ATHEROS®

There is Here.™

AR8327(N)/AR8328(N)

Seven-port Gigabit Ethernet Switch

Q&A

Catalog

List of Figure

1 Hardware

1.1 The difference between S17 v1.0 and v1.1

For S17 v1.0 and v1.1, some registers default value is changed. The difference is listed below.

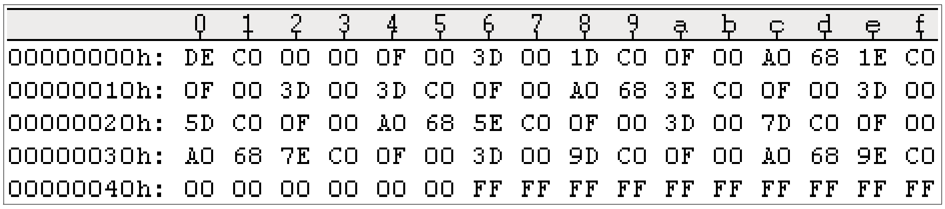
Issue	V1.0 (1L1E)	V1.1 (2L1E)
1.Can not forward on 1000M mode	Set PHY debug 0x3d=0x68a0. Please refer to FAQ1.2	Set PHY debug 0x3d=0x68e0 (default) Please remove FAQ1.2 workaround
2.100M waveform fail	Set PHY debug 0x0=0x2ea	Set PHY debug 0x0=0x2ee (default)
3.SGMII enable	Set 0xe0=c74164de	Set 0xe0=c74164d0
4.Serdes Auto-Negotiation	Not support	Set 0x10[7]=0 to enable AN(default)
5.802.3AZ EEE Control	LPI_EN=0 is disable (default)	LPI_EN=0 is enable (default)

1.2 The workaround for AR8327(N)/AR8328(N)-1L1E

For AR8327(N)/AR8328(N)-1L1E, the follow initialization must be done to the embedded PHY to make sure the switch can work correctly.

	PHY Address	REG Address	Value
1	0x0	0x1d	0x3d
	0x0	0x1e	0x68a0
2	0x1	0x1d	0x3d
	0x1	0x1e	0x68a0
3	0x2	0x1d	0x3d
	0x2	0x1e	0x68a0
4	0x3	0x1d	0x3d
	0x3	0x1e	0x68a0
5	0x4	0x1d	0x3d
	0x4	0x1e	0x68a0

The user also can use EEPROM to initial the chip.



```
             0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: DE C0 00 00 0F 00 3D 00 1D C0 0F 00 A0 68 1E C0
00000010h: 0F 00 3D 00 3D C0 0F 00 A0 68 3E C0 0F 00 3D 00
00000020h: 5D C0 0F 00 A0 68 5E C0 0F 00 3D 00 7D C0 0F 00
00000030h: A0 68 7E C0 0F 00 3D 00 9D C0 0F 00 A0 68 9E C0
00000040h: 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF
```

The user also can use CmdIssue-S17 to initial the chip.

```
or 3c c01d003d
or 3c c01e68a0

or 3c c03d003d
or 3c c03e68a0

or 3c c05d003d
or 3c c05e68a0

or 3c c07d003d
or 3c c07e68a0

or 3c c09d003d
or 3c c09e68a0
```

1.3 How to configure AR8327(N)/AR8328(N) MAC0 interface?

The following table shows the operation mode of AR8327 interface.

	Mode 1	Mode2	Mode3
P0(MAC0)	RGMII		GMII
P1~P4	Embedded GPHY		
P5(PHY4)	Embedded GPHY	RGMII Single GPHY	Embedded GPHY
P6(MAC6)	RGMII		x

The following table shows the operation mode of AR8328 interface.

	Mode 1	Mode2	Mode 3	Mode4	Mode5
P0(MAC0)	GMII/RGMII				
P1~P4	Embedded GPHY				
P5(PHY4)	Embedded GPHY	Embedded GPHY	RGMII	RGMII	GMII Single GPHY
P6(MAC6)	RGMII	GMII	RGMII	RGMII Single GPHY	

The following table shows the register setting to configure AR8327/AR8328 MAC0 as RGMII mode.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327 only
2	0x624	0x007f7f7f	
3	0x4	0x07600000	RGMII
4	0xc	0x01000000	
5	0x7c	0x7e	

When using the RGMII interface, user needs to configure the timing delay properly in order to fit the timing requirement on the receive side.

The MAC0 timing control is in the Port0 PAD Mode Control Register (offset 0x0004):

1. Bit 24: enable the timing delay for the output path of AR8327/AR8328. Set 1 to add 2ns delay in 1000 mode. (Need to enable 0xc[24] to take effect)
2. Bit [21:20]: select the delay time for the output path in 10/100 mode.
3. Bit 25: enable the timing delay for the input path of AR8327/AR8328 in 1000 mode.

4. Bit [23:22]: select the delay time for the input path in 1000 mode.

   00: 0.2ns

   01: 1.2ns

   10: 2.1ns

   11: 3.1ns

The following table shows the register setting to configure AR8327/AR8328 MAC0 as GMII PHY mode or GMII MAC mode.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327 only
2	0x624	0x007f7f7f	
3	0x4	0x00004000	GMII PHY mode
		0x00000040	GMII MAC mode
4	0x7c	0x7e	

AR8327/AR8328 provides the option registers to invert clock in GMII PHY mode.

	Address Offset/Bit	Description
1	0x4 bit[13]	Invert clock input
2	0x4 bit[12]	Invert clock output

AR8327/AR8328 provides the option registers to invert clock in GMII MAC mode.

	Address Offset/Bit	Description
1	0x4 bit[5]	Invert clock output
2	0x4 bit[4]	Invert clock input

The following table shows the register setting to configure AR8327/AR8328 MAC0 as MII PHY mode or MII MAC mode.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327 only
2	0x624	0x007f7f7f	
3	0x4	0x00000500	MII PHY mode
		0x00000004	MII MAC mode
4	0x7c	0x7d	

AR8327/AR8328 provides the option registers to invert clock in MII PHY mode.

	Address Offset/Bit	Description
1	0x4 bit[11]	Clock Edge for rxpipe
2	0x4 bit[9]	Invert clock output
3	0x4 bit[8]	Invert clock output

AR8327/AR8328 provides the option registers to invert clock in MII MAC mode.

	Address Offset/Bit	Description
1	0x4 bit[1]	Invert clock input
2	0x4 bit[0]	Invert clock input

1.4 How to configure AR8327(N)/AR8328(N) MAC6 interface?

The following table shows the operation mode of AR8327 interface.

	Mode 1	Mode2	Mode3
P0(MAC0)	RGMII		GMII
P1~P4	Embedded GPHY		
P5(PHY4)	Embedded GPHY	RGMII Single GPHY	Embedded GPHY
P6(MAC6)	RGMII		x

The following table shows the operation mode of AR8328 interface.

	Mode 1	Mode2	Mode 3	Mode4	Mode5
P0(MAC0)	GMII/RGMII				
P1~P4	Embedded GPHY				
P5(PHY4)	Embedded GPHY	Embedded GPHY	RGMII	RGMII	GMII Single GPHY
P6(MAC6)	RGMII	GMII	RGMII	RGMII Single GPHY	

The following table shows the register setting to configure AR8327/AR8328 MAC6 as RGMII mode.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327 only
2	0xc	0x07600000	RGMII
3	0x94	0x7e	

When using the RGMII interface, user needs to configure the timing delay properly in order to fit the timing requirement on the receive side.

The MAC6 timing control is in the Port6 PAD Mode Control Register (offset 0x000c):

1. Bit 24: enable the timing delay for the output path of AR8327/AR8328. Set 1 to add 2ns delay in 1000 mode.
2. Bit [21:20]: select the delay time for the output path in 10/100 mode.
3. Bit 25: enable the timing delay for the input path of AR8327/AR8328 in 1000 mode.
4. Bit [23:22]: select the delay time for the input path in 1000 mode.

   00: 0.2ns

   01: 1.2ns

   10: 2.1ns

   11: 3.1ns

The PHY4 timing control is in the debug register of PHY4:

1. Debug register offset 0x00 bit 15: RXCLK(output) delay control. Set 1 to add 2ns delay.
2. Debug register offset 0x05 bit 8: GTXCLK(input) delay control Set 1 to enable GTXCLK delay.
3. Debug register offset 0x0b bit [6:5]: GTXCLK delay time selection.

   00: 0.2ns

   01: 1.2ns

   10: 2.1ns

   11: 3.0ns

The following table shows the register setting to configure AR8327/AR8328 PHY4 as a single PHY.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327 only
2	0xc	0x00040000	MII mode
		0x00020000	RGMII mode
		0x00010000	GMII mode AR8328 only
3	0x90	0x0	
4	0x94	0x0	

The following setting is for PHY4 RGMII mode.

Item	PHY address	REG address	Value	Description
1	0x4	0x1d	0x12	Debug register offset 0x12[3]=1
		0x1e	0x4c0c	
2	0x4	0x1d	0x0	Debug register offset 0x0[15]=1
		0x1e	0x82ee	
3	0x4	0x1d	0x5	Debug register offset 0x5[8]=1
		0x1e	0x3d46	
4	0x4	0x1d	0xb	Debug register offset 0xb[6:5]=01
		0x1e	0xbc20	

The following table shows the register setting to configure AR8328 MAC6 as GMII PHY mode or GMII MAC mode.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327 only
2	0xc	0x00004000	GMII PHY mode
		0x00000040	GMII MAC mode
3	0x94	0x7e	

AR8328 provides the option registers to invert clock in GMII PHY mode.

	Address Offset/Bit	Description
1	0xc bit[13]	Invert clock input
2	0xc bit[12]	Invert clock output

AR8328 provides the option registers to invert clock in GMII MAC mode.

	Address Offset/Bit	Description
1	0xc bit[5]	Invert clock output
2	0xc bit[4]	Invert clock input

The following table shows the register setting to configure AR8327/AR8328 MAC6 as MII PHY mode or MII MAC mode.

	Address Offset	Value	Note
1	0x10	0x40000000	For AR8327
2	0xc	0x00000500	MII PHY mode
		0x00000004	MII MAC mode
3	0x94	0x7d	

AR8327/AR8328 provides the option registers to invert clock in MII PHY mode.

	Address Offset/Bit	Description
1	0xc bit[11]	Clock Edge for rxpipe
2	0xc bit[9]	Invert clock output
3	0xc bit[8]	Invert clock output

AR8327/AR8328 provides the option registers to invert clock in MII MAC mode.

	Address Offset/Bit	Description
1	0xc bit[1]	Invert clock input
2	0xc bit[0]	Invert clock input

1.5 How to configure AR8328(N) MAC5 interface?

The following table shows the operation mode of AR8328 interface.

	Mode 1	Mode2	Mode 3	Mode4	Mode5
P0(MAC0)	GMII/RGMII				
P1~P4	Embedded GPHY				
P5(PHY4)	Embedded GPHY	Embedded GPHY	RGMII	RGMII	GMII Single GPHY
P6(MAC6)	RGMII	GMII	RGMII	RGMII Single GPHY	

The following table shows the register setting to configure AR8328 MAC5 as RGMII mode.

	Address Offset	Value	Note
1	0x8	0x07600000	RGMII
2	0xc	0x01000000	
3	0x90	0x7e	

When using the RGMII interface, user needs to configure the timing delay properly in order to fit the timing requirement on the receive side.

The MAC5 timing control is in the Port5 PAD Mode Control Register (offset 0x0008):

1. Bit 24: enable the timing delay for the output path of AR8328. Set 1 to add 2ns delay in 1000 mode. (Need to enable 0xc[24] to take effect)
2. Bit [21:20]: select the delay time for the output path in 10/100 mode.
3. Bit 25: enable the timing delay for the input path of AR8328 in 1000 mode.
4. Bit [23:22]: select the delay time for the input path in 1000 mode.

   00: 0.2ns

   01: 1.2ns

   10: 2.1ns

   11: 3.1ns

The following table shows the register setting to configure AR8328 MAC5 as MII PHY mode or MII MAC mode.

	Address Offset	Value	Note
1	0x8	0x00000400	MII PHY mode
		0x00000004	MII MAC mode
2	0x90	0x7d	

AR8328 provides the option registers to invert clock in MII PHY mode.

	Address Offset/Bit	Description
1	0x8 bit[9]	Invert clock output
2	0x8 bit[8]	Invert clock output

AR8328 provides the option registers to invert clock in MII MAC mode.

	Address Offset/Bit	Description
1	0x8 bit[1]	Invert clock input
2	0x8 bit[0]	Invert clock input

1.6 How does MAC0 (or MAC6) work at SerDes/SGMII?

AR8327/AR8328 supports 1.25G SerDes/SGMII interface. The following table shows the register setting to configure either MAC0 or MAC6 as SerDes/SGMII mode.

	Address Offset	Value	Note
1	0x4	0x00080080	For MAC0
2	0xc 0x4	0x00000080 0x00080000	For MAC6
3	0xe0	0xc70167de	Serdes mode (v1.0)
	0xe0 0x10	0xc70167d0 0x20261320	Serdes mode AN enable (v1.1)
	0xe0 0x10	0xc70167d0 0x202613a0	Serdes mode AN disable (v1.1)
4	0xe0	0xc74164de	SGMII mode (v1.0)
	0xe0	0xc74164d0	SGMII mode (v1.1)

1.7 Packet buffer memory

AR8327/AR8328 has 128Kbytes packet buffer memory. The packet buffer is divided into 512 blocks and each block is 256 bytes.

1.8 Maximum Packet size

AR8327/AR8328 supports Jumbo Packet and the packet size is configured by the register 0x78 bit[13:0] MAX_FRAME_SIZE. The default maximum packet size is 1518bytes for untagged packet, 1522 bytes for tagged packet.

2 MDC/MDIO access

2.1 How does CPU access AR8327/AR8328 internal register by MDC/MDIO?

The Fig 2-1: MDC/MDIO access format shows the detail format and procedure to access the internal register by MDC/MDIO. Basically, there are two steps to access the register.

- Write the high address, this step can be omitted if the high address is unchanged.
- Read or Write the Register Data, It is allowed to access 16-bit data once instead of twice if the upper or lower 16-bit data is unchanged. Where the OP code is 01_B for write operation and 10_B for read operation.

The following describe the detail format used while accessing the registers:

- The first row is the standard format of MDIO access format.
- The second row is the register address of the register, A[18:0]
- The third row is the format to write the high address.
- The fourth and fifth rows are the format to read or write data of the register.
- The sixth row shows how to construct the 32 bits data from two 16 bits data.

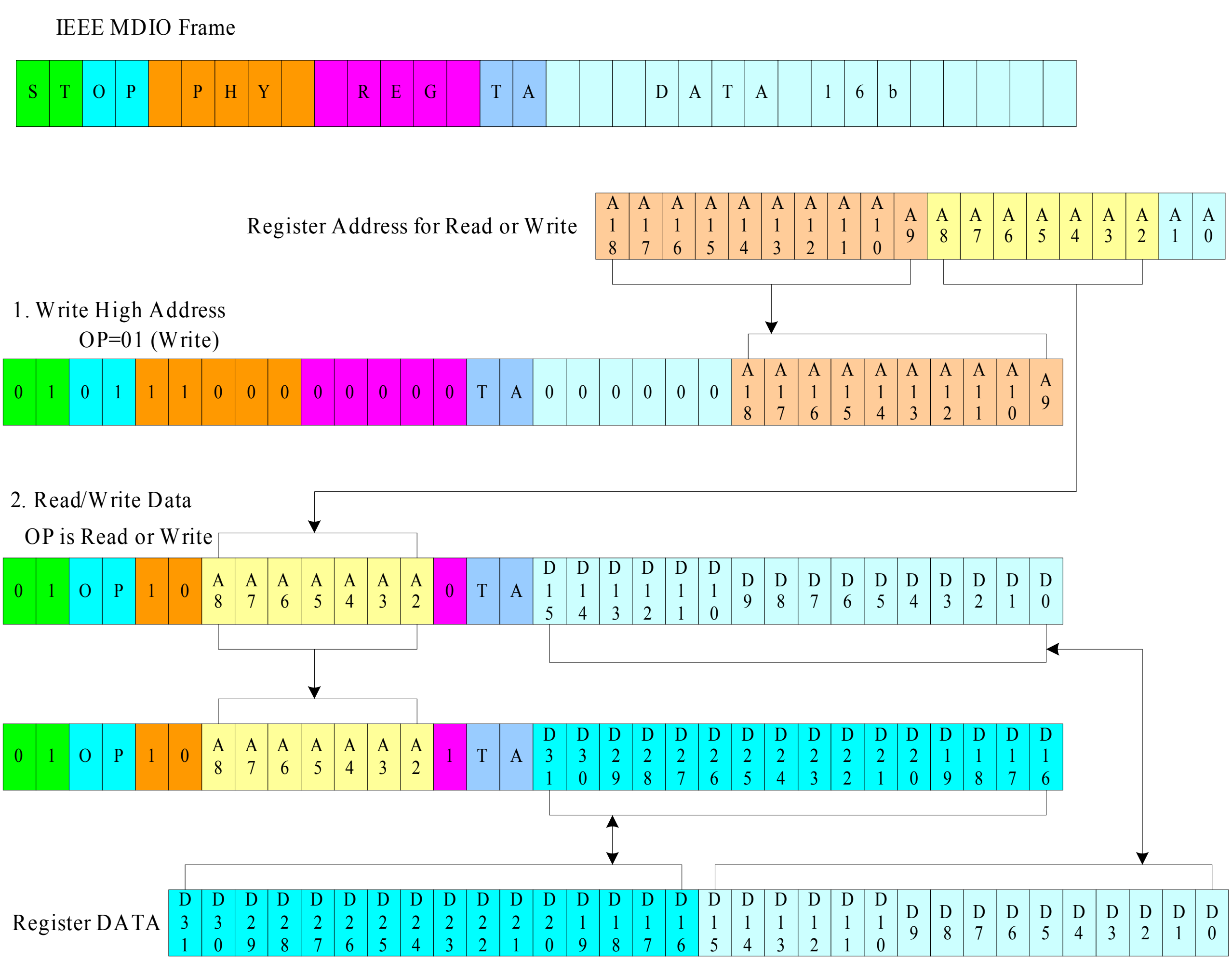

IEEE MDIO Frame
S T O P P H Y R E G T A D A T A 1 6 b
Register Address for Read or Write
A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
1. Write High Address
OP=01 (Write)
0 1 0 1 1 1 0 0 0 0 0 0 0 0 T A 0 0 0 0 0 0 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9
2. Read/Write Data
OP is Read or Write
0 1 O P 1 0 A8 A7 A6 A5 A4 A3 A2 0 T A D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
0 1 O P 1 0 A8 A7 A6 A5 A4 A3 A2 1 T A D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16
Register DATA
D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16 D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0

Fig 2-1: MDC/MDIO access format

2.2 How does CPU access AR8327/AR8328 internal PHY register by MDC/MDIO?

The Fig 2-2: Internal PHY Register access shows the format to access the AR8327/AR8328 internal PHY register through MDC/MDIO. The PHY_Addr[2:0] indicated which PHY port to be accessed, when 0 indicated PHY 0 and 1 indicated PHY 1, etc. The Op code is 10_B for read command and 01_B for write command.

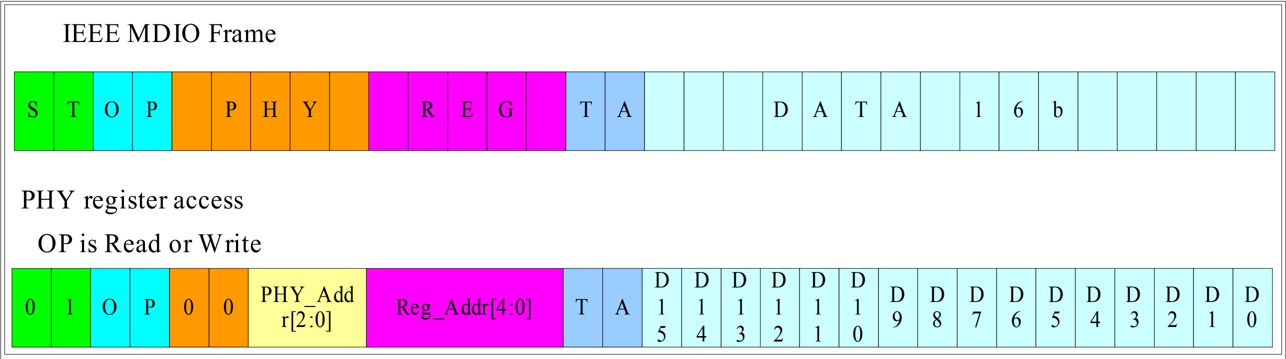


Fig 2-2: Internal PHY Register access

2.3 Can the MDC/MDIO of AR8327/AR8328 run to more than 10MHz?

The MDC of AR8327/8328 can support up to 10MHz clock input.

2.4 How to control the external PHY and AR8327/AR8328 by the same MDC/MDIO?

For 5-bit PHY address of MDC/MDIO command, address 0x0~0x4 is used by AR8327/AR8328 internal PHY and address 0x10~0x18 is also used by AR8327/AR8328 in-direct switch register access. So the address of the external PHY can only use 0x5~0xf.

2.5 How to turn off isolate mode of internal PHY when power-down?

AR8327/AR8328 internal PHY supports isolate mode to control the output pins (mdio, crs, col, txclk, rxclk), when PHY Reg.0x0[11] Power Down bit is set. The follow procedure can disable the isolate mode to allow PHY register read when power-down.

	PHY Address	REG Address	Value
1	0x0~0x4	0x1d	0x3
	0x0~0x4	0x1e	0x3eff

2.6 How to set the internal PHY to different speed/duplex mode?

IEEE 802.3 standard mode

PHY mode	PHY Reg. 0x0	PHY Reg. 0x4	PHY Reg. 0x9
Auto	0x3100	0x1de1	0x0200
Auto 1000F	0x3100	0x1c01	0x0200
Force 100F	Reg.0x9=0x0 → Reg.0x0=0x3300 → Reg.0x0=0x2100		
Force 100H	Reg.0x9=0x0 → Reg.0x0=0x3300 → Reg.0x0=0x2000		
Force 10F	0x0100	-	-
Force 10H	0x0000	-	-

non-IEEE 802.3 standard mode

PHY mode	PHY Reg. 0x0	PHY Reg. 0x4	PHY Reg. 0x9
Auto	0x3100	0x1de1	0x0200
Auto 1000F	0x3100	0x1c01	0x0200
Auto 100F	0x3100	0x1d81	0x0000
Auto 100H	0x3100	0x1c81	0x0000
Auto 10F	0x3100	0x1c61	0x0000
Auto 10H	0x3100	0x1c21	0x0000

3 Interrupt

3.1 How does CPU detect AR8327/AR8328 PHY status change by Interrupt pin?

AR8327/AR8328 supports PHY interrupt function and CPU can configure register to enable PHY_INT to detect PHY status change.

To enable PHY_INT,.

Step	Description
1	Read PHY Reg.0x13 to clear PHY interrupt status
2	Set PHY Reg.0x12 to enable PHY interrupt
3	Set SWITCH Reg.0x2c to enable SWITCH interrupt

To detect PHY status change,.

