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

FSL91030(M) chip

SDK command line configuration service example

Manual version: **V1.33**

Wuhan Feisiling Microelectronics Technology Co., Ltd.

April 2023

Table of contents

1 Port Numbering........5
2 Port Description........6
3 Equipment Operation Instructions........8
3.1 Start Command Line........ 8
3.2 Configuration Files........ 8
3.3 Access Interface Selection........9
3.4 Debug Print Switch........9
4 Register and Table Configuration Command Lines........ 10
4.1 Register Configuration Commands........10
4.1.1 reglist 10
4.1.2 getreg........10
4.1.3 setreg........10
4.1.4 modreg 10
4.2 Table Configuration Commands........11
4.2.1 memlist........11
4.2.2 dump........ 11
4.2.3 modify 11
5 Basic Configuration........13
5.1 Global Configuration........13
5.2 Port attribute configuration........13
5.3 Vlan attribute configuration........15
6 Vlan and Port Test........17
6.1 Adding and Deleting stpids 17
6.2 Creating a VLAN Domain........ 17
6.3 Port-based VLAN deletion........17
6.4 Port-based VLAN modification........18

6.5 VLAN deletion based on port + vlan..19

6.6 VLAN modification based on port + vlan..19

6.7 VLAN-based basic forwarding configuration..20

6.8 Status Setting, Query and Related Display.. 20

6.9 Port Isolation Service Test.. 23

6.10 Pdu Function Test.. 24

6.11 Packet Loss Statistics Function...25

7 Stg Function...28

7.1 Introduction to Stg Instruction.. 28

7.1.1 istag instruction description..28

7.1.2 estg Command Description...29

7.2 stg Configuration Example.. 29

7.2.1 istg Configuration Example..30

7.2.2 estg Configuration Example..33

8 Erps Function...37

8.1 Erps Instruction Introduction.. 37

8.1.1 ierps instruction description...37

8.1.2 eerps command description.. 38

8.2 erps configuration example...38

8.2.1 ierps Configuration Example..38

8.2.2 eerps Configuration Example.. 40

9 Lag Test.. 41

10 MAC Address Learning...43

11 MAC Address Learning Limitation... 44

12 MAC Address Aging..45

13 Acl, eAcl and vfp.. 47

13.1 Create related commands...48

13.2 Deletion related commands....................52

13.3 Display related commands....................52

13.4 Command Line Examples.................... 52

14 Mirroring.................... 57

14.1 Input/Output Ports, Input/Output Forwarding VLANID Mirroring.................... 57

14.1.1 Creating a Mirror Mode.................... 61

14.1.2 Deleting an Image.................... 57

14.1.3 Configuring the Input Mirror Port....................57

14.1.4 Configuring the Output Mirror Port....................57

14.1.5 Display Mirror Configuration.................... 61

14.2 MAC Address Mirroring....................58

14.2.1 Flow Mirroring.................... 61

15 Flow Control....................59

15.1 Command Line Introduction.................... 61

15.2 Configuration Example....................61

15.2.1 Example 1 59

15.2.2 Example 2 60

16 TM Test.................... 61

16.1 SP Mode (Priority Mode)....................61

16.2 WFQ Mode (Weight Allocation Mode).................... 62

17 Storm Suppression....................64

18 PKT DMA Function.................... 65

18.1 Introduction to functest pktdma command....................65

18.2 functest pktdma configuration example....................65

18.2.1 Application layer packet sending and receiving test configuration example.................... 65

18.2.2 Protocol stack packet sending and receiving test configuration example.................... 66

18.2.3 Application layer packet test configuration example....................67

19 Revision Information...69

1Port Number

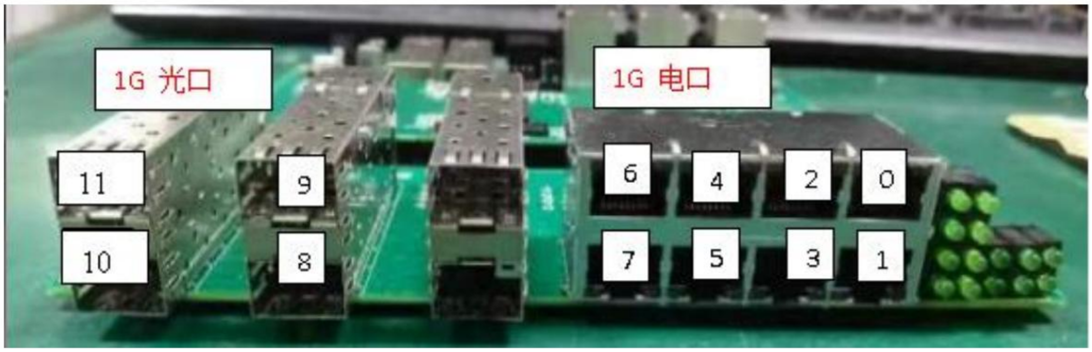
After the Demo board is powered on, log in through the serial port:

Login: root

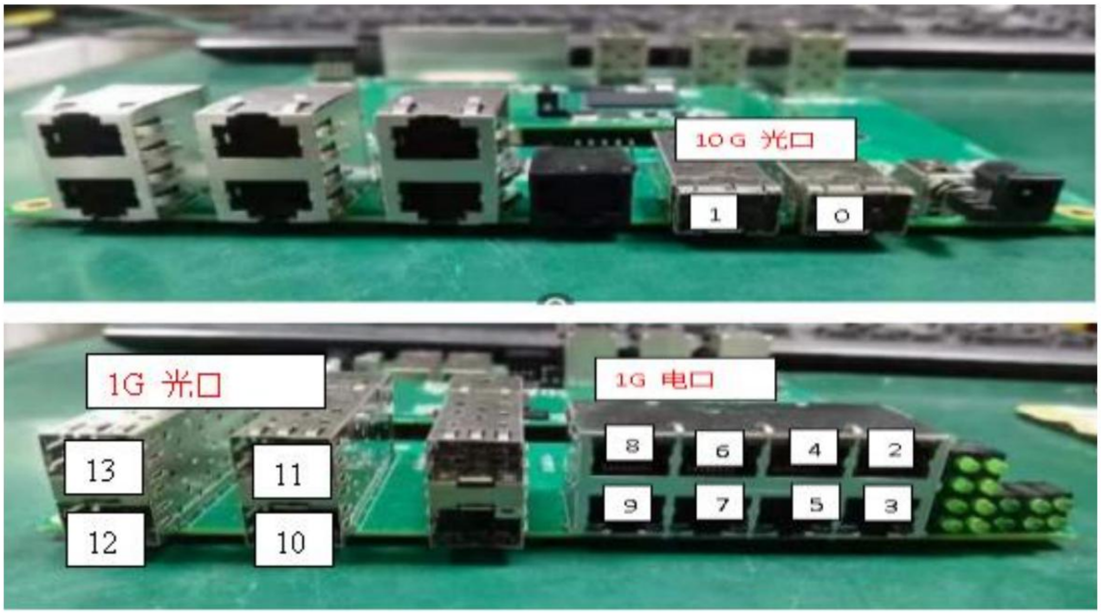
Password: fsl

Run the command fhcli: ./fhcli

Please refer to section 2.2 to switch the demo board port configuration to 8+4 mode: 8 electrical ports plus 4 1G optical ports (chip Initialization must be performed).



Please refer to section 2.2 to switch the demo board port configuration to 8+4+2 mode: 8 electrical ports plus 4 1G optical ports plus 2 10G optical ports. port mode (necessary for chip initialization).



2 Port Description

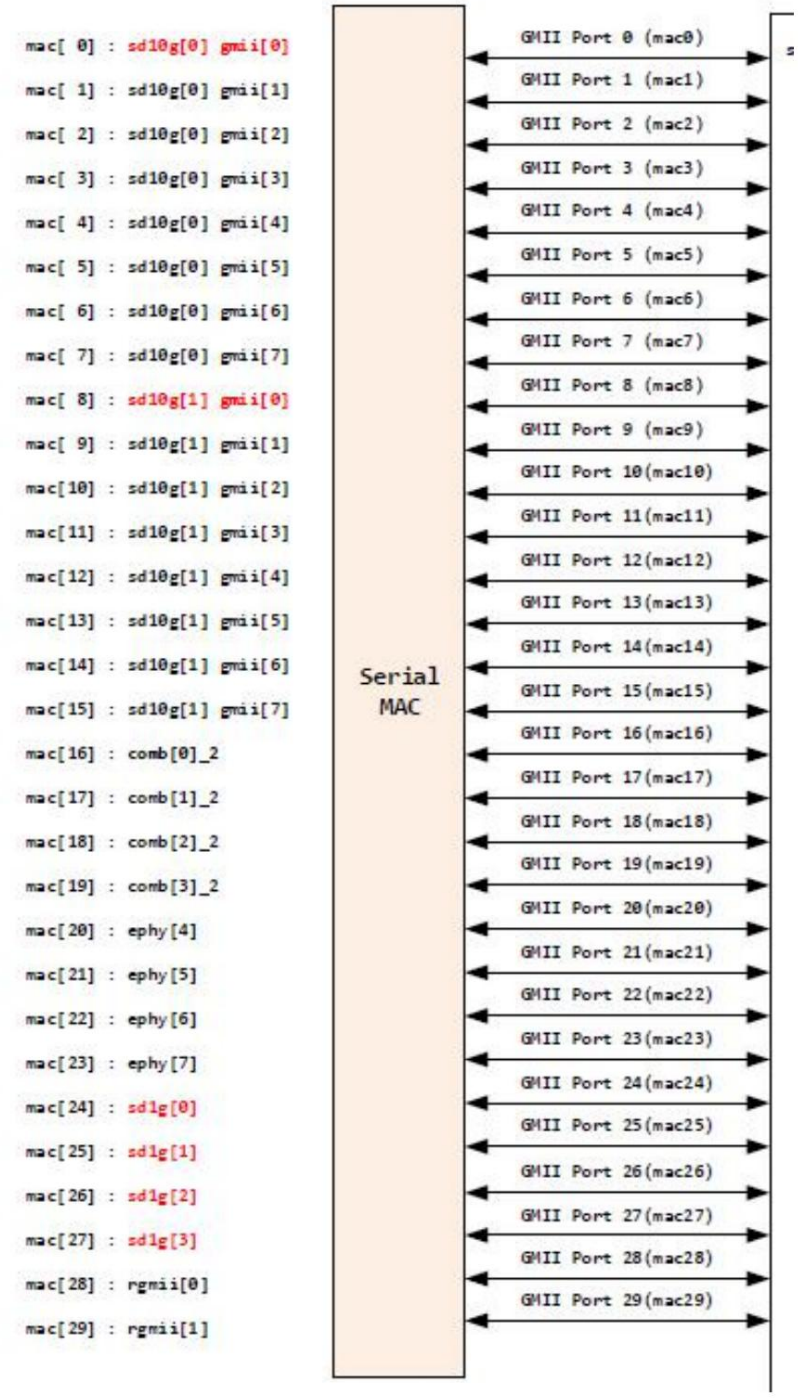
Port configuration is the first stage of the ingress pipeline, and its main purpose is to obtain the port configuration information related to the data packet entering the chip. The FSL91030M chip has different port concepts, namely physical ports and packet processing ports. In a broad sense, the physical port ID number corresponds to the MAC ID one by one, and the packet processing port corresponds to the logical port (and the panel port) one by one.

After powering on, FSL91030M first determines the chip's operating mode and port form based on the application scenario and configuration file information, and then determines The configuration information of a specific port.

The following table shows several common working modes and port configuration information of FSL91030M single chip:

Business Portal **Physical port/mac_id** Port form		**4** electrical ports		**8** electrical ports		**6*1G** optical ports		**2*10G** optical ports **+ 8** electrical ports		**4*1G** optical ports **+ 8** electrical ports		**2*10G** optical ports **+ 4** electrical ports **+** **4comb**	
0		GE0	16	GE0	16	GE0	0	XE0	0	GE0	16	XE0	0
1		GE1	17	GE1	17	GE1	8	XE1	8	GE1	17	XE1	8
2		GE2	18	GE2	18	GE2	24	GE0	16	GE2	18	Comb0	16
3		GE3	19	GE3	19	GE3	25	GE1	17	GE3	19	Comb1	17
4				GE4	20	GE4	26	GE2	18	GE4	20	Comb2	18
5				GE5	21	GE5	27	GE3	19	GE5	21	Comb3	19
6				GE5	22			GE4	20	GE6	22	GE0	20
7				GE7	23			GE5	21	GE7	23	GE1	21
8								GE6	22	GE8	24	GE2	22
9								GE7	23	GE9	25	GE3	23
10										GE10	26		
11										GE11	27		
31	CPU												
28	RGMII0												
29	RGMII1												

