```
/* NAT entry attribute flags */
#define FAL_NAT_ENTRY_PROTOCOL_TCP          0x1
#define FAL_NAT_ENTRY_PROTOCOL_UDP          0x2
#define FAL_NAT_ENTRY_PROTOCOL_PPTP         0x4
#define FAL_NAT_ENTRY_PROTOCOL_ANY          0x8
#define FAL_NAT_ENTRY_TRANS_IPADDR_INDEX    0x10
#define FAL_NAT_ENTRY_PORT_CHECK            0x20
#define FAL_NAT_HASH_KEY_PORT               0x40
#define FAL_NAT_HASH_KEY_IPADDR             0x80

/* NAT entry operation flags */
#define FAL_NAT_ENTRY_ID_EN                  0x1
#define FAL_NAT_ENTRY_SRC_IPADDR_EN          0x2
#define FAL_NAT_ENTRY_TRANS_IPADDR_EN        0x4
#define FAL_NAT_ENTRY_KEY_EN                 0x8
#define FAL_NAT_ENTRY_PUBLIC_IP_EN           0x10
#define FAL_NAT_ENTRY_SOURCE_IP_EN           0x20
#define FAL_NAT_ENTRY_AGE_EN                 0x40
```

2.13.2 Enumeration type documentation

```
/* NAPT operation mode */
typedef enum
{
    FAL_NAPT_FULL_CONE = 0,
    FAL_NAPT_STRICT_CONE,
    FAL_NAPT_PORT_STRICT,
    FAL_NAPT_SYNMETRIC,
} fal_napt_mode_t;
```

2.13.3 Struct documentation

```
/* This struct defines the NAPT entry. */
typedef struct
{
    a_uint32_t     entry_id;
    a_uint32_t     flags;
    a_uint32_t     status;
    fal_ip4_addr_t src_addr;
    fal_ip4_addr_t dst_addr;
    a_uint16_t     src_port;
    a_uint16_t     dst_port;
    fal_ip4_addr_t trans_addr;
    a_uint16_t     trans_port;
    a_uint16_t     rsv;
    a_bool_t       mirror_en;
    a_bool_t       counter_en;
    a_uint32_t     counter_id;
    a_uint32_t     ingress_packet;
    a_uint32_t     ingress_byte;
    a_uint32_t     egress_packet;
    a_uint32_t     egress_byte;
    fal_fwd_cmd_t  action;
    a_uint32_t     load_balance;
    a_uint32_t     flow_cookie;
    a_uint32_t     vrf_id;
    a_uint32_t     aging_sync;
    a_bool_t       priority_en;
    a_uint32_t     priority_val;
} fal_napt_entry_t;

/* This struct defines the NAT entry. */
typedef struct
{
    a_uint32_t     entry_id;
    a_uint32_t     flags;
    a_uint32_t     status;
    fal_ip4_addr_t src_addr;
    fal_ip4_addr_t trans_addr;
    a_uint16_t     port_num;
    a_uint16_t     port_range;
    a_uint32_t     slct_idx;
    a_bool_t       mirror_en;
    a_bool_t       counter_en;
    a_uint32_t     counter_id;
    a_uint32_t     ingress_packet;
```

```
    a_uint32_t     ingress_byte;
    a_uint32_t     egress_packet;
    a_uint32_t     egress_byte;
    fal_fwd_cmd_t  action;
    a_uint32_t     vrf_id;
} fal_nat_entry_t;

/* This struct defines the public address. */
typedef struct
{
    a_uint32_t     entry_id;
    fal_ip4_addr_t pub_addr;
} fal_nat_pub_addr_t;

typedef struct
{
    a_uint32_t     proto;
    fal_ip4_addr_t src_addr;
    fal_ip4_addr_t dst_addr;
    a_uint16_t     src_port;
    a_uint16_t     dst_port;
    a_uint32_t     flow_cookie;
}  fal_flow_cookie_t;

typedef struct
{
    a_uint32_t     proto;  /*1 tcp; 2 udp*/
    fal_ip4_addr_t src_addr;
    fal_ip4_addr_t dst_addr;
    a_uint16_t     src_port;
    a_uint16_t     dst_port;
    a_uint8_t     load_balance;
} fal_flow_rfs_t;
```


2.13.4 Function documentation

2.13.4.1 fal_nat_add

Definition	Add one NAT entry to one particular device.	
Prototype	fal_nat_add(	
	a_uint32_t dev_id,	Device ID
	fal_nat_entry_t * nat_entry	NAT entry
	)	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.13.4.2 fal_nat_del

Definition	Delete one NAT entries from one particular device.	
Prototype	`fal_nat_del(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t del_mode,`	NAT entry delete operation mode
	`fal_nat_entry_t * nat_entry`	NAT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.3 fal_nat_get

Definition	Get one NAT entries from one particular device.	
Prototype	`fal_nat_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t get_mode,`	NAT entry get operation mode
	`fal_nat_entry_t * nat_entry`	NAT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.4 fal_nat_next

Definition	Get next NAT entries from one particular device.	
Prototype	`fal_nat_next(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t next_mode,`	NAT entry next operation mode
	`fal_nat_entry_t * nat_entry`	NAT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.5 fal_nat_counter_bind

Definition	Bind one counter entry to one NAT entry to one particular device.	
Prototype	`fal_nat_counter_bind(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t entry_id,`	NAT entry ID
	`a_uint32_t cnt_id,`	counter entry ID
	`a_bool_t enable`	TRUE or FALSE
	`)`	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.13.4.6 fal_napt_add

Definition	Add one NAPT entry to one particular device.	
Prototype	`fal_napt_add(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_napt_entry_t * napt_entry`	NAPT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.7 fal_napt_del

Definition	Delete one NAPT entries from one particular device.	
Prototype	`fal_napt_del(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t del_mode,`	NAPT entry delete operation mode
	`fal_napt_entry_t * napt_entry`	NAPT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.8 fal_napt_get

Definition	Get one NAPT entry from one particular device.	
Prototype	`fal_napt_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t get_mode,`	NAPT entry get operation mode
	`fal_napt_entry_t * napt_entry`	NAPT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.9 fal_napt_next

Definition	Get next NAPT entries from one particular device.	
Prototype	`fal_napt_next(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t next_mode,`	NAPT entry next operation mode
	`fal_napt_entry_t * napt_entry`	NAPT entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.10 fal_napt_counter_bind

Definition	Bind one counter entry to one NAPT entry to one particular device.	
Prototype	`fal_napt_counter_bind(`	
	`  a_uint32_t dev_id,`	Device ID
	`  a_uint32_t entry_id,`	NAPT entry ID
	`  a_uint32_t cnt_id,`	Counter entry ID
	`  a_bool_t enable`	TRUE or FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.11 fal_nat_status_set

Definition	Set working status of NAT engine on a particular device.	
Prototype	`fal_nat_status_set(`	
	`  a_uint32_t dev_id,`	Device ID
	`  a_bool_t enable`	TRUE or FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.12 fal_nat_status_get

Definition	Get working status of NAT engine on a particular device.	
Prototype	`fal_nat_status_get(`	
	`  a_uint32_t dev_id,`	Device ID
	`  a_bool_t * enable`	TRUE or FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.13 fal_nat_hash_mode_set

Definition	Set NAT hash mode on a particular device.	
Prototype	`fal_nat_hash_mode_set(`	
	`  a_uint32_t dev_id,`	Device ID
	`  a_uint32_t mode`	NAT hash mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.14 fal_nat_hash_mode_get

Definition	Get NAT hash mode on a particular device.	
Prototype	`fal_nat_hash_mode_get(`	
	`  a_uint32_t dev_id,`	Device ID

	a_uint32_t * mode	NAT hash mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.15 fal_napt_status_set

Definition	Set working status of NAPT engine on a particular device.	
Prototype	fal_napt_status_set(	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	TRUE or FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.16 fal_napt_status_get

Definition	Get working status of NAPT engine on a particular device.	
Prototype	fal_napt_status_get(	
	a_uint32_t dev_id,	Device ID
	fal_nat_entry_t * nat_entry	TRUE or FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.17 fal_napt_mode_set

Definition	Set working mode of NAPT engine on a particular device.	
Prototype	fal_napt_mode_set(	
	a_uint32_t dev_id,	Device ID
	fal_napt_mode_t mode	NAPT mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.18 fal_napt_mode_get

Definition	Get working mode of NAPT engine on a particular device.	
Prototype	fal_napt_mode_get(	
	a_uint32_t dev_id,	Device ID
	fal_napt_mode_t * mode	NAPT mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.19 fal_nat_prv_base_addr_set

Definition	Set IP4 private base address on a particular device.

Prototype	fal_nat_prv_base_addr_set(	
	a_uint32_t dev_id,	Device ID
	fal_ip4_addr_t addr	Private base address
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.20 fal_nat_prv_base_addr_get

Definition	Get IP4 private base address on a particular device.	
Prototype	fal_nat_prv_base_addr_get(	
	a_uint32_t dev_id,	Device ID
	fal_ip4_addr_t * addr	Private base address
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.21 fal_nat_prv_base_mask_set

Definition	Set IP4 private base address mask on a particular device.	
Prototype	fal_nat_prv_base_mask_set(	
	a_uint32_t dev_id,	Device ID
	fal_ip4_addr_t addr	Private base address mask
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.22 fal_nat_prv_base_mask_get

Definition	Get IP4 private base address mask on a particular device.	
Prototype	fal_nat_prv_base_mask_get(	
	a_uint32_t dev_id,	Device ID
	private base address mask	Private base address mask
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.23 fal_nat_prv_addr_mode_set

Definition	Set IP4 private base address mode on a particular device.	
Prototype	fal_nat_prv_addr_mode_set(	
	a_uint32_t dev_id,	Device ID
	a_bool_t map_en	Private base mapping mode

	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.24 fal_nat_prv_addr_mode_get

Definition	Get IP4 private base address mode on a particular device.	
Prototype	fal_nat_prv_addr_mode_get(	
	a_uint32_t dev_id,	Device ID
	a_bool_t * map_en	Private base mapping mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.25 fal_nat_pub_addr_add

Definition	Add one public address entry to one particular device.	
Prototype	fal_nat_pub_addr_add(	
	a_uint32_t dev_id,	Device ID
	fal_nat_pub_addr_t * entry	Public address entry parameter
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.26 fal_nat_pub_addr_del

Definition	Delete one public address entry from one particular device.	
Prototype	fal_nat_pub_addr_del(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t del_mode,	Delete operation mode
	fal_nat_pub_addr_t * entry	Private base address
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.27 fal_nat_pub_addr_next

Definition	Get the next public address entries from one particular device.	
Prototype	fal_nat_pub_addr_next(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t next_mode,	Next operation mode
	fal_nat_pub_addr_t * entry	Public address entry parameter
	)	

Return Value	Returns SW_OK on success and sw_error_t on failure.

