



武汉飞思灵微电子技术有限公司
Wuhan FisiLink Microelectronics Technology Co., Ltd

FSL91030(M)芯片

SDK 接口文档

版本: V1.4

武汉飞思灵微电子技术有限公司

2022 年 12 月

目 录

1	VLAN Management	1
1.1	int fsl_vlan_create	1
1.2	int fsl_vlan_port_add	1
1.3	int fsl_vlan_destroy	2
1.4	int fsl_vlan_destroy_all	2
1.5	int fsl_vlan_port_remove	3
1.6	int fsl_vlan_control_set	3
1.7	int fsl_vlan_control_get	4
1.8	int fsl_vlan_port_ingress_erps_set	4
1.9	int fsl_vlan_port_egress_erps_set	5
1.10	int fsl_vlan_port_default_action_set	5
1.11	int fsl_vlan_port_default_action_get	6
1.12	int fsl_vlan_port_egress_default_action_set	7
1.13	int fsl_vlan_port_egress_default_action_get	7
1.14	int fsl_vlan_port_default_action_delete	8
1.15	int fsl_vlan_port_egress_default_action_delete	8
1.16	int fsl_vlan_port_protocol_action_add	9
1.17	int fsl_vlan_port_protocol_action_delete	9
1.18	int fsl_vlan_port_protocol_action_get	10
1.19	int fsl_vlan_port_protocol_action_delete_all	10
1.20	int fsl_vlan_translate_action_add	11
1.21	int fsl_vlan_translate_action_get	11
1.22	int fsl_vlan_translate_action_delete	12
1.23	int fsl_vlan_translate_egress_action_add	13
1.24	int fsl_vlan_translate_egress_action_delete	13
1.25	int fsl_vlan_translate_egress_action_get	14
1.26	int fsl_vlan_translate_action_range_add	15
1.27	int fsl_vlan_translate_action_range_delete	16

1.28	int fsl_vlan_translate_action_range_get.....	16
1.29	int fsl_vlan_mac_action_add	17
1.30	int fsl_vlan_mac_action_delete	18
1.31	int fsl_vlan_mac_action_get	18
1.32	int fsl_vlan_ip_action_add	19
1.33	int fsl_vlan_ip_action_delete	20
1.34	int fsl_vlan_ip_action_get	20
1.35	int fsl_mac_ip_bind_miss_action_set.....	21
1.36	int fsl_pdu_option_set.....	21
1.37	int fsl_pdu_config_add.....	22
1.38	int fsl_pdu_config_delete.....	22
1.39	int fsl_pdu_config_get.....	23
2	Policer Configuration	24
2.1	int fsl_policing_ctl_set.....	24
2.2	int fsl_policing_ctl_get.....	25
2.3	int fsl_macro_policing_update_set	25
2.4	int fsl_macro_policing_update_get.....	26
2.5	int fsl_flow_policing_update_set.....	26
2.6	int fsl_flow_policing_update_get	27
2.7	int fsl_macro_policing_enable_set	28
2.8	int fsl_macro_policer_create.....	28
2.9	int fsl_macro_policer_delete.....	30
2.10	int fsl_flow_policer_create	31
2.11	int fsl_flow_policer_delete	32
3	Quality of Service	34
3.1	int fsl_qos_profile_create.....	34
3.2	int fsl_qos_profile_delete.....	35
3.3	int fsl_vlan_priority_map_set	35
3.4	int fsl_vlan_priority_map_get.....	36
3.5	int fsl_dscp_map_set.....	36

3.6	int fsl_vlan_priority_map_get.....	37
3.7	int fsl_pri_remark_enable_set	38
3.8	int fsl_remark_profile_create	38
3.9	int fsl_remark_profile_delete	39
3.10	int fsl_dscp_unmap_set.....	39
3.11	int fsl_vlanpri_unmap_set.....	40
3.12	int fsl_dscp_unmap_get.....	41
3.13	int fsl_vlanpri_unmap_get.....	41
4	Strom Control.....	43
4.1	int fsl_storm_control_enable_set.....	43
4.2	int fsl_storm_control_enable_get	43
4.3	int fsl_storm_control_global_set.....	44
4.4	int fsl_storm_control_global_get.....	44
4.5	int fsl_storm_control_update_set	45
4.6	int fsl_storm_control_update_get	46
4.7	int fsl_storm_control_set.....	46
4.8	int fsl_storm_control_get	47
5	Field Processor.....	49
5.1	int fsl_field_init	49
5.2	int fsl_field_detach	49
5.3	int fsl_field_group_create_mode_id	49
5.4	int fsl_field_group_destroy.....	50
5.5	int fsl_field_entry_create_id.....	51
5.6	int fsl_field_entry_install	52
5.7	int fsl_field_entry_remove.....	52
5.8	int fsl_field_entry_destroy	53
5.9	int fsl_field_entry_destroy_all(int unit)	53
5.10	int fsl_field_action_add	54
5.11	int fsl_field_action_remove	55
5.12	int fsl_field_qualify_DstIp.....	55

5.13	int fsl_field_qualify_DstIp6.....	56
5.14	int fsl_field_qualify_DstIpRange.....	57
5.15	int fsl_field_qualify_DstIp6Range.....	57
5.16	int fsl_field_qualify_DstMac.....	58
5.17	int fsl_field_qualify_OuterVlanId.....	59
5.18	int fsl_field_qualifier_delete.....	59
5.19	int fsl_field_qualify_clear.....	60
5.20	int fsl_field_qualify_data.....	61
5.21	void fsl_field_data_qualifier_init.....	61
5.22	int fsl_field_data_qualifier_create.....	62
5.23	int fsl_field_data_qualifier_destroy.....	63
5.24	int fsl_field_data_qualifier_destroy_all.....	63
5.25	void fsl_field_data_packet_format_t_init.....	64
5.26	int fsl_field_data_qualifier_packet_format_add.....	64
5.27	int fsl_field_data_qualifier_packet_format_delete.....	65
6	Trunking(Link Aggregation).....	66
6.1	int fsl_trunk_init.....	66
6.2	int fsl_trunk_detach.....	66
6.3	int fsl_trunk_create_id.....	66
6.4	int fsl_trunk_set.....	67
6.5	int fsl_trunk_destroy.....	68
6.6	int fsl_trunk_get.....	68
6.7	int fsl_trunk_psc_set.....	69
6.8	int fsl_trunk_psc_get.....	70
6.9	int fsl_trunk_hash_alg_set.....	70
6.10	int fsl_trunk_hash_alg_get.....	71
6.11	int fsl_trunk_failover_set.....	72
7	Layer 2 Address Management.....	73
7.1	int fsl_l2_addr_add.....	73
7.2	int fsl_l2_addr_delete.....	73

7.3	int fsl_l2_addr_get	74
7.4	int fsl_l2_addr_delete_by_port	74
7.5	int fsl_l2_addr_delete_by_vlan	75
7.6	int fsl_l2_age_timer_get	75
7.7	int fsl_l2_age_timer_set	76
7.8	int fsl_l2_fast_age_enable_set	76
8	Layer 2 Multicast Management	78
8.1	int fsl_mcast_create	78
8.2	int fsl_mcast_delete	78
8.3	int fsl_mcast_bitmap_del	79
8.4	int fsl_mcast_addr_add	79
8.5	int fsl_mcast_addr_remove	80
9	Port Configuration	81
9.1	int fsl_port_control_set	81
9.2	int fsl_port_control_get	81
9.3	int fsl_port_tpid_init	82
9.4	int fsl_port_tpid_add	82
9.5	int fsl_port_tpid_delete	83
9.6	int fsl_port_tpid_set	83
9.7	int fsl_ingress_port_stp_set	84
9.8	int fsl_egress_port_stp_set	84
9.9	int fsl_egress_port_stp_get	85
9.10	int fsl_ingress_stp_erps_enable_set	85
9.11	int fsl_ingress_stp_erps_enable_get	86
9.12	int fslsoc_stat_get	87
10	Set filter, send and receive packets 注：仅适用 1030M	88
10.1	fsl_rx_filter_create	88
10.2	fsl_rx_filter_list	88
10.3	fsl_rx_filter_get	89
10.4	fsl_rx_filter_destroy	89