Step	Description
1	Check INTERRUPT pin until the state is changed to LOW
2	If the state is LOW, read SWITCH Reg.0x24[15] to check whether PHY interrupt is happened
3	If the PHY_INT=1, read PHY Reg.0x13 to clear PHY interrupt status
4	Set SWITCH Reg.0x24=0x8000 to clear SWITCH interrupt via Fig 14-7: Write AR8327/AR8328 register by Atheros header

4 EEPROM 93LC66 Format

4.1 What is the 93LC46/66 data format?

The hardware strapping (SPI_EN & SPI_SIZE) must be set and is latched at power on.

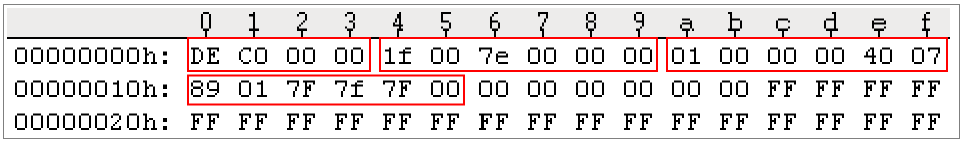
For example, use EEPROM to enable AR8327/AR8328 MAC0 RGMII interface.

	Register Address Offset	Value
1	0x7c	0x0000007e
2	0x4	0x07400000
3	0x624	0x007f7f7f

Transfer register address offset bit[17:2] to EEPROM format.

	EEPROM format	Value
1	0x1f (0x7c >> 2)	0x0000007e
2	0x1 (0x4 >> 2)	0x07400000
3	0x189 (0x624 >> 2)	0x007f7f7f

Use UltraEdit to edit 93LC66 EEPROM file.



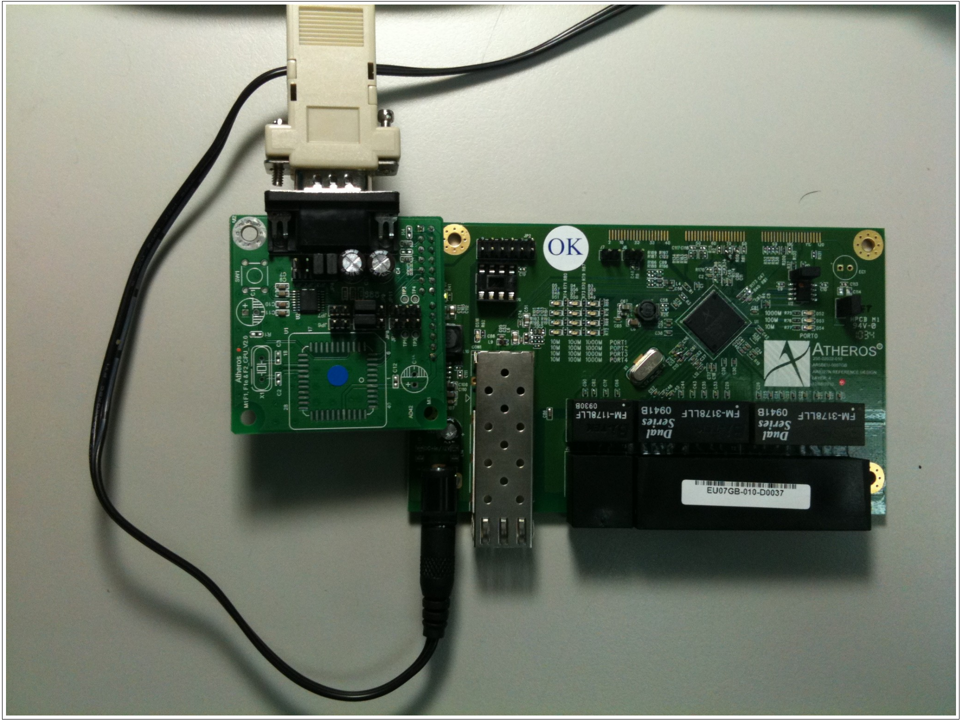
```
             0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: DE C0 00 00 1f 00 7e 00 00 00 01 00 00 00 40 07
00000010h: 89 01 7F 7f 7F 00 00 00 00 00 00 00 FF FF FF FF
00000020h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Note: The last register should be at address 0, and the LOAD_EEPROM bit is written to 1'b0 to stop loading EEPROM.

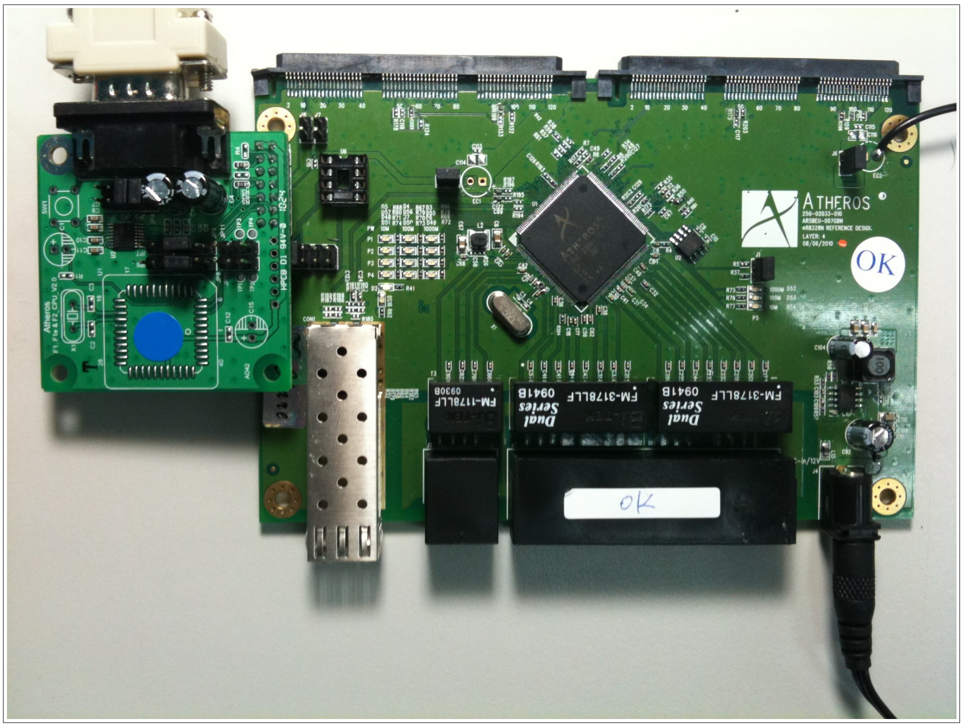
5 Green Ethernet

5.1 How to enable 802.3az (Energy Efficient Ethernet) on S17 v1.0?

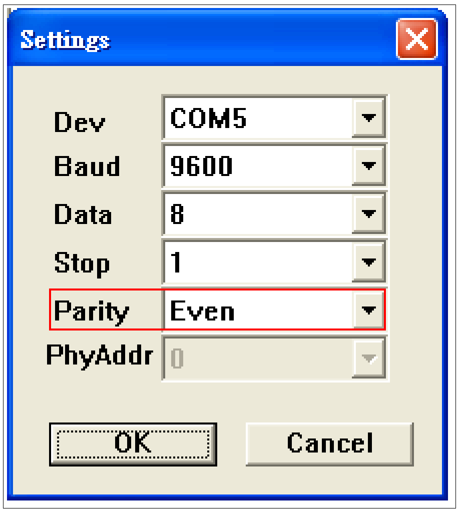
Please setup AR8327 evaluation board as the following diagram.



Please setup AR8327 evaluation board as the following diagram.

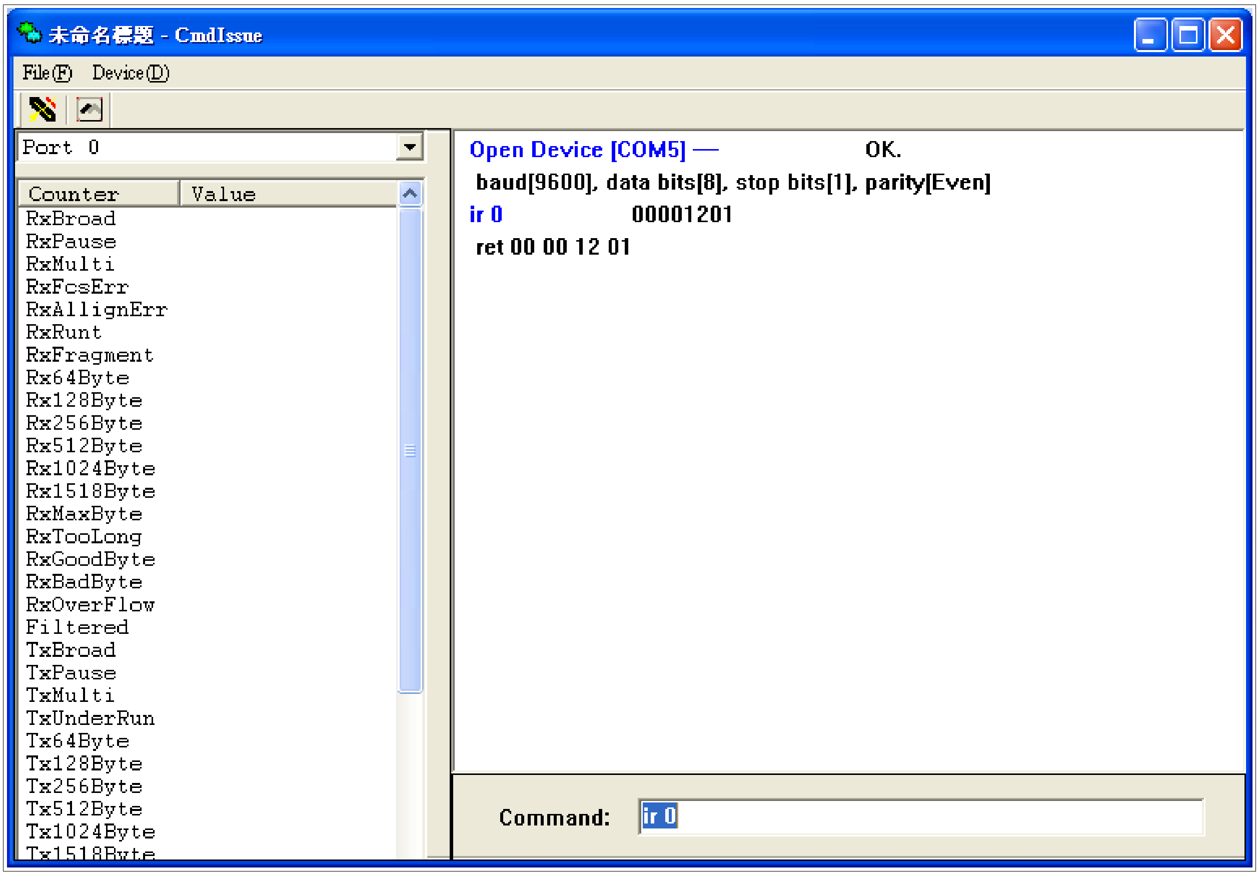


Execute CmdIssue for S17 and configure the COM port setting.



Dev COM5
Baud 9600
Data 8
Stop 1
Parity Even
PhyAddr 0

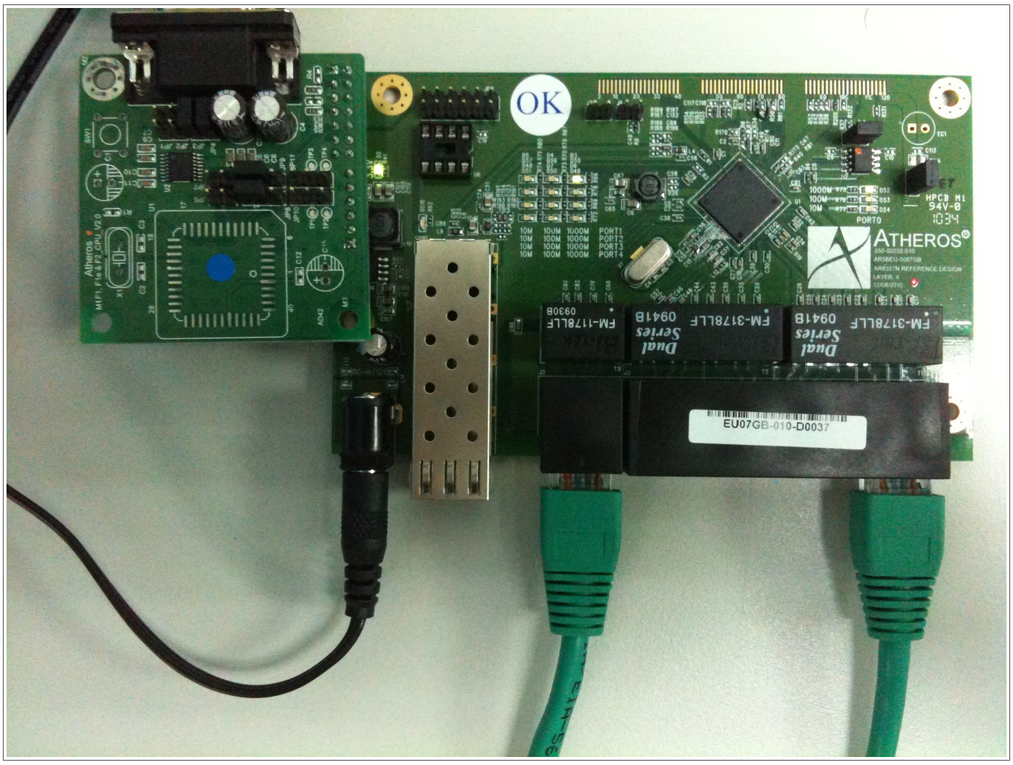
Once the COM port is opened successfully, use “ir 0” command to check DEVICE_ID and REV_ID.



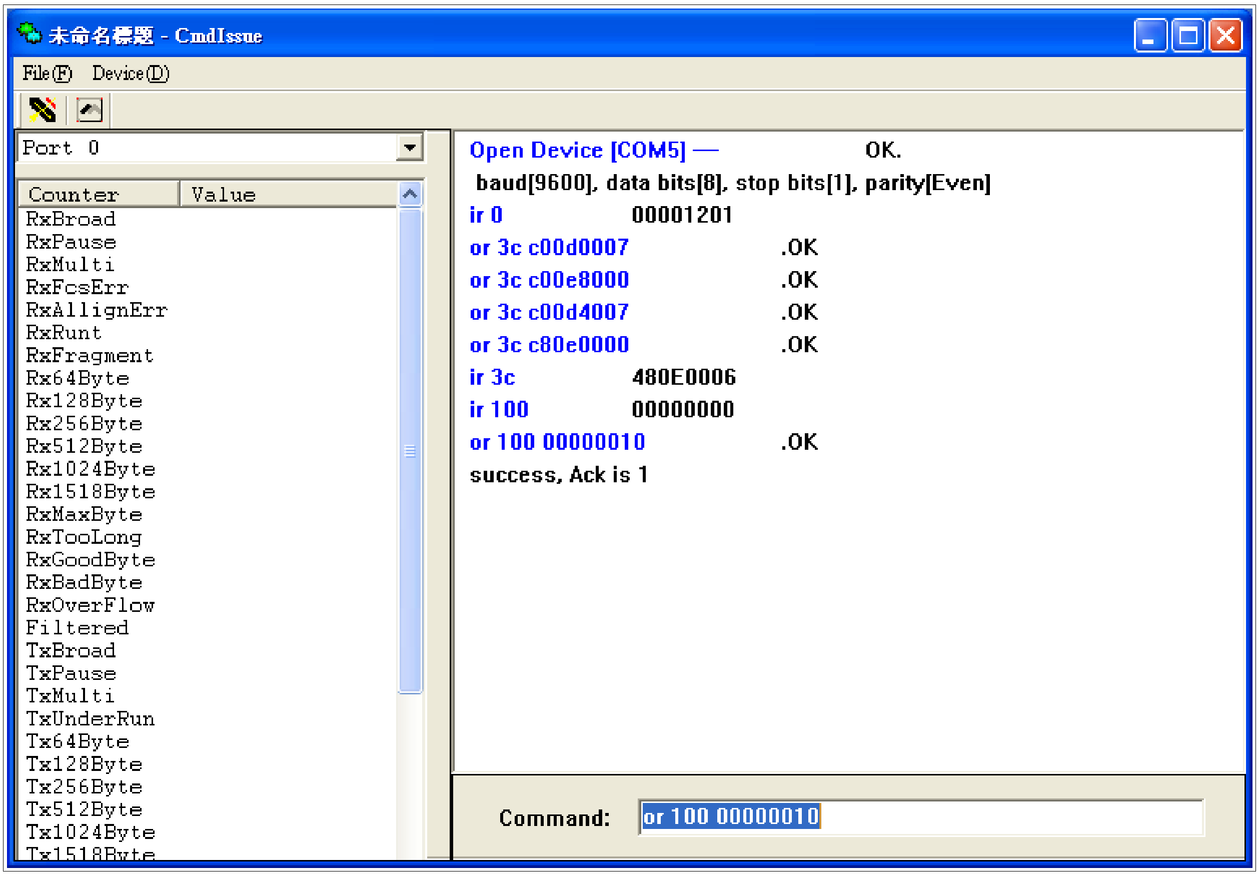
```
Open Device [COM5] —                 OK.
 baud[9600], data bits[8], stop bits[1], parity[Even]
ir 0                 00001201
 ret 00 00 12 01
```

Command: ir 0

You can use the follow method to check whether the DUT and the link partner support 802.3az operation or not. Use one Ethernet cable to connect Port1/PHY0 and Port5/PHY4 directly.



Read **EEE ability auto-negotiation result** (Device address=7, Address offset=0x8000)



```
Open Device [COM5] —              OK.
 baud[9600], data bits[8], stop bits[1], parity[Even]
ir 0              00001201
or 3c c00d0007              .OK
or 3c c00e8000              .OK
or 3c c00d4007              .OK
or 3c c80e0000              .OK
ir 3c             480E0006
ir 100            00000000
or 100 00000010             .OK
success, Ack is 1
```

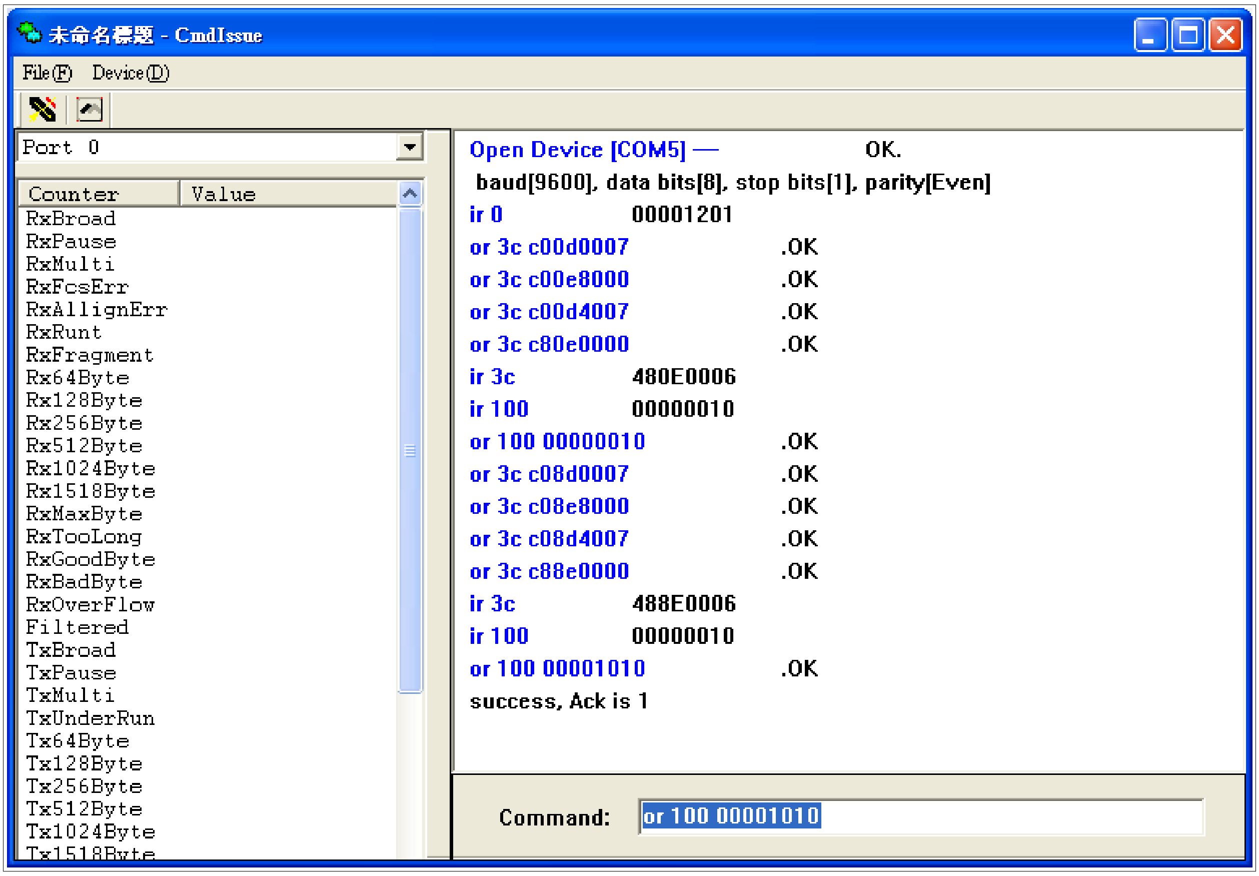
Command: or 100 00000010

	PHY Address	REG Address	Value
1	0x0	0xd	0x7
	0x0	0xe	0x8000
2	0x0	0xd	0x4007
	0x0	0xe	Read value

If the bit[2:1] of the read value is 01,10 or 11, it means the link partner supports EEE operation for 1000BT (bit2) or 100BT (bit1). And then you can set LPI_EN at switch register 0x100. Follow the same procedure for Port1/PHY0 and Port5/PHY4.

	Register Address Offset	Value
1	0x100	0x00000010

Note: 0x100[12]=LPI_EN_5, 0x100[10]=LPI_EN_4, 0x100[8]=LPI_EN_3, 0x100[6]=LPI_EN_2, 0x100[4]=LPI_EN_1



```
Counter    Value
RxBroad
RxPause
RxMulti
RxFcsErr
RxAllignErr
RxRunt
RxFragment
Rx64Byte
Rx128Byte
Rx256Byte
Rx512Byte
Rx1024Byte
Rx1518Byte
RxMaxByte
RxTooLong
RxGoodByte
RxBadByte
RxOverFlow
Filtered
TxBroad
TxPause
TxMulti
TxUnderRun
Tx64Byte
Tx128Byte
Tx256Byte
Tx512Byte
Tx1024Byte
Tx1518Byte
```