2.13.4.28 fal_nat_unk_session_cmd_set

Definition	Set forwarding command for those packets miss NAT entries on a particular device.	
Prototype	`fal_nat_unk_session_cmd_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.29 fal_nat_unk_session_cmd_get

Definition	Get forwarding command for those packets miss NAT entries on a particular device.	
Prototype	`fal_nat_unk_session_cmd_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_fwd_cmd_t * cmd`	Forwarding command
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.30 fal_nat_global_set

Definition	Set HNAT helper status to one particular device.	
Prototype	`fal_nat_global_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_bool_t enable`	TRUE or FALSE
	`a_bool_t sync_cnt_enable`	Sync counter
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.31 fal_flow_add

Definition	Add one flow entry to one particular device.	
Prototype	`fal_flow_add(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_napt_entry_t * napt_entry`	Flow entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.32 fal_flow_del

Definition	Delete one flow entries from one particular device.	
Prototype	`fal_napt_del(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t del_mode,`	NAPT entry delete operation mode
	`fal_napt_entry_t * napt_entry`	Flow entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.33 fal_flow_get

Definition	Get one flow entry from one particular device.	
Prototype	`fal_napt_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t get_mode,`	NAPT entry get operation mode
	`fal_napt_entry_t * napt_entry`	Flow entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.34 fal_flow_next

Definition	Get next flow entries from one particular device.	
Prototype	`fal_flow_next(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t next_mode,`	NAPT entry next operation mode
	`fal_napt_entry_t * napt_entry`	Flow entry
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.35 fal_flow_cookie_set

Definition	Set flow cookie entries from one particular device.	
Prototype	`fal_flow_cookie_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_flow_cookie_t * flow_cookie`	Flow cookie
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.13.4.36 fal_flow_rfs_set

Definition	Set flow RFS entries from one particular device.	
Prototype	fal_flow_rfs_set(	
	a_uint32_t dev_id,	Device ID
	a_uint8_t action	Set or delete
	fal_flow_rfs_t * rfs	Flow RFS
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14 FAL_PORT_CTRL

```
// auto negotiation advtisement ability
#define FAL_PHY_ADV_10T_HD        0x01
#define FAL_PHY_ADV_10T_FD        0x02
#define FAL_PHY_ADV_100TX_HD      0x04
#define FAL_PHY_ADV_100TX_FD      0x08
#define FAL_PHY_ADV_1000T_FD      0x200
#define FAL_PHY_ADV_FE_SPEED_ALL (FAL_PHY_ADV_10T_HD | \
                                  FAL_PHY_ADV_10T_FD | \
                                  FAL_PHY_ADV_100TX_HD | \
                                  FAL_PHY_ADV_100TX_FD)
#define FAL_PHY_ADV_GE_SPEED_ALL (FAL_PHY_ADV_10T_HD | \
                                  FAL_PHY_ADV_10T_FD | \
                                  FAL_PHY_ADV_100TX_HD |\
                                  FAL_PHY_ADV_100TX_FD |\
                                  FAL_PHY_ADV_1000T_FD)
#define FAL_PHY_ADV_PAUSE         0x10
#define FAL_PHY_ADV_ASY_PAUSE     0x20
#define FAL_PHY_FE_ADV_ALL       (FAL_PHY_ADV_FE_SPEED_ALL | \
                                  FAL_PHY_ADV_PAUSE | \
                                  FAL_PHY_ADV_ASY_PAUSE)
#define FAL_PHY_GE_ADV_ALL       (FAL_PHY_ADV_GE_SPEED_ALL | \
                                  FAL_PHY_ADV_PAUSE | \
                                  FAL_PHY_ADV_ASY_PAUSE)
```

2.14.1 Enumeration type documentation

2.14.1.1 enum fal_port_duplex_t

This enum defines port duplex mode type.

Enumeration values:

FAL_HALF_DUPLEX half-duplex mode

FAL_FULL_DUPLEX full-duplex mode

2.14.1.2 enum fal_port_speed_t

This enum defines port speed type.

Enumeration values:

FAL_SPEED_10 10Mbps

FAL_SPEED_100 100Mbps

FAL_SPEED_1000 1000Mbps

FAL_SPEED_10000 1Gbps

2.14.1.3 enum fal_port_head_mode_t

This enum defines packet Atheros header mode type.

Enumeration values:

FAL_NO_HEADER_EN No header.

FAL_ONLY_MANAGE_FRAME_EN Only management with header, must be under 4 bytes header mode.

FAL_ALL_TYPE_FRAME_EN All frame with header

2.14.1.4 enum fal_cable_status_t

This enum defines CDT result of cable status.

Enumeration values:

FAL_CABLE_STATUS_NORMAL Cable connected to this port is normal.

FAL_CABLE_STATUS_SHORT Cable connected to this port is too short.

FAL_CABLE_STATUS_OPENED No cable is connected to this port.

FAL_CABLE_STATUS_INVALID The CDT test result is invalid.

FAL_CABLE_STATUS_BUTT end of enum definition

2.14.1.5 enum fal_port_mdix_mode_t

This enum defines crossover mode.

```
typedef enum {
   PHY_MDIX_AUTO = 0,
   /**< Auto MDI/MDIX */
   PHY_MDIX_MDI = 1,
```

```
    /**< Fixed MDI */
    PHY_MDIX_MDIX = 2
    /**< Fixed MDIX */
} fal_port_mdix_mode_t;
```

2.14.1.6 enum fal_port_mdix_status_t

This enum defines crossover status.

```
typedef enum {
    PHY_MDIX_STATUS_MDI = 0,
    /**< Fixed MDI */
    PHY_MDIX_STATUS_MDIX = 1
    /**< Fixed MDIX */
} fal_port_mdix_status_t;
```

2.14.1.7 enum fal_port_medium_t

This enum defines port medium type.

```
typedef enum {
    PHY_MEDIUM_COPPER = 0,
    /**< Copper */
    PHY_MEDIUM_FIBER = 1,
    /**< Fiber */
} fal_port_medium_t;
```

2.14.1.8 enum fal_port_fiber_mode_t

This enum defines port fiber mode.

```
typedef enum {
    PHY_FIBER_100FX = 0,
    /**< 100FX fiber mode */
    PHY_FIBER_1000BX = 1,
    /**< 1000BX fiber mode */
} fal_port_fiber_mode_t;
```

2.14.1.9 enum fal_port_interface_mode_t

This enum defines port interface mode

```
typedef enum {
    PHY_PSGMII_BASET = 0,
    /**< PSGMII mode */
    PHY_PSGMII_BX1000 = 1,
```

```
    /**< PSGMII BX1000 mode */
    PHY_PSGMII_FX100 = 2,
    /**< PSGMII FX100 mode */
    PHY_PSGMII_AMDET = 3,
    /**< PSGMII Auto mode */
    PHY_SGMII_BASET = 4,
    /**< SGMII mode */
} fal_port_interface_mode_t;
```

2.14.2 Function documentation

2.14.2.1 fal_port_autoneg_adv_get

Definition	Get auto negotiation advisement ability on a particular port.	
Prototype	`sw_error_t fal_port_autoneg_adv_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t *autoadv`	Auto negotiation advisement ability
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.2 fal_port_autoneg_adv_set

Definition	Set auto negotiation advisement ability on a particular port.	
Prototype	`sw_error_t fal_port_autoneg_adv_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t autoadv`	Auto negotiation advisement ability
	`)`	
Description	Auto negotiation advisement ability refers to Macros defined before Enumeration Type Documentation.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.3 fal_port_autoneg_enable

Definition	Enable auto negotiation status on a particular port.	
Prototype	`sw_error_t fal_port_autoneg_enable (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id`	Port ID
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.4 fal_port_autoneg_restart

Definition	Restart auto negotiation procedure on a particular port.	
Prototype	`sw_error_t fal_port_autoneg_restart (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id`	Port ID
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.5 fal_port_autoneg_status_get

Definition	Get auto negotiation status on a particular port.	
Prototype	`sw_error_t fal_port_autoneg_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *status`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.6 fal_port_duplex_get

Definition	Get duplex mode on a particular port.	
Prototype	`sw_error_t fal_port_duplex_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_duplex_t *pduplex`	Duplex mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.7 fal_port_duplex_set

Definition	Set duplex mode on a particular port.	
Prototype	`sw_error_t fal_port_duplex_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_duplex_t duplex`	Duplex mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.8 fal_port_speed_get

Definition	Get speed on a particular port.

Prototype	sw_error_t fal_port_speed_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_speed_t *pspeed	Speed
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.9 fal_port_speed_set

Definition	Set speed on a particular port.	
Prototype	sw_error_t fal_port_speed_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_speed_t speed	Speed
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.10 fal_port_link_status_get

Definition	Get link status on particular port.	
Prototype	sw_error_t fal_port_link_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *status	Up (A_TRUE) or down (A_FALSE)
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.11 fal_port_link_forcemode_get

Definition	Get link forcemode on particular port.	
Prototype	sw_error_t fal_port_link_forcemode_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.12 fal_port_link_forcemode_set

Definition	Set link forcemode on particular port.	
Prototype	sw_error_t fal_port_link_forcemode_set (	

	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If link forcemode is enabled, MAC can be configured by software, else MAC will autonegotiate with PHY, and following APIs will not take effect: fal_port_txmac_status_set fal_port_rxmac_status_set	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.13 fal_port_txmac_status_get

Definition	Get status of Tx MAC on particular port.	
Prototype	sw_error_t fal_port_txmac_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.14 fal_port_txmac_status_set

Definition	Set status of Tx MAC on particular port.	
Prototype	sw_error_t fal_port_txmac_status_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If fal_port_link_forcemode_set is set to disabled, this API together with fal_port_rxmac_status_set will not take effect.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.15 fal_port_rxmac_status_get

Definition	Get status of Rx MAC on particular port.	
Prototype	sw_error_t fal_port_rxmac_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.16 fal_port_rxmac_status_set

Definition	Set status of Rx MAC on particular port.	
Prototype	`sw_error_t fal_port_rxmac_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If fal_port_link_forcemode_set is set to disabled, this API with fal_port_txmac_status_set will not take effect.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.17 fal_port_flowctrl_forcemode_get

Definition	Get flow control force mode on a particular port.	
Prototype	`sw_error_t fal_port_flowctrl_forcemode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.18 fal_port_flowctrl_forcemode_set

Definition	Set flow control force mode on a particular port.	
Prototype	`sw_error_t fal_port_flowctrl_forcemode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If flowctrl forcemode is enabled, MAC can be configured by software, else MAC will autonegotiate flow ctrl configuration with PHY, and following APIs will not take effect: ▪ fal_port_flowctrl_set ▪ fal_port_txfc_status_set ▪ fal_port_rxfc_status_set	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.19 fal_port_flowctrl_get