The FSL91030M has two 10G serdes, four 1G serdes and eight GEPHYs, physically connected to the mac topology as follows:


mac[0] : sd10g[0] gmii[0]
mac[1] : sd10g[0] gmii[1]
mac[2] : sd10g[0] gmii[2]
mac[3] : sd10g[0] gmii[3]
mac[4] : sd10g[0] gmii[4]
mac[5] : sd10g[0] gmii[5]
mac[6] : sd10g[0] gmii[6]
mac[7] : sd10g[0] gmii[7]
mac[8] : sd10g[1] gmii[0]
mac[9] : sd10g[1] gmii[1]
mac[10] : sd10g[1] gmii[2]
mac[11] : sd10g[1] gmii[3]
mac[12] : sd10g[1] gmii[4]
mac[13] : sd10g[1] gmii[5]
mac[14] : sd10g[1] gmii[6]
mac[15] : sd10g[1] gmii[7]
mac[16] : comb[0]_2
mac[17] : comb[1]_2
mac[18] : comb[2]_2
mac[19] : comb[3]_2
mac[20] : ephy[4]
mac[21] : ephy[5]
mac[22] : ephy[6]
mac[23] : ephy[7]
mac[24] : sd1g[0]
mac[25] : sd1g[1]
mac[26] : sd1g[2]
mac[27] : sd1g[3]
mac[28] : rgmii[0]
mac[29] : rgmii[1]
Serial MAC
GMII Port 0 (mac0)
GMII Port 1 (mac1)
GMII Port 2 (mac2)
GMII Port 3 (mac3)
GMII Port 4 (mac4)
GMII Port 5 (mac5)
GMII Port 6 (mac6)
GMII Port 7 (mac7)
GMII Port 8 (mac8)
GMII Port 9 (mac9)
GMII Port 10(mac10)
GMII Port 11(mac11)
GMII Port 12(mac12)
GMII Port 13(mac13)
GMII Port 14(mac14)
GMII Port 15(mac15)
GMII Port 16(mac16)
GMII Port 17(mac17)
GMII Port 18(mac18)
GMII Port 19(mac19)
GMII Port 20(mac20)
GMII Port 21(mac21)
GMII Port 22(mac22)
GMII Port 23(mac23)
GMII Port 24(mac24)
GMII Port 25(mac25)
GMII Port 26(mac26)
GMII Port 27(mac27)
GMII Port 28(mac28)
GMII Port 29(mac29)
s

3 Equipment operation instructions

3.1 Start the command line

SDK storage directory:

/var

document:

libsdk.so, fhcli, fh_pcie2pata.ko

After the device starts, enter the following command in the Linux interface:

cd /var

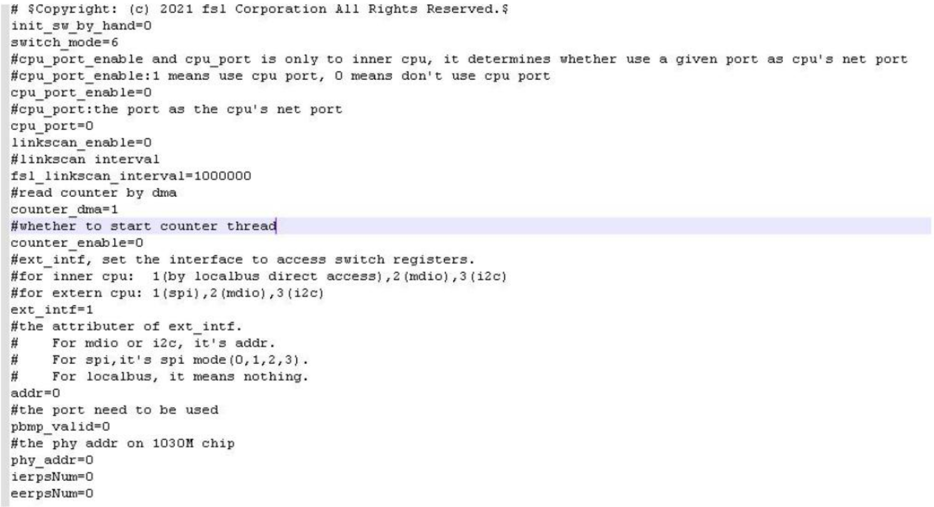
export LD_LIBRARY_PATH=/var

./fhcli

3.2 Configuration File

About FSL91030M chip demo board initialization:

Initialize by reading the configuration file config.fhme (configuration file config.fhme is in the /mnt/mtd path). Before starting fhcli, change init_sw_by_hand to 0, switch_mode=6 and run fhcli to initialize the chip to 8+4 mode (configuration file defaults to 0, configured to 8+4 mode). Change switch_mode to 7 to initialize it to 8+2 mode.



```
# $Copyright: (c) 2021 fsl Corporation All Rights Reserved.$
init_sw_by_hand=0
switch_mode=6
#cpu_port_enable and cpu_port is only to inner cpu, it determines whether use a given port as cpu's net port
#cpu_port_enable:1 means use cpu port, 0 means don't use cpu port
cpu_port_enable=0
#cpu_port:the port as the cpu's net port
cpu_port=0
linkscan_enable=0
#linkscan interval
fsl_linkscan_interval=1000000
#read counter by dma
counter_dma=1
#whether to start counter thread
counter_enable=0
#ext_intf, set the interface to access switch registers.
#for inner cpu:  1(by localbus direct access),2(mdio),3(i2c)
#for extern cpu: 1(spi),2(mdio),3(i2c)
ext_intf=1
#the attributer of ext_intf.
#    For mdio or i2c, it's addr.
#    For spi,it's spi mode(0,1,2,3).
#    For localbus, it means nothing.
addr=0
#the port need to be used
pbmp_valid=0
#the phy addr on 1030M chip
phy_addr=0
ierpsNum=0
eerpsNum=0
```

If you need to use the cpu port, you need to set cpu_port_enable=1 and specify a valid cpu port.

If you need to use the cpu port, you need to set cpu_port_enable=1 and specify a valid cpu port.

The corresponding relationship between several commonly used switch_mode values and port forms is shown in the following table:

switch_mode	6	7	13	15	17
Port form description	8+4 mode: 8 Electrical port plus 4 1G Optical port	8+2 mode: 8 Electrical port plus 2 10G mouth	24+4 mode: 24 Electrical ports plus 4 1 Optical port (for expansion phy scenario)	8+6 mode: 8 Electrical ports plus 6 1G Optical port	8+4+2 mode: 8 Electrical ports plus 4 1G optical port plus 2 10 Gigabit Port
Support RGMII	yes	yes	yes	yes	no

3.3 Access interface selection

1. Set up register access interface

The built-in CPU accesses the switch chip through the localbus bus to read and write directly. The default value can be used without setting. The external CPU accesses the switch chip through the localbus bus to read and write directly. The chip can be accessed via spi, mdio or i2c, and the access interface can be viewed or set via the ext_intf command.

2. View access interface information: ext_intf set

```
//Set to spi:

ext_intf set mode=1

//Set to mdio:

ext_intf set mode=2 addr=0x10

//Set to i2c:

ext_intf set mode=2 addr=0x54
```

3.4 Debug Print Switch

Set print information:

You can use the debug command to set whether to display debugging information.

Enable print information when reading and writing registers: debug + REG

Disable print information when reading and writing registers: debug - REG

4 Register and table configuration command line

4.1 Register Configuration Commands

4.1.1 reglist

/*List register names, "[]" parameter is optional. When no parameter is given, all registers will be listed; when a parameter is given, only registers containing the parameter string will be listed*/

```
reglist [keywords]
```

Example:

```
//Will list all register names containing "I_NET"

reglist I_NET
```

4.1.2 getreg

//Get the register field value, "{}" is a required parameter.

```
getreg {reg_name}
```

Example:

```
//Get the value of all fields of the iNetDefVlanCtl register

getreg I_NET_DEF_VLAN_CTL
```

4.1.3 setreg

//Set the register field value. The value of the unset field is 0.

```
setreg {reg_name} {field1=val1} [field2=val2] [... ]
```

Example:

```
/*Set the portBmp field value in the iNetDefVlanCtl register to 0xff*/

setreg I_NET_DEF_VLAN_CTL PORT_BMP=0xff
```

4.1.4 modreg

//Modify the register field value.

```
modreg {reg_name} {field1=val1} [field2=val2] [...]
```

Example:

//Set the portBmp field value of the iNetDefVlanCtl register to 0xff, and leave other unset field values unchanged

```
modreg I_NET_DEF_VLAN_CTL PORT_BMP=0xff
```

4.2 Table Configuration Commands

4.2.1 memlist

/*List table item names, "[]" parameter is optional. When no parameter is given, all table item names will be listed; when a parameter is given, only table item names containing the parameter string will be listed*/

memlist [keywords]

Example:

//Will list all table item names containing "I_NET"

```
memlist I_NET
```

4.2.2 dump

/*Get the domain value of the table entry. When the parameters index and cnt are not specified, all entries in the table entry will be listed (the number of entries listed depends on the depth of the table entry); when index is specified, the entry with index index will be listed; when index and cnt are specified, the cnt entries with base address index index will be listed*/

dump {mem_name} [index] [cnt]

Example:

//Get the value of the field of the 100th entry in the iNetVlanSrm table.

```
dump I_NET_VLAN_SRM 100
```

//Get the three entries of the iNetVlanSrm table starting from the 100th entry

```
dump I_NET_VLAN_SRM 100 3
```

4.2.3 modify

//Modify the table entry field value.

modify {mem_name} {index} {cnt} {field1=val1} [field2=val2] [...]

Example:

```
/*Modify the 100th entry in the iNetVlanSrm table. The destination portBmp field value is 0xff. Other unset field values remain unchanged*/

modify I_NET_VLAN_SRM 100 1 PORT_BMP=0xff

/*Modify the portBmp field value of the three entries starting from the 100th entry in the iNetVlanSrm table to 0xff, and the other unset field values are not set.
Change*/

modify I_NET_VLAN_SRM 100 3 PORT_BMP=0xff
```

5 Basic Configuration

5.1 Global Configuration

//type: port attribute type. See the following table for values and meanings

//val: port attribute value

Global set type={type} value={val}

Global get type={type}

Global setting **type**	**Type** Description
passLportLkp=0	enables the lag port function. Generally, the management chip enables the Lag function by default. If the configuration is wrong, The lag function can be forcibly enabled through this configuration; the unmanaged type does not support the lag function.
stpChkEn=1	Enable global stp check
stpDisableDropEn=2	Stp disable state discard enable
stpDisableChkEn=3	Stp disable status check enable
iFwdLrnLmtCtl=4	iFwd learning threshold enable

5.2 Port attribute configuration

//port_id: port number. Value range: 0~29

//type: port attribute type. See the following table for values and meanings

//val: port attribute value

Port attrset port={port_id} type={type} value={val}

Port type type	**Type** Description Value
fslPortControlXlateEn0 = 0,	Xlate0 Enable
fslPortControlXlateEn1,	Xlate1 Enable
fslPortControlFlowVlan0En,	VLAN Flow Enable
fslPortControlProtoVlanEn,	Protocol vlan enable
fslPortControlSvlanRangeEn,	Svlan Range Enable
fslPortControlCvlanRangeEn,	Cvlan Range Enable
fslPortControlScosMapEn,	Svlan cos mapping enable
fslPortControlCcosMapEn,	Cvlan cos mapping enable
fslPortControliVtEditEn,	Inbound vlan edit enable
fslPortControlLrnDisable,	Not learning instructions
fslPortControlPriVld,	Port priority valid flag

fslPortControlLeftAlgTp0,	The left hash table algorithm is specifically coded as follows: 0x0: Use the low bit of crc32 operation result 0x1: Use crc16-BISYNC operation result low bit 0x2: Use crc16-CCITT operation result low bit 0x3: Use the low bit of the key value
fslPortControlRightAlgTp0,	Right hash table algorithm, the specific encoding is as follows: 0x0: Use the low bit of crc32 operation result 0x1: Use crc16-BISYNC operation result low bit 0x2: Use crc16-CCITT operation result low bit 0x3: Use the low bit of the key value
fslPortControlLeftAlgTp1,	The left hash table algorithm is specifically coded as follows: 0x0: Use the low bit of crc32 operation result 0x1: Use the high bit of crc32 operation result 0x2: Use crc16 operation result 0x3: Use key value
fslPortControlRightAlgTp1,	Right hash table algorithm, the specific encoding is as follows: 0x0: Use the low bit of crc32 operation result 0x1: Use the high bit of crc32 operation result 0x2: Use crc16 operation result 0x3: Use key value
/*inet*/	
fslPortControlerpsLkpEn = 20,	Ring network protection check enable
fslPortControlinPortMirEn,	Input port mirroring enable
fslPortControlLrnUpdDisable, fslPortControlSmvFlag,	Learning update disable
	Site shift control is enabled. The meaning of each bit is described as follows: [0]:Discard Enable [1]: trap to CPU enable [2]: Site shift enable
fslPortControlHmClass,	Port priority in case of site relocation
fslPortControlAlwSameClsSmv,	Allows ports with the same priority to perform site shift operations
fslPortControlBrgEn,	Layer 2 bridge search enable, high effective
fslPortControlVlanFilterEn,	Vlan filtering enable
fslPortControlStpChEn,	Spanning tree check enable
fslPortControlUsePktSvid,	Use the SVID of the original message as the internal VLANId
fslPortControlUseUpdSvid,	Use the modified svid as the internal vlanId
/*ifwd*/	
fslPortControlDropUkwUc = 40, fslPortControlProtId,	Unknown unicast discard enable
fslPortControlProtInd,	Protection group id, you can create 4 protection groups
	Protection port indication, 2'b00, non-protection group, 2'b10: 1:1 protection group; 2'b11: 1+1 protection group
fslPortControlLrnMissToCpu,	Mac address original address lookup failure discard enable
fslPortControlLrnMissToDrop,	Mac address original address lookup failure trap to cpu enable

fslPortControlProGroup, protection port indication, the inbound port is a protection port, and the message is a non-OAM type message, or	Enable discard OAM indication, and the message is discarded.
fslPortControlAllowPortToSrc,	Loopback enable, allowing loopback on the same port
fslPortControlPortIsotEnBmp,	Port isolation is enabled. The specific description of each bit is as follows: [0]: known unicast [1]:Unknown unicast [2]: Known multicast [3]: Unknown multicast [4]: Broadcast
/*iacl*/	
fslPortControliAcl0LkpVld = 60,	iAcl0 query switch
fslPortControliAcl1LkpVld, /*eee*/	iAcl1 query switch
fslPortControlEvlanFilterEn = 70,	Egress VLAN filtering
fslPortControlEerpsLkEn,	Export ring network protection query switch
fslPortControlEoutPortMirEn,	Export mirror
fslPortControlEvtEditEn, /*epf*/	Egress VLAN Edit
fslPortControlUpdEn = 80,	Trunk port failure sharing enabled
fslPortControlOutStpChkEn,	Export stp check enable
fslPortControlOutPortIsUnStag,	Export stripping outer Stag
fslPortControlOutPortIsUnCtag,	Export stripping inner Ctag
fslPortControlPortIsotEn,	Port isolation enable
fslPortControlNetworkPort,	Split horizon enable
/*eacl*/	
fslPortControleAcl0LkpVld =90,	eAcl0 query switch
fslPortControleAcl1LkpVld,	eAcl1 query switch

5.3 **Vlan** attribute configuration

//vlan_id: vlan ID. Value range: 0~4091

//type: vlan attribute type. See the following table for values and meanings.