```
Open Device [COM5] —              OK.
 baud[9600], data bits[8], stop bits[1], parity[Even]
ir 0               00001201
or 3c c00d0007                 .OK
or 3c c00e8000                 .OK
or 3c c00d4007                 .OK
or 3c c80e0000                 .OK
ir 3c              480E0006
ir 100             00000010
or 100 00000010                .OK
or 3c c08d0007                 .OK
or 3c c08e8000                 .OK
or 3c c08d4007                 .OK
or 3c c88e0000                 .OK
ir 3c              488E0006
ir 100             00000010
or 100 00001010                .OK
success, Ack is 1
```

Command: or 100 00001010

Note: For AR8327/AR8328-1L1E, you will find the LEDs are blinking after you configure enable Port1/PHY0 and Port5/PHY4 register to enable 802.3az.

5.2 How to disable 802.3az (Energy Efficient Ethernet) on S17 v1.1?

You can use the follow method to disable PHY 802.3az operation.

	PHY Address	REG Address	Value
1	0x0	0xd	0x7
	0x0	0xe	0x3c
2	0x0	0xd	0x4007
	0x0	0xe	0x0

5.3 Power Saving Mechanism

Except IEEE 802.az, AR8327/AR8328 also supports various advanced power saving mechanism to reduce the operating power consumption.

- No Link Sleep Mode

  When there is no link partner, the PHY will enter its sleep mode and turn off most parts of the PHY. Only the carrier detection is alive ant the PHY will send out a link pulse in one fixed period to wake-up a link partner.

	3.3V	1.1V	Total Power
1000M (All Active)	284mA	800mA	1817.2mW
100M (All Active)	106mA	216mA	567.4mW
10M (All Active)	126mA	92mA	517mW
Link Down	27mA	58mA	152.9mW

- 1000Base-T Short Cable Mode

  The PHY will turn off some circuits in the receive path depending and also reduce amplitude in the transmit path upon the cable length when operated in 1000Base-T mode. The cable length detection will classify the length with under 20M, less than 100M, and more than 100M.

	3.3V	1.1V	Total Power
1000M (100 meter)	284mA	800mA	1817.2mW
1000M (1 meter)	301mA	584mA	1635.7mW

- 10Base-Te Idle Mode

  For the 10Base-T mode, when there is no data to transmit, the PHY will also turn off the transmitter and reduce its power consumption.

	3.3V	1.1V	Total Power
10M (All Active)	126mA	92mA	517mW
10M (All Linkup)	35mA	74mA	196.9mW

5.4 WOL (Wake-on-LAN) Interrupt

The computer Wake-on-LAN is implemented using a special network message called a magic packet. The magic packet is a broadcast frame followed by sixteen repetitions of the target computer's 48-bit MAC address.

A standard magic packet has the following basic limitations:

- Require destination computer MAC address
- Does not provide a delivery confirmation
- May not work outside of the local network

AR8327/AR8328 supports Wake-on-LAN Interrupt by ACL. User can configure ACL to classify MAC pattern, IPv4 pattern, IPv6 pattern to general interrupt to wake up network processor from low-power suspend mode. For detailed configuration, please refer to 16.5ACL interrupt.

For example, The uPNP Low Power Architecture uses SSDP (Simple Service Discovery Protocol) as the basis of the discovery protocol. The uPNP Low Power Architecture defines a set of new SSDK headers that allow the devices to communicate the power state changes and the sleep period.

In this example, we use AR8327/AR8328 ACL to detect SSDP packet to generate WOL interrupt to network processor, and also control SSDP packet to forward or drop.

	Address Offset	Value	Description
1	0x30	0x80000302	Bit 2 ACL_EN=1
ACL Rule --- IPv4 Pattern (Please refer to Table2-9)			
2	0x404	0xeffffffa	DIP=239.255.255.250
	0x408	0x00000000	
	0x40c	0x00000000	
	0x410	0x00000000	
	0x414	0x0000007f	Byte 16 [6:0] Source Port=all port
	0x400	0x80000001	
ACL Mask --- IPv4 Mask (Please refer to Table2-10)			
3	0x404	0xffffffff	DIP mask, DIP=239.255.255.250
	0x408	0x00000000	
	0x40c	0x00000000	
	0x410	0x00030000	
	0x414	0x000000c2	RULE VALID=2'b11, RULE TYPE=2
	0x400	0x80000101	

	Address Offset	Value	Description
ACL Action (Please refer to Table2-6)			
4	0x404	0x00000000	
	0x408	0x40000000	Bit[67:61] DES_PORT=P1
	0x40c	0x00010010	Bit[68] DES_PORT_OVER_EN Bit[80] ACL_MATCH_INT_EN
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000201	

5.5 The workaround for Marvell 88E1543 and 88E1545 Interoperability

For AR8327(N)/AR8328(N)-2L1E, the following software workaround can solve the interoperability when connecting to Marvel 88E1543/88E1545.

1. Initiation state:

	Register Address Offset	Value
1	0x100	0x00001550
For MAC1		
2	0x80	0x7e
3	0x9b4	0x4a → 0x0 → 0x4a → 0x3a004a
4	0x80	0x1280 (if PHY0 is link down) 0x1200 (if PHY0 is link up)

0x80/0x9b4 is for MAC1, 0x84/0x9b8 is for MAC2, 0x88/0x9bc is for MAC3, 0x8c/0x9c0 is for MAC4, 0x90/0x9c4 is for MAC5

2. PHY state is changed from link down to link up:

	PHY Address	REG Address	Value
1	0x0	0x11[10]	1:Link is up
2	0x0	0xd	0x7
	0x0	0xe	0x8000
3	0x0	0xd	0x4007
	0x0	0xe	Read value
If the read value is 0x2 or 0x6, enable MAC's LPI=0			

	PHY Address	REG Address	Value
	Register Address Offset	Value	
4	0x100	0x00001540	MAC1's LPI at 0x100[4]

3. PHY state is changed from link up to link down:

	PHY Address	REG Address	Value
1	0x0	0x11[10]	0:Link is down
	Register Address Offset	Value	
2	0x100	0x00001550	MAC1's LPI at 0x100[4]

For MAC1		
	Register Address Offset	Value
1	0x80	0x7e
2	0x9b4	0x4a → 0x0 → 0x4a → 0x3a004a
3	0x80	0x1280 (if PHY0 is link down)

6 Flow control function

6.1 How does user set MAC flow control function?

Please set the bit[4:5] and bit12 of register 0x7c/0x80/0x84/0x88/0x8c/0x90/0x94 for MAC flow control enable/disable. Therefore the bit[4:5] must be set to 11_B and the bit[12] must be set to 0_B.

1. Bit[4] : the ability to transmit pause frame to the link partner
   a. 0_B : disable TXMAC flow control for the MAC0.
   b. 1_B : enable TXMAC flow control for the MAC0.
2. Bit[5] : the ability to receive pause frame and stop transmitting
   a. 0_B : disable RXMAC flow control for the MAC0.
   b. 1_B : enable RXMAC flow control for the MAC0.
3. Bit[12] :
   a. 0_B : flow control ability is configured by bit[5:4]
   b. 1_B : flow control ability is configured by PHY Auto-negotiation

6.2 How does user set Twisted-Pair port flow control function?

Please set the bit12 of register 0x7c/0x80/0x84/0x88/0x8c to 1_B and configure PHY MII register 0x4 (Auto-negotiation Advertisement Register) bit[11:10] to $11_{B.}$

1. Bit[11] : the ability to transmit pause frame to the link partner
   a. 0_B : No asymmetric Pause.
   b. 1_B : Asymmetric Pause.
2. Bit[10] :
   c. 0_B : MAC Pause is not implemented.
   d. 1_B : MAC Pause is implemented.

7 Address Table Operation

7.1 How many entries do AR8327/AR8328 support?

AR8327/AR8328 has 2K address table which is separated to one 1K entries table and two 512 entries tables to improve hash violation. If hash violation is happened in first table, the learned MAC address is stored in second table. You also can configure the bit30 of register 0x618 to decide whether the new learned mac address can replace old one or not, when hash violation is happened.

7.2 How to configure learning and aging ability?

Please configure the bit20 of register 0x660/0x66c/0x678/0x684/0x690/0x69c/0x6a8 to enable/disable per port learning ability. Please configure the bit19 of register 0x618 to enable/disable global aging-out ability of the address table. The address table age timer is configured by bit[15:0] and the default timer is about 300 seconds.

[ssdk_sh to enable/disable port learning] ssdk_sh fdb portLearn get 0

ssdk_sh fdb portLearn set 0 disable

[ssdk_sh to configure aging-out timer] ssdk_sh fdb ageTime get

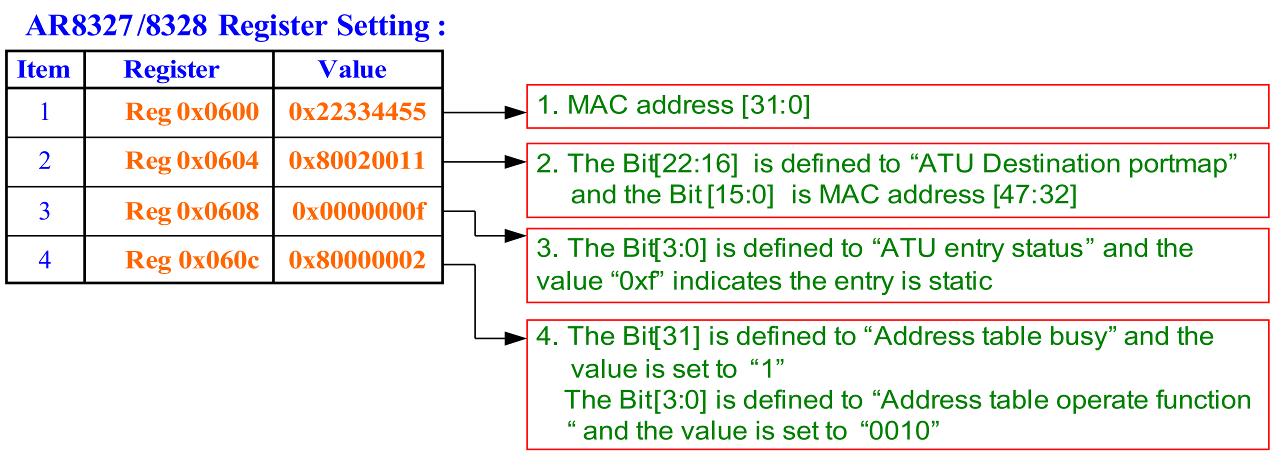
ssdk_sh fdb ageTime set 300

7.3 How to load a static MAC address entry to address table?

Please set the register 0x600, 0x604, 0x608 and 0x60c of ATU FUNCTION REGISTER. The Fig 7-1: Load a static MAC address entry to address table is an example to load an static MAC address "00 11 22 33 44 55" to address table.

1. 0x600 : this is Bit[31:0] of the MAC address.
2. 0x604 :
   a. Bit[27] : the entry is IVL or SVL learned.
   b. Bit[22:16] : the portmap is associated with this MAC address.
   c. Bit[15:0] : this is Bit[47:32] of the MAC address
3. 0x608 :
   a. Bit[19:8] : the MAC address is the member of ATU_VID group. (IVL learned)
   b. Bit[3:0] : f_H , indicates the entry is static and valid.
4. 0x60c :
   c. Bit[31] : 1_B Start an AT operation and cleared to zero after the operation is completed.
   d. Bit[3:0] : 0010_B , the CPU loads an entry into ATU table.

[ssdk_sh to add ATU entry] ssdk_sh fdb entry add 00-11-22-33-44-55 0 forward forward 1 yes no no no no no no no



AR8327/8328 Register Setting :

Item	Register	Value
1	Reg 0x0600	0x22334455
2	Reg 0x0604	0x80020011
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000002

1. MAC address [31:0]

2. The Bit[22:16] is defined to "ATU Destination portmap" and the Bit [15:0] is MAC address [47:32]

3. The Bit[3:0] is defined to "ATU entry status" and the value "0xf" indicates the entry is static

4. The Bit[31] is defined to "Address table busy" and the value is set to "1"
The Bit[3:0] is defined to "Address table operate function " and the value is set to "0010"

Fig 7-1: Load a static MAC address entry to address table

7.4 How to search all MAC addresses in the address table?

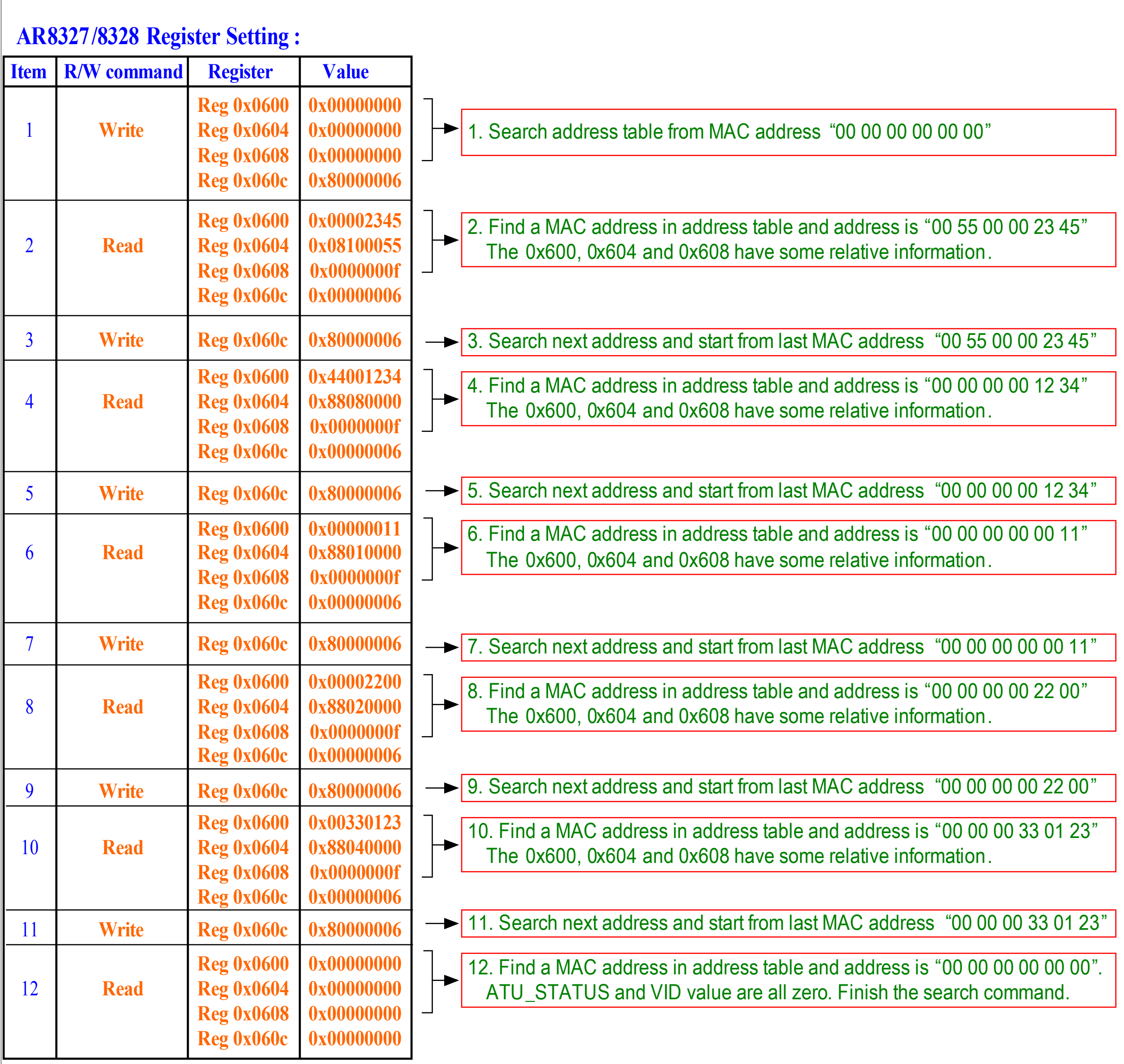
AR8327/AR8328 supports the following options to search MAC addresses in the address table.

1. Search by VID
2. Search by PORT
3. Search by Multicast address or all entries

The Fig 7-2: Search address table is an example for search all MAC addresses.

1. Fill in MAC address to address table for number of 5 and has two methods
   a. Static address: Please see Fig 7-1: Load a static MAC address entry to address table.
   b. Dynamic address : Address learning from received packet
   c. Address Table has number of 5 MAC address as follows:
      ➔00 00 00 00 00 11
      ➔00 00 00 00 22 00
      ➔00 00 00 33 01 23
      ➔00 00 44 00 12 34
      ➔00 55 00 00 23 45
2. Set MAC address = "00 00 00 00 00 00", VID = "0" and ATU_STATUS = 0000$_B$.
3. Set AT_BUSY = 1$_B$ and AT_FUNC = 0110$_B$ to search the first valid entry in the address table.
4. Read the register 0x600, 0x604 and 0x608 of ATU FUNCTION REGISTER for the next entry.
5. Repeat step3 & 4, until the returned value of MAC address, VID and ATU_STATUS are all zero.

[ssdk_sh to add ATU entry] ssdk_sh fdb entry show



AR8327/8328 Register Setting :

Item	R/W command	Register	Value	
1	Write	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x00000000 0x00000000 0x00000000 0x80000006	1. Search address table from MAC address "00 00 00 00 00 00"
2	Read	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x00002345 0x08100055 0x0000000f 0x00000006	2. Find a MAC address in address table and address is "00 55 00 00 23 45" The 0x600, 0x604 and 0x608 have some relative information.
3	Write	Reg 0x060c	0x80000006	3. Search next address and start from last MAC address "00 55 00 00 23 45"
4	Read	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x44001234 0x88080000 0x0000000f 0x00000006	4. Find a MAC address in address table and address is "00 00 00 00 12 34" The 0x600, 0x604 and 0x608 have some relative information.
5	Write	Reg 0x060c	0x80000006	5. Search next address and start from last MAC address "00 00 00 00 12 34"
6	Read	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x00000011 0x88010000 0x0000000f 0x00000006	6. Find a MAC address in address table and address is "00 00 00 00 00 11" The 0x600, 0x604 and 0x608 have some relative information.
7	Write	Reg 0x060c	0x80000006	7. Search next address and start from last MAC address "00 00 00 00 00 11"
8	Read	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x00002200 0x88020000 0x0000000f 0x00000006	8. Find a MAC address in address table and address is "00 00 00 00 22 00" The 0x600, 0x604 and 0x608 have some relative information.
9	Write	Reg 0x060c	0x80000006	9. Search next address and start from last MAC address "00 00 00 00 22 00"
10	Read	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x00330123 0x88040000 0x0000000f 0x00000006	10. Find a MAC address in address table and address is "00 00 00 33 01 23" The 0x600, 0x604 and 0x608 have some relative information.
11	Write	Reg 0x060c	0x80000006	11. Search next address and start from last MAC address "00 00 00 33 01 23"
12	Read	Reg 0x0600 Reg 0x0604 Reg 0x0608 Reg 0x060c	0x00000000 0x00000000 0x00000000 0x00000000	12. Find a MAC address in address table and address is "00 00 00 00 00 00". ATU_STATUS and VID value are all zero. Finish the search command.

Fig 7-2: Search address table

8 VLAN Table Operation

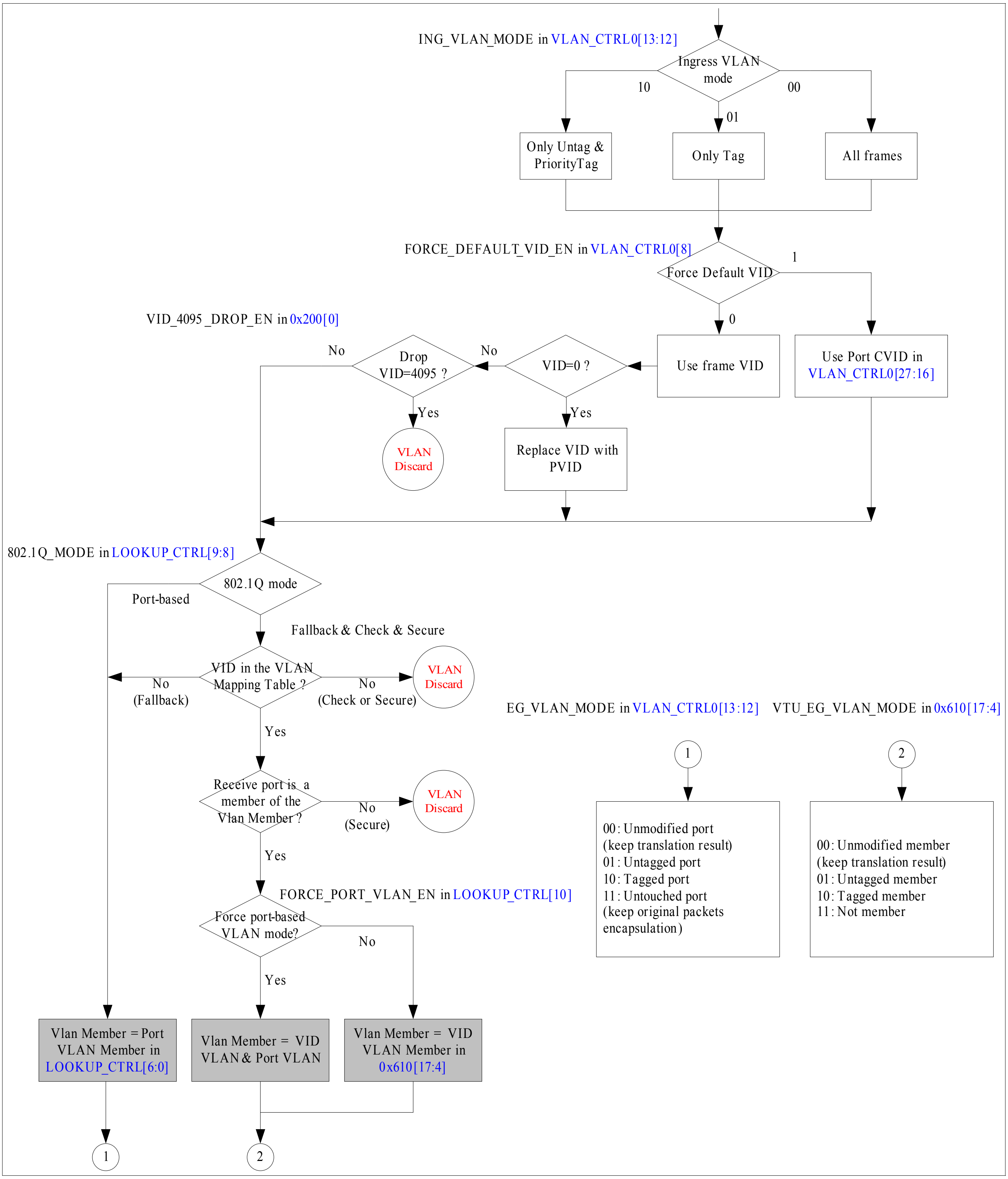

ING_VLAN_MODE in VLAN_CTRL0[13:12]
Ingress VLAN mode
10
01
00
Only Untag & PriorityTag
Only Tag
All frames
FORCE_DEFAULT_VID_EN in VLAN_CTRL0[8]
Force Default VID
1
0
Use frame VID
Use Port CVID in VLAN_CTRL0[27:16]
VID_4095_DROP_EN in 0x200[0]
No
Drop VID=4095 ?
No
VID=0 ?
Yes
VLAN Discard
Yes
Replace VID with PVID
802.1Q_MODE in LOOKUP_CTRL[9:8]
802.1Q mode
Port-based
Fallback & Check & Secure
VID in the VLAN Mapping Table ?
No (Fallback)
No (Check or Secure)
VLAN Discard
Yes
Receive port is a member of the Vlan Member ?
No (Secure)
VLAN Discard
Yes
FORCE_PORT_VLAN_EN in LOOKUP_CTRL[10]
Force port-based VLAN mode?
No
Yes
Vlan Member = Port VLAN Member in LOOKUP_CTRL[6:0]
Vlan Member = VID VLAN & Port VLAN
Vlan Member = VID VLAN Member in 0x610[17:4]
1
2
EG_VLAN_MODE in VLAN_CTRL0[13:12]
VTU_EG_VLAN_MODE in 0x610[17:4]
1
2
00: Unmodified port (keep translation result)
01: Untagged port
10: Tagged port
11: Untouched port (keep original packets encapsulation)
00: Unmodified member (keep translation result)
01: Untagged member
10: Tagged member
11: Not member

Fig 8-1: VLAN flow chart

]

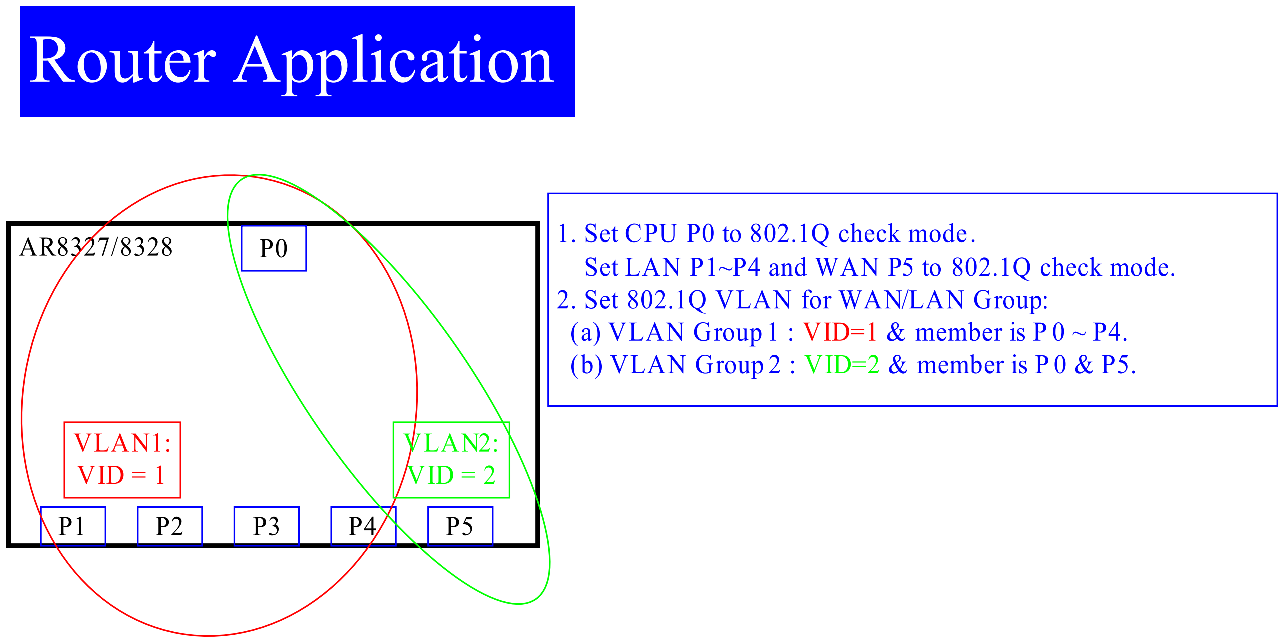
The following table is the address offset of PORT VLAN CONTROL REGISTER for each ports.

Port	PORT_VLAN_CTRL0	PORT_VLAN_CTRL1
0	0x0420	0x0424
1	0x0428	0x042c
2	0x0430	0x0434
3	0x0438	0x043c
4	0x0440	0x0444
5	0x0448	0x044c
6	0x0450	0x0454

The following table is the address offset of PORT LOOKUP CONTROL REGISTER for each ports.

Port	PORT_LOOKUP_CTRL0
0	0x0660
1	0x066c
2	0x0678
3	0x0684
4	0x0690
5	0x069c
6	0x06a8

8.1 How to configure VLAN to implement WAN/LAN application?



Router Application

1. Set CPU P0 to 802.1Q check mode.
   Set LAN P1~P4 and WAN P5 to 802.1Q check mode.
2. Set 802.1Q VLAN for WAN/LAN Group:
   (a) VLAN Group 1 : VID=1 & member is P 0 ~ P4.
   (b) VLAN Group 2 : VID=2 & member is P 0 & P5.

Fig 8-2: VLAN for Router Application

For router application, CPU is connected to AR8327/AR8328 MAC0. CPU can configure AR8327/AR8328 VLAN to implement 4*LAN and 1*WAN. The is an example for router application.

	Address Offset	Value	Description
1	0x660	0x0014027e	802.1Q check mode
	0x66c	0x0014027d	
	0x678	0x0014027b	
	0x684	0x00140277	
	0x690	0x0014026f	
	0x69c	0x0014025f	
2	0x420	0x00010001	P0~P4 Port CVID=1
	0x428	0x00010001	
	0x430	0x00010001	
	0x438	0x00010001	
	0x440	0x00010001	
3	0x448	0x00020001	P5 Port CVID=2
4	0x610	0x001bd560	VLAN(VID=1) With P0/1/2/3/4
	0x614	0x80010002	
5	0x610	0x001b7fe0	VLAN(VID=2) With P0/5
	0x614	0x80020002	

	Address Offset	Value	Description
6	0x424	0x00002040	Insert Tag to CPU port
7	0x42c	0x00001040	Remove Tag to WAN/LAN port
	0x434	0x00001040	
	0x43c	0x00001040	
	0x444	0x00001040	
	0x44c	0x00001040	

[ssdk_sh to set port0 802.1Q check mode] ssdk_sh portVlan ingress set 0 check

[ssdk_sh to set port1 802.1Q check mode] ssdk_sh portVlan ingress set 1 check

[ssdk_sh to set port2 802.1Q check mode] ssdk_sh portVlan ingress set 2 check

[ssdk_sh to set port3 802.1Q check mode] ssdk_sh portVlan ingress set 3 check

[ssdk_sh to set port4 802.1Q check mode] ssdk_sh portVlan ingress set 4 check

[ssdk_sh to set port5 802.1Q check mode] ssdk_sh portVlan ingress set 5 check

[ssdk_sh to set port5 CVID=2] ssdk_sh portVlan defaultCVid set 5 2

[ssdk_sh to create VLAN1] ssdk_sh vlan entry append 1 1 0,1,2,3,4 0 1,2,3,4 default default default

[ssdk_sh to create VLAN2] ssdk_sh vlan entry append 2 2 0,5 0 5 default default default

[ssdk_sh to set port0 tagged port] ssdk_sh portVlan egress set 0 tagged

[ssdk_sh to set port1 untagged port] ssdk_sh portVlan egress set 1 untagged

[ssdk_sh to set port2 untagged port] ssdk_sh portVlan egress set 2 untagged

[ssdk_sh to set port3 untagged port] ssdk_sh portVlan egress set 3 untagged

[ssdk_sh to set port4 untagged port] ssdk_sh portVlan egress set 4 untagged

[ssdk_sh to set port5 untagged port] ssdk_sh portVlan egress set 5 untagged

8.2 How to use core-port and s-tag mode

If the received packets of WAN/LAN are tagged, you can insert s-tag to separate WAN/LAN and keep the original c-tag of the received packets.

	Address Offset	Value	Description
1	0x424	0x00002240	Core port
	0x660	0x0014027e	Check mode
2	0x48	0x00028100	S-tag mode and header is 0x8100
3	0x420	0x00010001	P0~P4 Port SVID=1
	0x428	0x00010001	
	0x430	0x00010001	
	0x438	0x00010001	
	0x440	0x00010001	
4	0x448	0x00020002	P5 Port SVID=2

8.3 MAC clone

AR8327/AR8328 supports independent VLAN Learning (IVL) and Share VLAN Learning (SVL) architecture, so the same MAC address with different VIDs will create two independent ARL entries. If you want to bridge WAN and LAN interface, or clone LAN PC mac address to WAN interface, you must create the VLAN entry with VTU_IVL_EN=1.

8.4 VLAN tag insert/remove between S16, S27/26e and S17

AR8327/AR8328 supports VLAN tag insert/remove by port-based EG_VLAN_MODE or rule-based VTU_EG_VLAN_MODE.

EG_VLAN_MODE (port-based)	AR831x(S16)	AR822x(S27) /AR8236(S26e)	AR832x(S17)
unmodified	v	v	v
without VLAN	v	v	v
with VLAN	v	v	v
hybrid	-	v	-
untouched	-	-	v
VTU_EG_VLAN_ MODE(rule-based)	AR831x(S16)	AR822x(S27) /AR8236(S26e)	AR832x(S17)
unmodified	-	-	v
without VLAN	-	-	v
with VLAN	-	-	v

AR822x/AR8236 supports hybrid mode to transmit untagged and tagged packet to the same destination port. If VID of the egress packet is equal to the PVID of destination port, the egress packet will be untagged.

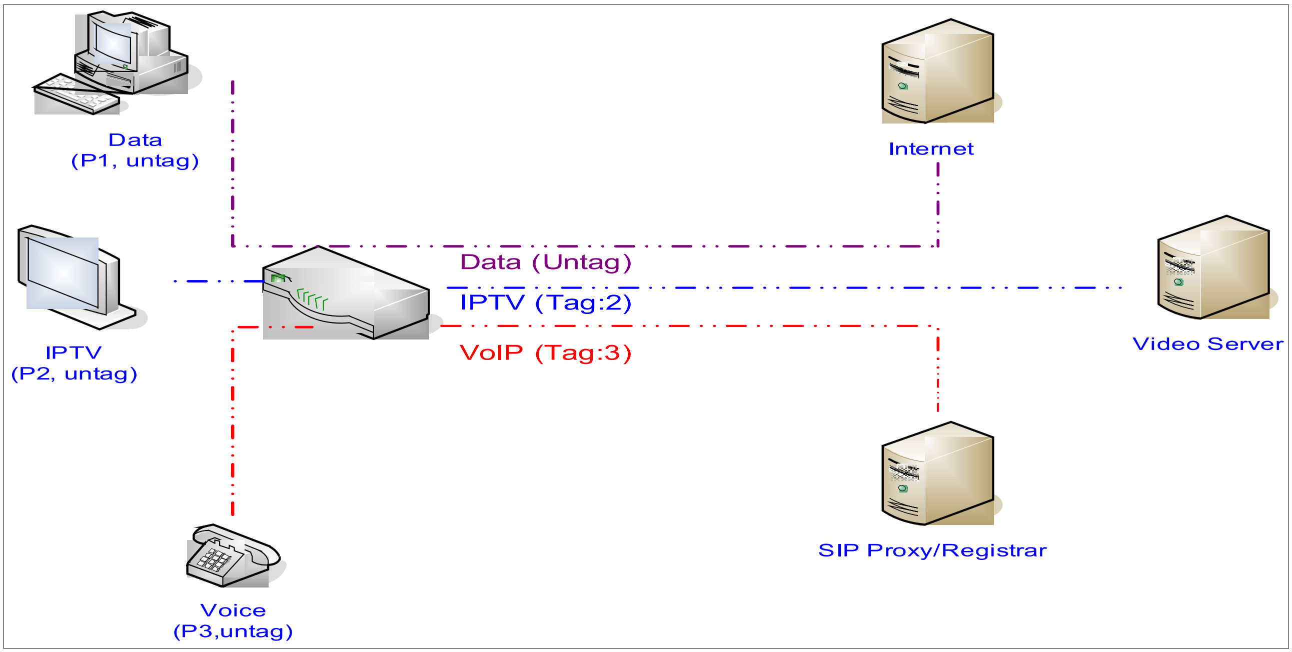


Fig 8-3: Flexible VLAN tag insert/remove (Up-link port is P0)

	Address Offset	Value	Description
1	0x660	0x0014027e	802.1Q check mode
	0x66c	0x0014027d	
	0x678	0x0014027b	
	0x684	0x00140277	
2	0x420	0x00010001	P0 Port CVID=1
	0x428	0x00010001	P1 Port CVID=1
	0x430	0x00020001	P2 Port CVID=2
	0x438	0x00030001	P3 Port CVID=3
3	0x610	0x001bff50	VLAN(VID=1) P0:untag, P1:untag
	0x614	0x80010002	
4	0x610	0x001bfde0	VLAN(VID=2) P0:tag, P2:untag
	0x614	0x80020002	
5	0x610	0x001bf7e0	VLAN(VID=3) P0:tag, P3:untag
	0x614	0x80030002	

For VTU_EG_VLAN_MODE, it is the register 0x610 bit[17:4]. Bit[5:4] is for P0, bit[7:6] is for P1 and so on.

8.5 How to search all valid VLAN entries in the VTU table?

AR8327/AR8328 supports the following options to search MAC addresses in the address table.

1. Search valid entries by VID
2. Search all valid entries

The Fig 8-4: Search VLAN table is an example for search all valid VLAN entries.

1. Set VID = 12'b0 and VT_FUNC = 101_B.
2. Set VT_BUSY = 1_B to search the first valid entry in the address table.
3. Read the register 0x610 and 0x614 of VTU FUNCTION REGISTER for the next entry.
4. Repeat step2 & 3, until the returned value of VID and VT_BUSY are all zero.

AR8327/8328 Register Setting :

Item	R/W command	Register	Value
1	Write	Reg 0x0614	0x80000005
2	Read	Reg 0x0610 Reg 0x0614	0x001bff10 0x00010005
3	Write	Reg 0x0614	0x80010005
4	Read	Reg 0x0610 Reg 0x0614	0x001bff20 0x00020005
5	Write	Reg 0x0614	0x80020005
6	Read	Reg 0x0610 Reg 0x0614	0x001bff40 0x00030005
7	Write	Reg 0x0614	0x80030005
8	Read	Reg 0x0610 Reg 0x0614	0x001bff80 0x00040005
9	Write	Reg 0x0614	0x80040005
10	Read	Reg 0x0610 Reg 0x0614	0x001bf1f0 0x00050005
11	Write	Reg 0x060c	0x80050005
12	Read	Reg 0x0610 Reg 0x0614	0x0003fff0 0x00000005

1. Search VTU table from VID = 12'b0

2. Find first valid entry in VTU table
The 0x610 and 0x614 have some relative information.

3. Search next address and start from last VTU entry "00 01 00 05"

4. Find first valid entry in VTU table
The 0x610 and 0x614 have some relative information.

5. Search next address and start from last VTU entry "00 02 00 05"

6. Find first valid entry in VTU table
The 0x610 and 0x614 have some relative information.

7. Search next address and start from last VTU entry "00 03 00 05"

8. Find first valid entry in VTU table
The 0x610 and 0x614 have some relative information.

9. Search next address and start from last VTU entry "00 04 00 05"

10. Find first valid entry in VTU table
The 0x610 and 0x614 have some relative information.

11. Search next address and start from last VTU entry "00 05 00 05"

12. Find a entry in VTU table and VID = 12'b0.
VTU_BUSY and VID value are all zero. Finish the search command.

Fig 8-4: Search VLAN table

9 Hardware IGMP/MLD Snooping

9.1 How to enable AR8327/AR8328 hardware IGMP/MLD snooping?

AR8327/AR8328 supports Ipv4 IGMP snooping (v1/v2/v3-Lite) and Ipv6 MLD snooping (v1/v2-Lite) by hardware. The following IGMP/MLD protocol packets can be recognized and the ARL table can be updated by hardware.

- IGMP v1 Join
- IGMP v2 Join/Leave
- IGMP v3 Include/Exclude (Group address=1 & Source IP=0)
- MLD v1 Join/Leave
- MLD v2 Include/Exclude (Group address=1 & Source IP=0)
- Group address aging timer (7-level)
- Group address priority remap
- Smart Leave (0x620[14])

	Address Offset	Value	Description
1	0x624	0x017f7f7f	IGMP_JOIN_LEAVE_DP=P0
2	0x620	0x000000f0	If 0x620[10]=0
		0x000004f8	If 0x620[10]=1
3	0x618	0x1ff8002b	Enable PRI=7, Static entry for v3
		0x1f78002b	Enable PRI=7, Aging entry for v2
4	0x214	0x01060606	Enable v3 mode
		0x00060606	Disable v3 mode
5	0x210	0x06060606	
6	0x664/0x670/0x67c/0x688/0x694/0x6a0/0x6ac	0x00040018	Enable DA priority

[ssdk_sh to set router IGMP_JOIN_LEAVE_DP] ssdk_sh igmp rp set 0x1

[ssdk_sh to enable CPU port] ssdk_sh misc cpuPort set enable

[ssdk_sh to copy IGMP to CPU] ssdk_sh igmp cmd set cpycpu

[ssdk_sh to set IGMP/MLD PRI] ssdk_sh igmp queue set enable 7

[ssdk_sh to set static entry] ssdk_sh igmp static set enable

[ssdk_sh to enable global hardware IGMP snooping] ssdk_sh igmp createStatus set enable

[ssdk_sh to enable IGMPv3/MLDv2] ssdk_sh igmp version3 set enable

[ssdk_sh to enable per port hardware join/leave] ssdk_sh igmp portJoin set 0 enable
ssdk_sh igmp portLeave set 0 enable

[ssdk_sh to enable DA priority] ssdk_sh qos ptMode set 0 da enable

The following configuration is for global control:

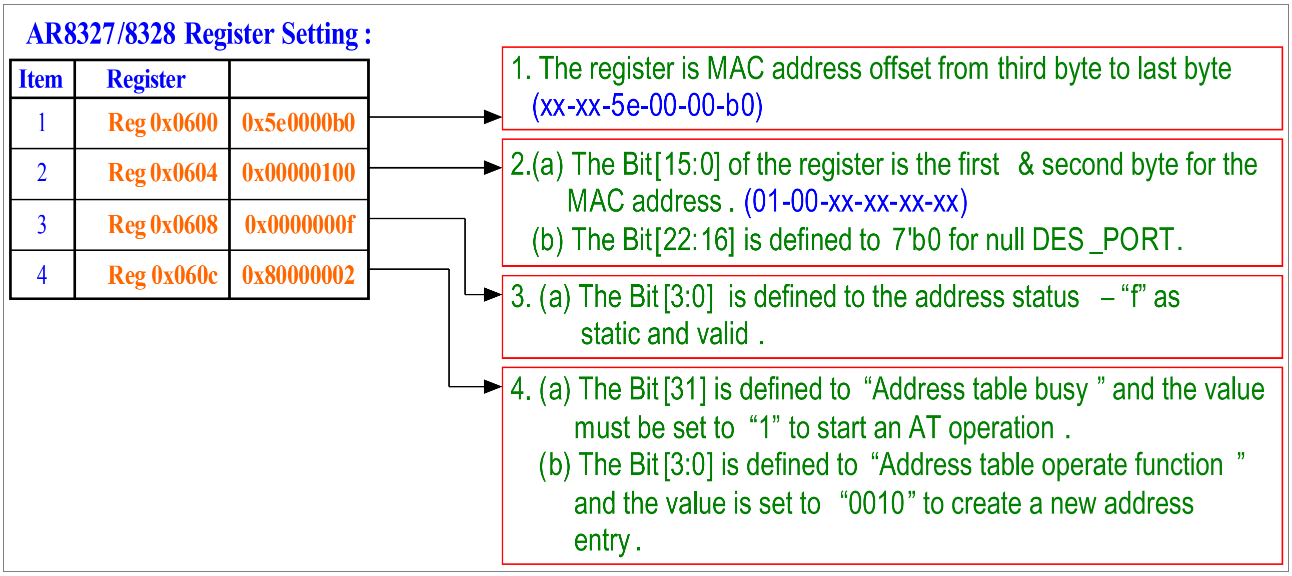
1. GLOBAL_FW_CONTROL0 Register 0x620[14]: AR8327/AR8328 will recognize some IGMP/MLD packets to update ARL table. After updating the portmap of ARL entry (IGMP/MLD group address), bit 14 is used to control dropping or forwarding IGMP/MLD leave packet to IGMP_JOIN_LEAVE_DP, if the portmap of ARL entry is not empty.
2. GLOBAL_FW_CONTROL1 Register 0x624[30:24]: The destination port map for IGMP/MLD join or leave packets. Normally the value is 0x01, the join/leave packets are forwarded to CPU port. For router application, the join/leave packets are forwarded to CPU port and WAN port.
3. ARL_CTRL Register 0x618[27:24]: bit 27 is used to enable/disable group address priority remap function and bit[26:24] is used to define the remap priority queue.
4. ARL_CTRL Register 0x618[23:20]: The 7-level group address aging timer can be configured. When the value is 0xf, the group address is never aged-out before receiving the leave packet. When the value is 0x7, the group address aging timer is about 300 seconds.
5. ARL_CTRL Register 0x618[28]: enable hardware add/remove group address to ARL table for IGMPv1/v2 and MLDv1 join/leave packet.
6. FRAME_ACK_CTRL1 Register 0x214[24]: enable hardware add/remove group address to ARL table for IGMPv3 and MLDv2 include/exclude packet.
7. FRAME_ACK_CTRL1 Register 0x210[2, 10, 18, 26] and 0x214[2, 10, 18]: When this bit is set, that port will remove the membership on receiving leave packet.
8. FRAME_ACK_CTRL1 Register 0x210[1, 9, 17, 25] and 0x214[1, 9, 17]: When this bit is set, that port will add the membership on receiving join packet.
9. FRAME_ACK_CTRL1 Register 0x210[0, 8, 16, 24] and 0x214[0, 8, 16]: enable IGMP/MLD packets.

9.2 How does AR8327/AR8328 forward the IP multicast packets?

When AR8327/AR8328 receives the IP multicast packets, if the DA is not contained in ARL table (i.e., no one joins this group address), it is treated as unknown multicast packets.

You have two methods to prevent the un-join multicast packet from flooding.

1. MULTI_FLOOD_DP : Flood Mask Register 0x624[14:8]
2. Create a static group address entry with null DES_PORT



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b0
2	Reg 0x0604	0x00000100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000002

1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b0)

2.(a) The Bit[15:0] of the register is the first & second byte for the MAC address . (01-00-xx-xx-xx-xx)
(b) The Bit[22:16] is defined to 7'b0 for null DES_PORT.

3. (a) The Bit[3:0] is defined to the address status – "f" as static and valid .

4. (a) The Bit[31] is defined to "Address table busy" and the value must be set to "1" to start an AT operation .
(b) The Bit[3:0] is defined to "Address table operate function" and the value is set to "0010" to create a new address entry.

Fig 9-1: Create a new address "01-00-5e-00-00-b0" and member is null

9.3 How does AR8327/AR8328 implement Software IGMP Snooping?

If AR8327/AR8328 hardware IGMP/MLD snooping can not meet your requirement, you can also implement IGMP/MLD snooping by software. CPU needs to implement IGMP/MLD protocol and maintain ARL table.

The following examples describe several ARL table operations when Join/Leave is happened.

1. Join / leave IGMP group procedure: There are two IP streaming packets, and its DA= 01-00-5e-00-00-b0 and DA= 01-00-5e 00 00 b1. Please refer to the below step1 ~ step6.
   a. Step1: First, no ports join to the two groups.

      ➔ The two IP streaming will be forwarded to all ports.

   b. Step2: P1 receives IGMP Join and the group address is "01-00-5e-00-00-b0". CPU needs to create a new entry. Fig 9-2: Create a new address "01-00-5e-00-00-b0" and member is P1

      ➔ Then IP streaming packet (DA: 01-00-5e-00-00-b0) will be forwarded to P1 and other ports won't receive this IP streaming packet.

   c. Step3 : P2 receives IGMP Join and the group address is "01-00-5e-00-00-b1". CPU needs to create another entry. Fig 9-3: Create another address "01-00-5e-00-00-b1" and member is P2

      ➔ Then IP streaming packet (DA: 01-00-5e-00-00-b1) will be forwarded to P2 and other ports won't receive this IP streaming packet.

   d. Step4 : P3 receives IGMP Join and the group address is "01-00-5e-00-00-b0". CPU needs to update the existed entry. Fig 9-4: Update IGMP group member as P1&P3 for the multicast address 01-00-5e-00-00-b0

      ➔ Then IP streaming packet (DA: 01-00-5e-00-00-b0) will be forwarded to P1/P3 and other ports won't receive this IP streaming packet.

   e. Step5 : P1 receives IGMP Leave and the group address is "01-00-5e-00-00-b0". CPU needs to update the existed entry. Fig 9-5: Update IGMP group member as P3 for the multicast address 01-00-5e-00-00-b0(remove P1 member)

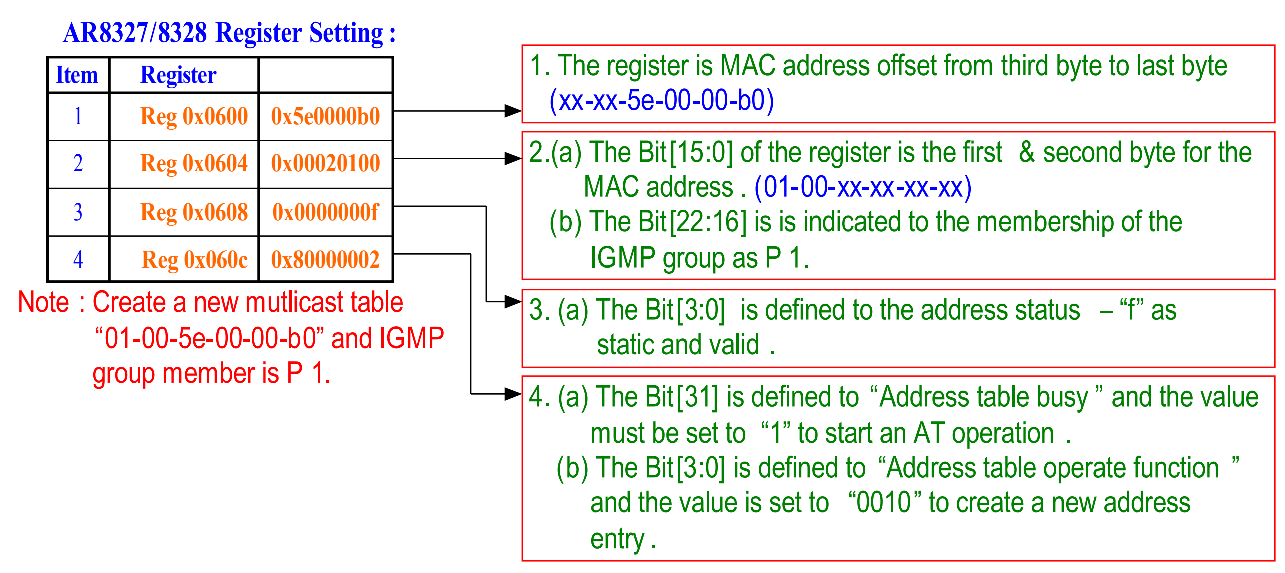
      ➔ Then IP streaming packet (DA: 01-00-5e-00-00-b0) will be forwarded to P3 and other ports won't receive the IP streaming packet.

   f. Step6 : P3 receives IGMP Leave and the group address is "01-00-5e-00-00-b0". Because P3 is the last member, CPU needs to remove the entry. Fig 9-6: Remove the address 01-00-5e-00-00-b0 in the multicast table

      ➔ If the multicast DA is not contained in ARL table, the IP streaming packet is treated as unknown multicast packets. So, the IP streaming packet (DA: 01-00-5e-00-00-b0) will be forwarded to all ports.

   g. Step7 : P2 receives IGMP Leave and the group address is "01-00-5e-00-00-b1". Because P2 is the last member, CPU needs to remove the entry. Fig 9-7: Remove the address 01-00-5e-00-00-b1 in the multicast table)

      ➔ So, the IP streaming packet (DA: 01-00-5e-00-00-b1) will be forwarded to all ports.



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b0
2	Reg 0x0604	0x00020100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000002

Note : Create a new mutlicast table "01-00-5e-00-00-b0" and IGMP group member is P 1.

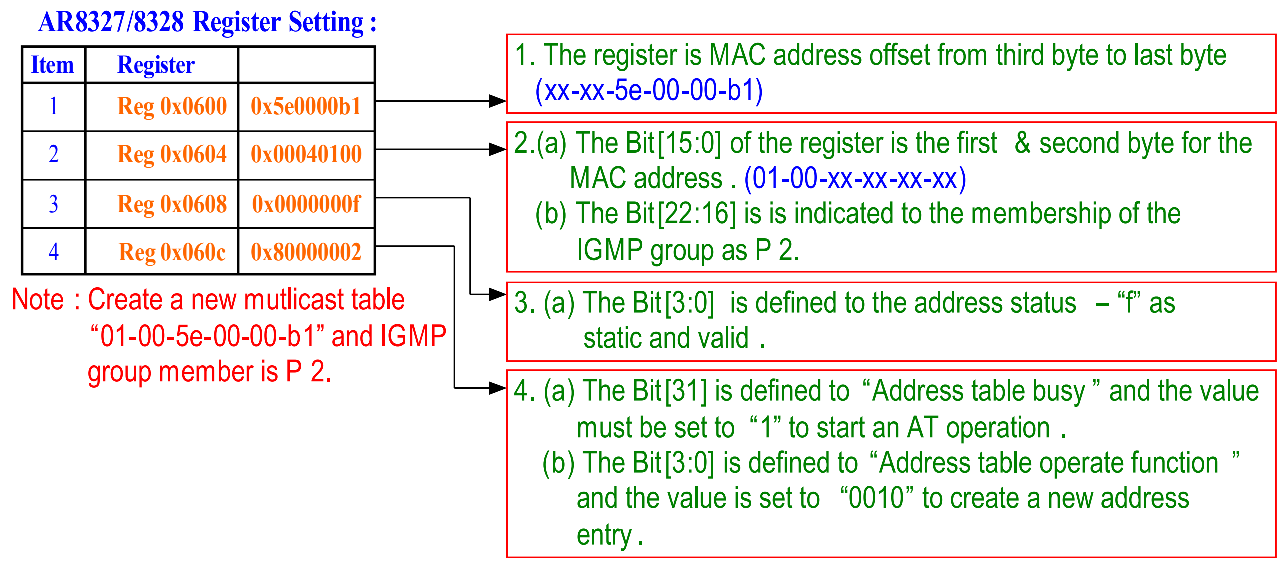
1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b0)

2.(a) The Bit[15:0] of the register is the first & second byte for the MAC address . (01-00-xx-xx-xx-xx)
(b) The Bit[22:16] is is indicated to the membership of the IGMP group as P 1.

3. (a) The Bit[3:0] is defined to the address status – "f" as static and valid .

4. (a) The Bit[31] is defined to "Address table busy " and the value must be set to "1" to start an AT operation .
(b) The Bit[3:0] is defined to "Address table operate function " and the value is set to "0010" to create a new address entry .

Fig 9-2: Create a new address "01-00-5e-00-00-b0" and member is P1



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b1
2	Reg 0x0604	0x00040100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000002

Note : Create a new mutlicast table "01-00-5e-00-00-b1" and IGMP group member is P 2.

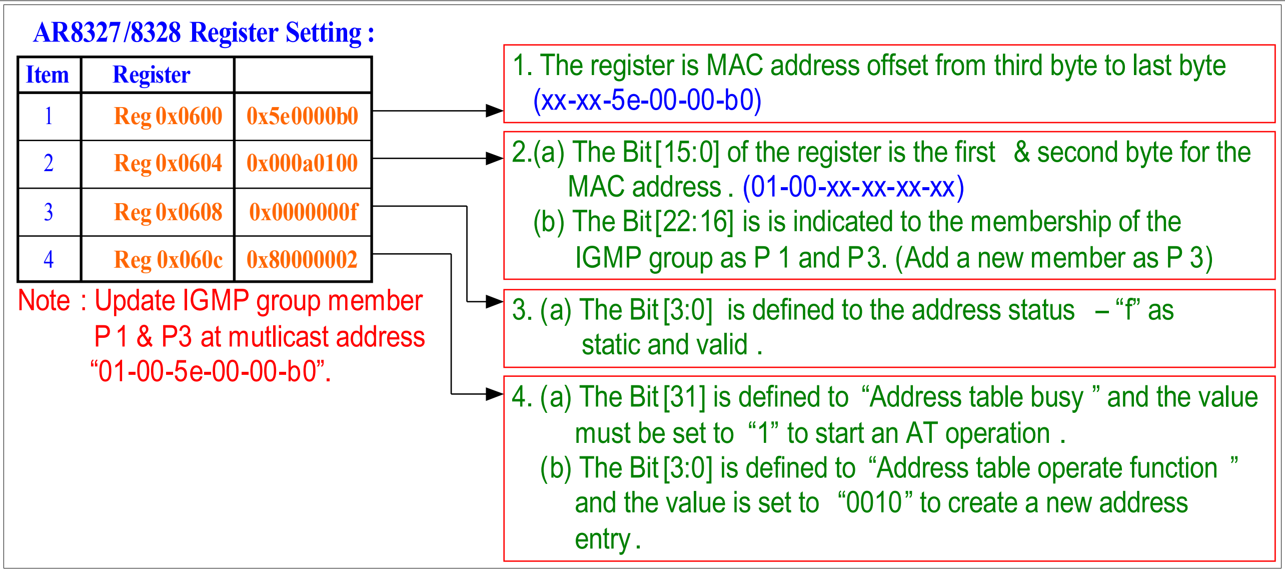
1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b1)

2.(a) The Bit[15:0] of the register is the first & second byte for the MAC address . (01-00-xx-xx-xx-xx)
(b) The Bit[22:16] is is indicated to the membership of the IGMP group as P 2.

3. (a) The Bit[3:0] is defined to the address status – "f" as static and valid .

4. (a) The Bit[31] is defined to "Address table busy " and the value must be set to "1" to start an AT operation .
(b) The Bit[3:0] is defined to "Address table operate function " and the value is set to "0010" to create a new address entry .

Fig 9-3: Create another address "01-00-5e-00-00-b1" and member is P2



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b0
2	Reg 0x0604	0x000a0100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000002

Note : Update IGMP group member P1 & P3 at mutlicast address "01-00-5e-00-00-b0".

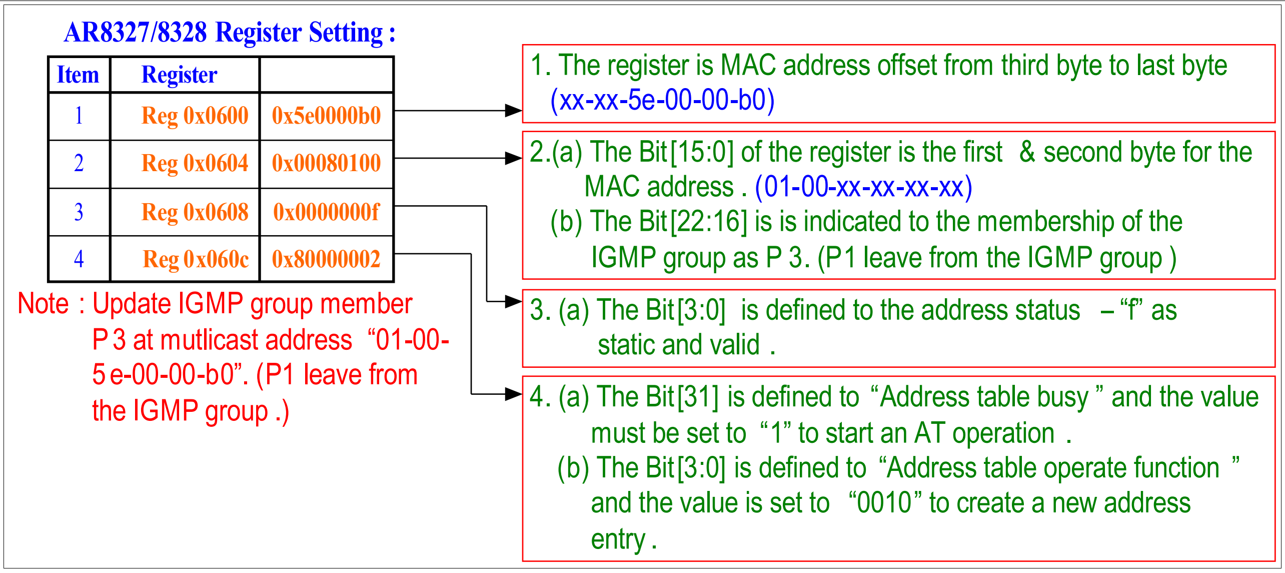
1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b0)

2.(a) The Bit[15:0] of the register is the first & second byte for the MAC address. (01-00-xx-xx-xx-xx)
(b) The Bit[22:16] is is indicated to the membership of the IGMP group as P1 and P3. (Add a new member as P3)

3. (a) The Bit[3:0] is defined to the address status – "f" as static and valid.

4. (a) The Bit[31] is defined to "Address table busy" and the value must be set to "1" to start an AT operation.
(b) The Bit[3:0] is defined to "Address table operate function" and the value is set to "0010" to create a new address entry.

Fig 9-4: Update IGMP group member as P1&P3 for the multicast address 01-00-5e-00-00-b0



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b0
2	Reg 0x0604	0x00080100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000002

Note : Update IGMP group member P3 at mutlicast address "01-00-5e-00-00-b0". (P1 leave from the IGMP group.)

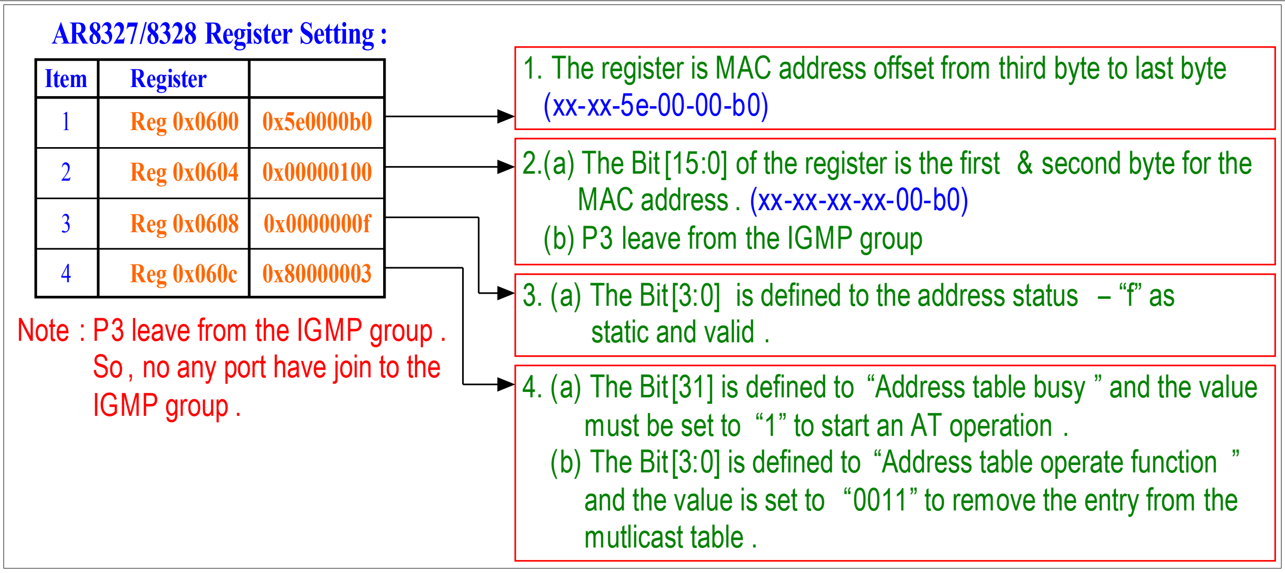
1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b0)

2.(a) The Bit[15:0] of the register is the first & second byte for the MAC address. (01-00-xx-xx-xx-xx)
(b) The Bit[22:16] is is indicated to the membership of the IGMP group as P3. (P1 leave from the IGMP group)

3. (a) The Bit[3:0] is defined to the address status – "f" as static and valid.

4. (a) The Bit[31] is defined to "Address table busy" and the value must be set to "1" to start an AT operation.
(b) The Bit[3:0] is defined to "Address table operate function" and the value is set to "0010" to create a new address entry.

Fig 9-5: Update IGMP group member as P3 for the multicast address 01-00-5e-00-00-b0(remove P1 member)



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b0
2	Reg 0x0604	0x00000100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000003

Note : P3 leave from the IGMP group . So , no any port have join to the IGMP group .

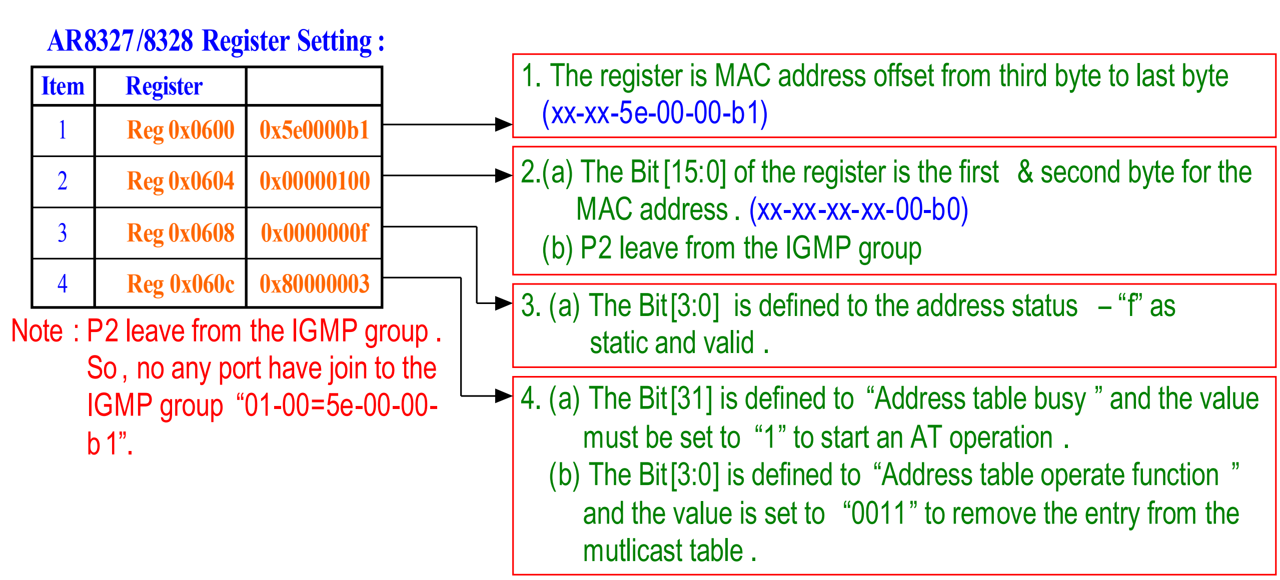
1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b0)

2.(a) The Bit [15:0] of the register is the first & second byte for the MAC address . (xx-xx-xx-xx-00-b0)
(b) P3 leave from the IGMP group

3. (a) The Bit [3:0] is defined to the address status – “f” as static and valid .

4. (a) The Bit [31] is defined to “Address table busy ” and the value must be set to “1” to start an AT operation .
(b) The Bit [3:0] is defined to “Address table operate function ” and the value is set to “0011” to remove the entry from the mutlicast table .

Fig 9-6: Remove the address 01-00-5e-00-00-b0 in the multicast table



AR8327/8328 Register Setting :

Item	Register	
1	Reg 0x0600	0x5e0000b1
2	Reg 0x0604	0x00000100
3	Reg 0x0608	0x0000000f
4	Reg 0x060c	0x80000003

Note : P2 leave from the IGMP group . So , no any port have join to the IGMP group “01-00=5e-00-00-b1”.

1. The register is MAC address offset from third byte to last byte (xx-xx-5e-00-00-b1)

2.(a) The Bit [15:0] of the register is the first & second byte for the MAC address . (xx-xx-xx-xx-00-b0)
(b) P2 leave from the IGMP group

3. (a) The Bit [3:0] is defined to the address status – “f” as static and valid .

4. (a) The Bit [31] is defined to “Address table busy ” and the value must be set to “1” to start an AT operation .
(b) The Bit [3:0] is defined to “Address table operate function ” and the value is set to “0011” to remove the entry from the mutlicast table .

Fig 9-7: Remove the address 01-00-5e-00-00-b1 in the multicast table

10 Rate Limit Control

AR8237/AR8328 supports the following rate limit mechanism.

1. Ingress Port Rate Limit
2. Egress Port Rate Limit
3. Egress Queue Rate Limit
4. Ingress Policer: Two Rate Three Color (compliant with Metro Ethernet Forum) and Two Single Rate
5. Egress Policer: Two Rate Three Color (compliant with MEF)

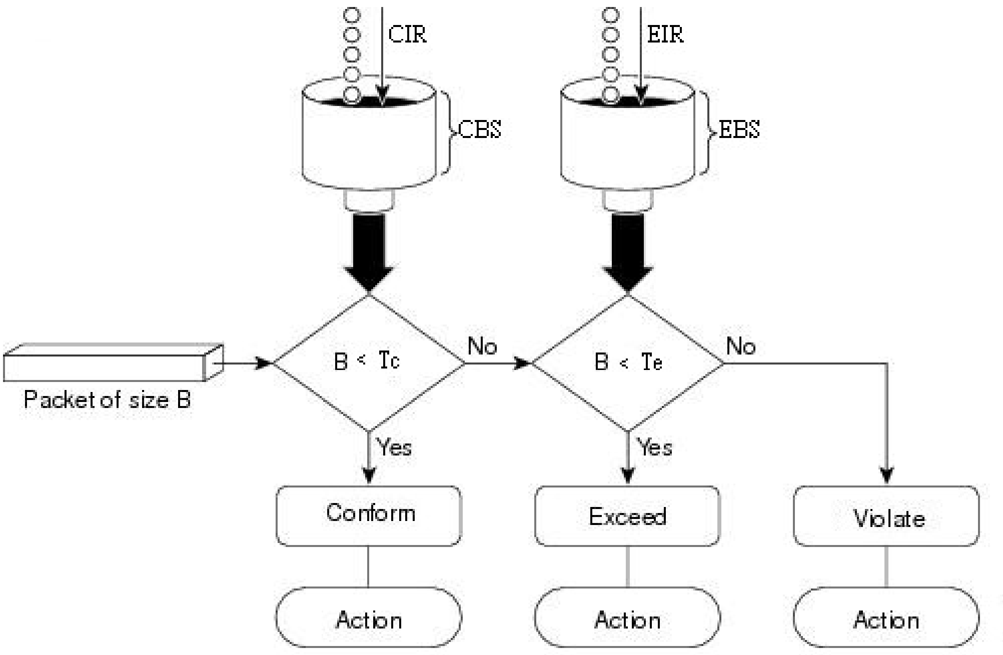


Fig 10-1: Two Rate Three Color

<u>Color Blind Algorithm Skeleton:</u>