Definition	Get flow control status on a particular port.	
Prototype	`sw_error_t fal_port_flowctrl_get (`	
	`a_uint32_t dev_id,`	Device ID

	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.20 fal_port_flowctrl_set

Definition	Set flow control status on a particular port.	
Prototype	sw_error_t fal_port_flowctrl_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If fal_port_flowctrl_forcemode_set is set to disabled, this API together with fal_port_rxfc_status_set and fal_port_txfc_status_set will not take effect.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.21 fal_port_txfc_status_get

Definition	Get status of Tx flow control on a particular port.	
Prototype	sw_error_t fal_port_txfc_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.22 fal_port_txfc_status_set

Definition	Set status of Tx flow control on a particular port.	
Prototype	sw_error_t fal_port_txfc_status_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If fal_port_flowctrl_forcemode_set is set to disabled, this API together with fal_port_flowctrl_set and fal_port_rxfc_status_set will not take effect.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.23 fal_port_rxfc_status_get

Definition	Get status of Rx flow control on a particular port.	
Prototype	`sw_error_t fal_port_rxfc_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.24 fal_port_rxfc_status_set

Definition	Set status of Rx flow control on a particular port.	
Prototype	`sw_error_t fal_port_rxfc_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	If fal_port_flowctrl_forcemode_set is set to disabled, this API together with fal_port_flowctrl_set and fal_port_txfc_status_set will not take effect.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.25 fal_port_bp_status_get

Definition	Get status of back pressure on a particular port.	
Prototype	`sw_error_t fal_port_bp_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.26 fal_port_bp_status_set

Definition	Set status of back pressure on a particular port.	
Prototype	`sw_error_t fal_port_bp_status_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.27 fal_header_type_get

Definition	Get status of Atheros header type value on a particular device.	
Prototype	sw_error_t fal_header_type_get (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable,	A_TRUE or A_FALSE
	a_uint32 *type	Header type
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.28 fal_header_type_set

Definition	Set status of Atheros header type value on a particular device.	
Prototype	sw_error_t fal_header_type_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable,	A_TRUE or A_FALSE
	a_uint32 type	Header type
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.29 fal_port_hdr_status_get

Definition	Get status of Atheros header packets parsed on a particular port.	
Prototype	sw_error_t fal_port_hdr_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.30 fal_port_hdr_status_set

Definition	Set status of Atheros header packets parsed on a particular port.	
Prototype	sw_error_t fal_port_hdr_status_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.31 fal_port_rxhdr_mode_get

Definition	Get status of Atheros header Rx packets parsed on a particular port.	
Prototype	`sw_error_t fal_port_rxhdr_mode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_header_mode_t *mode`	Header mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.32 fal_port_rxhdr_mode_set

Definition	Set status of Atheros header Rx packets parsed on a particular port.	
Prototype	`sw_error_t fal_port_rxhdr_mode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_header_mode_t mode`	Header mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.33 fal_port_txhdr_mode_get

Definition	Get status of Atheros header Tx packets parsed on a particular port.	
Prototype	`sw_error_t fal_port_txhdr_mode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_header_mode_t *mode`	Header mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.34 fal_port_txhdr_mode_set

Definition	Set status of Atheros header Tx packets parsed on a particular port.	
Prototype	`sw_error_t fal_port_txhdr_mode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_header_mode_t mode`	Header mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.35 fal_port_mac_loopback_get

Definition	Get loopback on a particular port.

Prototype	sw_error_t fal_port_mac_loopback_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.36 fal_port_mac_loopback_set

Definition	Set loopback on a particular port.	
Prototype	sw_error_t fal_port_mac_loopback_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.37 fal_port_cdt

Definition	Cable diagnostic test on a particular port.	
Prototype	sw_error_t fal_port_cdt (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t mdi_pair,	MDI pair ID
	fal_cable_status_t *cable_status,	Cable status
	a_uint32_t *cable_len	Cable length
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.38 fal_port_hibernate_get

Definition	Get hibernate status on a particular port.	
Prototype	sw_error_t fal_port_hibernate_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.39 fal_port_hibernate_set

Definition	Set hibernate status on a particular port.	
Prototype	sw_error_t fal_port_hibernate_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.40 fal_port_powersave_get

Definition	Get powersaving status on a particular port.	
Prototype	sw_error_t fal_port_powersave_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.41 fal_port_powersave_set

Definition	Set powersaving status on a particular port.	
Prototype	sw_error_t fal_port_powersave_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.42 fal_port_congestion_drop_set

Definition	Set congestion drop on a particular port.	
Prototype	sw_error_t fal_port_congestion_drop_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t queue_id,	Queue ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.43 fal_port_congestion_drop_get

Definition	Get congestion drop on a particular port.	
Prototype	`sw_error_t fal_port_congestion_drop_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t queue_id,`	Queue ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.44 fal_ring_flow_ctrl_thres_set

Definition	Set ring flow control threshold on a particular ring.	
Prototype	`sw_error_t fal_ring_flow_ctrl_thres_set (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t ring_id,`	Ring ID
	`a_uint8_t on_thres,`	On threshold
	`a_uint8_t off_thres`	Off threshold
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.45 fal_ring_flow_ctrl_thres_get

Definition	Get ring flow control threshold on a particular ring.	
Prototype	`sw_error_t fal_ring_flow_ctrl_thres_get (`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t ring_id,`	Ring ID
	`a_uint8_t *on_thres,`	On threshold
	`a_uint8_t *off_thres`	Off threshold
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.46 fal_port_8023az_set

Definition	Set 802.3az ability on a particular port.	
Prototype	`sw_error_t fal_port_8023az_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID

	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.47 fal_port_8023az_get

Definition	Get 802.3az ability on a particular port.	
Prototype	sw_error_t fal_port_8023az_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.48 fal_port_mdix_set

Definition	Set MDIX on a particular port.	
Prototype	sw_error_t fal_port_mdix_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_mdix_mode_t mode	Mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.49 fal_port_mdix_get

Definition	Get MDIX on a particular port.	
Prototype	sw_error_t fal_port_mdix_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_mdix_mode_t * mode	Mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.50 fal_port_mdix_status_get

Definition	Get current MDIX status on a particular port.	
Prototype	sw_error_t fal_port_mdix_status_get (	
	a_uint32_t dev_id,	Device ID

	fal_port_t port_id,	Port ID
	fal_port_mdix_mode_t * mode	Mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.51 fal_port_combo_prefer_medium_set

Definition	Set prefer medium on a particular port.	
Prototype	sw_error_t fal_port_combo_prefer_medium_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_medium_t medium	Medium
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.52 fal_port_combo_prefer_medium_get

Definition	Get prefer medium type on a particular port.	
Prototype	sw_error_t fal_port_combo_prefer_medium_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_medium_t * medium	Medium
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.53 fal_port_combo_medium_status_get

Definition	Get prefer medium type on a particular port.	
Prototype	sw_error_t fal_port_combo_medium_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_medium_t * medium	Medium
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.54 fal_port_combo_fiber_mode_set

Definition	Set fiber mode on a particular port.	
Prototype	sw_error_t fal_port_combo_fiber_mode_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	fal_port_fiber_mode_t mode	Mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.55 fal_port_combo_fiber_mode_get

Definition	Get prefer medium type on a particular port.	
Prototype	sw_error_t fal_port_combo_fiber_mode_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_medium_t * mode	Mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.56 fal_port_local_loopback_set

Definition	Set local loopback on a particular port.	
Prototype	sw_error_t fal_port_local_loopback_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	Enable
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.57 fal_port_local_loopback_get

Definition	Get local loopback on a particular port.	
Prototype	sw_error_t fal_port_local_loopback_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	Enable
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.58 fal_port_remote_loopback_set

Definition	Set remote loopback on a particular port.	
Prototype	sw_error_t fal_port_remote_loopback_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	Enable

	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.59 fal_port_remote_loopback_get

Definition	Get local remote loopback on a particular port.	
Prototype	sw_error_t fal_port_remote_looppback_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	Enable
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.60 fal_port_reset

Definition	Reset port on a particular port.	
Prototype	sw_error_t fal_port_reset (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.61 fal_port_power_off

Definition	Power off port on a particular port.	
Prototype	sw_error_t fal_port_power_off (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.62 fal_port_power_on

Definition	Power on port on a particular port.	
Prototype	sw_error_t fal_port_power_on (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.63 fal_port_phy_id_get

Definition	Get phy id on a particular port.	
Prototype	sw_error_t fal_port_phy_id_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint16_t * org_id	Org_id
	a_uint16_t * rev_id	Rev_id
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.64 fal_port_wol_status_set

Definition	Set wol status on a particular port.	
Prototype	sw_error_t fal_port_wol_status_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	enable
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.65 fal_port_wol_status_get

Definition	Get wol status on a particular port.	
Prototype	sw_error_t fal_port_wol_status_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	enable
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.66 fal_port_magic_frame_mac_set

Definition	Set magic frame mac address on a particular port.	
Prototype	sw_error_t fal_port_magic_frame_mac_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_mac_addr_t * mac	mac
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.67 fal_port_magic_frame_mac_get

Definition	Get magic frame mac address on a particular port.	
Prototype	`sw_error_t fal_port_magic_frame_mac_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_mac_addr_t * mac`	mac
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.68 fal_port_interface_mode_set

Definition	Set interface mode on a particular port.	
Prototype	`sw_error_t fal_port_interface_mode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_port_interface_mode_t mode`	mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.69 fal_port_interface_mode_get

Definition	Get interface mode on a particular port.	
Prototype	`sw_error_t fal_port_interface_mode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_mac_addr_t * mode`	mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.14.2.70 fal_port_interface_mode_status_get

Definition	Get current interface mode on a particular port.	
Prototype	`sw_error_t fal_port_interface_mode_status_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_mac_addr_t * mode`	mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15 FAL_PORT_VLAN

2.15.1 Enumeration type documentation

2.15.1.1 enum fal_pt_1qmode_t

This enum defines 802.1q mode type.

Enumeration values:

FAL_1Q_DISABLE 802.1q mode disable, port based VLAN

FAL_1Q_SECURE secure mode, packets which vid isn't in VLAN table or source port isn't in VLAN port member will be discarded.

FAL_1Q_CHECK check mode, packets which vid isn't in VLAN table will be discarded, packets which source port isn't in VLAN port member will forward base on VLAN port member.

FAL_1Q_FALLBACK fallback mode, packets which vid isn't in VLAN table will forwarded base on port VLAN, packet's which source port isn't in VLAN port member will forward base on VLAN port member.

2.15.1.2 enum fal_pt_invlan_mode_t

This enum defines receive packets tagged mode.

Enumeration values:

FAL_INVLAN_ADMIT_ALL receive all packets include tagged and untagged

FAL_INVLAN_ADMIT_TAGGED only receive tagged packets

FAL_INVLAN_ADMIT_UNTAGGED only receive untagged packets include priority tagged

2.15.1.3 enum fal_pt_egmode_t

This enum defines transmit packets tagged mode.