//val: vlan attribute value

vlan vlanset vlanid={vlan_id} **type={type} value={val}**

Vlan Type type	**Type** Description Value
fslVlanDropUnknown = 0, fslVlanMacList,	Unknown unicast packet discard enable
	Mac address blacklist and whitelist mode 1: Blacklist 2: Whitelist

fslVlanFidStmCtlVld,	FID-based storm control enablement
fslVlanMacLearnDisable,	Disable mac address learning
fslVlanMacLearnUp, fslVlanIsotIdx	Disable mac address learning update enable
= 5, fslVlanIsotEn,	VLAN-based port isolation pointer
fslVlanPduBypassStp,	Enable VLAN-based port isolation
fslVlanErpsId, fslVlanStpId = 20,	Identify PDU bypass spanning tree check enable
fslVlanPriVld,	ERPS ring network protection table ID
fslVlanPriority, fslVlanColor,	STP Spanning Tree ID
fslVlanQosProfileVld,	Internal priority and color indication
fslVlanQosProfileIdx,	Internal Priority
fslVlanInFpolVld,	Color: 0x00: RED 0x01: YELLOW 0x02: GREEN
fslVlanTrapEn, fslVlanDropEn,	Qos template valid indication
fslVlanBypassEn, fslVlanDot1xEn,	Qos Template Index
fslVlanInMirEn,	Inlet hierarchical metering, effective indication of small and medium pipelines
fslVlanOutErpsId = 40,	Trap to CPU enable
fslVlanOutStpId,	Discard Enable
fslVlanOutFpolVld,	Bypass Enable
	1x authentication enabled, high effective
	Ingress mirroring enabled
	Egress port ERPS ring network protection table ID
	Outbound port STP spanning tree ID
	Export hierarchical metering, effective indication of small and medium pipelines

6 **Vlan** and **Port** Test

6.1 Adding and removing **stpid**

1. Inbound direction:

```
port addtpid Port=14 Tpid=0x8100
```

```
port deltpid Port=20 Tpid=0x8100
```

2. Outbound direction:

```
port settpid Port=14 Tpid=0x9100
```

6.2 Create **a VLAN** domain

1. Port 5 uses the svid of the original message as the internal vlanId

```
port attrset port=5 type=29 value=1
```

2. Logical port members

```
//100 represents VLAN ID, 0xfff represents the added logical port members 0~11
```

```
vlan create 100 pbmp=0xfff
```

3. Strip the VLAN from the outbound ports in the VLAN domain

```
/*100 indicates VLAN ID, 0x1 indicates untag the outbound port 0, please note that when configuring ubmp, pbmp
Members must also be brought along, because the VLAN principle stipulates that ubmp must be a subset of pbmp*/
```

```
vlan create 100 pbmp=0xfff ubmp=0x1
```

4. Delete port members in VLAN domain

```
//100 represents VLAN ID, 0x2 represents deleting port member 1 of vlan 100
```

```
vlan remove 100 pbmp=0x2
```

6.3 Port - based **VLAN** deletion

1. Business configuration

Edit 8 unicast Ethernet messages on port 6 (electrical port) in the format of MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

2. Configuration commands:

```
//pbmp: one port occupies 1 bit, 1111110

vlan create 0x100 pbmp=0x7E

//type=8 is for vlan edit enable

 port attrset port=6 type=8 value=1

//Port 6 port default VLAN editing operation is to delete the outer VLAN

vlan action port default add port=6 sotovid=delete
```

3. Expected results:

The message is received at ports 1~5, and the message VLAN is stripped.

6.4 **Port** - based **VLAN** modification

1. Business configuration:

Edit 8 unicast Ethernet messages on port 6 (electrical port) in the format of MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

2. Configuration commands:

```
vlan create 0x100 pbmp=0x7E

//type=8 is for vlan edit enable

 port attrset port=6 type=8 value=1

//Port 6 port default vlan editing operation is to modify the outer Vlan id to 0x64

vlan action port default add Port=6 newsvid=0x64 sotovid=replace
```

3. Expected results:

The message is received at ports 1~5, and the message VLANID is modified to 0x64.

6.5 **VLAN** deletion based on port **+ vlan**

1. Business configuration

Edit 8 unicast Ethernet messages on port 6 (electrical port) in the format of MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

2. Configuration commands

```
vlan create 0x100 pbmp=0x7E

//type=8 is for vlan edit enable

port attrset port=6 type=8 value=1

//type=0 enables vlanxlate0

port attrset port=6 type=0 value=1

//Xlate0 uses the port+vlanid key to delete the matching outer VLAN

vlan action translate add port=6 keytype=PortOuter oldoutervLan=0x100                sotovid=delete
```

3. Expected Results

The message is received at ports 1~5, and the message VLAN is stripped.

6.6 **VLAN** modification based on **port +** vlan

1. Business configuration

Edit 8 unicast Ethernet messages on port 6 (electrical port) in the format of MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

2. Configuration commands

```
vlan create 0x100 pbmp=0x7E

//type=8 is for vlan edit enable

port attrset port=6 type=8 value=1

//type=0 enables vlanxlate0

port attrset port=6 type=0 value=1
```

```
//Xlate0 uses the port+vlanid key to change the matching outer VLAN ID to 0x64

vlan action translate add port=6 keytype=PortOuter oldoutervLan=0x100 newsvid=0x64 sotovid=replace
```

3. Expected Results

The message is received at ports 1~5, and the message VLANID is modified to 0x64.

6.7 VLAN- based basic forwarding configuration

1. Create a VLAN domain

```
//Port 5 uses the original message's SVID as the internal VLANId

port attrset port=5 type=29 value=1

//100 represents VLANID, 0xfff represents the added logical port members 0~11

vlan create 100 pbmp=0xfff
```

2. Strip the VLAN function from the outbound ports in the VLAN domain

```
//100 indicates VLAN ID, 0x1 indicates untag the outbound port 0.

vlan create 100 ubmp=0x1
```

3. Delete port members in VLAN domain

```
//100 represents VLAN ID, 0x2 represents deleting port member 1 of vlan 100

vlan remove 100 pbmp=0x2
```

6.8 Status setting, query and related display

1. Port status setting

```
//Configure port 1 to enable, 1 to enable, 0 to disable

PORT ctlset port=1 enable=0/1

//Configure the port auto-negotiation status 1 to enable, 0 to disable

PORT ctlset port=1 an=0/1

//Configure the port rate (currently supports electrical ports), unit: M
```

PORT ctlset port=1 speed=10/100/1000

//Configure port duplex status (currently supports electrical ports), 0 for half-duplex, 1 for full-duplex

PORT ctlset port=1 duplex=0/1

//Configure the negotiation capability of the port when auto-negotiation is enabled.

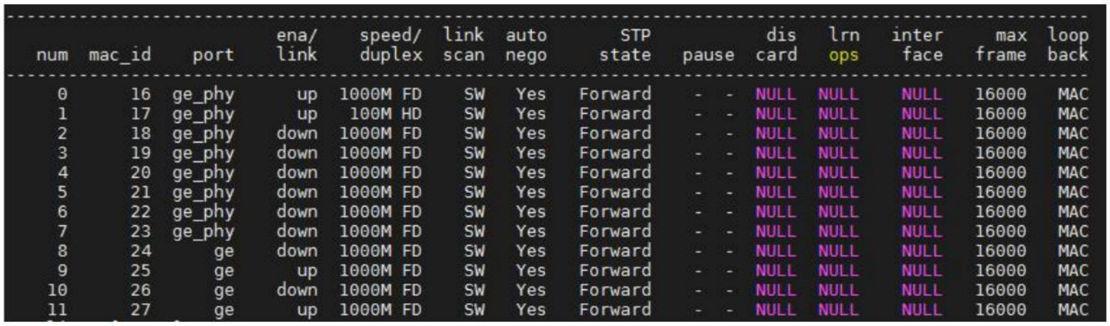
PORT ctlset port=<> ability=<0x7e0>(bit[5]:10M H bit[6]:10M F bit[7]:100M H bit[8]:100M F bit[9]:1000M H bit[10]:1000M F)

//Reset phy

PORT ctlset port=1 resetphy=<0|1>

2. Port status display command

Portsta



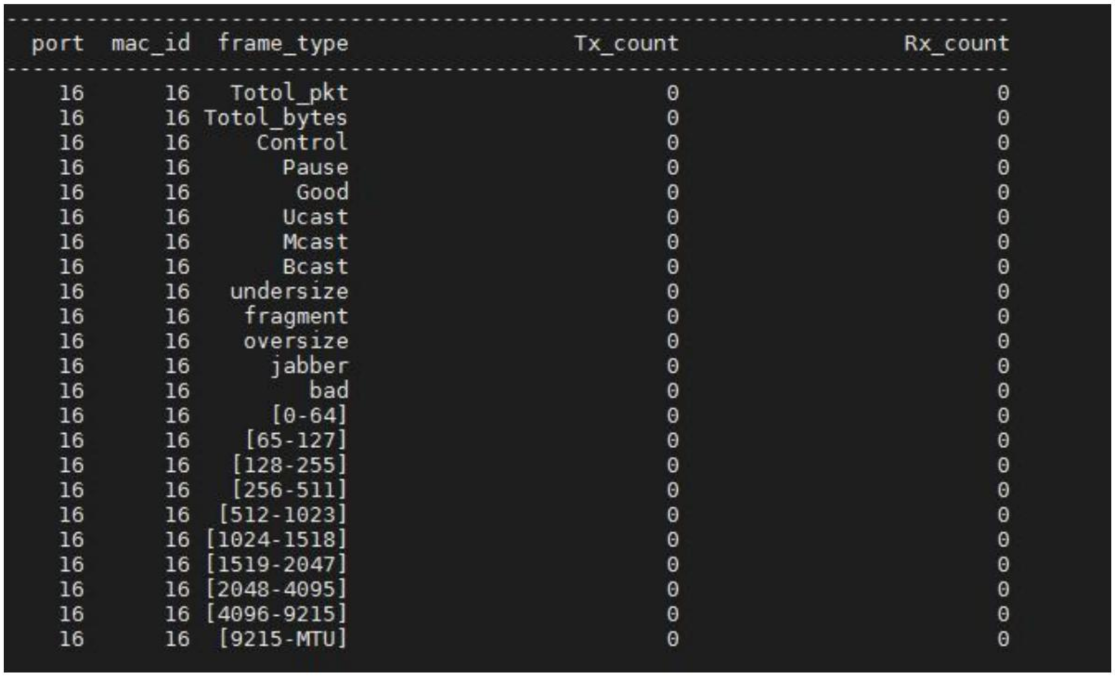
```
--------------------------------------------------------------------------------------------------------------------
                          ena/    speed/  link  auto        STP               dis   lrn    inter     max  loop
  num  mac_id     port    link    duplex  scan  nego      state    pause     card   ops     face   frame  back
--------------------------------------------------------------------------------------------------------------------
    0      16  ge_phy       up  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    1      17  ge_phy       up   100M HD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    2      18  ge_phy     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    3      19  ge_phy     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    4      20  ge_phy     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    5      21  ge_phy     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    6      22  ge_phy     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    7      23  ge_phy     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    8      24      ge     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
    9      25      ge       up  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
   10      26      ge     down  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
   11      27      ge       up  1000M FD    SW   Yes    Forward    -   -    NULL  NULL     NULL   16000   MAC
```

3. Port packet count display command

1) Read the register directly to display the count of a port:

//The port number is the mac_id in the figure

port showcounter port=16



```
------------------------------------------------------------------------------
  port  mac_id  frame_type                 Tx_count                 Rx_count
------------------------------------------------------------------------------
    16      16   Totol_pkt                        0                        0
    16      16 Totol_bytes                        0                        0
    16      16     Control                        0                        0
    16      16       Pause                        0                        0
    16      16        Good                        0                        0
    16      16       Ucast                        0                        0
    16      16       Mcast                        0                        0
    16      16       Bcast                        0                        0
    16      16   undersize                        0                        0
    16      16    fragment                        0                        0
    16      16    oversize                        0                        0
    16      16      jabber                        0                        0
    16      16         bad                        0                        0
    16      16      [0-64]                        0                        0
    16      16    [65-127]                        0                        0
    16      16   [128-255]                        0                        0
    16      16   [256-511]                        0                        0
    16      16  [512-1023]                        0                        0
    16      16 [1024-1518]                        0                        0
    16      16 [1519-2047]                        0                        0
    16      16 [2048-4095]                        0                        0
    16      16 [4096-9215]                        0                        0
    16      16  [9215-MTU]                        0                        0
```