```
If (Service Frame length is less than C-Bucket tokens)
        {declare green; remove tokens from C-Bucket}
else if (Service Frame length is less than E-Bucket tokens)
        {declare yellow; remove tokens from E-Bucket}
else declare red
```

10.1 Ingress/Egress Port Rate Limit

AR8327/AR8328 supports Ingress and Egress Port Rate Limit.

Rate Limit is implemented by a token bucket scheme where each bucket is implemented as a 15-bit counter. (ING_CIR, ING_EIR, EG_PRI0_CIR, EG_PRI0_EIR)

For each port, the user can specify the following parameters.

- **CIR/EIR: Token replenish rate in 32kbps*N. On every time slot, the tokens are added to the token bucket.**
- **Slot Time T: For the Time Slot, there are four options 100us/1ms/10ms/100ms.**
- **Ingress Burst Size CBS/EBS: For the burst size, there are seven options 4k/32k/128k/512k/2M/8M/32M byte.**
- **Egress Burst Size CBS/EBS: For the burst size, there are seven options 2k/4k/8k/16k/32k/128k/512k byte.**

Because CBS=T*CIR, the following table lists the rate setting for various scenarios.

Rate	CIR	T	CBS
1Gbps	N*32kbps	1ms	128kbyte
100Mbps	N*32kbps	1ms	32kbyte
10Mbps	N*32kbps	1ms	4kbyte

For example, use CIR to set the P0 ingress rate control to 100Mbps. 100Mbps=32Kbps*3200

	Address Offset	Value	Description
1	0xb00	0x18410c80	CBS=32kbyte
2	0xb08	0x0000fe00	All packets
3	0xb04	0x00047fff	If P0 flow-control is disable
	0xb04	0x00847fff	If P0 flow-control is enable

1. PORT0_ING_RATE_CTRL0 0xb00[14:0]: The rate step is 32Kbps.
2. PORT0_ING_RATE_CTRL0 0xb00[17:15]: Select burst size to 32kbyte.

[ssdk_sh to set ingress port policer] ssdk_sh rate portpolicer set 0 no yes no no no yes 102400 32768 0xfe 1ms no 0 0 0x0 1ms

For example, use CIR to set the egress rate control to 100Mbps. 100Mbps=32Kbps*3200, N=1600

	Address Offset	Value	Description
1	0x8ac	0x0000000a	EGRESS_RATE_EN_0=1
2	0x8a8	0x00000050	CBS=32kbyte
3	0x890	0x7fff0c80	EG_PRI0_CIR_0

1. PORT0_EG_RATE_CTRL0 0x890[14:0]: The rate step is 32Kbps.
2. PORT0_EG_RATE_CTRL6 0x8a8[6:4]: Select burst size to 32kbyte.
3. PORT0_EG_RATE_CTRL7 0x8ac[3]: Enable port base rate limit.

[ssdk_sh to set egress port shaper] ssdk_sh rate portshaper set 0 enable yes 102400 32768 0 0

10.2 Egress Queue Rate Limit

AR8327/AR8328 supports Egress Rate Limit by different priority queues.

For example, set the EG_PRI3_CIR=10Mbps, EG_PRI2_CIR=20Mbps, EG_PRI1_CIR=30Mbps and EG_PRI0_CIR=40Mbps

	Address Offset	Value	Description
1	0x8ac	0x00000002	EGRESS_RATE_EN_0=0
2	0x8a8	0x50505050	CBS=32kbyte
3	0x894	0x01400280	
4	0x890	0x03c00500	
5	0x664	0x00020018	Enable VLAN priority for ING_PORT_PRI for untag packet
	0x670	0x00020018	
	0x67c	0x00020018	
	0x688	0x00020018	
	0x694	0x00020018	
	0x6a0	0x00020018	
	0x6ac	0x00020018	

1. PORT0_EG_RATE_CTRL0 0x890[14:0]: The rate step is 32Kbps.
2. PORT0_EG_RATE_CTRL6 0x8a8[6:4]: Select burst size to 32kbyte.
3. PORT0_EG_RATE_CTRL7 0x8ac[3]: Enable port base rate limit.

[ssdk_sh to enable port 0 VLAN priority] ssdk_sh qos ptMode set 0 up enable

[ssdk_sh to enable port 1 VLAN priority] ssdk_sh qos ptMode set 1 up enable

[ssdk_sh to enable port 2 VLAN priority] ssdk_sh qos ptMode set 2 up enable

[ssdk_sh to enable port 3 VLAN priority] ssdk_sh qos ptMode set 3 up enable

[ssdk_sh to enable port 4 VLAN priority] ssdk_sh qos ptMode set 4 up enable

[ssdk_sh to set egress queue3 shaper] ssdk_sh rate queueshaper set 0 3 enable yes 10240 32768 0 0

[ssdk_sh to set egress queue2 shaper] ssdk_sh rate queueshaper set 0 2 enable yes 20480 32768 0 0

[ssdk_sh to set egress queue1 shaper] ssdk_sh rate queueshaper set 0 1 enable yes 30720 32768 0 0

[ssdk_sh to set egress queue0 shaper] ssdk_sh rate queueshaper set 0 0 enable yes 40960 32768 0 0

11 Storm Control

11.1 Storm control

AR8327/AR8328 supports CIR/EIR to implement storm control for broadcast, unicast, unknown unicast, multicast and unknown multicast packets.

For example, use CIR to set the P0 storm control to 10Mbps. 10Mbps=32Kbps*320

	Address Offset	Value	Description
1	0xb00	0x18410140	CBS=32kbyte
2	0xb04	0x00047fff	Bit 22 ING_RATE_UNIT=0, bytes
		0x00447fff	Bit 22 ING_RATE_UNIT=1, packets
3	0xb08	0x00003800	Unknown unicast/multicast and broadcast

[ssdk_sh to set ingress port policer] ssdk_sh rate portpolicer set 0 no yes no no no yes 10240 32768 0x38 1ms no 0 0 0x0 1ms

1. PORT0_ING_RATE_CTRL2 0xb08[15]: Enable multicast packet is calculated to CIR.
2. PORT0_ING_RATE_CTRL2 0xb08[14]: Enable unicast packet is calculated to CIR.
3. PORT0_ING_RATE_CTRL2 0xb08[13]: Enable unknown multicast packet is calculated to CIR.
4. PORT0_ING_RATE_CTRL2 0xb08[12]: Enable unknown unicast packet is calculated to CIR.
5. PORT0_ING_RATE_CTRL2 0xb08[11]: Enable broadcast packet is calculated to CIR.

The same configuration also can be applied to EIR.

1. PORT0_ING_RATE_CTRL2 0xb08[7]: Enable multicast packet is calculated to EIR.
2. PORT0_ING_RATE_CTRL2 0xb08[6]: Enable unicast packet is calculated to EIR.
3. PORT0_ING_RATE_CTRL2 0xb08[5]: Enable unknown multicast packet is calculated to EIR.
4. PORT0_ING_RATE_CTRL2 0xb08[4]: Enable unknown unicast packet is calculated to EIR.
5. PORT0_ING_RATE_CTRL2 0xb08[3]: Enable broadcast packet is calculated to EIR.

12 QoS Control

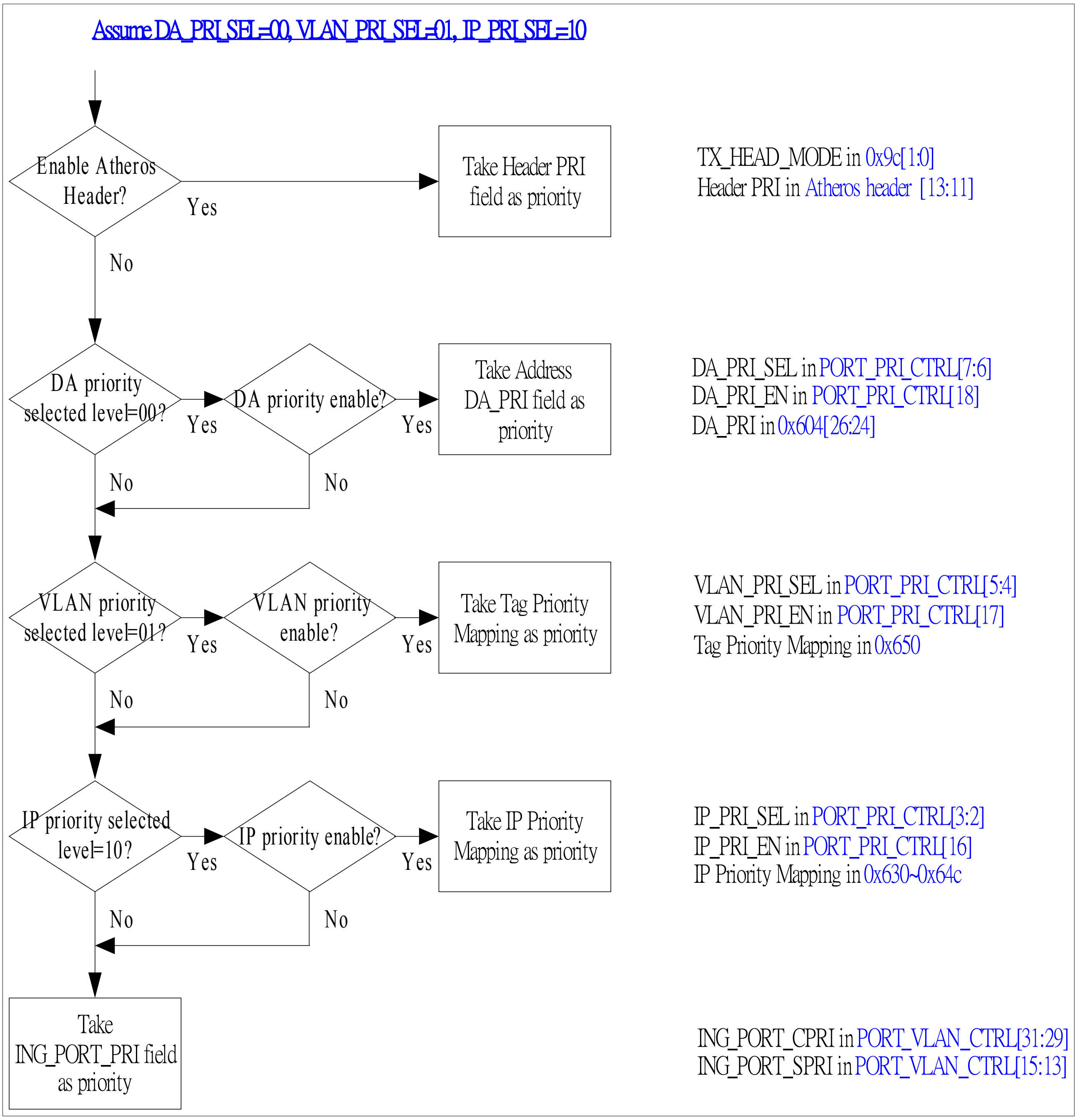

Assume DA_PRI_SEL=00, VLAN_PRI_SEL=01, IP_PRI_SEL=10
Enable Atheros Header?
Yes
Take Header PRI field as priority
TX_HEAD_MODE in 0x9c[1:0]
Header PRI in Atheros header [13:11]
No
DA priority selected level=00?
Yes
DA priority enable?
Yes
Take Address DA_PRI field as priority
DA_PRI_SEL in PORT_PRI_CTRL[7:6]
DA_PRI_EN in PORT_PRI_CTRL[18]
DA_PRI in 0x604[26:24]
No
No
VLAN priority selected level=01?
Yes
VLAN priority enable?
Yes
Take Tag Priority Mapping as priority
VLAN_PRI_SEL in PORT_PRI_CTRL[5:4]
VLAN_PRI_EN in PORT_PRI_CTRL[17]
Tag Priority Mapping in 0x650
No
No
IP priority selected level=10?
Yes
IP priority enable?
Yes
Take IP Priority Mapping as priority
IP_PRI_SEL in PORT_PRI_CTRL[3:2]
IP_PRI_EN in PORT_PRI_CTRL[16]
IP Priority Mapping in 0x630~0x64c
No
No
Take ING_PORT_PRI field as priority
ING_PORT_CPRI in PORT_VLAN_CTRL[31:29]
ING_PORT_SPRI in PORT_VLAN_CTRL[15:13]

Fig 12-1: QoS priority classification flow chart

12.1 How to configure Port base priority?

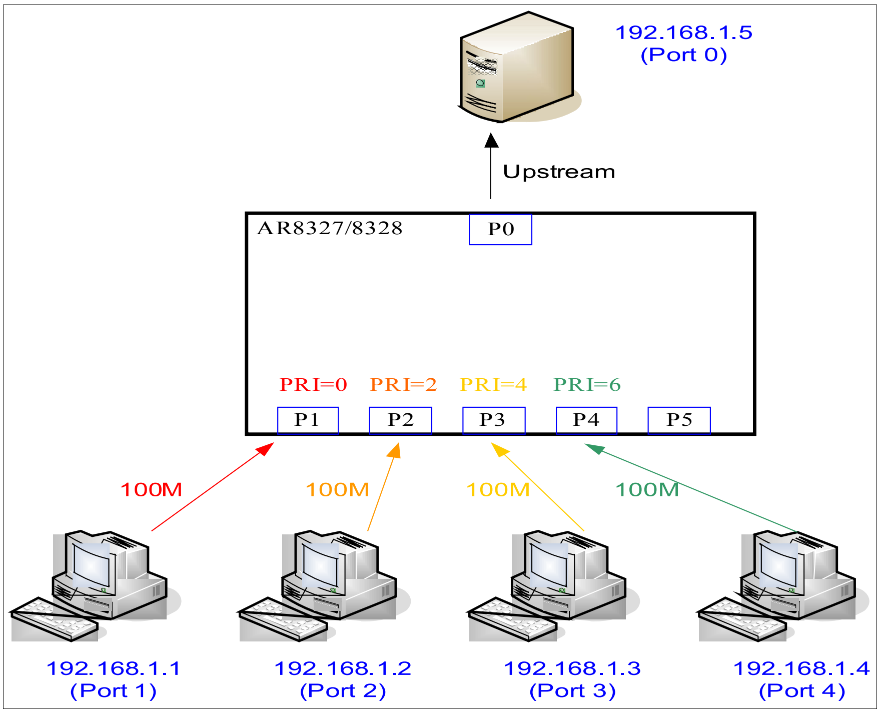


Fig 12-2: Port base priority

	Address Offset	Value	Description
1	0x664	0x00020018	Enable VLAN priority for ING_PORT_PRI of untag packet
	0x670	0x00020018	
	0x67c	0x00020018	
	0x688	0x00020018	
	0x694	0x00020018	
2	0x428	0x00010001	PORT_CPRI=0
	0x430	0x40010001	PORT_CPRI=2
	0x438	0x80010001	PORT_CPRI=4
	0x440	0xc0010001	PORT_CPRI=6
3	0x810	0x54332201	WAN Port PRI to Queue Mapping
4	0x830	0x10841041	Default
	0x830	0x00000000	SP
	0x830	0x40841041	MIX(1SP+5WRR)
	0x830	0x80041041	MIXPLUS(2SP+4WRR)
	0x830	0xd0841041	WRR (Q0~Q5 1:2:4:8:8:8)

AR8327/8328 supports 6 queues for Port0/5/6 and 4 queues for Port 1/2/3/4. The final queue mapping is defined by 0x810 (WAN Port PRI to Queue Mapping) and 0x814 (LAN Port PRI to Queue Mapping).

AR8327/AR8328 supports 4 kinds of priority scheduling mechanisms which is controlled by the WEIGHT_PRI_CTRL of the egress port.

1. Strict priority mode
2. 1SP+3(or 5) WRR mode I: Queue 3 is strict priority and Queue 2/1/0 are WRR scheme with programmable weight
3. 2SP+2(or 4) WRR mode II: Queue 3 and Queue 2 are strict priority and Queue 1/0 are WRR scheme with programmable weight
4. Weighted Round Robin mode: The weight is programmable.

[ssdk_sh to enable port 0 VLAN priority] ssdk_sh qos ptMode set 0 up enable

[ssdk_sh to enable port 1 VLAN priority] ssdk_sh qos ptMode set 1 up enable

[ssdk_sh to enable port 2 VLAN priority] ssdk_sh qos ptMode set 2 up enable

[ssdk_sh to enable port 3 VLAN priority] ssdk_sh qos ptMode set 3 up enable

[ssdk_sh to enable port 4 VLAN priority] ssdk_sh qos ptMode set 4 up enable

[ssdk_sh to enable port 1 default C-priority] ssdk_sh qos ptDefaultCpri set 1 0

[ssdk_sh to enable port 2 default C-priority] ssdk_sh qos ptDefaultCpri set 2 2

[ssdk_sh to enable port 3 default C-priority] ssdk_sh qos ptDefaultCpri set 3 4

[ssdk_sh to enable port 4 default C-priority] ssdk_sh qos ptDefaultCpri set 4 6

[ssdk_sh to enable port 0 egress SP priority] ssdk_sh qos ptschMode set 0 sp 1,2,4,8,8,8

[ssdk_sh to enable port 0 egress WRR priority] ssdk_sh qos ptschMode set 0 wrr 1,2,4,8,8,8

[ssdk_sh to enable port 0 egress MIX priority] ssdk_sh qos ptschMode set 0 mix 1,2,4,8,8,8

[ssdk_sh to enable port 0 egress MIXPLUS priority] ssdk_sh qos ptschMode set 0 mixplus 1,2,4,8,8,8

12.2 How to enable HOL for port/queue depth?

AR8327/8328 supports ingress port buffer and egress port/queue buffer control to handle Head-On-Line blocking. The following configuration is the recommended value.

	Address Offset	Value	Description
1	0x970	0x2a888888	
	0x974	0x000000c6	
	0x978	0x2a888888	
	0x97c	0x000000c6	
	0x980	0x2a888888	
	0x984	0x000000c6	
	0x988	0x2a888888	
	0x98c	0x000000c6	
	0x990	0x2a888888	
	0x994	0x000000c6	
	0x998	0x2a888888	
	0x99c	0x000000c6	
	0x9a0	0x2a888888	
	0x9a4	0x000000c6	

13 MIB Counter

13.1 Direct access

You can enable MIB counter on register 0x30 bit[0] and access MIB counter directly on the following memory map. The MIB counter will be clear after read in default. If you want to keep counting and not to clear, you can set 0x34[20]=1.

Address Offset	Description
0x01000 ~ 0x010a7	Port 0 MIB counter
0x01100 ~ 0x011a7	Port 1 MIB counter
0x01200 ~ 0x012a7	Port 2 MIB counter
0x01300 ~ 0x013a7	Port 3 MIB counter
0x01400 ~ 0x014a7	Port 4 MIB counter
0x01500 ~ 0x015a7	Port 5 MIB counter
0x01600 ~ 0x016a7	Port 6 MIB counter

For some undefined MIB counters, you can use the follow formulas to get the value.

TxFcsErr=TxOverSize+TxUnderRun+TxLateCol

TxOK= Tx64Byte+ Tx128Byte+ Tx256Byte+ Tx512Byte+ Tx1024Byte+ Tx1518Byte + TxMaxByte- TxLateCol -TxAbortCol- TxUnderRun- TxExcDefer

TxUnicast= TxOK- TxBroad- TxPause- TxMulti

RxOK= Rx64Byte+ Rx128Byte+ Rx256Byte+ Rx512Byte+Rx1024Byte+Rx1518Byte+ RxMaxByte- RxFcsErr - RxAllignErr

RxUnicast= RxOK- RxBroad- RxPause- RxMulti

[ssdk_sh to enable MIB function] ssdk_sh mib status set enable

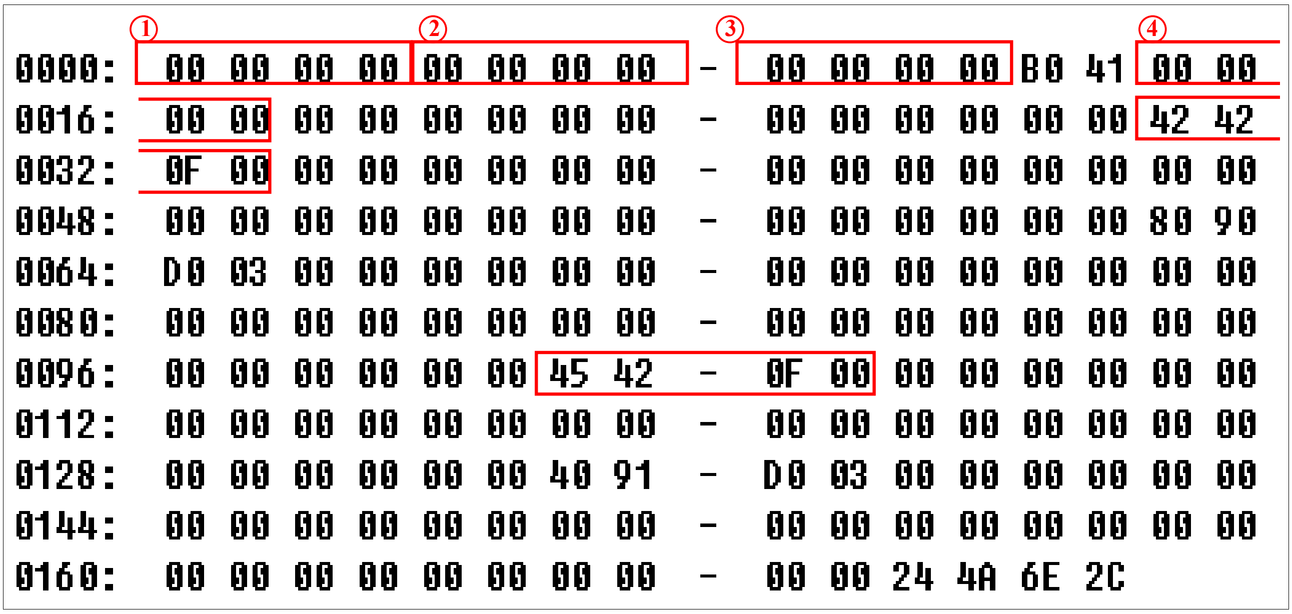
[ssdk_sh to get port 0 MIB counter] ssdk_sh mib statistics get 0

13.2 Autocast

AR8327/AR8328 supports “Autocast” to generate 7 MIB counter frames to CPU port. The packet format of MIB counter frame is

1. 174bytes, include 2 byte (header), 168 byte (MIB counter) and 4 bytes (CRC).
2. 176bytes, include 4 byte (header), 168 byte (MIB counter) and 4 bytes (CRC).

	Address Offset	Value	Description
1	0x620	0x000004f0	Enable CPU port
2	0x98	0x00000000	2-bytes header
		0x00010000	4-bytes header
3	0x9c	0x00000002	Enable Atheros TX header
4	0x30	0x80000301	Enable MIB
5	0x34	0x01020000	Clean all counters for all ports
	0x34	0x03020100	Autocast