Enumeration values:

FAL_EG_UNMODIFIED egress transmit packets unmodified (keep translation result)

FAL_EG_UNTAGGED egress transmit packets without VLAN tag

FAL_EG_TAGGED egress transmit packets with VLAN tag

FAL_EG_HYBRID egress transmit packets in hybrid tag mode

FAL_EG_UNTOUCHED egress transmit packets untouched (keep original packets encapsulation)

2.15.1.4 enum fal_vlan_propagation_mode_t

This enum defines VLAN propagation mode.

Enumeration values:

FAL_VLAN_PROPAGATION_DISABLE VLAN propagation disable

FAL_VLAN_PROPAGATION_CLONE VLAN propagation mode is clone

FAL_VLAN_PROPAGATION_REPLACE VLAN propagation mode is replace

2.15.2 Struct documentation

```
/* This struct defines VLAN translation entry. */
typedef struct
{
    a_uint32_t o_vid;          // original vlan id
    a_uint32_t s_vid;          // service vlan id
    a_uint32_t c_vid;          // custom vlan id
    a_bool_t   bi_dir;         // forward and reverse direction
    a_bool_t   forward_dir;    // forward direction
                               // from o_vid to s_vid and/or c_vid
    a_bool_t   reverse_dir;    // reverse direction
                               // from s_vid and/or c_vid to o_vid
    a_bool_t   o_vid_is_cvid;  // if enabled, o_vid means c_vid
                               // else o_vid means s_vid
    a_bool_t   s_vid_enable;   // s_vid in entry is valid
    a_bool_t   c_vid_enable;   // c_vid in entry is valid
    a_bool_t   one_2_one_vlan; // the entry used for 1:1 vlan
} fal_vlan_trans_entry_t;
```

2.15.3 Function documentation

2.15.3.1 fal_port_1qmode_get

Definition	Get 802.1q work mode on a particular port.	
Prototype	`sw_error_t fal_port_1qmode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_pt_1qmode_t *pport_1qmode`	Port VLAN mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.2 fal_port_1qmode_set

Definition	Set 802.1q work mode on a particular port.	
Prototype	`sw_error_t fal_port_1qmode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_pt_1qmode_t port_1qmode`	Port VLAN mode
	`)`	
Description	See also fal_port_force_portvlan_set.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.3 fal_port_default_cvid_get

Definition	Get default c-vid on a particular port.	
Prototype	`sw_error_t fal_port_default_cvid_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t *vid`	c-vid
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.4 fal_port_default_cvid_set

Definition	Set default c-vid on a particular port.	
Prototype	`sw_error_t fal_port_default_cvid_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t vid`	c-vid
	`)`	
Description	Port default CVID. Untagged frames when transmitted from this port will be tagged this default CVID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.5 fal_port_default_svid_get

Definition	Get default s-vid on a particular port.	
Prototype	`sw_error_t fal_port_default_svid_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t *vid`	s-vid
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.6 fal_port_default_svid_set

Definition	Set default s-vid on a particular port.	
Prototype	`sw_error_t fal_port_default_svid_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t vid`	s-vid
	`)`	
Description	Port default SVID. Untagged frames when transmitted from this port will be tagged this default SVID.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.7 fal_port_default_vid_get

Definition	Get default VID on a particular port.	
Prototype	`sw_error_t fal_port_default_vid_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t *vid`	Default VID
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.8 fal_port_default_vid_set

Definition	Set default VID on a particular port.	
Prototype	`sw_error_t fal_port_default_vid_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t vid`	Default VID
	`)`	
Description	For AR8337N, this API is not supported. Use fal_port_default_cvid_set and fal_port_default_svid_set.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.9 fal_port_force_default_vid_get

Definition	Get force default VLAN ID status on a particular port.	
Prototype	`sw_error_t fal_port_force_default_vid_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.10 fal_port_force_default_vid_set

Definition	Set force default VLAN ID status on a particular port.	
Prototype	`sw_error_t fal_port_force_default_vid_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Description	Force to use port default VID for received frame, when 802.1Q mode is not disabled. Port default VID is set by fal_port_default_cvid_set and fal_port_default_svid_set.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.11 fal_port_invlan_mode_get

Definition	Get ingress VLAN mode on a particular port.	
Prototype	`sw_error_t fal_port_invlan_mode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_pt_invlan_mode_t *mode`	Ingress VLAN mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.12 fal_port_invlan_mode_set

Definition	Set ingress VLAN mode on a particular port.	
Prototype	`sw_error_t fal_port_invlan_mode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_pt_invlan_mode_t mode`	Ingress VLAN mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.13 fal_port_egvlanmode_get

Definition	Get egress VLAN mode on a particular port.	
Prototype	`sw_error_t fal_port_egvlanmode_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_pt_1q_egmode_t *pport_egvlanmode`	Egress VLAN mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.14 fal_port_egvlanmode_set

Definition	Set egress VLAN mode on a particular port.	
Prototype	`sw_error_t fal_port_egvlanmode_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_pt_1q_egmode_t port_egvlanmode`	Egress VLAN mode
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.15 fal_port_force_portvlan_get

Definition	Get force port based VLAN status on a particular port.	
Prototype	`sw_error_t fal_port_force_portvlan_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t *enable`	A_TRUE or A_FALSE
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.16 fal_port_force_portvlan_set

Definition	Set force port based VLAN status on a particular port.	
Prototype	`sw_error_t fal_port_force_portvlan_set (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`)`	
Description	Force to use port based VLAN in a particular port.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.17 fal_portvlan_member_add

Definition	Add member of port based VLAN on a particular port.	
Prototype	`sw_error_t fal_portvlan_member_add (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t mem_port_id`	Member port ID
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.18 fal_portvlan_member_del

Definition	Delete member of port based VLAN on a particular port.	
Prototype	sw_error_t fal_portvlan_member_del (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t mem_port_id	Member port ID
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.19 fal_portvlan_member_get

Definition	Get member of port based VLAN on a particular port.	
Prototype	sw_error_t fal_portvlan_member_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_pbmp_t *mem_port_map	Member port bitmap
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.20 fal_portvlan_member_update

Definition	Update member of port based vlan on a particular port.	
Prototype	sw_error_t fal_portvlan_member_update (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_pbmp_t mem_port_map	Member port bitmap
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.21 fal_qinq_mode_get

Definition	Get switch qinq work mode on a particular device.	
Prototype	sw_error_t fal_qinq_mode_get (	
	a_uint32_t dev_id,	Device ID
	fal_qinq_mode_t *mode	Qinq work mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.22 fal_qinq_mode_set

Definition	Set switch qinq work mode on a particular device.	
Prototype	sw_error_t fal_qinq_mode_set (	

	a_uint32_t dev_id,	Device ID
	fal_qinq_mode_t mode	Qinq work mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.23 fal_port_qinq_role_get

Definition	Get qinq role on a particular port.	
Prototype	sw_error_t fal_port_qinq_role_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_qinq_port_role_t *role	Port role
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.24 fal_port_qinq_role_set

Definition	Set qinq role on a particular port.	
Prototype	sw_error_t fal_port_qinq_role_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_qinq_port_role_t role	Port role
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.25 fal_port_nestvlan_get

Definition	Get nest VLAN feature status on a particular port.	
Prototype	sw_error_t fal_port_nestvlan_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.26 fal_port_nestvlan_set

Definition	Set nest VLAN feature status on a particular port.	
Prototype	sw_error_t fal_port_nestvlan_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.27 fal_nestvlan_tpid_get

Definition	Get nest VLAN TPID on a particular device.	
Prototype	sw_error_t fal_nestvlan_tpid_get (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t *tpid	Tag protocol identification
	)	
Description	For AR8337N, this API is not supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.28 fal_nestvlan_tpid_set

Definition	Set nest VLAN TPID on a particular device.	
Prototype	sw_error_t fal_nestvlan_tpid_set (	
	a_uint32_t dev_id,	Device ID
	a_uint32_t tpid	Tag protocol identification
	)	
Description	For AR8337N, this API is not supported.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.29 fal_port_pri_propagation_get

Definition	Get priority propagation status on a particular port.	
Prototype	sw_error_t fal_port_pri_propagation_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.30 fal_port_pri_propagation_set

Definition	Set priority propagation status on a particular port.	
Prototype	sw_error_t fal_port_pri_propagation_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.31 fal_port_vlan_propagation_get

Definition	Get VLAN propagation mode on a particular port.	
Prototype	sw_error_t fal_port_vlan_propagation_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_vlan_propagation_mode_t *mode	VLAN propagation mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.32 fal_port_vlan_propagation_set

Definition	Set VLAN propagation mode on a particular port.	
Prototype	sw_error_t fal_port_vlan_propagation_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_vlan_propagation_mode_t mode	VLAN propagation mode
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.33 fal_port_tls_get

Definition	Get TLS status on a particular port.	
Prototype	sw_error_t fal_port_tls_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.34 fal_port_tls_set

Definition	Set TLS status on a particular port.	
Prototype	sw_error_t fal_port_tls_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	a_bool_t enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.35 fal_port_vlan_trans_add

Definition	Add a VLAN translation entry to a particular port.	
Prototype	sw_error_t fal_port_vlan_trans_add (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_vlan_trans_entry_t *entry	VLAN translation entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.36 fal_port_vlan_trans_del

Definition	Delete a VLAN translation entry from a particular port.	
Prototype	sw_error_t fal_port_vlan_trans_del (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_vlan_trans_entry_t *entry	VLAN translation entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.37 fal_port_vlan_trans_get

Definition	Get a VLAN translation entry from a particular port.	
Prototype	sw_error_t fal_port_vlan_trans_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_vlan_trans_entry_t *entry	VLAN translation entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.38 fal_port_vlan_trans_iterate

Definition	Iterate all VLAN translation entries from a particular port.	
Prototype	sw_error_t fal_port_vlan_trans_iterate (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID

	a_uint32_t *iterator,	Translation entry index If it's zero means get the first entry.
	fal_vlan_trans_entry_t *entry	VLAN translation entry
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.39 fal_netisolate_get

Definition	Get NET isolate status on a particular device.	
Prototype	sw_error_t fal_netisolation_get (	
	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.40 fal_netisolate_set

Definition	Set NET isolate status on a particular device.	
Prototype	sw_error_t fal_netisolation_set (	
	a_uint32_t dev_id,	Device ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If net isolation is enabled, hardware should isolate private net and public net, thus route is not available between private net and public net without NAT.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.41 fal_eg_trans_filter_bypass_en_get

Definition	Get egress translation filter bypass status on a particular device.	
Prototype	sw_error_t fal_eg_trans_filter_bypass_en_get (	
	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.42 fal_eg_trans_filter_bypass_en_set