10.5 prepare_pkt..... 90

10.6 fsl_common_tx..... 90

10.7 fsl_common_rx_register 91

10.8 fsl_common_rx_unregister 91

10.9 fsl_common_rx_start 92

10.10 fsl_common_rx_shutdown..... 92

11 修订信息 94

1 VLAN Management

1.1 int fsl_vlan_create

描述

创建一条 vlan。

语法

```
int fsl_vlan_create(int unit, fsl_vlan_t vid)
```

参数

unit	设备号
vid	vlan id

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.2 int fsl_vlan_port_add

描述

为一条 vlan 添加入向和出向的端口成员。

语法

```
int fsl_vlan_port_add(int unit, fsl_vlan_t vid, fsl_pbmp_t pbmp, fsl_pbmp_t ubmp,fsl_pbmp_t lbmp)
```

参数

unit	设备号
vid	vlan id
pbmp	待添加端口成员
upmp	待添加 untag 端口成员
lbmp	待添加 trunk 成员

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.3 int fsl_vlan_destroy

描述

删除一条指定 vlan。

语法

```
int fsl_vlan_destroy(int unit, fsl_vlan_t vid)
```

参数

unit 设备号

vid vlan id

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.4 int fsl_vlan_destroy_all

描述

删除所有 vlan。

语法

```
int fsl_vlan_destroy_all(int unit)
```

参数

unit 设备号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.5 int fsl_vlan_port_remove

描述

移除特定 vlan 的端口成员。

语法

int fsl_vlan_port_remove(int unit, fsl_vlan_t vid, fsl_pbmp_t pbmp,fsl_pbmp_t lbmp)

参数

unit	设备号
vid	vlan id
pbmp	待删除端口成员，高为有效指示
lbmp	待删除 trunk 成员，高为有效指示

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.6 int fsl_vlan_control_set

描述

设置 vlan 对应属性的值。

语法

int fsl_vlan_control_set(int unit, fsl_vlan_control_t type,fsl_vlan_t vid, int value)

参数

unit	设备号
type	vlan 属性枚举类型
vid	vlan id

value 待设置属性值

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.7 **int fsl_vlan_control_get**

描述

获取 vlan 对应属性的值

语法

int fsl_vlan_control_get(int unit, fsl_vlan_control_t type, fsl_vlan_t vid,int *value)

参数

unit 设备号

type vlan 属性枚举类型

vid vlan id

value vlan 属性值

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.8 **int fsl_vlan_port_ingress_erps_set**

描述

入方向环网保护设置。

语法

int fsl_vlan_port_ingress_erps_set(int unit, fsl_vlan_t vid, fsl_pbmp_t pbmp,fsl_pbmp_t lbmp)

参数

unit	设备号
vid	vlan id
pbmp	待保护端口成员
lbmp	待保护 trunk 成员

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.9 int fsl_vlan_port_egress_erps_set

描述

出方向环网保护设置。

语法

int fsl_vlan_port_egress_erps_set(int unit, fsl_vlan_t vid, fsl_pbmp_t pbmp,fsl_pbmp_t lbmp)

参数

unit	设备号
vid	vlan id
pbmp	待保护端口成员
lbmp	待保护 trunk 成员

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.10 int fsl_vlan_port_default_action_set

描述

创建入向端口默认的 **vlan** 处理动作

语法

```
int fsl_vlan_port_default_action_set(int unit, int port, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
port	入向物理端口
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.11 int fsl_vlan_port_default_action_get

描述

获取入向端口默认的 **vlan** 处理动作。

语法

```
int fsl_vlan_port_default_action_get(int unit, int port, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
port	入向物理端口
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.12 int fsl_vlan_port_egress_default_action_set

描述

创建出向端口默认的 vlan 处理动作。

语法

```
int fsl_vlan_port_egress_default_action_set(int unit, int port, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
port	出向物理端口
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.13 int fsl_vlan_port_egress_default_action_get

描述

获取出向端口默认的 vlan 处理动作。

语法

```
int fsl_vlan_port_egress_default_action_get(int unit, int port, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
port	出向物理端口
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_XXX 错误

1.14 int fsl_vlan_port_default_action_delete

描述

删除入向端口默认的 vlan 处理动作。

语法

```
int fsl_vlan_port_default_action_delete(int unit, int port)
```

参数

unit	设备号
port	入向物理端口

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.15 int fsl_vlan_port_egress_default_action_delete

描述

删除出向端口默认的 vlan 处理动作。

语法

```
int fsl_vlan_port_egress_default_action_delete(int unit, int port)
```

参数

unit	设备号
port	出向物理端口

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.16 int fsl_vlan_port_protocol_action_add

描述

增加端口的协议 vlan 动作。

语法

```
int fsl_vlan_port_protocol_action_add(int unit, int inIsIag, int inPort,
int ethType, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
inPort	入端口
ethType	以太网类型
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.17 int fsl_vlan_port_protocol_action_delete

描述

删除端口的协议 vlan 动作。

语法

```
int fsl_vlan_port_protocol_action_delete(int unit, int inIsIag, int inPort, int ethType)
```

参数

unit	设备号
inPort	入端口
ethType	以太网类型

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.18 int fsl_vlan_port_protocol_action_get

描述

获取端口的协议 vlan 动作。

语法

int fsl_vlan_port_protocol_action_get(int unit, int inIsIag, int inPort, int ethType, fsl_vlan_action_set_t *action)

参数

unit	设备号
inPort	入端口
ethType	以太网类型
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.19 int fsl_vlan_port_protocol_action_delete_all

描述

删除所有端口的协议 vlan 动作。

语法

int fsl_vlan_port_protocol_action_delete_all(int unit)

参数

unit	设备号
------	-----

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.20 int fsl_vlan_translate_action_add

描述

增加入向基于 vlan 的 vlan 翻译动作。

语法

int fsl_vlan_translate_action_add(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, int outer_vlan, int inner_vlan, fsl_vlan_action_set_t *action)

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan	外层 vlan id 或者 tag
inner_vlan	内层 vlan id 或者 tag
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.21 int fsl_vlan_translate_action_get

描述

获取入向基于 vlan 的 vlan 翻译动作。

语法

```
int fsl_vlan_translate_action_get(int unit,int xlate, int gport, fsl_vlan_translate_key_t key_mode, int
outer_vlan, int inner_vlan, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan	外层 vlan id 或者 tag
inner_vlan	内层 vlan id 或者 tag
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.22 int fsl_vlan_translate_action_delete

描述

删除入向基于 vlan 的 vlan 翻译动作。

语法

```
int fsl_vlan_translate_action_delete(int unit,int xlate, int gport, fsl_vlan_translate_key_t key_mode,
int outer_vlan, int inner_vlan)
```

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型

outer_vlan	外层 vlan id 或者 tag
inner_vlan	内层 vlan id 或者 tag

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.23 int fsl_vlan_translate_egress_action_add

描述

增加出向基于 vlan 的 vlan 翻译动作。

语法

int fsl_vlan_translate_egress_action_add(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, int outer_vlan, int inner_vlan, fsl_vlan_action_set_t *action)

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan	外层 vlan id 或者 tag
inner_vlan	内层 vlan id 或者 tag
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.24 int fsl_vlan_translate_egress_action_delete

描述

删除出向基于 vlan 的 vlan 翻译动作。

语法

```
int fsl_vlan_translate_egress_action_delete(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, int outer_vlan, int inner_vlan)
```

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan	外层 vlan id 或者 tag
inner_vlan	内层 vlan id 或者 tag

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.25 int fsl_vlan_translate_egress_action_get

描述

获取出向基于 vlan 的 vlan 翻译动作。

语法

```
int fsl_vlan_translate_egress_action_get(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, int outer_vlan, int inner_vlan, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号

key_mode	vlan 翻译的 key 的类型
outer_vlan	外层 vlan id 或者 tag
inner_vlan	内层 vlan id 或者 tag
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.26 int fsl_vlan_translate_action_range_add

描述

增加加入向基于 vlan range 的 vlan 翻译动作。

语法

int fsl_vlan_translate_action_range_add(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, int outer_vlan_lo, int outer_vlan_hi,int inner_vlan_lo, int inner_vlan_hi, fsl_vlan_action_set_t *action)

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan_lo	外层 vlan id 最小值
outer_vlan_hi	外层 vlan id 最大值
inner_vlan_lo	内层 vlan id 最小值
inner_vlan_hi	内层 vlan id 最大值
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.27 int fsl_vlan_translate_action_range_delete

描述

删除入向基于 vlan range 的 vlan 翻译动作。

语法

int fsl_vlan_translate_action_range_delete(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, int outer_vlan_lo, int outer_vlan_hi, int inner_vlan_lo, int inner_vlan_hi)

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan_lo	外层 vlan id 最小值
outer_vlan_hi	外层 vlan id 最大值
inner_vlan_lo	内层 vlan id 最小值
inner_vlan_hi	内层 vlan id 最大值

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.28 int fsl_vlan_translate_action_range_get

描述

获取入向基于 vlan range 的 vlan 翻译动作。

语法