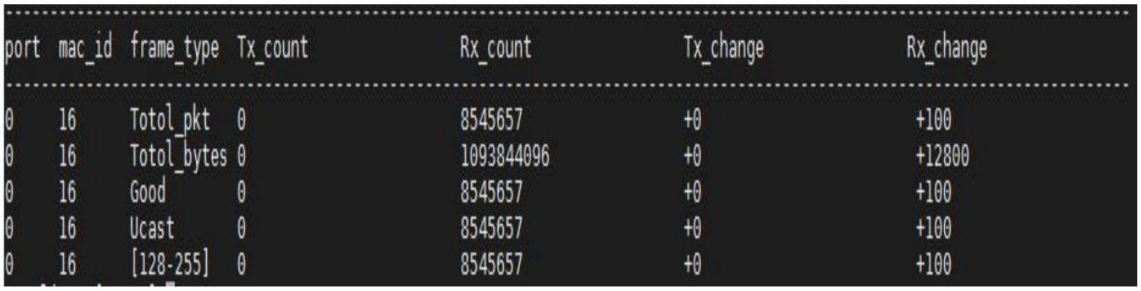
2) Read the count via DMA

Before starting fhcli, you need to set counter_dma=1 in the configuration file config.fhme

Order:

//Port 0 shows the change of counts twice

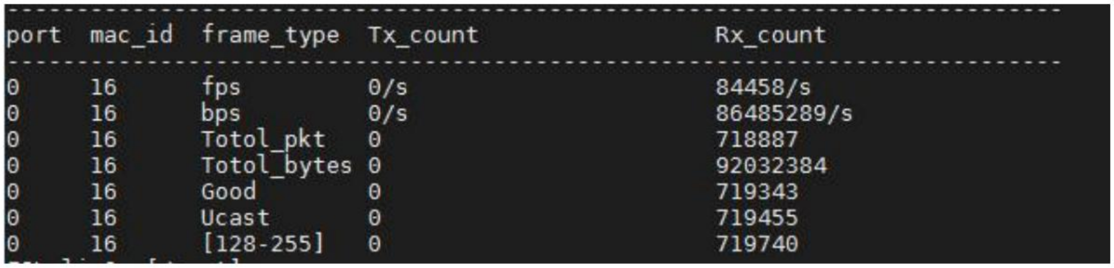
Show cc pbmp (display all ports when the command does not include **pbmp)**



```
----------------------------------------------------------------------------------------------
port  mac_id  frame_type  Tx_count          Rx_count          Tx_change          Rx_change
----------------------------------------------------------------------------------------------
0     16      Totol_pkt   0                 8545657           +0                 +100
0     16      Totol_bytes 0                 1093844096        +0                 +12800
0     16      Good        0                 8545657           +0                 +100
0     16      Ucast       0                 8545657           +0                 +100
0     16      [128-255]   0                 8545657           +0                 +100
```

//Show the count in port 0 is not zero

Show c nz pbmp (display all ports when the command does not include **pbmp)**

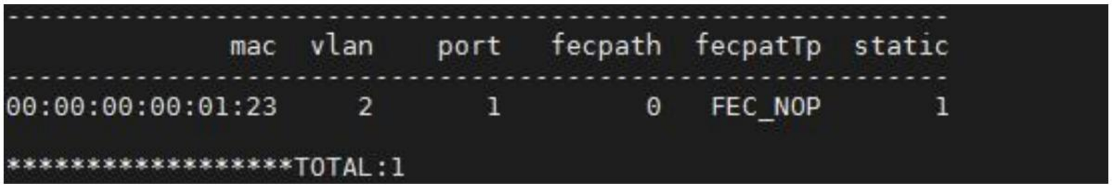


```
-----------------------------------------------------------------------------
port  mac_id  frame_type  Tx_count                  Rx_count
-----------------------------------------------------------------------------
0     16      fps         0/s                       84458/s
0     16      bps         0/s                       86485289/s
0     16      Totol_pkt   0                         718887
0     16      Totol_bytes 0                         92032384
0     16      Good        0                         719343
0     16      Ucast       0                         719455
0     16      [128-255]   0                         719740
```

4. Mac address display

Order:

l2 show



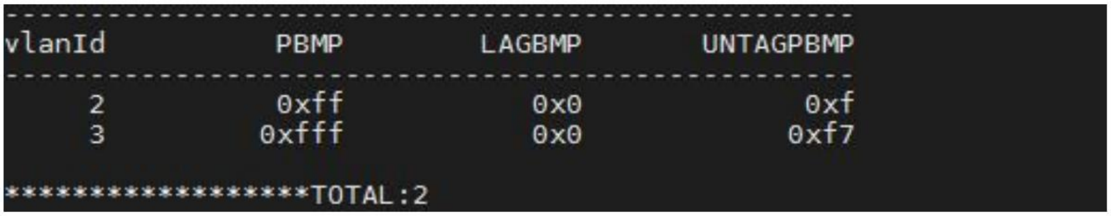
```
------------------------------------------------------------
              mac  vlan    port   fecpath  fecpatTp  static
------------------------------------------------------------
00:00:00:00:01:23     2       1         0   FEC_NOP       1

******************TOTAL:1
```

5. Vlan display

Order:

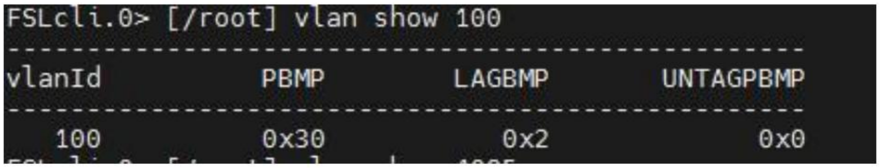
vlan show



```
-----------------------------------------------------
vlanId          PBMP        LAGBMP       UNTAGPBMP
-----------------------------------------------------
     2          0xff           0x0             0xf
     3         0xfff           0x0            0xf7

******************TOTAL:2
```

vlan show <vid>



```
FSLcli.0> [/root] vlan show 100
-----------------------------------------------------
vlanId          PBMP        LAGBMP       UNTAGPBMP
-----------------------------------------------------
   100          0x30           0x2             0x0
```

6.9 Port Isolation Service Test

The port isolation service mainly isolates the inbound port from certain ports;

Configuration command line:

Port portisolate inport=4 pbmp=0xf enable=1

Port 4 receives incoming messages, but ports 0, 1, 2, and 3 cannot receive messages.

//inport is the input port

//pbmp is the isolated port bitmap

//enable is port isolation enable (0 is off, 1 is on)

6.10 **Pdu** Function Test

Pdu supports 32 entries, matched in sequence, and the supported matching domains and corresponding masks are as follows:

Mask 0: no match, 1: match.

Match domain command line field smac		The	Command Line Fields
(global, 32 entries shared) MACaddress		mask is configured in the specific entry, the	mask_smac
dmac	dmac	mask is 0/1 16-47bit The mask is 0/1; 0-15bit is bit-by-bit mask, 0xFFFF, so dmac The mask setting range is 0-0x1FFFF	mask_dmac
ethType	ethType	0/1	mask_ethType
stag.isValid	stagVld	0/1	mask_stagVld
svid	svid	0/1	mask_svid
ctag.isValid	ctagVld	0/1	mask_ctagVld
cvid	cvid	0/1	mask_cvid
l2Tp	l2Tp	0/1	mask_l2Tp

Among the 32 pdu entries, the 0th entry is the default match for the message with dmac 0x0180C2000000 and dmac mask 0x1FFFE. The first one is the default match for the message with dmac 0x0180C2000000 and dmac mask 0x1FFF0, and the other entries are empty configurations.

1. Create vlan=100 forwarding domain

vlan create 100 pbmp=0xfff

2. Set the pdu global source mac.

vlan action pdu smac set MACaddress=0x2345678

3. Set the matching smac of pdu entry 5 to 0x2345678 and dmac to 0xaabbcc and send the message to the CPU

vlan action pdu config add index=5 dmac=0xaabbcc mask_dmac=0x1FFFF mask_smac=1

4. Set the PDU action of item 5, 0: do not process; 1: discard; 2: trap; 3: discard + trap;

vlan action pdu option set index=5 option=2

5. Set the trap port to the CPU port

```
dp trap set port=31
```

6.11 Packet loss statistics function

1. Packet loss statistics initialization (preset when the command line is started, no need for user call)

```
show drop init
```

2. Check the inbound packet loss of port 3

```
show drop count port=3 direc=0
```

3. Check outgoing packet loss on port 5

```
show drop count port=5 direc=1
```

4. Clear packet loss statistics for all ports

```
show drop clear
```

Summary of packet loss reasons:

0	NOP	--
1	IPR0ERR	Packet parsing error
2	MCSMACERR	Source MAC is the multicast MAC address
3	PORTPDUIDFY	PDU Matching Packet Loss
4	PORTDOT1XPROTO	Port-based 802.1x authentication packet loss
5	PORTDOT1X	Port-based 802.1x authentication fails
6	IAFT	AFT filtering of tag and utag packets causes packet loss
7	IXLATE	Inbound VLAN translation matching action packet loss
8	MACIPBIND	MAC, IP binding miss packet loss
9	IVLAN	Inbound vlan domain action packet loss
10	ISTPDISABLE	The inbound stp status is disable
11	VLANDOT1XPROTO	VLAN-based 802.1x authentication packet loss
12	VLANDOT1X	VLAN-based 802.1x authentication fails
13	IVLANFILTER	Inbound VLAN filtering
14	VLANPDUIDFY	PDU Matching Packet Loss
15	ISTPCHECK	The inbound stp status is non-forwarding
16	IPOPTION	IP packet port behavior with extension header packet loss
17	IPHDRERR	IP header error
18	V4HDRCHECKFAIL	IPv4 message ttl is equal to 0 or 1 or source ip is equal to destination ip
19	V6HDRCHECKFAIL	IPv6 message ttl is equal to 0 or 1 or source ip is equal to destination ip
20	L4HDRERR	--
21	TCPDOSCHECKFAIL	TCP or UDP DOS attack packets are dropped

twenty two	ICMPDOSCHECKFAIL	ICMP DOS attack message hits and is discarded
twenty three	RXPROT	OAM packets are discarded during ingress port protection
twenty four	DMAC	The destination mac address hit action is trap
25	DMACFILTER	The destination mac address hit action is drop
26	CCPORTNOMATCH	SCC/DCC source path check port fails when forwarding destination mac is SCC, DCC Type when hit discard
27	CCNOMATCH	When forwarding, if the destination mac is SCC or DCC type, it is discarded if it does not match.
28	PFMFILTER	Multicast pfm filtering packet loss
29	UCLOOPAVOIDPORT	The destination port of a unicast message is equal to the source port
30	UCLOOPAVOIDPORTCC	SCC, DCC forwarding type egress port equals source port
31	STORMCTL	Storm Suppression Packet Loss
32	SMAC	Source MAC hits trap during learning
33	LRNSMVCHKFAIL	Failed to shift site during learning
34	LRNBKTOVFW	Learn MAC address table overflow
35	SMACFILTER	Source MAC hits are discarded during learning
36	SMACMISSDROP	Source MAC address lookup failed and discarded
37	SMACBLACKLIST	Blacklist hit discard
38	SMACWHITELIST	Whitelist not hit discard
39	LRNSMV	Site shift trap
40	LRNSMVCLS	Site shift discard
41	LRNMACNUMLMT	MAC address learning entries exceed the threshold
42	IACL0LKP	Inbound acl0 matching action packet drop
43	IACL1LKP	Inbound acl1 matching action packet loss
44	IPOLDROP	Inbound Policing rate limit based on packet color and packet loss
45	FLDNOVLDBMP	There is no valid port in PortBitmap
46	EVLAN	Outgoing VLAN domain action packet loss
47	EXLATE	VLAN translation matching action packet loss occurs
48	EXLATEMISS	VLAN translation mismatch occurs and packet loss occurs
49	LAGNOMEMBER	The aggregation group has no port members.
50	OUTPORT	Egress Action Packet Loss
51	REMOTELOOPBACK	--
52	HORIZONSPLTPORT	Split Horizon
53	ICMPREDIR	--
54	MCLOOPAVOIDPORT	The destination port of the multicast message is equal to the source port
55	MCLOOPAVOIDPORTCC	Outbound SCC, DCC Forwarding Type Outbound port equals source port
56	ESTPDISABLE	The outbound stp status is disable
57	ESTPCHECK	The outbound stp status is non-forwarding
58	EVLANFILTER	Outgoing VLAN filtering
59	PORTISOLATE	Port Isolation
60	EACL0LKP	Outgoing acl0 matching action packet loss
61	EACL1LKP	Outgoing acl1 matching action packet loss
62	EPOLDROP	Outbound Policing rate limit based on packet color and packet loss
63	SAMEMAC	The source MAC of the message is equal to the destination MAC

7 **Stg** Function

According to the MSTP protocol, multiple different VLANs are combined to form an STP group (i.e., STG). VLANs in the same STG group share the STP status of the port. When implementing the STG function, the I_NET_VLAN_SRM table entry field STP_ID is understood as the STG ID, and the I_NET_STP_SRM table is searched through the ID to obtain the port status of the STG. Therefore, the STG ID must be less than the table entry depth of the I_NET_STP_SRM.

Stp is divided into istp and estp. To simplify the use, stg is divided into istg and estg. The specific instructions are as follows. “[ ]” indicates optional, and “|” indicates multiple selection 1.