Definition	Set egress translation filter bypass status on a particular device.	
Prototype	sw_error_t fal_eg_trans_filter_bypass_en_set (	

	a_uint32_t dev_id,	Device ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.43 fal_port_mac_vlan_xlt_get

Definition	Get MAC VLAN XLT status on a particular port.	
Prototype	sw_error_t fal_port_mac_vlan_xlt_get (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t *enable	A_TRUE or A_FALSE
	)	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.15.3.44 fal_port_mac_vlan_xlt_set

Definition	Set MAC VLAN XLT status on a particular port.	
Prototype	sw_error_t fal_port_mac_vlan_xlt_set (	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	)	
Description	If MAC VLAN XLT is enabled, hardware will use VLAN ID in FDB entry as the result of egress VLAN translation.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.16 FAL_QOS

2.16.1 Enumeration type documentation

2.16.1.1 fal_sch_mode_t

This enum defines traffic scheduling mode.

```
typedef enum {
    FAL_SCH_SP_MODE = 0,        //  strict priority scheduling mode
    FAL_SCH_WRR_MODE,           //  weight round robin scheduling mode
    FAL_SCH_MIX_MODE,           //  sp and wrr mixed scheduling mode
    FAL_SCH_MIX_PLUS_MODE,      //  sp and wrr mixed plus scheduling mode
    FAL_SCH_MODE_BUTT
```

```
 }
 fal_sch_mode_t;
```

2.16.1.2 fal_qos_mode_t

This enum defines QoS assignment mode.

```
typedef enum
{
    FAL_QOS_DA_MODE = 0,     //  qos assignment based on destination mac address
    FAL_QOS_UP_MODE,         //  qos assignment based on 802.1p field in vlan tag
    FAL_QOS_DSCP_MODE,       //  qos assignment based on dscp field in ip header
    FAL_QOS_PORT_MODE,       //  qos assignment based on port
    FAL_QOS_FLOW_MODE,       //  qos assignment based on flow
    FAL_QOS_MODE_BUTT
} fal_qos_mode_t;
```

2.16.2 Function documentation

2.16.2.1 fal_qos_sch_mode_set()

Definition	Set traffic scheduling mode on particular one device.	
Prototype	`sw_error_t fal_qos_sch_mode_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_sch_mode_t mode,`	Traffic scheduling mode
	`const a_uint32_t weight[]`	Weight value for each queue when in WRR mode
	`);`	
Description	Particular device can support parts of input options. Such as GARUDA doesn't support variable weight in WRR mode. When scheduling mode is SP, the weight is meaningless.	
Return Value	SW_OK or error code	

2.16.2.2 fal_qos_sch_mode_get()

Definition	Get traffic scheduling mode on particular device.	
Prototype	`sw_error_t fal_qos_sch_mode_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_sch_mode_t * mode,`	Traffic scheduling mode
	`a_uint32_t weight[]`	Weight value for each queue when in WRR mode
	`);`	
Return Value	SW_OK or error code	

2.16.2.3 fal_qos_queue_tx_buf_status_set()

Definition	Set queue Tx buffer aggsinment status of on a port.	
Prototype	`sw_error_t fal_qos_queue_tx_buf_status_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	Status 'A_TRUE' means every queue has fixed number Tx buffers. Status 'A_FALSE' means the queue shares the buffers with other queues.
	`);`	
Description	If the queue Tx buffer on one port is enabled, that means each queue of this port will have fixed number buffers when transmitting packets. Otherwise they share the whole buffers with the other queues in device.	
Return Value	SW_OK or error code	

2.16.2.4 fal_qos_queue_tx_buf_status_get ()

Definition	Get queue Tx buffer aggsinment status of a port.	
Prototype	`sw_error_t fal_qos_queue_tx_buf_status_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t * enable`	Status 'A_TRUE' means every queue has fixed number Tx buffers. Status 'A_FALSE' means the queue shares the buffers with other queues.
	`);`	
Return Value	SW_OK or error code	

2.16.2.5 fal_qos_queue_tx_buf_nr_set ()

Definition	Set max occupied Tx buffer number of a queue on a port.	
Prototype	`sw_error_t fal_qos_queue_tx_buf_nr_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_queue_t queue_id,`	Queue ID
	`a_uint32_t * number`	Queue max Tx buffer number. Function will return the actual buffer numbers in hardware because different device has different hardware granularity.
	`);`	
Return Value	SW_OK or error code	

2.16.2.6 fal_qos_queue_tx_buf_nr_get()

Definition	Get max occupied Tx buffer number of a queue on a port.	
Prototype	sw_error_t fal_qos_queue_tx_buf_nr_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_queue_t queue_id,	Queue ID
	a_uint32_t * number	Queue max Tx buffer number
	);	
Return Value	SW_OK or error code	

2.16.2.7 fal_qos_port_tx_buf_status_set()

Definition	Set buffer aggsinment status of transmitting port on one particular port.	
Prototype	sw_error_t fal_qos_port_tx_buf_status_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE. If the Tx buffer on one port is enabled, that means this port will have fixed number buffers when transmitting packets. Otherwise they will share the whole buffers with the other ports in device.
	);	
Return Value	SW_OK or error code	

2.16.2.8 fal_qos_port_tx_buf_status_get()

Definition	Get buffer aggsinment status of transmitting port on one particular port.	
Prototype	sw_error_t fal_qos_port_tx_buf_status_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	A_TRUE or A_FALSE. If the Tx buffer on one port is enabled, that means this port will have fixed number buffers when transmitting packets. Otherwise they will share the whole buffers with the other ports in device.
	);	
Return Value	SW_OK or error code	

2.16.2.9 fal_qos_port_red_en_set()

Definition	Set status of port red on one particular port.	
Prototype	`sw_error_t fal_qos_port_red_en_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t enable`	A_TRUE or A_FALSE
	`);`	
Return Value	SW_OK or error code	

2.16.2.10 fal_qos_port_red_en_get()

Definition	Get port red status of one particular port.	
Prototype	`sw_error_t fal_qos_port_red_en_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_bool_t * enable`	A_TRUE or A_FALSE
	`);`	
Return Value	SW_OK or error code	

2.16.2.11 fal_qos_port_tx_buf_nr_set()

Definition	Set max occupied buffer number of transmitting port on one particular port.	
Prototype	`sw_error_t fal_qos_port_tx_buf_nr_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t * number`	Port max Tx buffer number. Function will return the actual buffer numbers in hardware because different device has different hardware granularity.
	`);`	
Return Value	SW_OK or error code	

2.16.2.12 fal_qos_port_tx_buf_nr_get()

Definition	Get max occupied buffer number of transmitting port on one particular port.	
Prototype	`sw_error_t fal_qos_port_tx_buf_nr_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t * number`	Port max Tx buffer number.

	);	
Return Value	SW_OK or error code	

2.16.2.13 fal_qos_port_rx_buf_nr_set()

Definition	Set max reserved buffer number of receiving port on one particular port.	
Prototype	sw_error_t fal_qos_port_rx_buf_nr_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * number	Port max Rx buffer number. Function will the return actual buffer numbers in hardware because different device has different hardware granularity.
	);	
Return Value	SW_OK or error code	

2.16.2.14 fal_qos_port_rx_buf_nr_get()

Definition	Get max reserved buffer number of receiving port on one particular port.	
Prototype	sw_error_t fal_qos_port_rx_buf_nr_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * number	Port max RX buffer number.
	);	
Return Value	SW_OK or error code	

2.16.2.15 fal_cosmap_up_queue_set()

Definition	Set user priority to queue mapping.	
Prototype	sw_error_t fal_cosmap_up_queue_set(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t up,	802.1P
	fal_queue_t queue	Queue ID
	);	
Return Value	SW_OK or error code	

2.16.2.16 fal_cosmap_up_queue_get()

Definition	Get user priority to queue mapping.	
Prototype	sw_error_t fal_cosmap_up_queue_get(	

	a_uint32_t dev_id,	Device ID
	a_uint32_t up,	802.1P
	fal_queue_t * queue	Pinter to Queue ID
	);	
Return Value	SW_OK or error code	

2.16.2.17 fal_cosmap_dscp_queue_set()

Definition	Set DSCP to queue mapping.	
Prototype	sw_error_t fal_cosmap_dscp_queue_set(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t dscp,	DSCP
	fal_queue_t queue	Pinter to Queue ID
	);	

2.16.2.18 fal_cosmap_dscp_queue_get()

Definition	Get DSCP to queue mapping.	
Prototype	sw_error_t fal_cosmap_dscp_queue_get(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t dscp,	DSCP
	fal_queue_t * queue	Pinter to Queue ID
	);	
Return Value	SW_OK or error code	

2.16.2.19 fal_qos_port_mode_set()

Definition	Set port QoS mode on one particular port.	
Prototype	sw_error_t fal_qos_port_mode_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_qos_mode_t mode,	QoS mode
	a_bool_t enable	A_TRUE of A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.20 fal_qos_port_mode_get()

Definition	Get port QoS mode on one particular port.

Prototype	sw_error_t fal_qos_port_mode_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_qos_mode_t mode,	QoS mode
	a_bool_t * enable	A_TRUE of A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.21 fal_qos_port_mode_pri_set()

Definition	Set priority of one particular QoS mode on one particular port.	
Prototype	sw_error_t fal_qos_port_mode_pri_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_qos_mode_t mode,	QoS mode
	a_uint32_t pri	Priority. Smaller priority means higher priority.
	);	
Return Value	SW_OK or error code	

2.16.2.22 fal_qos_port_mode_pri_get()

Definition	Get priority of one particular QoS mode on one particular port.	
Prototype	sw_error_t fal_qos_port_mode_pri_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_qos_mode_t mode,	QoS mode
	a_uint32_t * pri	Priority. Smaller priority means higher priority.
	);	
Return Value	SW_OK or error code	

2.16.2.23 fal_qos_port_default_up_set()

Definition	Set default user priority on one particular port.	
Prototype	sw_error_t fal_qos_port_default_up_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t up	802.1P
	);	
Return Value	SW_OK or error code	

2.16.2.24 fal_qos_port_default_up_get()

Definition	Get default user priority on one particular port.	
Prototype	sw_error_t fal_qos_port_default_up_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * up	802.1P
	);	
Return Value	SW_OK or error code	

2.16.2.25 fal_qos_port_sch_mode_set()

Definition	Set traffic scheduling mode on particular one port.	
Prototype	sw_error_t fal_qos_port_sch_mode_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_sch_mode_t mode,	Traffic scheduling mode
	const a_uint32_t weight[]	Weight value for each queue when in WRR mode. Particular device may support parts of input options only. Some chip doesn't support variable weight in WRR mode. The weight is meaningless when scheduling mode is SP, usually it's zero.
	);	
Return Value	SW_OK or error code	

2.16.2.26 fal_qos_port_sch_mode_get()

Definition	Get traffic scheduling mode on particular port.	
Prototype	sw_error_t fal_qos_port_sch_mode_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_sch_mode_t * mode,	Traffic scheduling mode
	a_uint32_t weight[]	Weight value for each queue when in WRR mode.
	);	
Return Value	SW_OK or error code	

2.16.2.27 fal_qos_port_default_spri_set()

Definition	Set default stag priority on one particular port.