```
int fsl_vlan_translate_action_range_get(int unit, int xlate, int gport, fsl_vlan_translate_key_t
key_mode, int outer_vlan_lo, int outer_vlan_hi, int inner_vlan_lo, int inner_vlan_hi, fsl_vlan_action_set_t
*action)
```

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
outer_vlan_lo	外层 vlan id 最小值
outer_vlan_hi	外层 vlan id 最大值
inner_vlan_lo	内层 vlan id 最小值
inner_vlan_hi	内层 vlan id 最大值
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.29 int fsl_vlan_mac_action_add

描述

增加加入向基于 mac 的 vlan 翻译动作。

语法

```
int fsl_vlan_mac_action_add(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode,
fsl_mac_t mac, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
------	-----

xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
mac	vlan 翻译匹配的源 mac 地址
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.30 int fsl_vlan_mac_action_delete

描述

删除入向基于 mac 的 vlan 翻译动作。

语法

int fsl_vlan_mac_action_delete(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode, fsl_mac_t mac)

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
mac	vlan 翻译匹配的源 mac 地址

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.31 int fsl_vlan_mac_action_get

描述

获取入向基于 mac 的 vlan 翻译动作。

语法

```
int fsl_vlan_mac_action_get(int unit, int xlate, int gport, fsl_vlan_translate_key_t key_mode,
fsl_mac_t mac, fsl_vlan_action_set_t *action)
```

参数

unit	设备号
xlate	xlate0 和 xlate1 的选择
gport	虚拟或物理端口号
key_mode	vlan 翻译的 key 的类型
mac	vlan 翻译匹配的源 mac 地址
action	vlan tag 的动作设置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.32 int fsl_vlan_ip_action_add

描述

增加基于 ip 的 mac 绑定动作。

语法

```
int fsl_vlan_ip_action_add(int unit, fsl_vlan_ip_t *vlan_ip, fsl_mac_t mac, int gport)
```

参数

unit	设备号
vlan_ip	ip 匹配的相关配置
mac	与 ip 绑定的源 mac 地址

gport 与 ip 绑定的虚拟或物理端口号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.33 int fsl_vlan_ip_action_delete

描述

删除基于 ip 的 mac 绑定动作。

语法

int fsl_vlan_ip_action_delete(int unit, fsl_vlan_ip_t *vlan_ip)

参数

unit 设备号

vlan_ip ip 匹配的相关配置

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

1.34 int fsl_vlan_ip_action_get

描述

获取基于 ip 的 mac 绑定动作。

语法

int fsl_vlan_ip_action_get(int unit, fsl_vlan_ip_t *vlan_ip, fsl_mac_t *mac, int *is_lag, int *lport)

参数

unit 设备号

vlan_ip ip 匹配的相关配置

mac	与 ip 绑定的源 mac 地址
is_lag	与 ip 绑定的端口 lag 信息
lport	与 ip 绑定的物理端口号

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.35 int fsl_mac_ip_bind_miss_action_set

描述

设置 mac、ip 绑定失败时的全局报文处理动作。

语法

int fsl_mac_ip_bind_miss_action_set(int unit, int bypassen, int trapen, int dropen)

参数

unit	设备号
bypassen	bypass 使能
trapen	trap 使能
dropen	drop 使能

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.36 int fsl_pdu_option_set

描述

设置 l2pdu 全局处理动作。

语法

```
int fsl_pdu_option_set(int unit, uint64_t pdu_option)
```

参数

unit	设备号
pdu_option	全局 pdu 处理行为配置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.37 int fsl_pdu_config_add

描述

增加 pdu 配置条目。

语法

```
int fsl_pdu_config_add(int unit, int index, fsl_pdu_config_t *pdu_cfg)
```

参数

unit	设备号
index	pdu 配置条目 id
pdu_cfg	pdu 内容及掩码配置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.38 int fsl_pdu_config_delete

描述

删除 pdu 配置条目。

语法

```
int fsl_pdu_config_delete(int unit, int index)
```

参数

unit	设备号
index	pdu 配置条目 id

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

1.39 int fsl_pdu_config_get

描述

获取 pdu 条目的具体配置。

语法

```
int fsl_pdu_config_get(int unit, int index, fsl_pdu_config_t *pdu_cfg)
```

参数

unit	设备号
index	pdu 配置条目 id
pdu_cfg	pdu 内容及掩码配置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

2 Policer Configuration

2.1 int fsl_policing_ctl_set

描述

设置 policing 的全局配置。

语法

```
int fsl_policing_ctl_set(int unit, int direct, fsl_policing_ctl_t *pol_ctl)
```

参数

unit	设备号
direct	policing 方向，入向和出向
pol_ctl	policing 的全局配置数据
typedef struct fsl_policing_ctl_s {	
uint16_t	pktLenUsePkt; //当基于包做 policing 时的等效包长，字节为单位
uint8_t	macroPktBytes; //macro 模式
	// 0x0: 基于字节做 policing
	// 0x1: 基于包做 policing
uint8_t	flowPktBytes; //flow 模式
	// 0x0: 基于字节做 policing
	// 0x1: 基于包做 policing
uint8_t	meterGran; // 控制粒度，默认 0
uint8_t	preambleLen; //帧间隔和前导码等效包长，字节为单位
} fsl_policing_ctl_t;	

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_XXX 错误

2.2 int fsl_policing_ctl_get

描述

获取 policing 的全局配置。

语法

```
int fsl_policing_ctl_get(int unit, int direct, fsl_policing_ctl_t *pol_ctl)
```

参数

unit	设备号
direct	policing 方向，入向和出向
pol_ctl	policing 的全局配置数据

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

2.3 int fsl_macro_policing_update_set

描述

设置基于端口 policing 的全局更新配置。

语法

```
int fsl_macro_policing_update_set(int unit, int direct, fsl_policing_update_t *pol_upd)
```

参数

unit	设备号
direct	policing 方向，入向和出向
pol_upd	policing 更新配置数据

```
typedef struct fsl_policing_update_s {
```



```
uint8_t    updEn;           //填充令牌桶使能

uint16_t   updMaxIndex;     //更新最大条目数

uint8_t     timer0Num;      //令牌桶刷新周期参数

uint8_t     timer1Num;      //令牌桶刷新周期参数

uint16_t    timer0;         //令牌桶刷新周期参数

uint16_t    timer1;         //令牌桶刷新周期参数

} fsl_policing_update_t;
```

返回值

```
FSLRAL_E_NONE    成功

FSLRAL_E_XXX      错误
```

2.4 int fsl_macro_policing_update_get

描述

获取基于端口 policing 的全局更新配置。

语法

```
int fsl_macro_policing_update_get(int unit, int direct, fsl_policing_update_t *pol_upd)
```

参数

```
unit          设备号

direct        policing 方向，入向和出向

pol_upd       policing 更新配置数据
```

返回值

```
FSLRAL_E_NONE    成功

FSLRAL_E_XXX      错误
```

2.5 int fsl_flow_policing_update_set

描述

设置基于流 policing 的全局更新配置。

语法

```
int fsl_flow_policing_update_set(int unit, int direct, fsl_policing_update_t *pol_upd)
```

参数

```
unit          设备号

direct        policing 方向，入向和出向

pol_upd       policing 更新配置数据

typedef struct fsl_policing_update_s {
    uint8_t    updEn;          //填充令牌桶使能
    uint16_t    updMaxIndex;    //更新最大条目数
    uint8_t     timer0Num;      //令牌桶刷新周期参数
    uint8_t     timer1Num;      //令牌桶刷新周期参数
    uint16_t    timer0;         //令牌桶刷新周期参数
    uint16_t    timer1;         //令牌桶刷新周期参数
} fsl_policing_update_t;
```

返回值

```
FSLRAL_E_NONE    成功
FSLRAL_E_XXX      错误
```

2.6 int fsl_flow_policing_update_get

描述

获取基于流 policing 的全局更新配置。

语法

```
int fsl_flow_policing_update_get(int unit, int direct, fsl_policing_update_t *pol_upd)
```