7.1 Introduction to Stg Instruction

7.1.1 istag instruction description

/*istg is initialized. After initialization, the default stgid=0 will be created, including vlan1-vlan4095, and the stp status of all ports and lag ports of stg0 is diable (it will take effect only after the port stp_check is enabled, and it is in forward state before enabling). */

istg init

//istg function is turned off

istg deinit

// Enable stpcheck for the corresponding port

istg enable pbmp=<pbmp>

// Enable stpcheck of the corresponding port

istg disable pbmp=<pbmp>

//Create STGs, you can create multiple stgs

istg create [<id>] [...]

//Cancel STGs configuration

istg destroy <id> [...]

//View the configured STG(s), display the vlans contained in the stg and the corresponding stp status

istg show [<id>]

//Add VLAN(s) to a STG

istg add <id> <vlan_id> [...]

//Remove VLAN(s) from a STG

```
istg remove <id> <vlan_id> [...]
```

//Display all stg port status

```
istg stp
```

//Display the status of a stg port

```
istg stp <id>
```

//Set the stp status of a STG

```
istg stp <id> <pbmp=xx>[<lbmp=xx>] <state=disable|block|learn|forward>
```

7.1.2 estg command description

The estg instruction is similar to istg and will not be described in detail here.

```
estg init

estg deinit

estg enable pbmp=<pbmp>

estg disable pbmp=<pbmp>

estg create [<id>][...]

estg destroy <id>[...]

estg show [<id>]

estg add <id> <vlan_id> [...]

estg remove <id> <vlan_id> [...]

estg stp

estg stp <id>

estg stp <id> <pbmp=xx>[<lbmp=xx>] <state=disable|block|learn|forward>
```

7.2 stg configuration example

7.2.1 **istg** configuration example

Create 4 stgs, stg10 to stg13, and configure lag0 to include port2 and port3.

stg10 contains vlan 100-104, and configures port0 and lag0 to disable state

stg11 contains vlan 105-109, and configures port0 and lag0 to block state

stg12 contains vlan 110-114, and configures port0 and lag0 to learn state

stg13 contains vlan 115-119, and configures port0 and lag0 to forward state

1. Lag configuration

```
trunk init
```

//Create a lag group (lagID is 0), port members are 2 and 3, hashkey is DstIp and SrcMac.

```
trunk add Id=0 Rtag=0x2001 Pbmp=0xC
```

/* Enable **the lag** port function. Generally, the management chip enables **the Lag** function by default. If the configuration is wrong, you can force **the lag** to be enabled through this configuration. Function; unmanaged models do not support **lag** function*/

```
global set type=0 value=0
```

2. VLAN configuration

//Create VLAN domain 100-119, including port0-1 and lag0

```
vlan create 100 pbmp=0x3 lbmp=0x1
vlan create 101 pbmp=0x3 lbmp=0x1
vlan create 102 pbmp=0x3 lbmp=0x1
vlan create 103 pbmp=0x3 lbmp=0x1
vlan create 104 pbmp=0x3 lbmp=0x1
vlan create 105 pbmp=0x3 lbmp=0x1
vlan create 106 pbmp=0x3 lbmp=0x1
vlan create 107 pbmp=0x3 lbmp=0x1
vlan create 108 pbmp=0x3 lbmp=0x1
vlan create 109 pbmp=0x3 lbmp=0x1
vlan create 110 pbmp=0x3 lbmp=0x1

vlan create 111 pbmp=0x3 lbmp=0x1

vlan create 112 pbmp=0x3 lbmp=0x1

vlan create 113 pbmp=0x3 lbmp=0x1

vlan create 114 pbmp=0x3 lbmp=0x1

vlan create 115 pbmp=0x3 lbmp=0x1

vlan create 116 pbmp=0x3 lbmp=0x1

vlan create 117 pbmp=0x3 lbmp=0x1

vlan create 118 pbmp=0x3 lbmp=0x1

vlan create 119 pbmp=0x3 lbmp=0x1

//Use the VLAN information of the packet

port attrset port=0 type=29 value=1

port attrset port=1 type=29 value=1

port attrset port=2 type=29 value=1

port attrset port=3 type=29 value=1
```

3. istg configuration

```
istg init

//It is capable of port 0, 2, 3 stp check

istg enable pbmp=0xD

//Create istg 10-13

istg create 10 11 12 13

//Add vlan 100-104 in istg10

istg add 10 100 101 102 103 104

//Add vlan 105-109 in istg11

istg add 11 105 106 107 108 109

//Add VLAN 110-114 in istg12
```

```
istg add 12 110 111 112 113 114

//Add vlan 115-119 in istg13

istg add 13 115 116 117 118 119

//istg10 port0, lag0 stp status is diable

istg stp 10 pbmp=0x1 lbmp=0x1 state=disable

//istg11 port0, lag0 stp status is block

istg stp 11 pbmp=0x1 lbmp=0x1 state=block

//istg12 port0, lag0 stp status is learn

istg stp 12 pbmp=0x1 lbmp=0x1 state=learn

//istg13 port0, lag0 stp status is forward

istg stp 13 pbmp=0x1 lbmp=0x1 state=forward

//View stg configuration

istg show
```

Test 1:

Port0 configures SMAC and DMAC default values, increasing by 1 each time, VLAN increases from 100 to 119, and sends in burst mode 20, check the packet receiving status of port1-3.

Expected result: port0 sends 20 messages, port1 receives 5 messages, and port2 and 3 receive 5 messages together.

Test 2:

Port1 configures SMAC, DMAC and test 1 interchangeably, VLAN increases from 100 to 119 in sequence, sends 20 packets in burst mode, and checks the packet receiving status of port0 and port2-3.

Expected result: port0 receives 20 messages, and port2 and 3 receive 10 messages together.

Test 3:

Port2 configures SMAC, DMAC and test 1 interchangeably, VLAN increases from 100 to 119 in sequence, sends 20 packets in burst mode, and checks the packet receiving status of port0 and port1.

Expected result: port0 receives 5 packets, port1 receives no packets.

7.2.2 **estg** configuration example

Create 8 estgs, stg10 to stg17, and configure lag0 to include port2 and port3.

estg10 contains vlan 100-104, configure port1 to disable state, lag0 disable

estg11 contains vlan 105-109, configure port1 to block state, lag0 disable

estg12 contains vlan 110-114, configure port1 to learn state, lag0 disable

estg13 contains vlan 115-119, configure port1 to forward state, lag0 disable

estg14 contains vlan 120-124, configure lag0 to disable state, port1 disable

estg15 contains vlan 125-129, configure lag0 to block state, port1 disable

estg16 contains vlan 130-134, configure lag0 to learn state, port1 disable

estg17 contains vlan 135-139, configure lag0 to forward state, port1 disable

1. Clear **the istg** configuration

```
istg deinit
```

2. Lag configuration

```
trunk init
```

//Create a lag group (lagID is 0), port members are 2 and 3, hashkey is DstIp and SrcMac.

```
trunk add Id=0 Rtag=0x2001 Pbmp=0xC
```

3. VLAN configuration

//Create VLAN domain 100-139, including port0-1 and lag0

```
vlan create 100 pbmp=0x3 lbmp=0x1

vlan create 101 pbmp=0x3 lbmp=0x1

vlan create 102 pbmp=0x3 lbmp=0x1

vlan create 103 pbmp=0x3 lbmp=0x1

vlan create 104 pbmp=0x3 lbmp=0x1

vlan create 105 pbmp=0x3 lbmp=0x1
vlan create 106 pbmp=0x3 lbmp=0x1

vlan create 107 pbmp=0x3 lbmp=0x1

vlan create 108 pbmp=0x3 lbmp=0x1

vlan create 109 pbmp=0x3 lbmp=0x1

vlan create 110 pbmp=0x3 lbmp=0x1

vlan create 111 pbmp=0x3 lbmp=0x1

vlan create 112 pbmp=0x3 lbmp=0x1

vlan create 113 pbmp=0x3 lbmp=0x1

vlan create 114 pbmp=0x3 lbmp=0x1

vlan create 115 pbmp=0x3 lbmp=0x1

vlan create 116 pbmp=0x3 lbmp=0x1

vlan create 117 pbmp=0x3 lbmp=0x1

vlan create 118 pbmp=0x3 lbmp=0x1

vlan create 119 pbmp=0x3 lbmp=0x1

//Use the VLAN information of the packet

port attrset port=0 type=29 value=1

port attrset port=1 type=29 value=1

port attrset port=2 type=29 value=1

port attrset port=3 type=29 value=1
```

4. estg configuration

```
estg init

//It is possible to check the stp of ports 1, 2, and 3

istg enable pbmp=0xE

//Create estg 10-17

estg create 10 11 12 13 14 15 16 17

//Add vlan 100-104 in estg10

estg add 10 100 101 102 103 104
estg add 11 105 106 107 108 109
estg add 12 110 111 112 113 114
estg add 13 115 116 117 118 119
estg add 14 120 121 122 123 124
estg add 15 125 126 127 128 129
estg add 16 130 131 132 133 134
estg add 17 135 136 137 138 139
estg stp 10 pbmp=0x2 lbmp=0x1 state=disable
estg stp 11 lbmp=0x1 state=disable
estg stp 11 pbmp=0x2 state=block
estg stp 12 lbmp=0x1 state=disable
estg stp 12 pbmp=0x2 state=learn
estg stp 13 lbmp=0x1 state=disable
estg stp 13 pbmp=0x2 state=forward
estg stp 14 pbmp=0x2 lbmp=0x1 state=disable
estg stp 15 pbmp=0x2 state=disable
estg stp 15 lbmp=0x1 state=block
estg stp 16 pbmp=0x2 state=disable
estg stp 16 lbmp=0x1 state=learn
estg stp 17 pbmp=0x2 state=disable
estg stp 17 lbmp=0x1 state=forward
// Check if the configuration is as expected
estg show
```

Test 4:

Port0 configures SMAC and DMAC default values, increasing by 1 each time, VLAN increases from 100 to 139, and sends in burst mode 40, check the packet receiving status of port1-3.

Expected result: port0 sends 40 messages, port1 receives 5 messages, and port2 and 3 receive 5 messages together.

Test 5:

Port1 configures SMAC, DMAC and test 1 interchangeably, VLAN increases from 100 to 139, burst mode sends 40 messages, check

Check the packet receiving status of port0 and port2-3.

Expected result: port0 receives 40 messages, and port2 and 3 receive 0 messages.

Test 6:

Port2 configuration SMAC, DMAC and test 1 interchange, vlan from 100 to 139, burst mode sends 40, check

Check the packet receiving status of port0 and port1.

Expected result: port0 receives 40 messages, port1 receives 0 messages.

8 **Erps** function

The function of erps is similar to that of stp, and is also divided into two parts: ierps and eerps. The specific instructions are as follows. “[ ]” indicates optional, and “|” indicates multiple selection 1.

8.1 Introduction to Erps Instructions

8.1.1 ierps instruction description

/*ierps initialization. After initialization, the default ierpsid=0 will be created, including all ports and lags of vlan1-vlan4095 and ierps0 The erps status of the port is diable (it takes effect only after the port erps_check is enabled, and it is in forward status before enabling). */

ierps init

//ierps function is turned off

ierps deinit

// Enable erps check for the corresponding port

ierps enable pbmp=<pbmp>

// Enable the erps check of the corresponding port

ierps disable pbmp=<pbmp>

//Create ERPSs, you can create multiple erps

ierps create [<id>] [...]

//Cancel ERPSs configuration

ierps destroy <id> [...]

//View the configured ERPS(s), display the VLANs included in the erps and the corresponding erps status

ierps show [<id>]

//Add VLAN(s) to an ERPS

ierps add <id> <vlan_id> [...]

//Remove VLAN(s) from an ERPS

ierps remove <id> <vlan_id> [...]

//Display all erps status

ierps state

//Display a certain erps status

ierps state<id>

//Set the status of a certain ERPS

ierps state<id> <pbmp=xx>[<lbmp=xx>] <state=disable|forward>

8.1.2 **eerps** instruction description

The eerps instruction is similar to ierps and will not be described in detail here.

eerps init

eerps deinit

eerps enable pbmp=<pbmp>

eerps disable pbmp=<pbmp>

eerps create [<id>][...]

eerps destroy <id>[...]

eerps show [<id>]

eerps add <id> <vlan_id> [...]

eerps remove <id> <vlan_id> [...]

eerps state

eerps state<id>

eerps state<id> <pbmp=xx>[<lbmp=xx>] <state=disable|forward>

8.2 **erps** configuration example

8.2.1 **ierps** configuration example

Create two ierps, erps 2 and erps 3, and configure lag0 to include port2 and port3.

erps2 contains vlan 100-104, and configures port0 and lag0 to disable state

erps3 includes vlan 105-109, and configures port0 and lag0 to forward state

1. Lag configuration:

trunk init

//Create a lag group (lagID is 0), port members are 2 and 3, hashkey is DstIp and SrcMac.

trunk add Id=0 Rtag=0x2001 Pbmp=0xC

/* Enable the lag port function. Generally, the management chip enables the Lag function by default. If the configuration is wrong, you can force the lag to be enabled through this configuration. Function; unmanaged type does not support lag function*/

global set type=0 value=0

2. Vlan configuration:

//Create VLAN domain 100-119, including port0-1 and lag0

vlan create 100 pbmp=0x3 lbmp=0x1

vlan create 101 pbmp=0x3 lbmp=0x1

vlan create 102 pbmp=0x3 lbmp=0x1

vlan create 103 pbmp=0x3 lbmp=0x1

vlan create 104 pbmp=0x3 lbmp=0x1

vlan create 105 pbmp=0x3 lbmp=0x1

vlan create 106 pbmp=0x3 lbmp=0x1

vlan create 107 pbmp=0x3 lbmp=0x1

vlan create 108 pbmp=0x3 lbmp=0x1

vlan create 109 pbmp=0x3 lbmp=0x1

//Use the VLAN information of the packet

port attrset port=0 type=29 value=1

port attrset port=1 type=29 value=1

port attrset port=2 type=29 value=1

port attrset port=3 type=29 value=1

3.ierps configuration:

ierps init

//It can check ports 0, 2, and 3 erps

```
Ierps enable pbmp=0xD

//Create ierps 2 3

ierps create 2 3

//Add vlan 100-104 in ierps 2

ierps add 2100 101 102 103 104

//Add vlan 105-109 in ierps 3

ierps add 3105 106 107 108 109

//ierps 2 port0, lag0 erps status is diable

ierps state 2 pbmp=0x1 lbmp=0x1 state=disable

//ierps 3 port0, lag0 erps status is forward

ierps state 3 pbmp=0x1 lbmp=0x1 state=forward

//View ierps configuration

ierps show
```

Test 1:

Port0 configures SMAC and DMAC default values, increasing by 1 each time, VLAN increases from 100 to 109, and sends in burst mode 10, check the packet receiving status of port1-3.