Prototype	sw_error_t fal_qos_port_default_spri_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t spri	Service TAG priority
	);	
Return Value	SW_OK or error code	

2.16.2.28 fal_qos_port_default_spri_get()

Definition	Get default stag priority on one particular port.	
Prototype	sw_error_t fal_qos_port_default_spri_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * spri	Service TAG priority
	);	
Return Value	SW_OK or error code	

2.16.2.29 fal_qos_port_default_cpri_set()

Definition	Set default ctag priority on one particular port.	
Prototype	sw_error_t fal_qos_port_default_cpri_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t cpri	CTAG priority
	);	
Return Value	SW_OK or error code	

2.16.2.30 fal_qos_port_default_cpri_get()

Definition	Get default ctag priority on one particular port.	
Prototype	sw_error_t fal_qos_port_default_cpri_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * cpri	CTAG priority
	);	
Return Value	SW_OK or error code	

2.16.2.31 fal_qos_port_force_spri_status_set()

Definition	Set force stag priority flag on one particular port.	
Prototype	sw_error_t fal_qos_port_force_spri_status_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.32 fal_qos_port_force_spri_status_get()

Definition	Get force stag priority flag on one particular port.	
Prototype	sw_error_t fal_qos_port_force_spri_status_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.33 fal_qos_port_force_cpri_status_set()

Definition	Set force ctag priority flag on one particular port.	
Prototype	sw_error_t fal_qos_port_force_cpri_status_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.34 fal_qos_port_force_cpri_status_get()

Definition	Get force ctag priority flag on one particular port.	
Prototype	sw_error_t fal_qos_port_force_cpri_status_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.35 fal_qos_queue_remark_table_set()

Definition	Set egress queue based CoS remark on one particular port.	
Prototype	sw_error_t fal_qos_queue_remark_table_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_queue_t queue_id,	Queue ID
	a_uint32_t tbl_id,	Remark table ID
	a_bool_t enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.16.2.36 fal_qos_queue_remark_table_get()

Definition	Get egress queue based CoS remark on one particular port.	
Prototype	sw_error_t fal_qos_queue_remark_table_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_queue_t queue_id,	Queue ID
	a_uint32_t * tbl_id,	Remark table ID
	a_bool_t * enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.17 FAL_RATE

2.17.1 Enumeration type documentation

2.17.1.1 fal_storm_type_t

This enum defines storm type.

```
typedef enum {
    FAL_UNICAST_STORM = 0,      //  storm caused by unknown unicast packets
    FAL_MULTICAST_STORM,        //  storm caused by unknown multicast packets
    FAL_BROADCAST_STORM,        //  storm caused by broadcast packets
    FAL_STORM_TYPE_BUTT
}
fal_storm_type_t;
```

2.17.2 Function documentation

2.17.2.1 fal_rate_queue_egrl_set()

Definition	Set queue egress rate limit status on one particular port and queue.	
Prototype	`sw_error_t fal_rate_queue_egrl_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_queue_t queue_id,`	Queue ID
	`a_uint32_t * speed,`	Rate limit speed
	`a_bool_t enable`	A_TRUE or A_FALSE
	`);`	
Description	The granularity of speed is bps. Because different device has different hardware granularity, function will return the actual speed in hardware. When queue egress rate limit is disabled, the input parameter speed is meaningless.	
Return Value	SW_OK or error code	

2.17.2.2 fal_rate_queue_egrl_get()

Definition	Get queue egress rate limit status on one particular port and queue.	
Prototype	`sw_error_t fal_rate_queue_egrl_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_queue_t queue_id,`	Queue ID
	`a_uint32_t * speed,`	Rate limit speed
	`a_bool_t * enable`	A_TRUE or A_FALSE
	`);`	
Return Value	SW_OK or error code	

2.17.2.3 fal_rate_port_egrl_set()

Definition	Set port egress rate limit status on one particular port.	
Prototype	`sw_error_t fal_rate_port_egrl_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`a_uint32_t * speed,`	Rate limit speed
	`a_bool_t enable`	A_TRUE or A_FALSE
	`);`	
Description	The granularity of speed is bps. Because different device has different hardware granularity, function will return the actual speed in hardware. When port egress rate limit is disabled, the input parameter speed is meaningless.	
Return Value	SW_OK or error code	

2.17.2.4 fal_rate_port_egrl_get()

Definition	Get port egress rate limit status on one particular port.	
Prototype	sw_error_t fal_rate_port_egrl_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * speed,	Rate limit speed
	a_bool_t * enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.17.2.5 fal_rate_port_inrl_set()

Definition	Set port ingress rate limit status on one particular port.	
Prototype	sw_error_t fal_rate_port_inrl_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * speed,	Rate limit speed
	a_bool_t enable	A_TRUE or A_FALSE
	);	
Description	The granularity of speed is bps. Because different device has different hardware granularity, function will the return actual speed in hardware. When port ingress rate limit is disabled, the input parameter speed is meaningless.	
Return Value	SW_OK or error code	

2.17.2.6 fal_rate_port_inrl_get()

Definition	Get port ingress rate limit status on one particular port.	
Prototype	sw_error_t fal_rate_port_inrl_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * speed,	Rate limit speed
	a_bool_t * enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.17.2.7 fal_storm_ctrl_frame_set()

Definition	Set particular type storm control status on one particular port.	
Prototype	sw_error_t fal_storm_ctrl_frame_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_storm_type_t frame_type,	Packets type which causes storm

	a_bool_t enable	A_TRUE or A_FALSE
	);	
Description	When one particular packets type storm control is enabled, this type packets speed will be calculated in storm control.	
Return Value	SW_OK or error code	

2.17.2.8 fal_storm_ctrl_frame_get()

Definition	Get particular type storm control status on one particular port.	
Prototype	sw_error_t fal_storm_ctrl_frame_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_storm_type_t frame_type,	Packets type which causes storm
	a_bool_t * enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.17.2.9 fal_storm_ctrl_rate_set()

Definition	Set storm control speed on one particular port.	
Prototype	sw_error_t fal_storm_ctrl_rate_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t* rate	Storm control speed
	);	
Description	The granularity of speed is packets per second. Because different device has different hardware granularity, function will return the actual speed in hardware.	
Return Value	SW_OK or error code	

2.17.2.10 fal_storm_ctrl_rate_get()

Definition	Get storm control speed on one particular port.	
Prototype	sw_error_t fal_storm_ctrl_rate_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * rate	Storm control speed
	);	
Return Value	SW_OK or error code	

2.17.2.11 fal_rate_port_policer_set()

Definition	Set port ingress policer parameters on one particular port.	
Prototype	sw_error_t fal_rate_port_policer_set(	

	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_policer_t * policer	Pointer to struct of port ingress policer parameters.
	);	
Return Value	SW_OK or error code	

2.17.2.12 fal_rate_port_policer_get()

Definition	Get port ingress policer parameters on one particular port.	
Prototype	sw_error_t fal_rate_port_policer_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	fal_port_policer_t * policer	Pointer to struct of port ingress policer parameters.
	);	
Return Value	SW_OK or error code	

2.17.2.13 fal_rate_port_shaper_set()

Definition	Set port egress shaper parameters on one particular port.	
Prototype	sw_error_t fal_rate_port_shaper_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable,	A_TRUE or A_FALSE
	fal_egress_shaper_t * shaper	Pointer to struct of port egress shaper parameters.
	);	
Return Value	SW_OK or error code	

2.17.2.14 fal_rate_port_shaper_get()

Definition	Get port egress shaper parameters on one particular port.	
Prototype	sw_error_t fal_rate_port_shaper_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable,	A_TRUE or A_FALSE
	fal_egress_shaper_t * shaper	Pointer to struct of port egress shaper parameters.
	);	
Return Value	SW_OK or error code	

2.17.2.15 fal_rate_queue_shaper_set()

Definition	Set queue egress shaper parameters on one particular port.	
Prototype	`sw_error_t fal_rate_queue_shaper_set(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_queue_t queue_id,`	Queue ID
	`a_bool_t enable,`	A_TRUE or A_FALSE
	`fal_egress_shaper_t * shaper`	Pointer to struct of port egress shaper parameters.
	`);`	
Return Value	SW_OK or error code	

2.17.2.16 fal_rate_queue_shaper_get()

Definition	Get queue egress shaper parameters on one particular port.	
Prototype	`sw_error_t fal_rate_queue_shaper_get(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_port_t port_id,`	Port ID
	`fal_queue_t queue_id,`	Queue ID
	`a_bool_t * enable,`	A_TRUE or A_FALSE
	`fal_egress_shaper_t * shaper`	Pointer to struct of port egress shaper parameters.
	`);`	
Return Value	SW_OK or error code	

2.17.2.17 fal_rate_acl_policer_set()

Definition	Set ACL ingress policer parameters.	
Prototype	`sw_error_t fal_rate_acl_policer_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t policer_id,`	ACL policer ID
	`fal_acl_policer_t * policer`	Pointer to struct of ACL ingress policer parameters
	`);`	
Return Value	SW_OK or error code	

2.17.2.18 fal_rate_acl_policer_get()

Definition	Get ACL ingress policer parameters.	
Prototype	`sw_error_t fal_rate_acl_policer_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t policer_id,`	ACL policer ID
	`fal_acl_policer_t * policer`	Pointer to struct of ACL ingress policer parameters

	);	
Return Value	SW_OK or error code	

2.17.2.19 fal_rate_port_add_rate_byte_set()

Definition	Set bytes number which should be added to frame when calculate rate limit on one particular port.	
Prototype	sw_error_t fal_rate_port_add_rate_byte_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t number	Bytes number
	);	
Return Value	SW_OK or error code	

2.17.2.20 fal_rate_port_add_rate_byte_get()

Definition	Get bytes number which should be added to frame when calculate rate limit on one particular port.	
Prototype	sw_error_t fal_rate_port_add_rate_byte_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_uint32_t * number	Bytes number
	);	
Return Value	SW_OK or error code	

2.17.2.21 fal_rate_port_gol_flow_en_set()

Definition	Set global flow control enabled when global threshold is reached on one particular port.	
Prototype	sw_error_t fal_rate_port_gol_flow_en_set(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.17.2.22 fal_rate_port_gol_flow_en_get()

Definition	Get global flow control enabled when global threshold is reached on one particular port.	
Prototype	sw_error_t fal_rate_port_gol_flow_en_get(	
	a_uint32_t dev_id,	Device ID
	fal_port_t port_id,	Port ID
	a_bool_t * enable	A_TRUE or A_FALSE