参数

unit	设备号
direct	policing 方向，入向和出向
pol_upd	policing 更新配置数据

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

2.7 int fsl_macro_policing_enable_set

描述

设置端口 policing 的使能与否。

语法

```
int fsl_macro_policing_enable_set(int unit, int direct, int pol_id, int pol_en)
```

参数

unit	设备号
direct	policing 方向，入向和出向
pol_id	policing id，此处为物理端口号
pol_en	policing 使能标志位 0/1

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

2.8 int fsl_macro_policer_create

描述

创建端口 policing。

语法

```
int fsl_macro_policer_create(int unit, int direct, int pol_id, fsl_policer_config_t *pol_cfg)
```

参数

unit	设备号
direct	policing 方向，入向和出向
pol_id	policing id，此处为物理端口号
pol_cfg	policing 配置数据
typedef struct fsl_policer_config_s {	
fsl_policer_mode_t	mode; //fslPolicerModeTrTcm, /* RFC 2698 */ //fslPolicerModeTrTcmDs, /* RFC 4115 */ //fslPolicerModeSrTcm /* RFC 2697 */
fsl_color_sense_t	colorSense; //色盲和色敏感
uint8_t	globalCFlag; //全局 C 桶向 E 桶耦合标志
fsl_meter_sharing_mode_t	sharingMode; //共享模式，0：不共享模式 1：FSL_MIN_ONLY, //2：FSL_MAX_ONLY，3：FSL_MIN_MAX
uint32_t	cir; //c 桶令牌填充速率 kb
uint32_t	cirMax; //c 桶最大添加速率 kb
uint32_t	cbs; //c 桶桶深 kb
uint32_t	eir; //e 桶令牌填充速率 kb
uint32_t	eirMax; //e 桶最大添加速率 kb
uint32_t	ebs; //e 桶桶深
uint8_t	redPri; //红色报文（color == 2'b00)的新的 pri 值

uint8_t	yellowPri; //黄色报文 (color == 2'b01)的新的 pri 值
uint8_t	greenPri; //绿色报文 (color == 2'b10)的新的 pri 值
uint8_t	rChangePri; //红色报文更新 pri 使能
uint8_t	yChangePri; //黄色报文更新 pri 使能
uint8_t	gChangePri; //绿色报文更新 pri 使能
uint8_t	rChangeDrop; //红色报文 drop 使能
uint8_t	yChangeDrop; //黄色报文 drop 使能
uint8_t	gChangeDrop; //绿色报文 drop 使能
} fsl_policer_config_t;	

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

2.9 int fsl_macro_policer_delete

描述

删除端口 policing。

语法

int fsl_macro_policer_delete(int unit, int direct, int pol_id)

参数

unit	设备号
direct	policing 方向，入向和出向
pol_id	policing id，此处为物理端口号

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_XXX 错误

2.10 int fsl_flow_policer_create

描述

创建流 policing。

语法

int fsl_flow_policer_create(int unit, int direct, int pol_id, fsl_policer_config_t *pol_cfg)

参数

unit	设备号
direct	policing 方向，入向和出向
pol_id	policing id (-1 表示系统分配)
pol_cfg	policing 配置数据
typedef struct fsl_policer_config_s {	
fsl_policer_mode_t	mode; //fslPolicerModeTrTcm, /* RFC 2698 */
	//fslPolicerModeTrTcmDs, /* RFC 4115 */
	//fslPolicerModeSrTcm /* RFC 2697 */
fsl_color_sense_t	colorSense; //色盲和色敏感
uint8_t	globalCFlag; //全局 C 桶向 E 桶耦合标志
fsl_meter_sharing_mode_t	sharingMode; //共享模式，0：不共享模式 1：FSL_MIN_ONLY, //2：FSL_MAX_ONLY, 3：FSL_MIN_MAX
uint32_t	cir; //c 桶令牌填充速率 kb
uint32_t	cirMax; //c 桶最大添加速率 kb
uint32_t	cbs; //c 桶桶深 kb
uint32_t	eir; ///e 桶令牌填充速率 kb
uint32_t	eirMax; //e 桶最大添加速率 kb

uint32_t	ebs; //e 桶桶深
uint8_t	redPri; //红色报文 (color == 2'b00)的新的 pri 值
uint8_t	yellowPri; //黄色报文 (color == 2'b01)的新的 pri 值
uint8_t	greenPri; //绿色报文 (color == 2'b10)的新的 pri 值
uint8_t	rChangePri; //红色报文更新 pri 使能
uint8_t	yChangePri; //黄色报文更新 pri 使能
uint8_t	gChangePri; //绿色报文更新 pri 使能
uint8_t	rChangeDrop; //红色报文 drop 使能
uint8_t	yChangeDrop; //黄色报文 drop 使能
uint8_t	gChangeDrop; //绿色报文 drop 使能
} fsl_policer_config_t;	

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

2.11 int fsl_flow_policer_delete

描述

删除流 policing。

语法

int fsl_flow_policer_delete(int unit, int direct, int pol_id)

参数

unit	设备号
direct	policing 方向，入向和出向

pol_id policing id

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3 Quality of Service

3.1 int fsl_qos_profile_create

描述

创建优先级映射模板。

语法

```
int fsl_qos_profile_create(int unit, int qos_pro_index, fsl_qos_profile_t *qos_profile)
```

参数

```
unit                设备号

qos_pro_index       映射模板序号（-1 表示系统分配）

qos_profile         模板配置

typedef struct fsl_qos_profile_s {

    uint8_t    useDefault; //1:使用默认优先级
                                //0:其他

    uint8_t    useL2Info; //0x0:优先使用 L3 头信息
                                //0x1:使用 L2 头信息

    uint8_t    trustCtag; //0x0: 优先使用使用 STAG
                                //0x1: 使用 CTAG

    uint8_t    phbPtr; //PHB 表索引指针高 6 位,可用逻辑端口号

    uint8_t    use_flag; //默认 0，不需要配置

} fsl_qos_profile_t;
```

返回值

```
FSLRAL_E_NONE    成功

FSLRAL_E_XXX     错误
```

3.2 int fsl_qos_profile_delete

描述

删除优先级映射模板。

语法

```
int fsl_qos_profile_delete(int unit, int qos_pro_index)
```

参数

unit	设备号
qos_pro_index	映射模板序号

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.3 int fsl_vlan_priority_map_set

描述

设置 vlan 优先级到内部优先级的映射。

语法

```
int fsl_vlan_priority_map_set(int unit, int qos_pro_index, int pkt_pri, int cfi, int internal_pri, fsl_color_t color)
```

参数

unit	设备号
qos_pro_index	映射模板序号
pkt_pri	报文优先级
cfi	报文 cfi
internal_pri	内部优先级
color	颜色

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.4 int fsl_vlan_priority_map_get

描述

获取 vlan 优先级映射配置。

语法

```
int fsl_vlan_priority_map_get(int unit, int qos_pro_index, int pkt_pri, int cfi, int *internal_pri, fsl_color_t *color)
```

参数

unit	设备号
qos_pro_index	映射模板序号
pkt_pri	报文优先级
cfi	报文 cfi
internal_pri	内部优先级
color	颜色

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.5 int fsl_dscp_map_set

描述

设置 dscp 到内部优先级的映射。

语法

```
int fsl_dscp_map_set(int unit, int qos_pro_index, int dscp, int internal_pri, fsl_color_t color)
```

参数

unit	设备号
qos_pro_index	映射模板序号
pkt_pri	报文 dscp
internal_pri	内部优先级
color	颜色

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.6 int fsl_vlan_priority_map_get

描述

获取 dscp 映射的配置。

语法

```
int fsl_vlan_priority_map_get(int unit, int qos_pro_index, int pkt_pri, int cfi, int *internal_pri, fsl_color_t *color)
```

参数

unit	设备号
qos_pro_index	映射模板序号
pkt_pri	报文 dscp
internal_pri	内部优先级
color	颜色

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.7 int fsl_pri_remark_enable_set

描述

设置端口内部优先级重标记使能。

语法

```
int fsl_pri_remark_enable_set(int unit, int port, int rmk_en)
```

参数

unit	设备号
port	物理端口号
rmk_en	重标记使能标志位 0/1

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.8 int fsl_remark_profile_create

描述

创建重标记模板。

语法

```
int fsl_remark_profile_create(int unit, int rmkPriPtr, fsl_rmk_info_t *rmk_info)
```