```
0000:  00 00 00 00 00 00 00 00  -  00 00 00 00 B0 41 00 00
0016:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 42 42
0032:  0F 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0048:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 80 90
0064:  D0 03 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0080:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0096:  00 00 00 00 00 00 45 42  -  0F 00 00 00 00 00 00 00
0112:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0128:  00 00 00 00 00 00 40 91  -  D0 03 00 00 00 00 00 00
0144:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0160:  00 00 00 00 00 00 00 00  -  00 00 24 4A 6E 2C
```

Fig 13-1: Auto-cast packet of Port 1 MIB counter

” B0 41” is the Atheros header for MIB auto-cast packet. ”42 42 0F 00” is mapped to Rx64Byte and the value is 1,000,002; ”45 42 0F 00” is mapped to Tx64Byte and the value is 1,000,005.

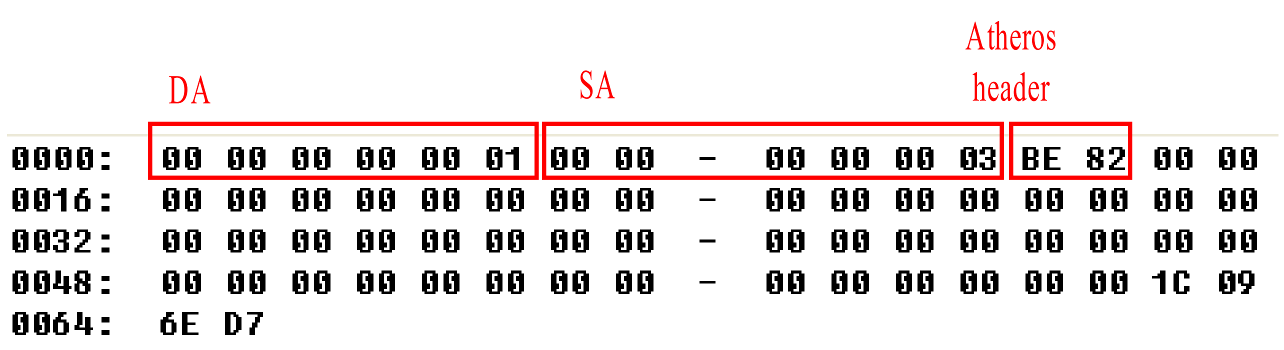
When “Autocast” MIB counters is enabled, the CPU’s MAC needs to runs in promiscuous mode as the related counters are put into the frame at locations which are usually used for the source and destination MAC addresses. If the CPU-MAC isn’t run in promiscuous mode, the related packets will be filtered.

14 Atheros Header

14.1 How does CPU get the source port of the received packet and forward the transmitted packet to dedicated port/group?

When you implement IGMP (or STP), CPU needs to know the source port of the received IGMP report (or BPDU). And CPU also needs to forward IGMP specify query (or BPDU) to dedicated port.

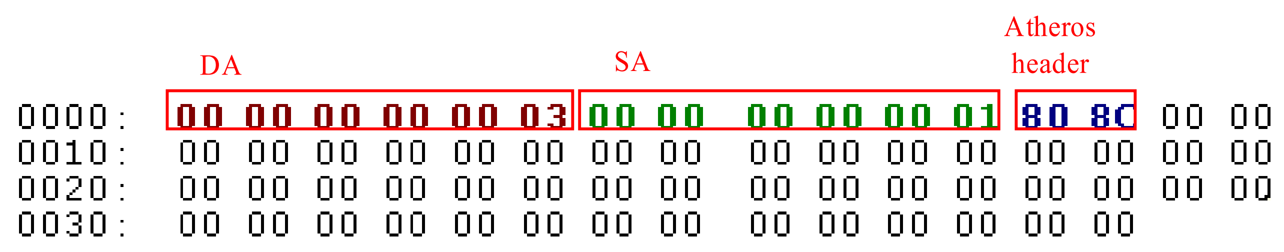
	Address Offset	Value	Description
1	0x620	0x000004f0	Enable CPU port
2	0x98	0x00000000	2-bytes header
3	0x9c	0x0000000a	Enable Atheros TX/RX header