	);	
Return Value	SW_OK or error code	

2.18 FAL_REG_ACCESS

2.18.1 Function documentation

2.18.1.1 fal_phy_get

Definition	Get PHY register value of specific PHY device.	
Prototype	sw_error_t fal_phy_get(	
	a_uint32_t device_id,	PHY device ID
	a_uint32_t phy_addr,	PHY device register ID
	a_uint32_t reg,	Pointer to the memory storing the value
	a_uint32_t * value,	PHY register value
	)	
Return Value	SW_OK or error code	

2.18.1.2 fal_phy_set

Definition	Set PHY register value of specific PHY device.	
Prototype	sw_error_t fal_phy_set(	
	a_uint32_t device_id,	PHY device ID
	a_uint32_t phy_addr,	PHY device register ID
	a_uint32_t reg,	Pointer to the memory storing the value
	a_uint32_t value,	PHY register value
	)	
Return Value	SW_OK or error code	

2.18.1.3 fal_reg_field_get

Definition	Get register specific field value of specific register.	
Prototype	sw_error_t fal_reg_field_get(	
	a_uint32_t device_id,	Device ID
	a_uint32_t reg_addr,	Device register address
	a_uint32_t bit_offset,	Field position in bit
	a_uint32_t field_len,	Field length in bit
	a_uint8_t value[],	Pointer to the memory storing the value
	a_uint32_t value_len,	Value length
	)	
Return Value	SW_OK or error code	

2.18.1.4 fal_reg_field_set

Definition	Set register specific field value of specific register.	
Prototype	sw_error_t fal_reg_field_set(	
	a_uint32_t device_id,	Device ID
	a_uint32_t reg_addr,	Device register address
	a_uint32_t bit_offset,	Field position in bit
	a_uint32_t field_len,	Field length in bit
	a_uint8_t value[],	pointer to the memory storing the value
	a_uint32_t value_len,	Value length
	)	
Return Value	SW_OK or error code	

2.18.1.5 fal_reg_get

Definition	Get the specific register value	
Prototype	sw_error_t fal_reg_get(	
	a_uint32_t device_id,	Device ID
	a_uint32_t reg_addr,	Device register address
	a_uint8_t value[],	pointer to the memory storing the value
	a_uint32_t value_len,	Value length
	)	
Return Value	SW_OK or error code	

2.18.1.6 fal_reg_set

Definition	Set the specific register value.	
Prototype	sw_error_t fal_reg_set(	
	a_uint32_t device_id,	Device ID
	a_uint32_t reg_addr,	Device register address
	a_uint8_t value[],	pointer to the memory storing the value
	a_uint32_t value_len,	Value length
	)	
Return Value	SW_OK or error code	

2.19 FAL_SEC

2.19.1 Enumeration type documentation

2.19.1.1 enum fal_norm_item_t

This enum defines normalization item type.

Enumeration values:

FAL_NORM_MAC_RESV_VID_CMD Frame with VID equal to 4095 should be dropped.

FAL_NORM_MAC_INVALID_SRC_ADDR_CMD Frame with SA is broadcast or multicast address should be dropped.

FAL_NORM_IP_INVALID_VER_CMD Frame with version field is not equal to 0x4 or 0x6 in IP header should be dropped.

FAL_NROM_IP_SAME_ADDR_CMD Frame with SIP equal to DIP should be dropped.

FAL_NROM_IP_TTL_CHANGE_STATUS Frame TTL change to IP_TTL_VALUE.

FAL_NROM_IP_TTL_VALUE IP_TTL_VALUE.

FAL_NROM_IP4_INVALID_HL_CMD Frame with IPv4 header length less than 20 byte should be dropped.

FAL_NROM_IP4_HDR_OPTIONS_CMD Frame with IP options exist should be dropped/redirect to CPU.

FAL_NROM_IP4_INVALID_DF_CMD Frame with DF=1 and offset or MF not zero should be dropped.

FAL_NROM_IP4_FRAG_OFFSET_MIN_LEN_CMD Frame with offset length less than IPV4_FRAG_MIN_SIZE should be dropped.

FAL_NROM_IP4_FRAG_OFFSET_MIN_SIZE IPV4_FRAG_MIN_SIZE

FAL_NROM_IP4_FRAG_OFFSET_MAX_LEN_CMD Frame with offset length more than max (Offset (13bits) x8+ IP TOTAL LEN（16bits）>= 64KB) should be dropped.

FAL_NROM_IP4_INVALID_FRAG_OFFSET_CMD Frame with IPv4 fragment (not the last fragment, mf = 1) and length check error ((IP len (LENGTH FIELD) - Header Len) % 8 ！= 0) should be dropped.

FAL_NROM_IP4_INVALID_SIP_CMD Frame with SIP[31:24] more than 0xE0 and less than 0xF0, or equal to 0x7F, or SIP[31:0] is 0x32'hFFFFFFFF should be dropped.

FAL_NROM_IP4_INVALID_DIP_CMD Frame with DIP all zero, or DIP[31:24] is 0x7F should be dropped.

FAL_NROM_IP4_INVALID_CHKSUM_CMD Frame with IPv4 checksum error should be dropped.

FAL_NROM_IP4_INVALID_PL_CMD Frame with short length (20(Min IPv4 Header Length) + 18 + VLAN + SNAP + PPPOE > FRAME LENGTH) should be dropped

FAL_NROM_IP4_DF_CLEAR_STATUS Frame IPv4 DF field cleared to zero.

FAL_NROM_IP4_IPID_RANDOM_STATUS Frame (not fragment) is sent out with random ID.

FAL_NROM_IP6_INVALID_PL_CMD Frame with short length (40(Min IPv6 Header Length) + 18 + VLAN + SNAP + PPPOE > FRAME LENGTH) should be dropped.

FAL_NROM_IP6_INVALID_SIP_CMD IPv6 frame with SIP is ::1 or ff00::/8 should be dropped.

FAL_NROM_IP6_INVALID_DIP_CMD IPv6 frame with DIP is ::1 or zero should be dropped.

FAL_NROM_TCP_BLAT_CMD TCP frame with SP equal to DP should be dropped.

FAL_NROM_TCP_INVALID_HL_CMD If frame with TCP header length less than TCP_HDR_MIN_SIZE, but not first of fragment, should be dropped.

FAL_NROM_TCP_MIN_HDR_SIZE TCP_HDR_MIN_SIZE

FAL_NROM_TCP_INVALID_SYN_CMD Frame with SYN = 1 & ACK = 0 & SP<1024 should be dropped.

FAL_NROM_TCP_SU_BLOCK_CMD Frame with SYN = 1 & URG = 1 should be dropped.

FAL_NROM_TCP_SP_BLOCK_CMD Frame with SYN = 1 & PSH = 1 should be dropped.

FAL_NROM_TCP_SAP_BLOCK_CMD Frame with SYN = 1 & ACK = 1 & PSH = 1 should be dropped.

FAL_NROM_TCP_XMAS_SCAN_CMD Frame FIN = 1, URG = 1, and PSH = 1 should be dropped.

FAL_NROM_TCP_NULL_SCAN_CMD Frame with all TCP FLAG zero should be dropped.

FAL_NROM_TCP_SR_BLOCK_CMD Frame with SYN = 1 & RST = 1 should be dropped.

FAL_NROM_TCP_SF_BLOCK_CMD Frame with SYN = 1 & FIN = 1 should be dropped.

FAL_NROM_TCP_SAR_BLOCK_CMD Frame with SYN = 0 & ACK = 0 & RST = 0 should be dropped.

FAL_NROM_TCP_RST_SCAN_CMD Frame with RST = 1 should be dropped.

FAL_NROM_TCP_SYN_WITH_DATA_CMD TCP frame with SYN = 1 & IP payload len > TCP header length should be dropped.

FAL_NROM_TCP_RST_WITH_DATA_CMD TCP frame with RST = 1 & IP payload len > TCP header length should be dropped.

FAL_NROM_TCP_FA_BLOCK_CMD Frame with FIN = 1 & ACK = 0 should be dropped.

FAL_NROM_TCP_PA_BLOCK_CMD Frame with PUSH = 1 & ACK = 0 should be dropped.

FAL_NROM_TCP_UA_BLOCK_CMD Frame with URG = 1 & ACK = 0 should be dropped.

FAL_NROM_TCP_INVALID_CHKSUM_CMD Frame with TCP checksum error should be dropped.

FAL_NROM_TCP_INVALID_URGPTR_CMD Frame with URG = 0 but pointer not zero should be dropped.

FAL_NROM_TCP_INVALID_OPTIONS_CMD Frame with SYN = 0 and IP header larger than 20 byte, should be dropped.

FAL_NROM_UDP_BLAT_CMD UDP frame with SP equal to DP should be dropped.

FAL_NROM_UDP_INVALID_LEN_CMD Frame with UDP length check error (UDP LEN + IP HDR != IP LEN) should be dropped.

FAL_NROM_UDP_INVALID_CHKSUM_CMD Frame with UDP checksum error should be dropped.

FAL_NROM_ICMP4_PING_PL_EXCEED_CMD Ping frame with IP payload length larger than ICMPV4_MAX_LEN should be dropped.

FAL_NROM_ICMP4_PING_MAX_PL_VALUE ICMPV4_MAX_LEN

FAL_NROM_ICMP4_PING_FRAG_CMD ICMPv4 frame with fragment should be dropped.

FAL_NROM_ICMP6_PING_PL_EXCEED_CMD Ping frame with IP payload length larger than ICMPV6_MAX_LEN should be dropped.

FAL_NROM_ICMP6_PING_MAX_PL_VALUE ICMPV6_MAX_LEN

FAL_NROM_ICMP6_PING_FRAG_CMD ICMPv6 frame with fragment should be dropped.

2.19.2 Function documentation

2.19.2.1 fal_sec_norm_item_get

Definition	Get normalization particular item types value.	
Prototype	`sw_error_t fal_sec_norm_item_get (`	
	`a_uint32_t dev_id,`	Device ID
	`fal_norm_item_t item,`	Normalization item type
	`void *value`	Normalization item value
	`)`	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.19.2.2 fal_sec_norm_item_set

Definition	Set normalization particular item types value.

Prototype	sw_error_t fal_sec_norm_item_set (	
	a_uint32_t dev_id,	Device ID
	fal_norm_item_t item,	Normalization item type
	void value	Normalization item value
	)	
Description	Normalization item refer to fal_norm_item_t.	
Return Value	Returns SW_OK on success and sw_error_t on failure.	

2.20 FAL_STP

2.20.1 Enumeration type documentation

2.20.1.1 enum fal_stp_state_t

This enum defines port state for spanning tree.