参数

unit	设备号
rmkPriPtr	重标记模板序号
rmk_info	重标记配置信息

```
typedef struct fsl_rmk_info_s {  
    uint8_t    index; //重映射的索引
```

```
uint8_t    scosRmkEn; //重映射 scos 使能

uint8_t    ccosRmkEn; //重映射 ccos 使能

uint8_t    brgChgTos; //重映射 tos 使能

uint8_t    onlyChgDscp; //只修改 dscp

uint8_t    use_flag; //默认 0，不需要配置

} fsl_rmk_info_t;
```

返回值

```
FSLRAL_E_NONE    成功

FSLRAL_E_XXX     错误
```

3.9 int fsl_remark_profile_delete

描述

删除重标记模板。

语法

```
int fsl_remark_profile_delete(int unit, int rmkPriPtr)
```

参数

```
unit          设备号

rmkPriPtr     重标记模板序号
```

返回值

```
FSLRAL_E_NONE    成功

FSLRAL_E_XXX     错误
```

3.10 int fsl_dscp_unmap_set

描述

设置报文 dscp 重标记。

语法

```
int fsl_dscp_unmap_set(int unit, int rmkPriPtr, int internal_pri, fsl_color_t color, int pkt_dscp)
```

参数

unit	设备号
internal_pri	内部优先级
color	颜色
pkt_dscp	报文 dscp

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.11 int fsl_vlanpri_unmap_set

描述

设置报文 vlan 优先级重标记

语法

```
int fsl_vlanpri_unmap_set(int unit, int rmkPriPtr, int internal_pri, fsl_color_t color, int cos, int cfi)
```

参数

unit	设备号
internal_pri	内部优先级
color	颜色
cos	报文 vlan pri
cfi	报文 vlan cfi

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_XXX 错误

3.12 int fsl_dscp_unmap_get

描述

获取报文 dscp 重标记配置。

语法

int fsl_dscp_unmap_get(int unit, int rmkPriPtr, int internal_pri, fsl_color_t color, int *pkt_dscp)

参数

unit	设备号
internal_pri	内部优先级
color	颜色
pkt_dscp	报文 dscp

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

3.13 int fsl_vlanpri_unmap_get

描述

获取报文 vlan 优先级重标记配置。

语法

int fsl_vlanpri_unmap_get(int unit, int rmkPriPtr, int internal_pri, fsl_color_t color, int *cos, int *cfi)

参数

unit	设备号
internal_pri	内部优先级
color	颜色

cos	报文 vlan pri
cfi	报文 vlan cfi

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4 Strom Control

4.1 int fsl_storm_control_enable_set

描述

设置风暴控制使能与否。

语法

```
int fsl_storm_control_enable_set(int unit, fsl_storm_control_mode_t mode, int arg,
fsl_forward_type_t fwd_type, int enable)
```

参数

unit	设备号
mode	风暴控制模式（3 种：system，port，forward id）
arg	风暴控制 id，对应三种模式
fwd_type	转发类型（单播，组播，广播）
enable	使能标识

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.2 int fsl_storm_control_enable_get

描述

获取风暴控制的使能标识。

语法

```
int fsl_storm_control_enable_get(int unit, fsl_storm_control_mode_t mode, int arg,
fsl_forward_type_t fwd_type, int *enable)
```

参数

unit	设备号
------	-----

mode	风暴控制模式（3 种：system，port，forward id）
arg	风暴控制 id，对应三种模式
fwd_type	转发类型（单播，组播，广播）
enable	使能标识

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.3 int fsl_storm_control_global_set

描述

设置风暴控制的全局配置。

语法

int fsl_storm_control_global_set(int unit, uint8_t meter_gran, uint8_t preamble_len)

参数

unit	设备号
meter_gran	控制粒度
preamble_len	前导码和帧间隙的等效包长

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.4 int fsl_storm_control_global_get

描述

获取风暴控制的全局配置。

语法

```
int fsl_storm_control_global_get(int unit, uint8_t *meter_gran, uint8_t *preamble_len)
```

参数

unit	设备号
meter_gran	控制粒度
preamble_len	前导码和帧间隙的等效包长

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.5 int fsl_storm_control_update_set

描述

设置风暴控制的更新配置。

语法

```
int fsl_storm_control_update_set(int unit, fsl_storm_control_mode_t mode, fsl_storm_ctl_global_t *global_ctl)
```

参数

unit	设备号
mode	风暴控制模式
global_ctl	风暴控制的更新配置

```
typedef struct fsl_storm_ctl_global_s {  
    uint32_t pktLenUsePkt; //当基于包做 metering 时的等效包长  
    uint8_t updEn; //填充令牌桶使能  
    uint32_t maxUpdIdx; //填充令牌桶的最大 index  
    uint32_t delayInterval; //填充令牌桶周期  
} fsl_storm_ctl_global_t;
```

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.6 int fsl_storm_control_update_get

描述

获取风暴控制的更新配置。

语法

```
int fsl_storm_control_update_get(int unit, fsl_storm_control_mode_t mode, fsl_storm_ctl_global_t *global_ctl)
```

参数

unit	设备号
mode	风暴控制模式
global_ctl	风暴控制的更新配置

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.7 int fsl_storm_control_set

描述

设置风暴控制的类型、限速值和突发尺寸。

语法

```
int fsl_storm_control_set(int unit, fsl_storm_control_mode_t mode, int arg, fsl_forward_type_t fwd_type, fsl_storm_policing_type_t pol_type, uint32_t limit, uint32_t burst_size)
```

参数

unit	设备号
------	-----

mode	风暴控制模式
arg	风暴控制 id，对应三种模式
fwd_type	转发类型
pol_type	policing 类型（包和字节）
limit	限速值
burst_size	突发尺寸

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

4.8 int fsl_storm_control_get

描述

获取风暴控制的类型、限速值和突发尺寸。

语法

```
int fsl_storm_control_get(int unit, fsl_storm_control_mode_t mode, int arg, fsl_forward_type_t fwd_type, fsl_storm_policing_type_t *pol_type, uint32_t *limit, uint32_t *burst_size)
```

参数

unit	设备号
mode	风暴控制模式
arg	风暴控制 id，对应三种模式
fwd_type	转发类型
pol_type	policing 类型（包和字节）
limit	限速值
burst_size	突发尺寸

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

5 Field Processor

5.1 int fsl_field_init

描述

fp 初始化，需最先被调用。

语法

```
int fsl_field_init(int unit)
```

参数

unit 设备号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

5.2 int fsl_field_detach

描述

清除所有 fp 相关配置，包含软件资源和硬件配置。

语法

```
int fsl_field_detach(int unit)
```

参数

unit 设备号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

5.3 int fsl_field_group_create_mode_id

描述

创建 field 组。

语法

```
int fsl_field_group_create_mode_id(int unit, fsl_field_qset_t qset, int pri, fsl_field_group_mode_t mode,uint16_t entry_num,fsl_field_key_tp_t key_tp,fsl_field_group_t group)
```

参数

unit	设备号
qset	指定配置的哪一个模块。仅支持 fslFieldQualifyStageIngress、fslFieldQualifyStageEgress、fslFieldQualifyStageLookup、fslFieldQualifyStageLookupEgress
pri	暂未使用 mode、entry_num、Key_tp
group	组 ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_RESOURCE	硬件资源不可用
FSLRAL_E_PARAM	参数错误
FSLRAL_E_EXISTS	group ID 或 key tp 已存在
FSLRAL_E_MEMORY	分配内存失败
FSLRAL_E_XXX	其他错误

5.4 int fsl_field_group_destroy

描述

销毁 field 组。调用此函数前，该 group 下的所有 entries 必须被销毁。

语法

```
int fsl_field_group_destroy(int unit, fsl_field_group_t group)
```

参数

unit	设备号
group	组 ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	该组不存在
FSLRAL_E_BUSY	该组中存在 entry
FSLRAL_E_XXX	其他错误

5.5 int fsl_field_entry_create_id

描述

创建 entry。

语法

```
int fsl_field_entry_create_id(int unit,fsl_field_group_t group,fsl_field_entry_t entry)
```

参数

unit	设备号
group	组 ID
entry	entry ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	组不存在
FSLRAL_E_EXISTS	entry ID 已存在
FSLRAL_E_MEMORY	分配内存失败

FSLRAL_E_RESOURCE entry 条目已满

5.6 int fsl_field_entry_install

描述

安装 entry 到硬件表项。entry 安装之前应使用 fsl_field_qualify_xxx 添加 qualify，使用 fsl_field_action_add 添加 actions。

语法

```
int fsl_field_entry_install(int unit, fsl_field_entry_t entry)
```

参数

unit	设备号
entry	entry ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_XXX	其他错误