```
0000:  00 00 00 00 00 01 00 00  -  00 00 00 03 BE 82 00 00
0016:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0032:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0048:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 1C 09
0064:  6E D7
```

Fig 14-1: Get source port by 2-bytes Atheros header

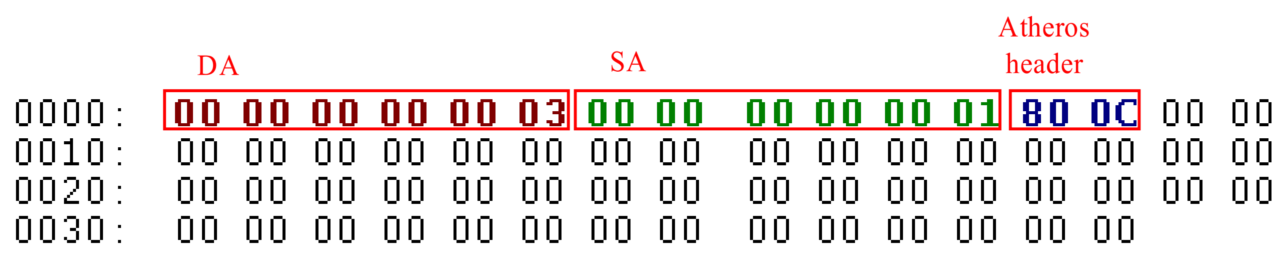
” BE 82 ” is the Atheros header for normal packet and the source port is port 2.