Expected result: port0 sends 10 messages, port1 receives 5 messages, and port2 and 3 receive 5 messages together.

Test 2:

Port1 configures SMAC, DMAC and test 1 interchangeably, VLAN increases from 100 to 109 in sequence, sends 10 packets in burst mode, and checks the packet receiving status of port0 and port2-3.

Expected result: port0 receives 10 messages, and port2 and 3 receive 5 messages together.

8.2.2 eerps configuration example

For the eerps configuration example, please refer to ierps, which will not be described in detail here.

9 **Lag** Test

Supports 8 lag groups, lag ID is 0 ~ 7. Each lag group supports up to 32 ports, and the same port can only belong to one lag group.

1. Create **a vlan domain, and the vlan** domain port is **the lag** port

//Create a vlan domain of 100, add three lag groups to the vlan domain, and the lagIDs are 0, 1, and 2

vlan create 100 lbmp=0x7

2. Lag function initialization

//When using the lag function, you should first initialize

trunk init

3. Lag function is turned off

trunk deinit

4. Create a **lag** group

trunk add Id=<lagID> Rtag=<psc> Pbmp=<pBmp>

Create a lag group and specify the port members and hash key in the lag group. The default hash algorithm is crc8. The hash key supports L2, L3, L4 layer partial domain combination, such as SrcMac, DstMac, SrcIp, DstIp, L4SrcPort, L4DstPort, etc. The corresponding psc values are as follows:

//SrcMac=0x1

//DstMac=0x2

//EthType=0x4

//UpdStag=0x8

//StagVid=0x10

//StagCfi=0x20

//StagCos=0x40

UpdCtag=0x80

CtagVid=0x100

CtagCfi=0x200

CtagCos=0x400

SrcIp=0x1000

DstIp=0x2000

Protocol=0x4000

Tos=0x8000

L4SrcPort=0x10000

L4DstPort=0x20000

//When UpdStag is set, StagVid/StagCfi/StagCos indicates that the updated stag value is used as //hashkey; UpdStag is not When set, StagVid/StagCfi/StagCos means using the stag //value in the message header as the hashkey. The same applies to UpdCtag

example:

/*Create a lag group (lagID is 2), port members are 1, 2, 3, hashkey is DstIp and SrcMac. If you send a message with random DstIp and SrcMac, the message will be output in a load-balanced manner on ports 1, 2, and 3. If you send a message with fixed DstIp and SrcMac, the message will be output from a fixed port among ports 1, 2, and 3. */

trunk add Id=2 Rtag=0x2001 Pbmp=0xE

5. Set **the hash key**

trunk psc Id=<lagID> Rtag=<psc>

6. Set **the hash** algorithm

trunk alg Id=<lagID> Hash=<alg>

The hash algorithm supports crc8, crc16, crc32, etc. The alg value can be: crc8=0, crc32Lo=1, crc32Hi=2, crc16Bs=3, crc16cc=4ÿxor16=5ÿcrc16ccHiXor8=6ÿcrc16ccHiXor4=7ÿcrc16ccHiXor2=8ÿcrc16ccHiXor1=9

example:

//Set the hash algorithm of lag group 3 to crc16Bs

trunk alg Id=3 Hash=3

7. Set up **lag** group member failure sharing

trunk failover Id=<lagID> Able=<1|0>

example:

/* Enable the failure sharing function of lag group 2. When the link status of a port member in lag group 2 changes, such as from up to down, the traffic of the port will be shared to other ports in the lag group for output. When the port link status changes from down to up, the traffic will resume output from the port. */

trunk failover Id=2 Able=1

8. Destroy **the trunk** group

trunk destroy Id=<trunk_id>

9. Display **trunk** group information

trunk show

10 MAC address learning

1. Business configuration

Edit a unicast Ethernet message on port 1 (optical port), MacSa increases, the format is: MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100).

2. Configuration Commands

The MAC address learning function is enabled by default and no related configuration is required.

3. Test **1**

MacSA increments, and the step mode starts from MacSa0 and sends 1000 messages.

Expected result 1

Ports 2~6 all receive 1000 messages from port 1.

//Multicast replication function.

4. Test **2**

Select a port from port 2 to 6 and edit the macDa message starting from MacSa0 in increments. The format is: MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

Expected Result 2

1000 packets are received from the selected port at port 1, and no packets are received at other ports.

//Learn 1000 mac addresses.

11 MAC address learning limit

1. Business configuration

Edit a unicast Ethernet message on port 1 (optical port), MacSa increases, the format is: MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100).

2. Configuration commands:

//Set the address learning limit to 500

```
L2 learnlimit global=1 limit=500
```

Enable address learning restriction:

```
global set type=4 value=1
```

3. Test **1**

MacSA increments, and the step mode starts from MacSa0 and sends 1000 messages.

Expected result 1

Ports 2~6 all receive 1000 messages from port 1.

//Multicast replication function.

4. Test **2**

Select a port from port 2 to 6 and edit the macDa message starting from MacSa0 in increments. The format is: MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

Expected result 2:

Port 1 receives 1000 packets from the selected port, of which 500 are unicast forwarded and the other 500 are flooded. Other ports receive 500 packets due to flooding.

// 500 MAC addresses are learned, and MAC addresses not in this range are flooded.

12 MAC address aging

1. Business configuration

Edit a unicast Ethernet message on port 1 (optical port), MacSa increases, the format is: MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100).

2. Configuration Commands

//Set the aging time to 60s, in seconds

age 60

3. Test **1**

MacSA increments, and the step mode starts from MacSa0 and sends 1000 messages.

Expected result 1

Ports 2~6 all receive 1000 messages from port 1.

//Multicast replication function

4. Test **2**

Select a port from port 2 to 6 and edit the macDa message starting from MacSa0 in increments. The format is: MacDa, MacSa, Vlan (tpid=0x8100, cos=0, cfi=0, vid=0x100);

Expected Result 2

1000 packets are received from the selected port at port 1, and no packets are received at other ports.

//Learn 1000 mac addresses.

5. Test **3**

Wait for more than 60 seconds and repeat packet 2.

Expected Result 3

Since the address has aged, no outgoing port can be found according to MacDa, and all ports receive 1000 messages.

Additional related commands:

Port aging enable switch:

```
age penable port=<port_id> value=<0|1>

//port port number

//value 1 is on, 0 is off, by default all ports are enabled
```

Fast aging based on matching rules

```
age rule Port=<port_id> MACaddress=<0x> Vlanid=<> TrunkGroupID=<> value=<0|1>
```

The above commands can perform fast aging based on port number, mac address, vlan id, and trunk id. Value 0 means closing the matching rule. A value of 1 enables the rule.

13 Acl, eAcl and vfp

The chip includes multiple flow identification engines, namely ACL, EACL and vfp, with maximum entry numbers of 512, 256 and 128 respectively. The engines all use TCAM table lookup.

The number of ACL entries is 512. It can be divided into MacKey, IPv4Key, IPv6Key and MixKey according to the configuration. Each table stores When the ACL table stores all complete MacKey or IPv4Key entries, the maximum width of the Key value will reach 320. bits, the number of entries is halved to 256. When the ACL table stores all complete IPv6Key or MixKey entries, the maximum width of the Key value is When the width of the key is 160 bits, the number of entries reaches the maximum. The number of entries is 512. ACL includes two flow search engines, ACL0 and ACL1, which share all ACL search entries.

ACL can classify traffic based on port, L2, L3, and L4 content. Different Keys can be used for different input packet types. The specific contents of various Key combinations are shown in Table 13-1.

Table 13- 1 ACL_KEY

Key Type	content
MacKey	DMAC[47:0],SMAC[47:0],STAG[15:0],CTAG[15:0], ETHERTYPE[15:0], PORTS[34:0], L3UDF[7:0], etc.
IPv4Key	DIP[31:0],SIP[31:0],PROTOCOL[7:0],TOS[7:0],PORTS[34:0], L4SrcPort[15:0], L4DsrPort[15:0], L3UDF[7:0], L4UDF[7:0], etc.
IPv6Key	DIP[127:0],SIP[127:0],PROTOCOL[7:0],TOS[7:0],PORTS[34:0],L4SrcPort[15:0],L4DsrPort[15:0], L3UDF[7:0], L4UDF[7:0], TTL[7:0], DMAC[47:0], TCPFLAG[7:0], STAG[15:0], CTAG[15:0], etc.
MixKey	DIP[127:0],SIP[127:0],PROTOCOL[7:0],TOS[7:0],PORTS[34:0],L4SrcPort[15:0],L4DsrPort[15:0], L3UDF[7:0],L4UDF[7:0],TTL[7:0],DMAC[47:0],SMAC[47:0],ICMPCODE[7:0],ICMPTYPE[7:0],TTL [7:0], TCPFLAG[7:0], STAG[15:0], CTAG[15:0], IPOPTION[0:0], IPHDRERR[0:0], L3TP[3:0], L4TP[2:0], etc.

By default, IPv4 and Arp packets look for IPv4Key, IPv6 packets look for IPv6Key, and other packets look for MacKey. Controls whether to force MacKey, IPv4Key, IPv6Key, or MixKey for ACL lookup based on the port. ACL0 and ACL1 You can configure different Key type searches.

Select the search key of ACL0 and ACL1 according to the different types and configurations of data packets. The search key contains different ACL templates. The two ACL search engines will obtain two processing behaviors of the business flow. Arbitration: The arbitration principle is that if there is a conflict between two processing behavior items, the processing behavior corresponding to the ACL1 search engine shall prevail.

The search method and key of EACL are basically the same as those of ACL.

Vfp can be divided into VlanKey, MacKey, IPv4Key and IPv6Key according to the configuration, and each table stores different types of Key values. The specific contents of various keys are shown in Table 13-2.

Table 13- 2 VFP_KEY

Key Type	content
VlanKey	SMAC[47:0],STAG[15:0],CTAG[15:0],SIP[17:0],L2Tp,L3Tp,L4Tp ETHERTYPE[15:0],GPORT, etc.
MacKey	SMAC[47:0],DMAC[47:0],STAG[15:0],CTAG[15:0],PORTS[34:0],

	ETHERTYPE[15:0],L3UDF[7:0],L4UDF[7:0],L2Tp,L3Tp,L4Tp, etc.
IPv4Key	DIP[31:0],SIP[31:0],PROTOCOL[7:0],TOS[7:0],PORTS[34:0],L4SrcPort[15:0],L4DsrPort[15:0],L3UDF[7:0],L4UDF[7:0],TTL[7:0],TCPFLAG[7:0],ipLen,ipOption,L3Tp,L4Tp, etc.
IPv6Key	DIP[127:0],SIP[127:0],PROTOCOL[7:0],TOS[7:0],PORTS[34:0],L4SrcPort[15:0],L4DsrPort[15:0], L3UDF[7:0],L4UDF[7:0],TTL[7:0],DMAC[47:0],ICMPCODE[7:0],ICMPTYPE[7:0],TCPFLAG[7:0], STAG[15:0], CTAG[15:0], ipOption, ipHdrErr, L3TP[3:0], L4TP[2:0], etc.

By default, IPv4 and Arp packets search for IPv4Key, IPv6 packets search for IPv6Key, and other packets search for MacKey. You can also control whether to force the use of VlanKey, MacKey, IPv4Key, and IPv6Key for searching based on the port.