5.7 int fsl_field_entry_remove

描述

从硬件表项中删除 entry。

语法

```
int fsl_field_entry_remove(int unit, fsl_field_entry_t entry)
```

参数

unit	设备号
entry	entry ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_XXX	其他错误

5.8 int fsl_field_entry_destroy

描述

销毁 entry。销毁软件 entry 相关资源，如果该 entry 已经安装至硬件表项，则函数会调用 fsl_field_entry_remove 接口清除硬件表项中的 entry，释放资源。

语法

```
int fsl_field_entry_destroy(int unit, fsl_field_entry_t entry)
```

参数

unit	设备号
entry	entry ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_XXX	其他错误

5.9 int fsl_field_entry_destroy_all(int unit)

描述

销毁所有 entry。

语法

```
int fsl_field_entry_destroy_all(int unit)
```

参数

unit 设备号

返回值

- FSLRAL_E_NONE 成功
- FSLRAL_E_INIT 未初始化
- FSLRAL_E_XXX 其他错误

5.10 int fsl_field_action_add

描述

添加 action 到 entry。一条 entry 可以添加多个 action。

语法

int fsl_field_action_add(int unit,fsl_field_entry_t entry,fsl_field_action_t action,uint32_t param0, uint32_t param1)

参数

- unit 设备号
- entry entry ID
- action action 类型
- param0 action 值。不使用时赋值为 0。
- param1 action 值。不使用时赋值为 0。

返回值

- FSLRAL_E_NONE 成功
- FSLRAL_E_INIT 未初始化
- FSLRAL_E_NOT_FOUND entry 不存在
- FSLRAL_E_MEMORY 分配内存失败
- FSLRAL_E_UNAVAIL 不支持该 action 类型
- FSLRAL_E_CONFIG action 类型冲突

FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.11 int fsl_field_action_remove

描述

entry 中删除 action。

语法

int fsl_field_action_remove(int unit, fsl_field_entry_t entry,fsl_field_action_t action)

参数

unit	设备号
entry	entry ID
action	action 类型

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.12 int fsl_field_qualify_DstIp

描述

添加 qualification 到 entry，匹配报文的 ipv4 地址。

语法

int fsl_field_qualify_DstIp(int unit, fsl_field_entry_t entry,fsl_ip_t data,fsl_ip_t mask)

参数

unit	设备号
entry	entry ID
data	数据
mask	掩码

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.13 int fsl_field_qualify_DstIp6

描述

添加 qualification 到 entry，匹配报文的 ipv6 地址。

语法

int fsl_field_qualify_DstIp6(int unit,fsl_field_entry_t entry,fsl_ip6_t data,fsl_ip6_t mask)

参数

unit	设备号
entry	entry ID
data	数据
mask	掩码

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化

FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.14 int fsl_field_qualify_DstIpRange

描述

添加 qualification 到 entry。范围匹配报文 ipv4 地址,支持 4 种范围配置，且配置范围值不能重合。

语法

int fsl_field_qualify_DstIpRange(int unit, fsl_field_entry_t entry,fsl_ip_t ipL, fsl_ip_t ipH)

参数

unit	设备号
entry	entry ID
ipL	最小 ipv4 地址
ipH	最大 ipv4 地址

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.15 int fsl_field_qualify_DstIp6Range

描述

添加 qualification 到 entry。范围匹配报文 ipv6 地址,支持 2 种范围配置，且配置范围值不能重合。

语法

```
int fsl_field_qualify_DstIp6Range(int unit, fsl_field_entry_t entry,fsl_ip6_t ipL, fsl_ip6_t ipH)
```

参数

unit	设备号
entry	entry ID
ipL	最小 ipv6 地址
ipH	最大 ipv6 地址

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.16 int fsl_field_qualify_DstMac

描述

添加 qualification 到 entry，匹配报文的目的 Mac 地址。

语法

```
int fsl_field_qualify_DstMac(int unit, fsl_field_entry_t entry,fsl_mac_t data,fsl_mac_t mask)
```

参数

unit	设备号
entry	entry ID
data	数据
mask	掩码

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.17 int fsl_field_qualify_OuterVlanId

描述

添加 qualification 到 entry，匹配报文的外层 Vlan ID。

语法

int fsl_field_qualify_OuterVlanId(int unit, fsl_field_entry_t entry, fsl_vlan_t data,fsl_vlan_t mask)

参数

unit	设备号
entry	entry ID
data	数据
mask	掩码

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.18 int fsl_field_qualifier_delete

描述

从指定 entry 中删除 qualification。

语法

```
int fsl_field_qualifier_delete(int unit, fsl_field_entry_t entry, fsl_field_qualify_t qual)
```

参数

unit	设备号
entry	entry ID
qual	qualifier ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.19 int fsl_field_qualify_clear

描述

从指定 entry 中删除所有的 qualification。

语法

```
int fsl_field_qualify_clear(int unit, fsl_field_entry_t entry)
```

参数

unit	设备号
entry	entry ID

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	entry 不存在
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.20 int fsl_field_qualify_data

描述

添加数据 qualification 到 entry，匹配 udf 值。

语法

int fsl_field_qualify_data(int unit,fsl_field_entry_t eid, int qual_id,uint8_t *data,uint8_t *mask, uint16_t length)

参数

unit	设备号
eid	entry ID
qual_id	同 fsl_field_data_qualifier_t 结构体的 qual_id
data	数据
mask	掩码
length	匹配数据的长度

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	qual ID 未创建
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

5.21 void fsl_field_data_qualifier_init

描述

初始化 `fsl_field_data_qualifier_t` 结构体。

语法

```
void fsl_field_data_qualifier_init(fsl_field_data_qualifier_t *data_qual)
```

参数

`data_qual` 结构体指针

返回值

无返回值。

5.22 `int fsl_field_data_qualifier_create`

描述

创建一个数据 `qualifier`。仅创建软件资源。

语法

```
int fsl_field_data_qualifier_create(int unit, fsl_field_data_qualifier_t *data_qualifier)
```

参数

`unit` 设备号
`data_qualifier` 结构体指针

返回值

<code>FSLRAL_E_NONE</code>	成功
<code>FSLRAL_E_INIT</code>	未初始化
<code>FSLRAL_E_EXISTS</code>	<code>qual ID</code> 存在
<code>FSLRAL_E_FULL</code>	支持的 <code>data qualifier</code> 条数已满
<code>FSLRAL_E_PARAM</code>	参数错误
<code>FSLRAL_E_XXX</code>	其他错误

5.23 int fsl_field_data_qualifier_destroy

描述

销毁一个数据 qualifier。

语法

```
int fsl_field_data_qualifier_destroy(int unit, int qual_id)
```

参数

unit	设备号
qual_id	qual ID

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	qual ID 未创建
FSLRAL_E_XXX	其他错误

5.24 int fsl_field_data_qualifier_destroy_all

描述

销毁所有数据 qualifier。

语法

```
int fsl_field_data_qualifier_destroy_all(int unit)
```

参数

unit	设备号
------	-----

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化

FSLRAL_E_XXX 其他错误

5.25 void fsl_field_data_packet_format_t_init

描述

初始化_field_data_packet_format_t 结构体。

语法

void fsl_field_data_packet_format_t_init (_field_data_packet_format_t *packet_format)

参数

packet_format支持 data qualifier 的数据包格式

返回值

无返回值。

5.26 int fsl_field_data_qualifier_packet_format_add

描述

将 qual id 对应的软件数据写入硬件表项。

语法

int fsl_field_data_qualifier_packet_format_add(int unit, int qual_id,_field_data_packet_format_t *packet_format)

参数

unit	设备号
qual_id	qual ID
packet_format	暂未使用（不能为 NULL）

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	qual ID 未创建

FSLRAL_E_XXX 其他错误

5.27 **int fsl_field_data_qualifier_packet_format_delete**

描述

将 qual id 对应的硬件表项删除。

语法

int fsl_field_data_qualifier_packet_format_delete(int unit,int qual_id,_field_data_packet_format_t *packet_format)

参数

unit	设备号
qual_id	qual ID
packet_format	暂未使用（不能为 NULL）

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	未初始化
FSLRAL_E_NOT_FOUND	qual ID 未创建
FSLRAL_E_XXX	其他错误