```
0000:  00 00 00 00 00 03 00 00  00 00 00 01 80 8C 00 00
0010:  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0020:  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030:  00 00 00 00 00 00 00 00  00 00 00 00 00 00
```

Fig 14-2: Forward to dedicated port by 2-bytes Atheros header

” 80 8C ” is the Atheros header for normal packet and forward to port 2 and port 3.



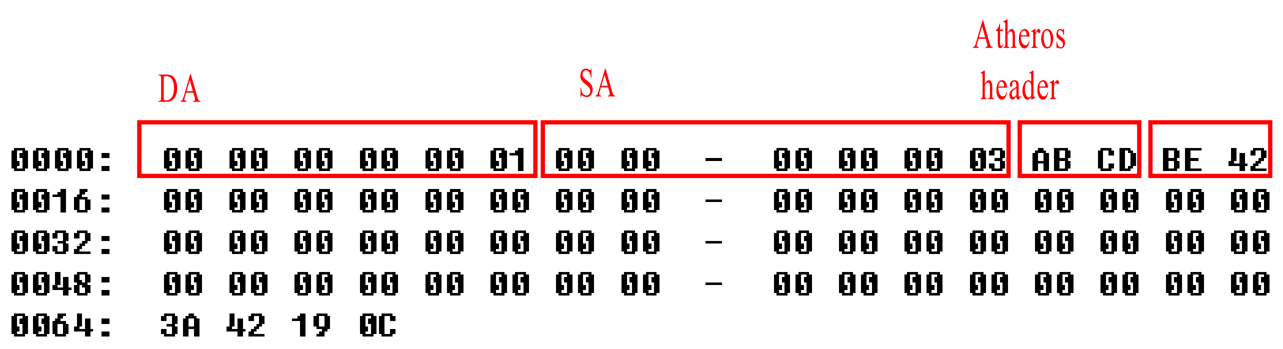
```
0000:  00 00 00 00 00 03 00 00  00 00 00 01 80 0C 00 00
0010:  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0020:  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030:  00 00 00 00 00 00 00 00  00 00 00 00 00 00
```

Fig 14-3: Forward to dedicated VLAN by 2-bytes Atheros header

” 80 0C ” is the Atheros header for normal packet and forward to VLAN with port 2 and port 3 members.

For 2-bytes Atheros header, all packets which are received and transmitted by CPU need to have 2-bytes Atheros header. It may impact CPU performance to forward normal packets. So AR8327/8328 supports new 4-bytes Atheros header and has the option to insert 4-bytes Atheros header for management packet only.

	Address Offset	Value	Description
1	0x620	0x000004f0	Enable CPU port
2	0x98	0x0001abcd	4-bytes header with 0xabcd
3	0x9c	0x0000000a	Enable Atheros TX/RX header for all packets
	0x9c	0x00000005	Enable Atheros TX/RX header for management only



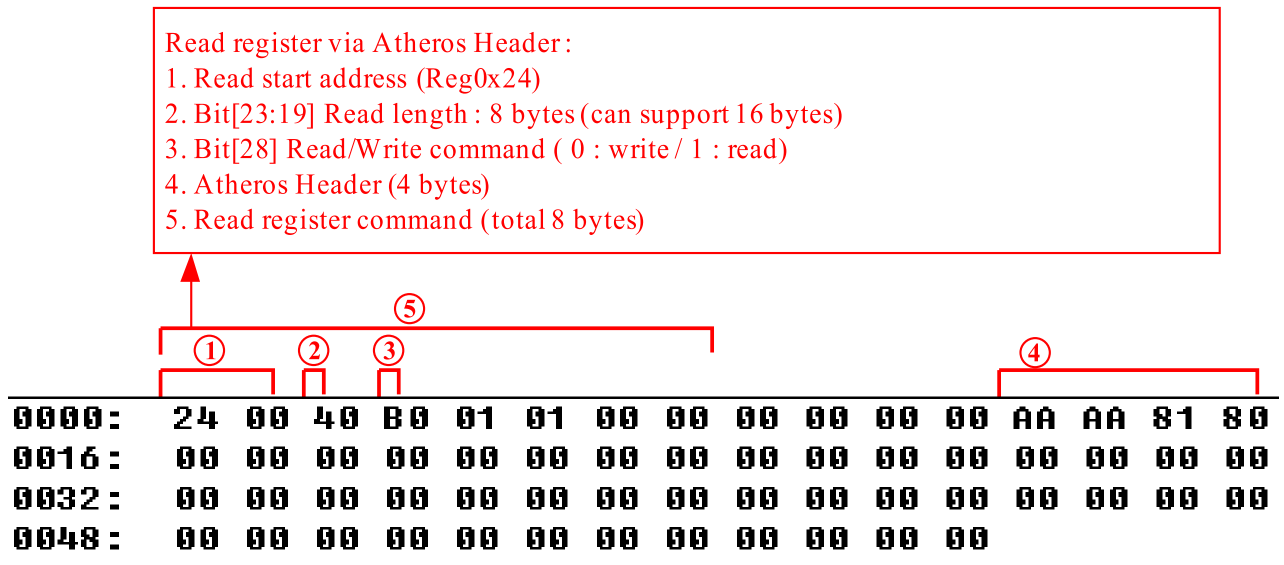
```
0000:  00 00 00 00 00 01 00 00  -  00 00 00 03 AB CD BE 42
0016:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0032:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0048:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0064:  3A 42 19 0C
```

Fig 14-4: Get source port by 4-bytes Atheros header

14.2 How does CPU use the packet to read/write registers?

Besides MDC/MDIO, AR8327/AR8328 also supports in-band registers access by Atheros header.

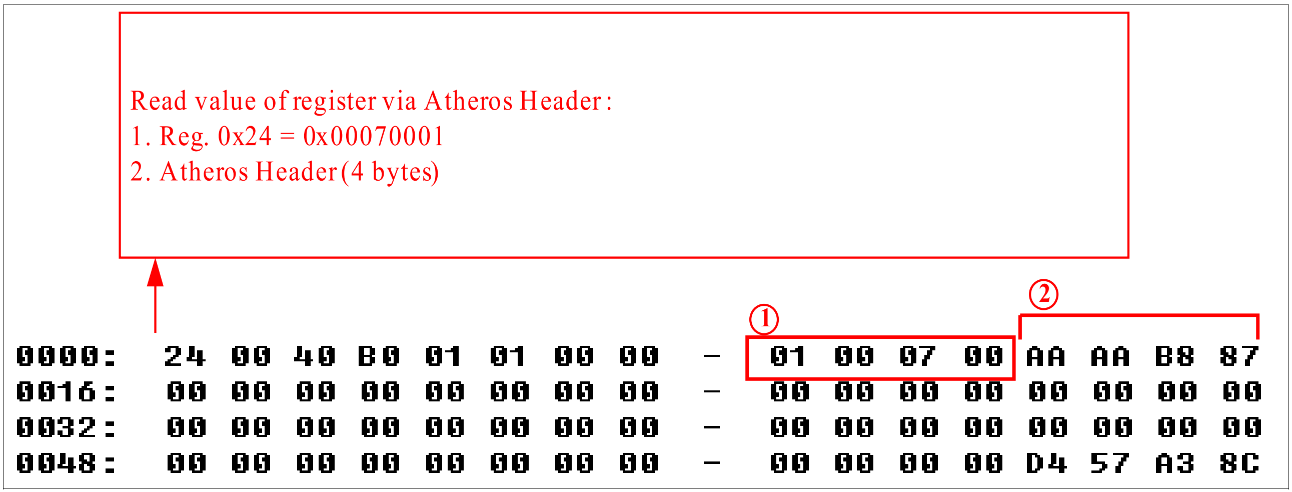
	Address Offset	Value	Description
1	0x620	0x000004f0	Enable CPU port
2	0x98	0x0001aaaa	4-bytes header with 0xaaaa
3	0x9c	0x00000005	Enable Atheros TX/RX header for management only



Read register via Atheros Header :
1. Read start address (Reg0x24)
2. Bit[23:19] Read length : 8 bytes (can support 16 bytes)
3. Bit[28] Read/Write command (0 : write / 1 : read)
4. Atheros Header (4 bytes)
5. Read register command (total 8 bytes)

```
0000:  24 00 40 B0 01 01 00 00 00 00 00 00 AA AA 81 80
0016:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0032:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0048:  00 00 00 00 00 00 00 00 00 00 00 00
```

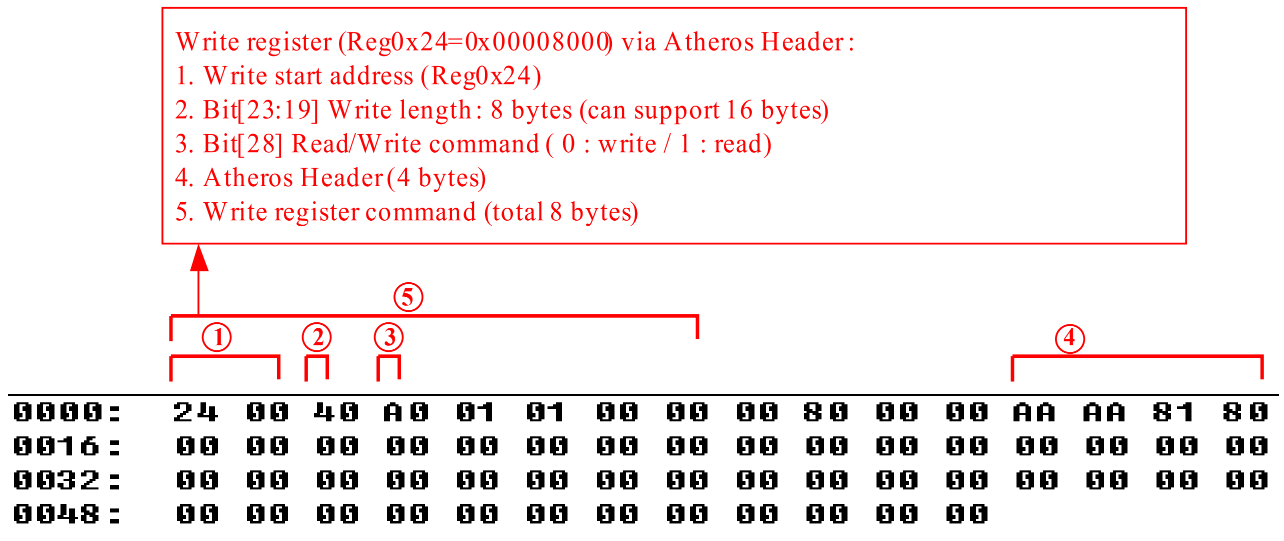
Fig 14-5: Read AR8327/AR8328 register by Atheros header



Read value of register via Atheros Header :
1. Reg. 0x24 = 0x00070001
2. Atheros Header (4 bytes)

```
0000:  24 00 40 B0 01 01 00 00  -  01 00 07 00 AA AA B8 87
0016:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0032:  00 00 00 00 00 00 00 00  -  00 00 00 00 00 00 00 00
0048:  00 00 00 00 00 00 00 00  -  00 00 00 00 D4 57 A3 8C
```

Fig 14-6: Ack for reading AR8327/AR8328 register by Atheros header



Write register (Reg0x24=0x00008000) via Atheros Header :
1. Write start address (Reg0x24)
2. Bit[23:19] Write length : 8 bytes (can support 16 bytes)
3. Bit[28] Read/Write command (0 : write / 1 : read)
4. Atheros Header (4 bytes)
5. Write register command (total 8 bytes)