13.1 Create related commands

1. Enable **ACL/EACL/VFP** query switch based on port

//port: port ID

//value: 1 means on, 0 means off

/*type: 60 means ACL0 search, 61 means ACL1 search; 90 means EACL0 search, 91 means EACL1 search; 2 means VFP search*/

port attrset port={portId} **value={0|1}** type={val}

2. Force the use of a certain **key** query **ACL** or **EACL** based on port configuration

//"{}" indicates required selection, "[]" indicates optional selection, "|" indicates one or more selections

```
fp forcekey port={portId} [ipv4ForceMacKey0=0|1] [ipv4ForceIpv6Key0=0|1]
 [ipv4ForceMixKey0=0|1] [ipv6ForceMacKey0=0|1] [ipv6ForceIpv4Key0=0|1]
 [ipv6ForceMixKey0=0|1] [macForceIpv6Key0=0|1] [macForceIpv4Key0=0|1]
 [macForceMixKey0=0|1] [ipv4ForceMacKey1=0|1] [ipv4ForceIpv6Key1=0|1]
 [ipv4ForceMixKey0=1|1] [ipv6ForceMacKey1=0|1] [ipv6ForceIpv4Key1=0|1]
 [ipv6ForceMixKey1=0|1] [macForceIpv6Key1=0|1] [macForceIpv4Key1=0|1]
 [macForceMixKey1=0|1]
```

3. Force the use of a certain **key** to query **VFP** based on port configuration

//"{}" indicates required selection, "[]" indicates optional selection, "|" indicates one or more selections

```
fp forcekey port={portId} [useVlanKey=0|1]
 [ipv4ForceMacKey=0|1][ipv4ForceIpv6Key=0|1]
```

[ipv6ForceMacKey=0|1] [ipv6ForceIpv4Key=0|1]

[macForceIpv4Key0=0|1][macForceIpv6Key=0|1]

4. Initialization should be called once first

fp init

5. Clear **ACL, EACL, and VFP** indication configuration

fp qset clear

6. Indicates that the current module is **ACL, EACL** or **VFP**

//StageIngress represents ACL

// StageEgress represents EACL

//StageLookup represents VFP

fp qset add {StageIngress | StageEgress | StageLookup}

7. Create Groups

/*

* Create Group

*pri: not used, always fill in 1

*gid: group ID, globally unique

*keyTp: 0(MacKey), 1(IPv4Key), 2(IPv6Key), 3(MixKey)

*size: indicates the number of entries currently assigned to this Key type, and the value is an integer multiple of 4

*mode: 0 (Key minimum width), 1 (two entries form a Key - 2 times the minimum width), 3 (four entries form a Key)

*/

fp group create {pri} {gid} {keyTp} {size} {mode}

8. Create **entry**

/*

*Create an Entry. The Entry ID specifies the priority, the smaller the ID, the higher the priority. The entry ID ranges of each module are as follows: iAcl: 1~512; eAcl: 513~768; vfp: 769~896; evfp: 897~928 *gid: group ID, globally unique

*eid: entry ID, globally unique

*/

fp entry create {gid} {eid}

9. Configure matching domain

/*

*eid: entry ID

*field: Matching domain. Contains SrcIp, DstIp, SrcIp6, DstIp6, IpProtocol, L4SrcPort, L4DstPort,

*Inports,InportL2Tp,L3Tp,L4Tp,L5Tp,L4SrcPortRng,L4DstPortRng,Color,

*OuterVlan,InnerVlan,Ttl,TcpControlSrcMac,DstMac,EtherType,SrcIpRng,

*DstIpRng,SrcIp6Rng,DstIp6Rng, etc.

*data: key value

*mask: keyMask

*/

fp qual {eid} {field} {data} {mask}

10. Post-matching behavior

/*

*eid: entryID, globally unique

*action: behavior after matching.

*Support RedirectPort (redirect to port, need to specify port ID),

*RedirectTrunk (redirect to the trunk group, need to specify the trunk group ID),

*RedirectMcast (redirect to the multicast group, need to specify the multicast group ID),

*UpdateCounter (enable hit count statistics), Drop (discard), Permit (forward), Trap (trap to the specified port),

*OuterVlanNew (modify or add outer vlan, need to specify vlanID), OuterVlanDelete (delete outer vlan)

*OuterVlanPrioNew (modify outer vlan pri, need to specify vlanID)

*InnerVlanNew (modify or add inner vlan, need to specify vlanID), InnerVlanDelete (delete inner vlan)

*InnerVlanPrioNew (modify inner vlan pri, need to specify vlanID) etc.

*VFP supports Drop, Trap, PrioIntNew (specify internal priority, need to specify priority value), DoNotLearn (do not learn),

*DtOuterVlanCopyInner (double tag message, copy inner vid to outer vid),

*DtOuterVlanPrioCopyInner (double tag message, copy inner pri to outer pri),

*DtOuterVlanCfiCopyInner,DtInnerVlanCopyOuter,DtInnerVlanPrioCopyOuter,DtInnerVlanCfiCopyOuter,

*SotInnerVlanCopyOuter (single outer layer message, copy outer layer vid to inner layer vid),

*SotInnerVlanPrioCopyOuter,SotInnerVlanCfiCopyOuter,

*SitOuterVlanCopyInner (single inner layer message, copy inner layer vid to outer layer vid),

*SitOuterVlanPrioCopyInner,SitOuterVlanCfiCopyInner,

*DtOuterVlanNew <vid> (double tag message, modify the outer vid, need to specify the vid parameter),

*DtOuterVlanPrioNew<pri>,DtOuterVlanCfiNew <cif>,

*DtInnerVlanNew,DtInnerVlanPrioNew,DtInnerVlanCfiNew,

*SotOuterVlanNew,SotOuterVlanPrioNew,SotOuterVlanCfiNew,

*SitInnerVlanNew,SitInnerVlanPrioNew,SitInnerVlanCfiNew,

*SotInnerVlanAdd,SotInnerVlanPrioAdd,SotInnerVlanCfiAdd,

*SitOuterVlanAdd (add outer VLAN to a single inner message, vid must be specified),

*SitOuterVlanPrioAdd,SitOuterVlanCfiAdd,

*UtInnerVlanAdd (without Tag message, add inner vlan, need to specify vid),

*UtInnerVlanPrioAdd,UtInnerVlanCfiAdd,

*UtOuterVlanAdd,UtOuterVlanPrioAdd,UtOuterVlanCfiAdd,

*DtOuterVlanDelete,DtInnerVlanDelete,SotOuterVlanDelete,SitInnerVlanDelete

*/

fp action add {eid} {action} [val]

11. Send hardware entries

fp entry install {eid}

12. Other commands can be viewed through **fp help**

13.2 Delete related commands

//Deinitialization

fp detach

//When eid is specified, only the corresponding entry is deleted. Without parameters, all entries are deleted.

fp entry destroy [eid]

//When gid is specified, only the corresponding group is deleted. Without parameters, all groups are deleted.

fp group destroy [gid]

13.3 Display related commands

//Display fp related configuration

fp show [group|entry] [id]

//Show all supported quals and actions

fp list actions|quals [ifp|efp|vfp]

Other commands can be viewed via **fp help**

13.4 Command Line Examples

// Enable the ACL0 query function of port 1

port attrset port=1 value=1 type=60

fp init

fp qset clear

//Indicates that the current configuration is for the iACL module

fp qset add StageIngress

```
// Layer 2 message forced query ipv4key

fp forcekey port=1 macForceIpv4Key0=1
```

```
/*Create group1 as IPv4Key, with 40 entries (entryID is 1~40), mode is 2-in-1 (two entries are combined into one Key, the actual number of ACL entries occupied is: 40*2=100*/

fp group create 1 1 1 40 1

//Based on group1, create entry1

fp entry create 1 1
```

```
//Match the five-tuple of messages

fp qual 1 DstIp 0x12345678 0xffffffff

fp qual 1 SrcIp 0x87654321 0xffffff00

fp qual 1 IpProtocol 0x33 0xff

fp qual 1 L4SrcPort 0x1244 0xffff

fp qual 1 L4DstPort 0x6789 0xffff
```

```
//Match the packets coming into ports 0, 1, and 2 (the mask is the inverse of the key value)

fp qual 1 inPorts 0x7 0xfffffff8
```

```
//The behavior is to redirect to port 31 (cpu port)

fp action add 1 redirectport 31
```

```
//Statistical hit count

fp action add 1 UpdateCounter
```

```
//Send hardware entries

fp entry install 1


//Get ACL hit count statistics

fp counter get 1


//Based on group1, create entry2

fp entry create 1 2


//Match tclflag

fp qual 2 TcpControl 0x12 0xff


fp action add 2 OuterVlanNew 1000

fp action add 2 InnerVlanNew 1001

fp entry install 2

...
```

/*Create group2 as IPv6Key, with 60 entries (entryID is 41~100), mode is 4 in 1 (four entries are combined into one Key), the actual number of ACL entries is: 60*4=240*/

```
fp group create 1 2 2 60 3

//Based on group2, create entry41

fp entry create 2 41

fp qual 41 SrcIp6 5566:6677:7788:8899:1122:2233:3344:4455 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

fp qual 41 DstMac 12:34:56:77:90:32 ff:ff:ff:ff:ff:00
```

//Behavior discard

```
fp action add 41 drop
```

//Send hardware entries

```
fp entry install 41
```

/*Create group3 as MixKey, with 40 entries (entryID is 101~140), mode is 4-in-1 (four entries are combined into one Key), the actual number of ACL entries is: 40*4=160*/

```
fp group create 1 3 3 40 3
```

//Based on group3, create entry110

```
fp entry create 3 110
```

//Match the dip of ipv6 message

```
fp qual 110 DstIp6 5566:6677:7788:8899:1122:2233:3344:4455 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

If you want to match the dip of ipv4 packets, you can configure it as follows:

```
fp qual 110 DstIp6 0:0:0:0:0:0:3344:4455 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff00
```

(The lower 32 bits represent ipv4, the upper 96 bits key is 0, the mask is all 1, and the matching domain can be added

fp qual 110 EtherType 0x0800 0xffff)

//Match message smac

```
fp qual 110 SrcMac 12:34:56:77:90:32 ff:ff:ff:ff:ff:00
```

//The behavior is trap, and the trap port is configured by other commands (for example, the trap port is the CPU port, dp trap set port=31)

```
fp action add 110 trap
```

//Send hardware entries

```
fp entry install 110
```

14 Mirror

This chip supports the mirroring function of input/output ports, input/output forwarding VLANID, MAC address and input/output service flow. The number of input/output mirroring ports is 1 each.

14.1 Input/output ports, input/output forwarding **VLAN ID** mirroring

14.1.1 Creating a mirror mode

//"{}" indicates required selection, "[]" indicates optional selection, "|" indicates one or more selections

//PortIngress: mirroring based on input port;

//PortEgress: based on output port mirroring

//VlanIngress: forward VLANID mirroring based on ingress;

//VlanEgress: Mirror based on egress forwarding VLANID

//src_port: mirrored port number. Valid when Mode is PortIngress or PortEgress

//vlan_id: Forwarding VLAN ID. Valid when Mode is VlanIngress or VlanEgress

mirror create Mode={PortIngress | PortEgress | VlanIngress | VlanEgress}

{ SrcPort=src_port | Vlan=vlan_id }

14.1.2 Deleting an Image

mirror destroy Mode={PortIngress | PortEgress | VlanIngress | VlanEgress}

{ SrcPort=src_port | Vlan=vlan_id }

14.1.3 Configuring the Input Mirror Port

mirror IDestPort {dst_port}

14.1.4 Configuring the Output Mirror Port

mirror EDestPort {dst_port}

Example:

/*Based on input mirroring on port 1, if the mirroring port is port 2, the traffic received on port 1 will be mirrored out from port 2*/

mirror create Mode=PortIngress SrcPort=1

```
mirror lDestPort 2
```

14.1.5 Display Mirror Configuration

```
mirror show [port | vlan]
```

14.2 MAC Address Mirroring

By searching the MAC address table, the mirroring operation is performed on the matching source MAC address and destination MAC address.

14.2.1 Flow Mirroring

In the ACL and EACL modules, you can set sampling or mirroring functions based on flows.

15. Flow Control

15.1 Command Line Introduction

The command to enable the port flow control function is as follows:

/*The flow control can be turned on and off by enabling the command*/

```
port pause set enable port=<port_id> [TxEn=<1|0>] [RxEn=<1|0>]
```

/*When sending traffic to one port through multiple ports, the sum of the port Sport thresholds will be greater than the Dport threshold, which will cause packet loss before the flow control function takes effect. Therefore, it is necessary to modify the Sport threshold of the flow control port to ensure that the sum of the Sport thresholds is less than Dport, and the thdon value is greater than the thdoff value. */

```
port pause get threshold[enable] port=<port_id>
port pause set threshold port=<port_id> [thdon=<xx>] [thdoff=<xx>]
```

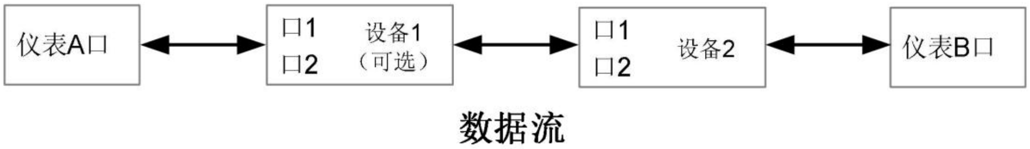
15.2 Configuration Example

15.2.1 Example 1

1. Business configuration

Connect the two devices in series and open the service channels as described in the previous chapters. The traffic is sent from instrument A port to instrument B port, with 1G traffic:

Assume the connection is as follows:



2. Device **2** configuration

```
port pause set enable port=0 TxEn=1
```

Device 1 (optional) configuration:

```
port pause set enable port=1 RxEn=1
port pause set enable port=0 TxEn=1
```

Shaping performs speed limiting on port 1 of device 2:

```
dp shape port set Port=1 Mode=0 FillRate=10000 BurstSize=0xffff Quantum=2
```

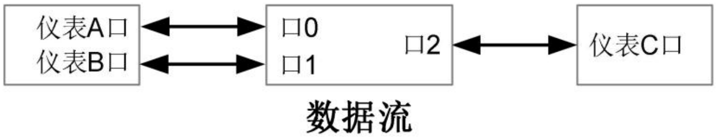
3. Expected results

The receiving rate of instrument B port is 10000 kbps, and instrument A port can receive pause frames. If instrument A supports flow control, the sending rate will be reduced to 10000 kbps; when the speed limit is cancelled, or the shaping rate is set to be greater than the inlet rate, instrument A port will receive pause stop frames.

15.2.2 Example **2**

1. Business configuration

As shown in the figure, the service channels are opened as described in the previous chapters, and the traffic is sent from the instrument A port and B port to the instrument C port, with 1G traffic each:



2. Flow control command configuration

Since the default Sport flow control opening threshold of the port is 600, the closing threshold is 400, and the Dport green packet cache threshold is 800. At this time, you can set the port Sport thdon=300, thdoff=200 for ports 0 and 1, and enable the flow control function for ports 0 and 1. The configuration instructions are as follows:

```
port pause set enable port=0 TxEn=1

port pause set enable port=1 TxEn=1

port pause set threshold port=0 thdon=300 thdoff=200

port pause set threshold port=1 thdon=300 thdoff=200
```

3. Expected results

The receiving rate of port C of the instrument is 1Gbps, and ports A and B of the instrument can receive pause frames. If the instrument supports flow control, the sending rates of ports A and B will be reduced to 500 Mbps; when the speed limit is cancelled, or the sum of the rates of ports A and B is less than 1Gbps, ports A and B of the instrument will receive pause stop frames.

16 TM Test

16.1 SP Mode (Priority Mode)

1. Business configuration

Edit 8 unicast Ethernet messages on port 1 (optical port) in the format of MacDa, MacSa, Vlan (tpid=0x8100, cos=0~7, cfi=0, vid=0x200).

2. Configuration Commands

```
vlan create 0x200 pbmp=0x7E

port attrset port=1 value=0 type=10

vlan vlanset vlanid=0x200 value=1 type=24

vlan vlanset vlanid=0x200 value=10 type=25

policing map profile create QosProIndex=10 UseL2Info=1 TrustCtag=0 PhbPtr=0

//Mapping of vlan priority to internal priority

policing map vlan pri map set QosProIndex=10 PktPri=0 InternalPri=0 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=1 InternalPri=1 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=2 InternalPri=2 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=3 InternalPri=3 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=4 InternalPri=4 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=5 InternalPri=5 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=6 InternalPri=6 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=7 InternalPri=7 Color=2

// Mapping of internal priority to queue

dp port queue map set Port=2 Pri=0 Queue=0

dp port queue map set Port=2 Pri=1 Queue=1

dp port queue map set Port=2 Pri=2 Queue=2

dp port queue map set Port=2 Pri=3 Queue=3

dp port queue map set Port=2 Pri=4 Queue=4
dp port queue map set Port=2 Pri=5 Queue=5

dp port queue map set Port=2 Pri=6 Queue=6

dp port queue map set Port=2 Pri=7 Queue=7

//Port speed limit

dp shape port set Port=2 Mode=0 FillRate=rate1 BurstSize=0xffff Quantum=2

//Disable port-based admission control and enable queue-based admission control

dp admin dport set port=2 enable=0 coloraware=1

dp admin global queue set enable=1
```

3. Expected results

When the port speed limit is rate1 kbps and the total traffic of 8 flows is rate2 kbps, if rate1 < rate2, high-priority packets will be passed first and no packets will be lost. Low-priority packets will be lost or will not be received at all (only part of the traffic will be allocated or no traffic will be allocated at all).

16.2 WFQ Mode (Weighted QoS Mode)

1. Business configuration

Edit 8 unicast Ethernet messages on port 1 (optical port) in the format of MacDa, MacSa, Vlan (tpid=0x8100, cos=0~7, cfi=0, vid=0x200).

2. Configuration Commands

```
vlan create 0x200 pbmp=0x7E

port attrset port=1 value=0 type=10

vlan vlanset vlanid=0x200 value=1 type=24

vlan vlanset vlanid=0x200 value=10 type=25

policing map profile create QosProIndex=10 UseL2Info=1 TrustCtag=0 PhbPtr=0

policing map vlan pri map set QosProIndex=10 PktPri=0 InternalPri=0 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=1 InternalPri=1 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=2 InternalPri=2 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=3 InternalPri=3 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=4 InternalPri=4 Color=2
policing map vlan pri map set QosProIndex=10 PktPri=5 InternalPri=5 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=6 InternalPri=6 Color=2

policing map vlan pri map set QosProIndex=10 PktPri=7 InternalPri=7 Color=2

dp port queue map set Port=2 Pri=0 Queue=0

dp port queue map set Port=2 Pri=1 Queue=1

dp port queue map set Port=2 Pri=2 Queue=2

dp port queue map set Port=2 Pri=3 Queue=3

dp port queue map set Port=2 Pri=4 Queue=4

dp port queue map set Port=2 Pri=5 Queue=5

dp port queue map set Port=2 Pri=6 Queue=6

dp port queue map set Port=2 Pri=7 Queue=7

dp shape port set Port=2 Mode=0 FillRate=rate1 BurstSize=0xffff Quantum=2

dp admin dport set port=2 enable=0 coloraware=1

dp admin global queue set enable=1

dp schedule queue set Port=2 pri0Weight=1 pri1Weight=2 pri2Weight=3 pri3Weight=4 pri4Weight=5 pri5Weight=6 pri6Weight=7 pri7Weight=8 wrrquantum=2 wrrpri=3 schmode=1 schbmp=0xFF
```

3. Expected results

When the port speed limit is rate1 kbps and the total traffic of 8 flows is rate2 kbps, if rate1 < rate2, the traffic will be The proportional distribution of weight1:weight2:…:weight8.

17 Storm Suppression

Supports three types of storm suppression: global, port, and fid. The following takes port-based storm suppression as an example to introduce its configuration command line.

To suppress the unknown multicast packets on port 2, the configuration is as follows:

1. Set the equivalent packet length and global configuration of the frame interval and preamble

```
policing storm control global set PreambleLen=20
```

2. Port-based storm control token update settings: update enable, maximum update entries 187, mode=1 is port-based model

```
policing storm control update set Mode=1 UpdEn=1 MaxUpdIdx=187 DelayInterval=10000
```

3. Enable the storm control function for unknown multicast. Index=2 means port 2, and FwdType=2 means unknown multicast (0: known multicast). 1: known multicast, 2: unknown multicast, 3: broadcast)

```
policing storm control enable set Mode=1 Index=2 FwdType=2 Enable=1
```

4. Specific speed limit configuration settings for Storm Control, PolType=0 means speed limit based on bytes, Limit is the speed limit value (in kb), The specific speed limit value is related to the clock frequency, and BurstSize is the bucket depth (in kb)

```
policing storm control config set Mode=1 index=2 FwdType=2 PolType=0 Limit=500000
BurstSize=10000000
```

Note: BurstSize cannot be set too small, it is best to be above 50 times Limit.

18 **PKT DMA** function

Note: This chapter is only applicable to FSL91030M. PKT DMA is used by the CPU to send and receive packets from the switch chip panel port (not the debug network port). The command is mainly used for debugging. Debug the application layer packet receiving and sending functions, and set filters (only for packets coming from non-debugging network ports).

18.1 Introduction to functest pktdma command

functest

functest is a command to test SDK functions. pktdma is a subcommand of functest. This subcommand has 10 subfunctions (mode=1~10).

Use the following command to view usage:

functest pktdma mode=0

18.2 functest pktdma configuration example

18.2.1 Application layer packet sending and receiving test configuration example

Configure the panel port port3 to forward the packet to the CPU, and the packet type 0x88f7 does not pass through the protocol stack, but is directly uploaded to the application layer. The application layer prints out the received message. After receiving the packet, the application layer forwards it along the original path and then prints out the sent message.

Channel configuration for sending and receiving packets:

1. Reset the switch chip

```
modreg TOP_CFG_REG_RESET_GLOBAL UPI_RST_GLB_LOGIC_N=0 UPI_RST_GLB_UPI_N=0

modreg TOP_CFG_REG_RESET_GLOBAL UPI_RST_GLB_LOGIC_N=1 UPI_RST_GLB_UPI_N=1

modreg PKT_DMA_REG_PKT_DMA_AXI_RD_CFG UPI_AXI4_RLEN_MAX=0xf

modreg PKT_DMA_REG_PKT_DMA_AXI_WR_CFG UPI_AXI4_WLEN_MAX=0xf
```

2. Set the **CPU** channel: forward the packets entering **port 3** to **the CPU port (port 31)**

```
modify I_VT_PORT_SRM 3 1 FWD_VLD=1 OUT_LPORT=31

modify I_NET_PORT_SRM 3 1 STP_CHK_EN=0 BRG_EN=0
```

3. Set **the CPU** delivery channel: the packets sent from **the CPU** port do not pass through **the PP**

```
modreg E_ACL_LOOP_CTL LOOP_BYPASS=0x80

modreg E_EE_LOOP_CTL LOOP_BYPASS=0x80
modreg E_PF_LOOP_CTL LOOP_BYPASS0=0x80 LOOP_BYPASS1=0x80

modreg E_POL_LOOP_CTL LOOP_BYPASS=0x80

modreg I_ACL_LOOP_CTL LOOP_BYPASS=0xbe

modreg I_FWD_LOOP_CTL LOOP_BYPASS0=0xbe LOOP_BYPASS1=0xbe LOOP_BYPASS2=0xbe

modreg I_NET_LOOP_CTL LOOP_BYPASS0=0xbe LOOP_BYPASS1=0xbe

modreg I_POL_LOOP_CTL LOOP_BYPASS0=0xbe LOOP_BYPASS1=0xbe

modreg I_PR0_LOOP_CTL LOOP_BYPASS=0x80

modreg I_VT_LOOP_CTL LOOP_BYPASS0=0x3e LOOP_BYPASS1=0x3e

modreg E_DST_LOOP_CTL LOOP_BYPASS=0x0 LOOP_BYPASS1=0x0

modreg I_DST_LOOP_CTL LOOP_BYPASS0=0x0 LOOP_BYPASS1=0x0
```

4. Send and receive packet filters

//dest=2, send directly to the application layer without going through the protocol stack

```
functest pktdma mode=7 type=0x88f7 dest=2
```

5. Application layer packet sending and receiving counts are cleared

```
functest pktdma mode=2
```

6. Start receiving and forwarding received packages

//prtpkt controls whether to print the message, bit0 controls the rx direction, bit1 controls the tx direction.

//mode=4 means receiving first and then forwarding to the source port, mode=3 means only receiving.

```
functest pktdma mode=4 prtpkt=3
```

7. Check the application layer send and receive packet counts

```
functest pktdma mode=1
```

8. Stop receiving

```
functest pktdma mode=6
```

18.2.2 Protocol stack packet sending and receiving test configuration example

Configure the packets of panel port 3 to be forwarded to the CPU, and the packets with dmac of 01:02:ff:1a:11:3e , or types of 0x0800 and 0x0806 are sent to the kernel protocol stack.

Channel configuration for sending and receiving packets:

1. Reset the switch chip

```
modreg TOP_CFG_REG_RESET_GLOBAL UPI_RST_GLB_LOGIC_N=0 UPI_RST_GLB_UPI_N=0

modreg TOP_CFG_REG_RESET_GLOBAL UPI_RST_GLB_LOGIC_N=1 UPI_RST_GLB_UPI_N=1

modreg PKT_DMA_REG_PKT_DMA_AXI_RD_CFG UPI_AXI4_RLEN_MAX=0xf

modreg PKT_DMA_REG_PKT_DMA_AXI_WR_CFG UPI_AXI4_WLEN_MAX=0xf
```

2. Set the **CPU** channel: forward the packets entering **port 3** to **the CPU port (port 31)**

```
modify I_VT_PORT_SRM 3 1 FWD_VLD=1 OUT_LPORT=31

modify I_NET_PORT_SRM 3 1 STP_CHK_EN=0 BRG_EN=0
```

3. Send and receive packet filters

```
 //dest=1, send to protocol stack

functest pktdma mode=7 type=0x0800 dest=1

functest pktdma mode=7 type=0x0806 dest=1

functest pktdma mode=7 mac=01:02:ff:1a:11:3e dest=1
```

18.2.3 Application layer packet test configuration example

Package

/*Set the CPU sending channel: the packets sent from the CPU port do not pass through the PP (ppbypass=0 when sending packets, no configuration is required)*/

```
modreg E_ACL_LOOP_CTL LOOP_BYPASS=0x80

modreg E_EE_LOOP_CTL LOOP_BYPASS=0x80

modreg E_PF_LOOP_CTL LOOP_BYPASS0=0x80 LOOP_BYPASS1=0x80

modreg E_POL_LOOP_CTL LOOP_BYPASS=0x80

modreg I_ACL_LOOP_CTL LOOP_BYPASS=0xbe

modreg I_FWD_LOOP_CTL LOOP_BYPASS0=0xbe LOOP_BYPASS1=0xbe LOOP_BYPASS2=0xbe

modreg I_NET_LOOP_CTL LOOP_BYPASS0=0xbe LOOP_BYPASS1=0xbe

modreg I_POL_LOOP_CTL LOOP_BYPASS0=0xbe LOOP_BYPASS1=0xbe

modreg I_PR0_LOOP_CTL LOOP_BYPASS=0x80
modreg I_VT_LOOP_CTL LOOP_BYPASS0=0x3e LOOP_BYPASS1=0x3e

modreg E_DST_LOOP_CTL LOOP_BYPASS=0x0 LOOP_BYPASS1=0x0

modreg I_DST_LOOP_CTL LOOP_BYPASS0=0x0 LOOP_BYPASS1=0x0

/*dmac is 01:02:ff:1a:11:3e, type is 0800, dest=3, dport is 3, id=1, ppbypass is 1*/

functest pktdma mode=5 mac=01:02:ff:1a:11:3e type=0x0800 dest=3 id=1
```

19 Revision Information

Revision time	Version	describe
2021.5.8	V1.0	initial version.
2022.12.22	V1.32	Content optimization.
2022.4.25	V1.33	Content optimization.