6 Trunking(Link Aggregation)

6.1 int fsl_trunk_init

描述

trunk 初始化。需最先被调用。

语法

```
int fsl_trunk_init(int unit)
```

参数

unit 设备号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

6.2 int fsl_trunk_detach

描述

清除所有 trunk 相关配置，包含软件资源和硬件配置。

语法

```
int fsl_trunk_detach(int unit)
```

参数

unit 设备号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

6.3 int fsl_trunk_create_id

描述

创建 trunk ID。

语法

```
int fsl_trunk_create_id(int unit, fsl_trunk_t tid)
```

参数

unit	设备号
tid	trunk ID (0~7)

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_EXISTS	trunk ID 已存在
FSLRAL_E_XXX	其他错误

6.4 int fsl_trunk_set

描述

指定 trunk 组中的端口及 hash key 和 hash 算法类型,默认 hash key 为源 Mac 地址和目的 Mac 地址。如果 trunk 组中已经有成员端口, 原有成员端口将被替换。

语法

```
int fsl_trunk_set(int unit, fsl_trunk_t tid, fsl_trunk_add_info_t *add_info)
```

参数

unit	设备号
tid	trunk ID (0~7)
add_info	结构体指针

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_NOT_FOUND	trunk ID 不存在
FSLRAL_E_XXX	其他错误

6.5 int fsl_trunk_destroy

描述

销毁 trunk 组。

语法

int fsl_trunk_destroy(int unit, fsl_trunk_t tid)

参数

unit	设备号
tid	trunk ID (0~7)

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_NOT_FOUND	trunk ID 不存在
FSLRAL_E_XXX	其他错误

6.6 int fsl_trunk_get

描述

获取 trunk 组中的端口成员及 hash key 和 hash 算法类型。

语法

int fsl_trunk_get(int unit, fsl_trunk_t tid,fsl_trunk_add_info_t *t_data)

参数

unit	设备号
tid	trunk ID (0~7)
t_data	保存 trunk 信息

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_XXX	其他错误

6.7 int fsl_trunk_psc_set

描述

设置 trunk 组的 hash key。

语法

int fsl_trunk_psc_set(int unit, fsl_trunk_t tid, int psc)

参数

unit	设备号
tid	trunk ID (0~7)
psc	hash key

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_NOT_FOUND	trunk ID 不存在

FSLRAL_E_XXX 其他错误

6.8 **int fsl_trunk_psc_get**

描述

获取 trunk 组的 hash key。

语法

```
int fsl_trunk_psc_get(int unit, fsl_trunk_t tid, int *psc)
```

参数

unit 设备号

tid trunk ID（0~7）

psc 获取的 hash key

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_INIT trunk 未初始化

FSLRAL_E_PARAM 参数错误

FSLRAL_E_NOT_FOUND trunk ID 不存在

FSLRAL_E_XXX 其他错误

6.9 **int fsl_trunk_hash_alg_set**

描述

设置 trunk 组的 hash 算法类型。

语法

```
int fsl_trunk_hash_alg_set(int unit,fsl_trunk_t tid,fsl_trunk_hash_alg_t alg)
```

参数

unit	设备号
tid	trunk ID（0~7）
alg	hash 算法类型

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_NOT_FOUND	trunk ID 不存在
FSLRAL_E_XXX	其他错误

6.10 int fsl_trunk_hash_alg_get

描述

获取 trunk 组的 hash 算法类型。

语法

int fsl_trunk_hash_alg_get(int unit, fsl_trunk_t tid, int *alg)

参数

unit	设备号
tid	trunk ID（0~7）
alg	获取的 hash 算法类型

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_NOT_FOUND	trunk ID 不存在

FSLRAL_E_XXX 其他错误

6.11 **int fsl_trunk_failover_set**

描述

设置 trunk 组端口失效分担使能。

语法

int fsl_trunk_failover_set(int unit, fsl_trunk_t tid, int able)

参数

unit	设备号
tid	trunk ID（0~7）
able	1:开启 0:关闭

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_INIT	trunk 未初始化
FSLRAL_E_PARAM	参数错误
FSLRAL_E_NOT_FOUND	trunk ID 不存在
FSLRAL_E_EMPTY	trunk 组未设置
FSLRAL_E_XXX	其他错误

7 Layer 2 Address Management

7.1 int fsl_l2_addr_add

描述

添加一条 mac 地址。

语法

```
int fsl_l2_addr_add(int unit, fsl_l2_addr_t *l2addr)
```

参数

unit	设备号
l2addr	l2 地址

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

7.2 int fsl_l2_addr_delete

描述

删除一条 mac 地址。

语法

```
int fsl_l2_addr_delete(int unit, fsl_mac_t mac, fsl_vlan_t vid,fslral_mem_t mem,uint32_t index)
```

参数

unit	设备号
mac	mac 地址
vid	vlan id
mem	mac key 表类型
index	mac key 表索引

返回值

- FSLRAL_E_NONE 成功
- FSLRAL_E_XXX 错误

7.3 **int fsl_l2_addr_get**

描述

获取一条 mac 地址表的类型和索引。

语法

```
int fsl_l2_addr_get(int unit, fsl_mac_t mac, fsl_vlan_t vid,fsl_l2_addr_t *l2addr,fslral_mem_t *mem,uint32_t *index)
```

参数

- unit 设备号
- mac mac 地址
- vid vlan id
- l2addr l2 地址,暂未使用
- mem mac key 表类型
- index mac key 表索引

返回值

- FSLRAL_E_NONE 成功
- FSLRAL_E_XXX 错误

7.4 **int fsl_l2_addr_delete_by_port**

描述

基于 mac 地址行为表的端口来删除 mac 地址。

语法

```
int fsl_l2_addr_delete_by_port(int unit, fsl_module_t mod, fsl_port_t port, uint32_t flags)
```

参数

unit	设备号
mod	暂未使用
port	端口号
flags	暂未使用

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

7.5 int fsl_l2_addr_delete_by_vlan

描述

基于 vlan 来删除 mac 地址，删除指定 vlan 下的所有 mac 地址。

语法

int fsl_l2_addr_delete_by_vlan(int unit, fsl_vlan_t vid, uint32_t flags)

参数

unit	设备号
vid	vlan id
flags	暂未使用

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

7.6 int fsl_l2_age_timer_get

描述

获取 mac 地址普通老化时间。

语法

```
int fsl_l2_age_timer_get(int unit, int *age_seconds)
```

参数

unit 设备号

age_seconds 普通老化时间

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

7.7 int fsl_l2_age_timer_set

描述

设置 mac 地址普通老化时间。

语法

```
int fsl_l2_age_timer_set(int unit, int age_seconds)
```

参数

unit 设备号

age_seconds 普通老化时间

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

7.8 int fsl_l2_fast_age_enable_set

描述

mac 地址快速老化使能设置。

语法

int fsl_l2_fast_age_enable_set(int unit, int value)

参数

unit	设备号
value	快速老化使能开关值

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

8 Layer 2 Multicast Management

8.1 int fsl_mcast_create

描述

创建一条组播组 id。

语法

```
int fsl_mcast_create(int unit, int group_id, fsl_pbmp_t pbmp)
```

参数

unit	设备号
group_id	组播组 id
pbmp	组播组端口成员

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

8.2 int fsl_mcast_delete

描述

删除一条组播组 id。

语法

```
int fsl_mcast_delete(int unit, int group_id)
```

参数

unit	设备号
group_id	组播组 id

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_XXX 错误

8.3 **int fsl_mcast_bitmap_del**

描述

删除对应组播组 id 的端口成员。

语法

```
int fsl_mcast_bitmap_del(int unit, uint16_t group_id, fsl_pbmp_t pbmp)
```

参数

unit	设备号
group_id	组播组 id
pbmp	待删除组播组成员，高为有效指示

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

8.4 **int fsl_mcast_addr_add**

描述

添加一条组播 mac 地址。

语法

```
int fsl_mcast_addr_add(int unit, fsl_mcast_addr_t *mcaddr)
```

参数

unit	设备号
mcaddr	组播地址信息结构体

返回值

FSLRAL_E_NONE	成功
---------------	----

FSLRAL_E_XXX 错误

8.5 int fsl_mcast_addr_remove

描述

删除一条组播 mac 地址。

语法

int fsl_mcast_addr_remove(int unit, sal_mac_addr_t mac, fsl_vlan_t vid)

参数

unit	设备号
mac	mac 地址
vid	vlan id

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

9 Port Configuration

9.1 int fsl_port_control_set

描述

设置端口对应属性的值。

语法

```
int fsl_port_control_set(int unit, fsl_port_t port, fsl_port_control_t type, int value)
```