Enumeration values:

FAL_STP_DISABLED disable state

FAL_STP_BLOKING blocking state

FAL_STP_LISTENING listening state

FAL_STP_LEARNING learning state

FAL_STP_FARWARDING forwarding state

2.20.2 Function documentation

2.20.2.1 fal_stp_port_state_get

Definition	Get port STP state on a particular spanning tree and port.	
Prototype	sw_error_t fal_stp_port_state_get(	
	a_uint32_t device_id,	Device ID
	a_uint32_t st_id,	Spanning tree ID
	fal_port_t port_id,	Port ID
	fal_stp_state_t * state,	Port state for spanning tree
	)	
Description	If the device supports the single spanning tree only, st_id should be FAL_SINGLE_STP_ID which is zero.	
Return Value	SW_OK or error code	

2.20.2.2 fal_stp_port_state_set

Definition	Set port STP state on a particular spanning tree and port.

Prototype	sw_error_t fal_stp_port_state_set(	
	a_uint32_t device_id,	Device ID
	a_uint32_t st_id,	Spanning tree ID
	fal_port_t port_id,	Port ID
	fal_stp_state_t state,	Port state for spanning tree
	)	
Description	If the device supports the single spanning tree only, st_id should be FAL_SINGLE_STP_ID which is zero.	
Return Value	SW_OK or error code	

2.21 FAL_TRUNK

2.21.1 Function documentation

2.21.1.1 fal_trunk_group_get

Definition	Get particular trunk group information on particular device.	
Prototype	sw_error_t fal_trunk_group_get(	
	a_uint32_t device_id,	Device ID
	a_uint32_t trunk_id,	Trunk group ID
	a_bool_t * enable,	Trunk group status (enable/disable)
	fal_pbmp_t * member	Port member information
	)	
Return Value	SW_OK or error code	

2.21.1.2 fal_trunk_group_set

Definition	Set particular trunk group information on particular device.	
Prototype	sw_error_t fal_trunk_group_set(	
	a_uint32_t device_id,	Device ID
	a_uint32_t trunk_id,	Trunk group ID
	a_bool_t enable,	Trunk group status (enable/disable)
	fal_pbmp_t member	Port member information
	)	
Return Value	SW_OK or error code	

2.21.1.3 fal_trunk_hash_mode_get

Definition	Get trunk hash mode on particular device.	
Prototype	sw_error_t fal_trunk_hash_mode_get(	
	a_uint32_t device_id,	Device ID
	a_uint32_t * hash_mode,	Trunk hash mode Definition as below: #define FAL_TRUNK_HASH_KEY_DA 0x1 #define FAL_TRUNK_HASH_KEY_SA 0x2 #define FAL_TRUNK_HASH_KEY_DIP 0x4 #define FAL_TRUNK_HASH_KEY_SIP 0x8
	)	
Return Value	SW_OK or error code	

2.21.1.4 fal_trunk_hash_mode_set

Definition	Set trunk hash mode on particular device.	
Prototype	sw_error_t fal_trunk_hash_mode_set(	
	a_uint32_t device_id,	Device ID
	a_uint32_t hash_mode,	Trunk hash mode Definition as below: #define FAL_TRUNK_HASH_KEY_DA 0x1 #define FAL_TRUNK_HASH_KEY_SA 0x2 #define FAL_TRUNK_HASH_KEY_DIP 0x4 #define FAL_TRUNK_HASH_KEY_SIP 0x8
	)	
Return Value	SW_OK or error code	

2.22 FAL_TYPE

2.22.1 Enumeration type documentation

2.22.1.1 enum fal_fwd_cmd_t

This enum defines several forwarding command type.

Enumeration values:

FAL_MAC_FRWRD packets are normally forwarded

FAL_MAC_DROP packets are dropped

FAL_MAC_CPY_TO_CPU packets are copied to CPU

FAL_MAC_RDT_TO_CPU packets are redirected to CPU

2.22.1.2 enum fal_fwd_cmd_t

This enum defines several forwarding command type.

Enumeration values:

FAL_MAC_FRWRD packets are normally forwarded

FAL_MAC_DROP packets are dropped

FAL_MAC_CPY_TO_CPU packets are copied to CPU

FAL_MAC_RDT_TO_CPU packets are redirected to CPU

2.22.1.3 enum fal_pt_1q_egmode_t

This enum defines packets transmitted out VLAN tagged mode.

Enumeration values:

FAL_EG_UNMODIFIED egress transmit packets unmodified

FAL_EG_UNTAGGED egress transmit packets without VLAN tag

FAL_EG_TAGGED egress transmit packets with VLAN tag

FAL_EG_HYBRID egress transmit packets in hybrid tag mode

2.22.1.4 enum fal_pt_1q_egmode_t

This enum defines packets transmitted out VLAN tagged mode.

Enumeration values:

FAL_EG_UNMODIFIED egress transmit packets unmodified

FAL_EG_UNTAGGED egress transmit packets without VLAN tag

FAL_EG_TAGGED egress transmit packets with VLAN tag

FAL_EG_HYBRID egress transmit packets in hybrid tag mode

2.23 FAL_VLAN

2.23.1 Structure documentation

2.23.1.1 fal_vlan_t

This structure defines VLAN entry.

```
typedef struct
{
```

```
    a_uint16_t vid;             //  vlan entry id
    a_uint16_t fid;             //  filter data base id
    fal_pbmp_t mem_ports;       //  member port bit map
    fal_pbmp_t tagged_ports;    //  bit map of tagged infomation for member
port
    fal_pbmp_t untagged_ports;  //  bit map of untagged infomation for
member port
    fal_pbmp_t unmodify_ports   //  bit map of unmodified infomation for
member port
    fal_pbmp_t u_ports;
    a_bool_t   learn_dis;       //  disable address learning
    a_bool_t   vid_pri_en;      //  enable 802.1p
    a_uint8_t  vid_pri;         //  vlaue of 802.1p when enable vid_pri_en
} fal_vlan_t;
```

2.23.2 Function documentation

2.23.2.1 fal_vlan_entry_append()

Definition	Append a VLAN entry on particular device.	
Prototype	`sw_error_t fal_vlan_entry_append(`	
	`a_uint32_t dev_id,`	Device ID
	`fal_vlan_t * vlan_entry`	Pointer to VLAN entry
	`);`	
Return Value	SW_OK or error code	

2.23.2.2 fal_vlan_create()

Definition	Create a VLAN entry through VLAN ID on a particular device.	
Prototype	`sw_error_t fal_vlan_create(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t vlan_id`	VLAN ID
	`);`	
Return Value	SW_OK or error code	

2.23.2.3 fal_vlan_next()

Definition	Get next VLAN entry through VLAN ID on a particular device.	
Prototype	`sw_error_t fal_vlan_next(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t vlan_id`	VLAN ID
	`fal_vlan_t * p_vlan`	Pointer to VLAN entry

	);	
Description	If the value of VID is zero, this operation will get the first entry.	
Return Value	SW_OK or error code	

2.23.2.4 fal_vlan_find()

Definition	Find a VLAN entry through VLAN ID on particular device.	
Prototype	sw_error_t fal_vlan_find(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t vlan_id	VLAN ID
	fal_vlan_t * p_vlan	Pointer to VLAN entry
	);	
Return Value	SW_OK or error code	

2.23.2.5 fal_vlan_member_update()

Definition	Update a VLAN entry member port through VLAN ID on a particular device.	
Prototype	sw_error_t fal_vlan_member_update(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t vlan_id	VLAN ID
	fal_pbmp_t member,	Port member bitmaps
	fal_pbmp_t u_member	Untagged port member bitmaps
	);	
Return Value	SW_OK or error code	

2.23.2.6 fal_vlan_delete()

Definition	Delete a VLAN entry through VLAN ID on a particular device.	
Prototype	sw_error_t fal_vlan_delete(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t vlan_id	VLAN ID
	);	
Return Value	SW_OK or error code	

2.23.2.7 fal_vlan_reset()

Definition	Reset FAL VLAN module on a particular device.	
Prototype	sw_error_t fal_vlan_reset(	
	a_uint32_t dev_id	Device ID
	);	
Return Value	SW_OK or error code	

2.23.2.8 fal_vlan_flush()

Definition	Flush all VLAN entries on a particular device.	
Prototype	`sw_error_t fal_vlan_flush(`	
	`a_uint32_t dev_id`	Device ID
	`);`	
Return Value	SW_OK or error code	

2.23.2.9 fal_vlan_init()

Definition	Init FAL VLAN module on a particular device.	
Prototype	`sw_error_t fal_vlan_init(`	
	`a_uint32_t dev_id`	Device ID
	`);`	
Return Value	SW_OK or error code	

2.23.2.10 fal_vlan_cleanup()

Definition	Clean up FAL VLAN module.	
Prototype	`sw_error_t fal_vlan_cleanup(void);`	
Return Value	SW_OK or error code	

2.23.2.11 fal_vlan_fid_set()

Definition	Set FID of a particular VLAN entry on a particular device.	
Prototype	`sw_error_t fal_vlan_fid_set(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t vlan_id`	VLAN ID
	`a_uint32_t fid`	FDB ID
	`);`	
Return Value	SW_OK or error code	

2.23.2.12 fal_vlan_fid_get()

Definition	Get FID of a particular VLAN entry on a particular device.	
Prototype	`sw_error_t fal_vlan_fid_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t vlan_id`	VLAN ID
	`a_uint32_t * fid`	FDB ID
	`);`	

Return Value	SW_OK or error code

2.23.2.13 fal_vlan_member_add()

Definition	Add a port member to a particular VLAN entry on a particular device.	
Prototype	sw_error_t fal_vlan_member_add(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t vlan_id	VLAN ID
	fal_port_t port_id,	Port ID
	fal_pt_1q_egmode_t port_info	Port tag information
	);	
Return Value	SW_OK or error code	

2.23.2.14 fal_vlan_member_del()

Definition	Delete a port member from a particular VLAN entry on a particular device.	
Prototype	sw_error_t fal_vlan_member_del(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t vlan_id	VLAN ID
	fal_port_t port_id,	Port ID
	);	
Return Value	SW_OK or error code	

2.23.2.15 fal_vlan_learning_state_set()

Definition	Set FDB learning status of a particular VLAN entry on a particular device.	
Prototype	sw_error_t fal_vlan_learning_state_set(	
	a_uint32_t dev_id,	Device ID
	a_uint32_t vlan_id	VLAN ID
	a_bool_t enable	A_TRUE or A_FALSE
	);	
Return Value	SW_OK or error code	

2.23.2.16 fal_vlan_learning_state_get()

Definition	Get FDB learning status of a particular VLAN entry on a particular device.	
Prototype	`sw_error_t fal_vlan_learning_state_get(`	
	`a_uint32_t dev_id,`	Device ID
	`a_uint32_t vlan_id`	VLAN ID
	`a_bool_t * enable`	A_TRUE or A_FALSE
	`);`	
Return Value	SW_OK or error code	