参数

unit	设备号
type	port 属性枚举类型
port	端口号
value	port 属性值

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

9.2 int fsl_port_control_get

描述

获取端口对应属性的值。

语法

```
int fsl_port_control_get(int unit, fsl_port_t port, fsl_port_control_t type, int *value)
```

参数

unit	设备号
type	port 属性枚举类型
port	端口号

value port 属性值

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

9.3 **int fsl_port_tpid_init**

描述

初始化获取各个端口 tpid 使用类型，初始化变量，在 tpid 处理时，最先被调用。

语法

int fsl_port_tpid_init(int unit)

参数

unit 设备号

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

9.4 **int fsl_port_tpid_add**

描述

入方向端口 tpid 添加。

语法

int fsl_port_tpid_add(int unit,fsl_port_t port,uint16_t tpid)

参数

unit 设备号

port 端口号

tpid tpid 类型值（默认为 0x88A8）

返回值

- FSLRAL_E_NONE 成功
- FSLRAL_E_XXX 错误

9.5 **int fsl_port_tpid_delete**

描述

入方向端口 tpid 删除。

语法

```
int fsl_port_tpid_delete(int unit,fsl_port_t port,uint16_t tpid)
```

参数

- unit 设备号
- port 端口号
- tpid Tpid 类型值

返回值

- FSLRAL_E_NONE 成功
- FSLRAL_E_XXX 错误

9.6 **int fsl_port_tpid_set**

描述

出方向端口 tpid 添加。

语法

```
int fsl_port_tpid_set(int unit,fsl_port_t port,uint16_t tpid)
```

参数

- unit 设备号
- port 端口号

tpid Tpid 类型值

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

9.7 **int fsl_ingress_port_stp_set**

描述

设置入方向端口 stp 状态。

语法

int fsl_ingress_port_stp_set(int unit, fsl_port_t gport, fsl_vlan_t vid, int stp_state)

参数

unit 设备号

gport 端口号(分为普通端口和 trunk 口)

stp_state 端口 stp 状态(共四种) FSL_STP_DISABLE = 0x0, FSL_STP_BLOCKING = 0x1, FSL_STP_LEARNING = 0x2, FSL_STP_FORWARDING = 0x3

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

9.8 **int fsl_egress_port_stp_set**

描述

设置出方向方向端口 stp 状态。

语法

int fsl_egress_port_stp_set(int unit, fsl_port_t gport, fsl_vlan_t vid, int stp_state)

参数

unit 设备号

gport 端口号(分为普通端口和 trunk 口)

stp_state 端口 stp 状态(共四种) FSL_STP_DISABLE = 0x0, FSL_STP_BLOCKING = 0x1, FSL_STP_LEARNING = 0x2, FSL_STP_FORWARDING = 0x3

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

9.9 int fsl_egress_port_stp_get

描述

获取出方向 stp 状态。

语法

int fsl_egress_port_stp_get(int unit, fsl_port_t gport, fsl_vlan_t vid, int *stp_state)

参数

unit 设备号

gport 端口号(分为普通端口和 trunk 口)

stp_state 端口 stp 状态(共四种) FSL_STP_DISABLE = 0x0, FSL_STP_BLOCKING = 0x1, FSL_STP_LEARNING = 0x2, FSL_STP_FORWARDING = 0x3

返回值

FSLRAL_E_NONE 成功

FSLRAL_E_XXX 错误

9.10 int fsl_ingress_stp_erps_enable_set

描述

设置 stpchck 和 erpschk 使能状态。

语法

int fsl_ingress_stp_erps_enable_set(int unit, fsl_port_t gport, fsl_port_control_t chk_type, int value)

参数

unit	设备号
gport	端口号(分为普通端口和 trunk 口)
chk_type	stp check erps check 类型（共四种） fslPortControlerpsLkpEn, fslPortControlStpChEn, fslPortControlEerpsLkEn, fslPortControlOutStpChkEn
value	使能状态值

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

9.11 int fsl_ingress_stp_erps_enable_get

描述

获取 stpchck 和 erpschk 使能状态。

语法

int fsl_ingress_stp_erps_enable_get(int unit ,fsl_port_t gport, fsl_port_control_t chk_type, int *value)

参数

unit	设备号
gport	端口号(分为普通端口和 trunk 口)
chk_type	stp check erps check 类型（共四种） fslPortControlerpsLkpEn, fslPortControlStpChEn, fslPortControlEerpsLkEn, fslPortControlOutStpChkEn
value	使能状态值

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

9.12 int fslsoc_stat_get

描述

获取端口不同类型的计数。

语法

```
int fslsoc_stat_get(int unit, fslsoc_port_t port, fslsoc_stat_val_t type, uint64_t *val)
```

参数

unit	设备号
port	端口号
type	不同计数类型
value	计数值

返回值

FSLRAL_E_NONE	成功
FSLRAL_E_XXX	错误

10 Set filter, send and receive packets

注：本章节仅适用 FSL91030M。

10.1 fsl_rx_filter_create

描述

创建过滤器

语法

```
int fsl_rx_filter_create(uint8_t pkt_data_offset,uint8_t pkt_data_size,uint8_t *data,uint8_t *mask,
uint8_t priority,char *desc,uint8_t desc_size, uint16_t desc_type)
```

参数

pkt_data_offset: 报文偏移。

pkt_data_size: 从报文偏移 **pkt_data_offset** 开始计算，需要过滤的报文字段长度。最长为 **KCOM_FILTER_BYTES_MAX**（256 字节）

data: 过滤器要过滤的报文字段，其长度等于 **pkt_data_size**。

mask: 用于要过滤的报文字段的掩码，其长度等于 **pkt_data_size**

priority: 优先级，数值越低，优先级越高。

desc: 过滤器名字。

desc_size: 过滤器名字的长度，最长为 **KCOM_FILTER_DESC_MAX**（32 字节）

desc_type: 过滤器行为，**KCOM_DEST_T_NULL**（不做处理），**KCOM_DEST_T_NETIF**（上传协议栈），**KCOM_DEST_T_API**（上传应用层）。

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.2 fsl_rx_filter_list

描述

查询过滤器的 id 和数量

语法

```
int fsl_rx_filter_list(uint8_t *filter_ids,uint32_t *filter_num)
```

参数

filter_ids: 查询到的过滤器 id 列表

filter_num: 查询到的过滤器的数量

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.3 fsl_rx_filter_get

描述

根据过滤器 id 获取过滤器信息

语法

```
int fsl_rx_filter_get(uint8_t id,kcom_filter_t *filter)
```

参数

id: 要查询的过滤器的 id

filter: 查询到的过滤器的信息

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.4 fsl_rx_filter_destroy

描述

根据过滤器 id 删除过滤器

语法


```
int fsl_rx_filter_destroy(uint8_t id)
```

参数

id: 要删除的过滤器 id

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.5 prepare_pkt

描述

设置包头待发送包的包头。

语法

```
int prepare_pkt(fsl_pkt_t *pkt)
```

参数

pkt 待发送的包

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.6 fsl_common_tx

描述

发包

语法

```
int fsl_common_tx(int unit, fsl_pkt_t *pkt, void *cookie)
```

参数

unit 设备号

pkt 待发送的包

cookie 预留，未使用

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.7 fsl_common_rx_register

描述

注册收包处理函数

语法

```
int fsl_common_rx_register(int unit, const char *name, fh_rx_cb_f callback, uint8_t priority, void *cookie, uint32_t flags)
```

参数

unit 设备号

name 收包处理函数的名称

callback 收包处理函数

priority 收包处理函数的优先级

cookie 预留，未使用

flags 预留，未使用

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.8 fsl_common_rx_unregister

描述

注销收包处理函数

语法

```
int fsl_common_rx_unregister(int unit, fh_rx_cb_f callback, uint8_t priority)
```

参数

callback 收包处理函数

priority 收包处理函数的优先级

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.9 fsl_common_rx_start

描述

开启收包。初始化，创建收包线程和包处理线程

语法

```
int fsl_common_rx_start(int unit)
```

参数

unit 设备号

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

10.10 fsl_common_rx_shutdown

描述

停止收包。注销收包线程和包处理线程。

语法

```
int fsl_common_rx_shutdown(int unit)
```

参数

unit 设备号

返回值

FSL_ERR_OK 成功

FSL_ERR_XXX 错误

11 修订信息

修订时间	版本	描述
2021.4.27	V1.0	初始版本。
2021.12.23	V1.4	内容优化。