```
0000:   24 00 40 A0 01 01 00 00 00 80 00 00 AA AA 81 80
0016:   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0032:   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0048:   00 00 00 00 00 00 00 00 00 00 00 00
```

Fig 14-7: Write AR8327/AR8328 register by Atheros header

15 Mirror

AR8327/AR8328 supports the following mirror functions.

- Copy the ingress packets to the mirror port
- Copy the egress packets to the mirror port
- Copy the ingress/egress packets to the mirror port
- Copy the dedicated DA packets to the mirror port

15.1 Port mirror

You can configure the register 0x620 bit[7:4] to assign the port number which packet should be mirrored to. For example, we want to copy the ingress/egress packets of P2 to P1.

	Address Offset	Value	Description
1	0x620	0x00000410	MIRROR_PORT is P1
2	0x678	0x0214007b	Bit 25 ING_MIRROR_EN=1
3	0x984	0x000100c2	Bit 16 EG_MIRROR_EN=1

[ssdk_sh to set MIRROR_PORT=P1] ssdk_sh mirror analyPt set 1

[ssdk_sh to mirror ingress packet of P2] ssdk_sh mirror ptIngress set 2 enable

[ssdk_sh to mirror egress packet of P2] ssdk_sh mirror ptEgress set 2 enable

16 Access Control List

AR8327/8328 provides 96 ACL rules to classify MAC pattern, IPv4 pattern, IPv6 pattern to execute the following actions.

1. Generate interrupt for WOL
2. Rate limit
3. Drop/Forward/Copy/Redirect to CPU
4. Mirror
5. Destination Port
6. Query remapping
7. Lookup by SVID or CVID
8. VID/PRI change for SVID/CVID
9. DSCP remapping

AR8327/8328 also supports Window Pattern for flexible application and Enhance MAC pattern for Q-in-Q application.

16.1 Change VID from 0x10 to 0x20 on P0 and forward to P1

	Address Offset	Value	Description
1	0x30	0x80000302	Bit 2 ACL_EN=1
2	0x660	0x0014017e	To recognize VID, the ingress port must enable 802.1Q mode
3	0x42c	0x00000040	If the egress port can not be set to untouched, the packet will not be modified.
ACL Rule --- MAC Pattern (Please refer to Table2-7)			
4	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x00000000	
	0x410	0x0000000a	Byte 13:12 [11:0] VID=0xa
	0x414	0x00000001	Byte 16 [6:0] Source Port=0x1
	0x400	0x80000001	
ACL Mask --- MAC Pattern Mask (Please refer to Table2-8)			
5	0x404	0x00000000	
	0x408	0x00000000	

	Address Offset	Value	Description
	0x40c	0x00000000	
	0x410	0x00000fff	Byte 13:12 [11:0] VID MASK
	0x414	0x000000c9	RULE VALID=2'b11, VID MASK=1 RULE TYPE=1
	0x400	0x80000101	
ACL Action (Please refer to Table2-6)			
6	0x404	0x00160000	CTAG=0x16
	0x408	0x40001000	DES_PORT=P1, TRANS_CTAG_CHANGE_EN=1
	0x40c	0x00000010	DES_PORT_OVER_EN=1
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000201	

[ssdk_sh to enable ACL function] ssdk_sh acl status set enable

[ssdk_sh to set port0 ingress to 802.1Q fallback mode] ssdk_sh portVlan ingress set 0 check

[ssdk_sh to set port1 egress to unmodified mode] ssdk_sh portVlan egress set 1 unmodified

16.2 IP DSCP to 802.1p remapping

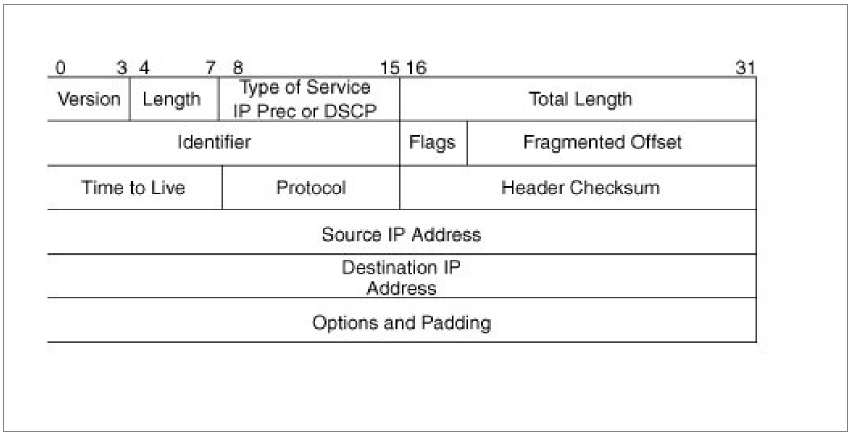


Fig 16-1: Ipv4 header format

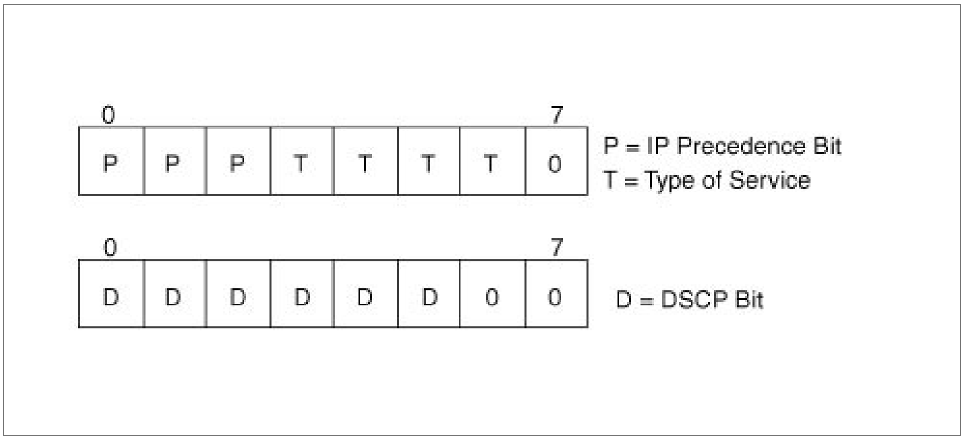


Fig 16-2: Ipv4 header type of service (ToS) field

	Address Offset	Value	Description
1	0x30	0x80000302	Bit 2 ACL_EN=1
2	0x660	0x0014017e	To recognize VID, the ingress port must enable 802.1Q mode
3	0x42c	0x00000040	If the egress port can not be set to untouched, the packet will not be modified.
ACL Rule --- IPv4 Pattern (Please refer to Table2-9)			
4	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x0000b800	Byte 9 [11:0] DSCP=2'b101110
	0x410	0x00000000	
	0x414	0x0000007f	Byte 16 [6:0] Source Port=all port
	0x400	0x80000001	
ACL Mask --- IPv4 Mask (Please refer to Table2-10)			

	Address Offset	Value	Description
5	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x0000ff00	Byte 9 [11:0] DSCP MASK
	0x410	0x00000000	
	0x414	0x000000c2	RULE VALID=2'b11, RULE TYPE=2
	0x400	0x80000101	
ACL Action (Please refer to Table2-6)			
6	0x404	0xe0000000	CTAG PRI=7
	0x408	0x40000200	CTAG_PRI_REMAP_EN=1
	0x40c	0x00000000	
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000201	

16.3 Block invalid Source IP except 10.21.1.53 and 10.21.1.54

	Address Offset	Value	Description
1	0x30	0x80000302	Bit 2 ACL_EN=1
ACL Rule 0 --- IPv4 Pattern (Please refer to Table2-9)			
2	0x404	0x00000000	
	0x408	0x0a150136	SIP=10.21.1.54
	0x40c	0x00000000	
	0x410	0x00000000	
	0x414	0x000000ff	RULE_RESULT_INVERT_EN=1
	0x400	0x80000000	Index-0
ACL Mask --- IPv4 Mask (Please refer to Table2-10)			
3	0x404	0x00000000	
	0x408	0xffffffff	
	0x40c	0x00000000	Byte 9 [11:0] DSCP MASK
	0x410	0x00030000	
	0x414	0x00000002	RULE VALID=2'b00 (start), RULE TYPE=2
	0x400	0x80000100	Index-0

	Address Offset	Value	Description
ACL Action (Please refer to Table2-6)			
4	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x000001c0	ACL_DP_ACT=111
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000200	Index-0
ACL Rule 1 --- IPv4 Pattern (Please refer to Table2-9)			
5	0x404	0x00000000	
	0x408	0x0a150135	SIP=10.21.1.53
	0x40c	0x00000000	
	0x410	0x00000000	
	0x414	0x000000ff	RULE_RESULT_INVERT_EN=1
	0x400	0x80000001	Index-1
ACL Mask --- IPv4 Mask (Please refer to Table2-10)			
6	0x404	0x00000000	
	0x408	0xffffffff	
	0x40c	0x00000000	Byte 9 [11:0] DSCP MASK
	0x410	0x00030000	
	0x414	0x00000082	RULE VALID=2'b10 (end), RULE TYPE=2
	0x400	0x80000101	Index-1
ACL Action (Please refer to Table2-6)			
7	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x000001c0	ACL_DP_ACT=111
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000201	Index-1

16.4 ACL rate limit or ACL counter

AR8327/8328 supports rate limit or counter by per ACL rules.

ACL Action (Please refer to Table2-6)			
1	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x00004200	Bit[78] ACL_RATE_EN Bit[77:73] ACL_RATE_SEL
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000201	

ACL POLICY/COUNTER MODE			
2	0x9f0	0x00000000	1:Counter, 0:Rate Limit
	0x9f4	0x00000002	1:Packet count, 0:Byte count
ACL RATE CONTROL			
3	0xa08	0x00028020	
	0xa0c	0x00040000	Bit[22] 1:Packet, 0:Byte If P0 flow-control is disable
	0xa0c	0x00840000	If P0 flow-control is enable

16.5 ACL interrupt

AR8327/8328 supports interrupt generation by ACL rule.

ACL Action (Please refer to Table2-6)			
1	0x404	0x00000000	
	0x408	0x00000000	
	0x40c	0x00010000	Bit[80] ACL_MATCH_INT_EN
	0x410	0x00000000	
	0x414	0x00000000	
	0x400	0x80000201	

17 Hardware NAT and PPPoE Header Remove/Add

AR8327/8328 provides hardware routing acceleration for IPv4

1. 16 PPPoE session table
2. 128 ARP entry
3. 1K NAT entry
4. 8 Router MAC address/VID table
5. 16 Public IP table

17.1 Static NAT

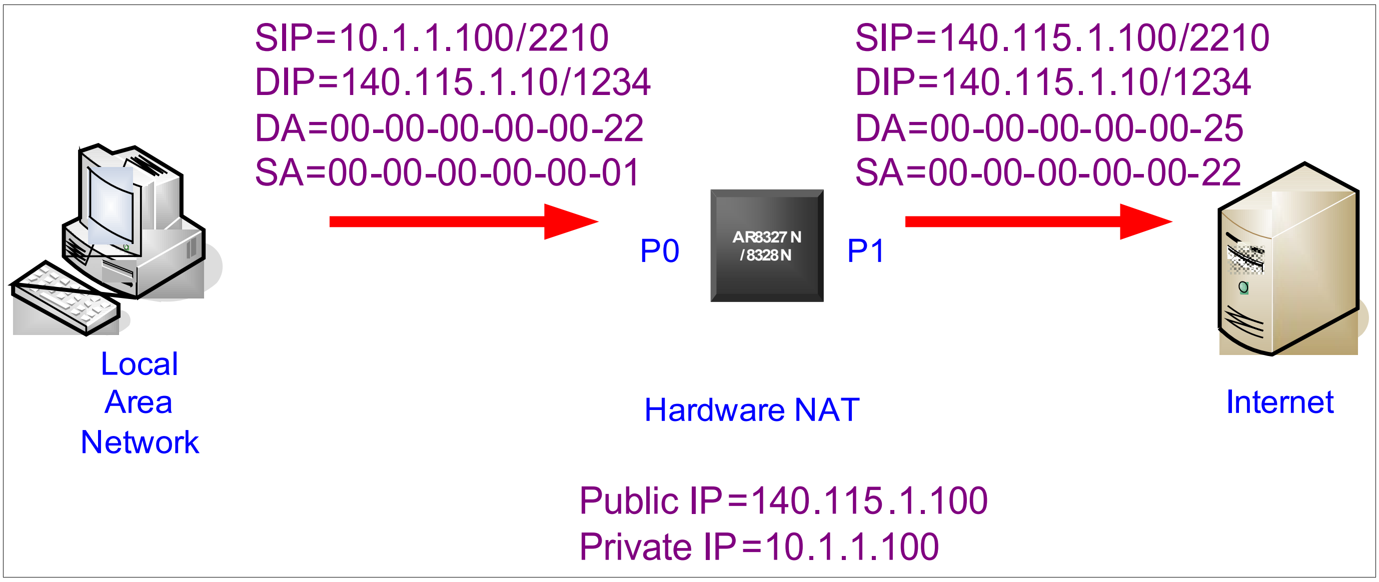


Fig 17-1: HNAT (LAN to WAN)

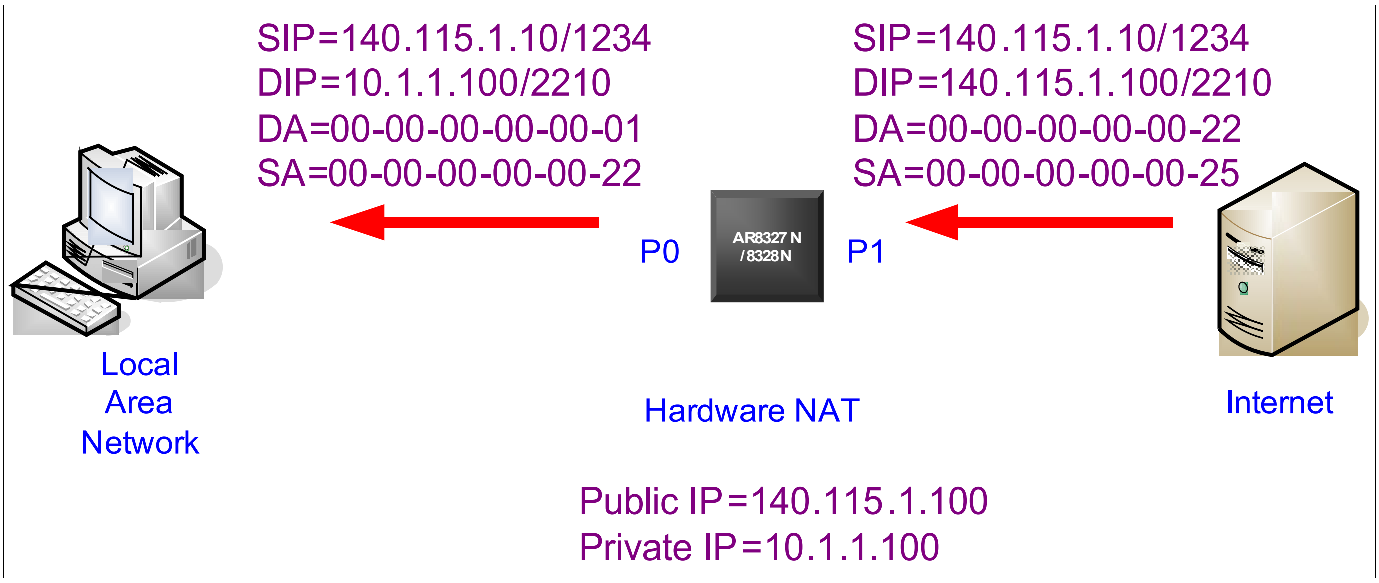
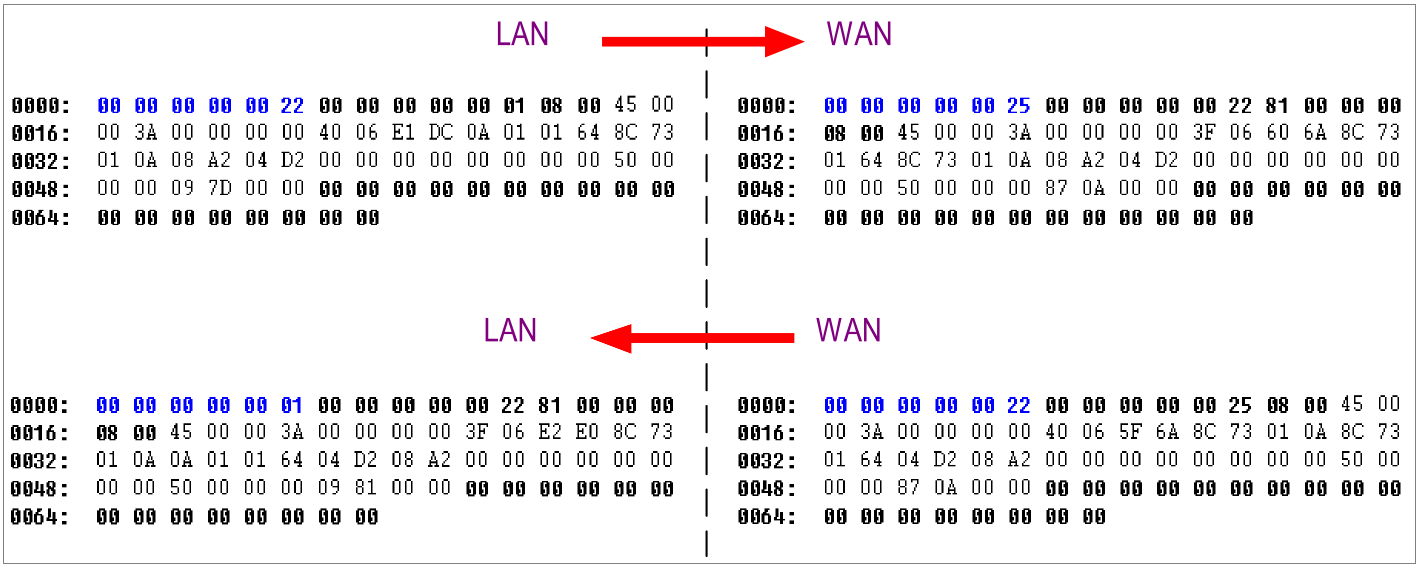


Fig 17-2: HNAT (WAN to LAN)



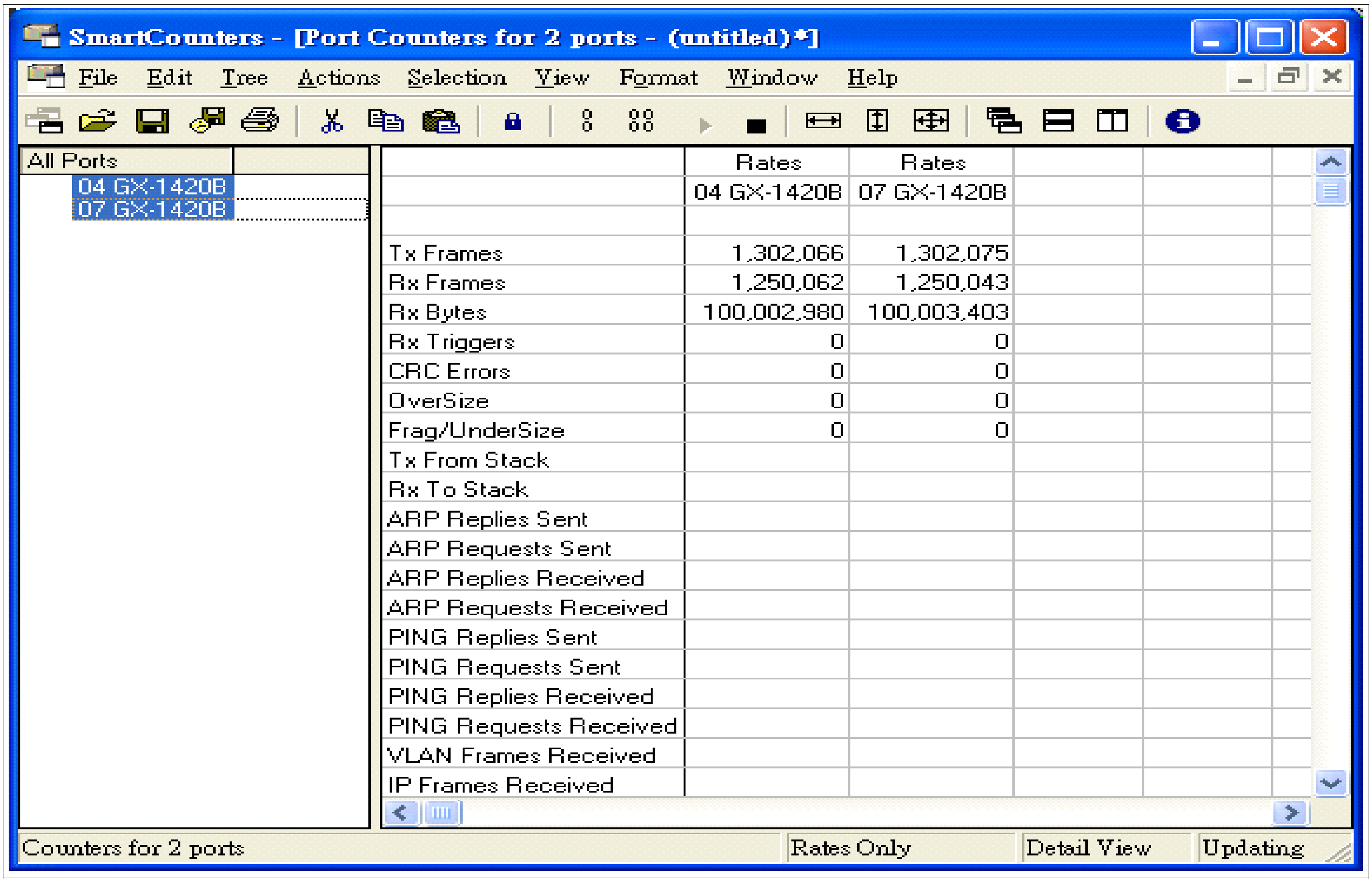
```
0000:  00 00 00 00 00 22 00 00 00 00 00 01 08 00 45 00
0016:  00 3A 00 00 00 00 40 06 E1 DC 0A 01 01 64 8C 73
0032:  01 0A 08 A2 04 D2 00 00 00 00 00 00 00 00 50 00
0048:  00 00 09 7D 00 00 00 00 00 00 00 00 00 00 00 00
0064:  00 00 00 00 00 00 00 00
```

```
0000:  00 00 00 00 00 25 00 00 00 00 00 22 81 00 00 00
0016:  08 00 45 00 00 3A 00 00 00 00 3F 06 60 6A 8C 73
0032:  01 64 8C 73 01 0A 08 A2 04 D2 00 00 00 00 00 00
0048:  00 00 50 00 00 00 87 0A 00 00 00 00 00 00 00 00
0064:  00 00 00 00 00 00 00 00 00 00 00 00
```

```
0000:  00 00 00 00 00 01 00 00 00 00 00 22 81 00 00 00
0016:  08 00 45 00 00 3A 00 00 00 00 3F 06 E2 E0 8C 73
0032:  01 0A 0A 01 01 64 04 D2 08 A2 00 00 00 00 00 00
0048:  00 00 50 00 00 00 09 81 00 00 00 00 00 00 00 00
0064:  00 00 00 00 00 00 00 00
```

```
0000:  00 00 00 00 00 22 00 00 00 00 00 25 08 00 45 00
0016:  00 3A 00 00 00 00 40 06 5F 6A 8C 73 01 0A 8C 73
0032:  01 64 04 D2 08 A2 00 00 00 00 00 00 00 00 50 00
0048:  00 00 87 0A 00 00 00 00 00 00 00 00 00 00 00 00
0064:  00 00 00 00 00 00 00 00
```

Fig 17-3: Packet Edit Engine


SmartCounters - [Port Counters for 2 ports - (untitled)*]
File Edit Tree Actions Selection View Format Window Help
All Ports
04 GX-1420B
07 GX-1420B

	Rates	Rates
	04 GX-1420B	07 GX-1420B
Tx Frames	1,302,066	1,302,075
Rx Frames	1,250,062	1,250,043
Rx Bytes	100,002,980	100,003,403
Rx Triggers	0	0
CRC Errors	0	0
OverSize	0	0
Frag/UnderSize	0	0
Tx From Stack		
Rx To Stack		
ARP Replies Sent		
ARP Requests Sent		
ARP Replies Received		
ARP Requests Received		
PING Replies Sent		
PING Requests Sent		
PING Replies Received		
PING Requests Received		
VLAN Frames Received		
IP Frames Received		

Counters for 2 ports | Rates Only | Detail View | Updating

Fig 17-4: Bi-direction Throughput (TCP minimum packet length=76bytes)

Note: 1,000,000,000 bit/(80*8 +64-bit SFD+96-bit Preamble)=1,250,000 fps

	Address Offset	Value	Description
1	0x30	0x80000005	Bit 2 L3_EN=1
2	0xe38	0x00200a53	Bit 1 HNAT_EN=1 Bit 0 HNAPT_EN=1
Hrouter_Pbased Control			
3	0xe04	0x00000001	P0 check SMAC&SIP for ARP
	0xe08	0x00000001	P0 check SMAC&SIP for Normal Packet
	0xe0c	0x00003f3f	Bit[22:16] IP_SP_UPDATE_EN Bit[14:8] ARP_REP_LEARN_EN Bit[6:0] ARP_REQ_LEARN_EN
Frame Ack Control & UNI_FLOOD_DP			
4	0x210	0x71717171	ARP_REQ_EN, ARP_ACK_EN, DHCP_EN, IGMP_MLD_EN
	0x214	0x03717171	
	0x624	0x7f7f7f7f	
Router MAC address (ACL and PKT Edit)			
5	0x5a900	0x00000022	Bit[47:0] Router MAC address
	0x5a904	0xf0000000	Bit[59:48]VID_LOW
	0x5a908	0x000001ff	Bit[71:60]VID_HIGH
	0x2000	0x00000022	
	0x2004	0xf0000000	
	0x2008	0x000001ff	
Routing SNAT			
6	0xe48	0x8c73010a	DIP=140.115.1.10
	0xe4c	0x00000025	DA=00-00-00-00-00-25
	0xe50	0x10000000	DP to P1
	0xe54	0x00007003	
	0xe58	0x80000032	Add one entry to ARP table
Routing for check SIP&SA; DNAT			
7	0xe48	0x0a010164	SIP=10.1.1.100
	0xe4c	0x00000001	SA=00-00-00-00-00-01
	0xe50	0x00000000	DP to P0
	0xe54	0x00007003	
	0xe58	0x80000032	Add one entry to ARP table

	Address Offset	Value	Description
NAT Function, fill Private IP=10.1.1.100, fill Public IP=140.115.1.100			
8	0x418	0x0000a010	bit[19:0]=Private_IP_BASE_[31:12]
	0xe5c	0x0000a010	
9	0xe48	0x8c730164	Public IP=140.115.1.100
	0xe4c	0x64000018	
	0xe50	0x00008301	
	0xe54	0x00000000	
	0xe58	0x80000022	Add one entry to NAT table
Public IP			
10	0x2100	0x8c730164	
	0x2a000	0x8c730164	
	0x2a040	0x0000ffff	
	0x5aa00	0x8c730164	
	0x2a004	0x0000f0001	
ROUTER EGRESS VLAN MODE			
11	0xc80	0x00000000	

17.2 PPPoE Header Remove/Add

To insert PPPoE header on P1 and remove PPPoE header on P0,

	Address Offset	Value	Description
Routing SNAT			
1	0xe48	0x8c73010a	DIP=140.115.1.10
	0xe4c	0x00000025	DA=00-00-00-00-00-25
	0xe50	0x10000000	DP to P1
	0xe54	0x00007803	bit[107] PPPoE EN bit[106:103] PPPoE INDEX
	0xe58	0x80000032	Add one entry to ARP table
PPPoE Session ID=0x1111			
2	0x214	0x02000000	Bit[25] PPPoE_EN
	0x02200	0x00001111	
	0x5f000	0x00021111	Use to remove PPPoE header

18 LED Control

AR8327/8328 supports LED control rules to program LED behavior. Each AR8327/8328 port has three LEDs and the default behavior of the LEDs are 1000_link_activity (LED0), 100_link_activity (LED1) and 10_link _activity (LED2).

18.1 How to program registers to have one Link Led and one Activity Led?

	Address Offset	Value	Description
1	0x50	0xcf05cf05	PHY0~3/PHY4 LED0 is link led
2	0x54	0xc835c835	PHY0~3/PHY4 LED1 is activity led

19 Production Test

19.1 IEEE Conformance Test

Please follow the below procedures to program 1000Base-T Test Mode.

	Switch Address		Value
1	0x10		0x20241320
	PHY Address	REG Address	Value
2	0x0	0x1d	0xb
	0x0	0x1e	0x5
3	0x0	0x0	0x8140
4	0x0	0x9	0x2200 (Mode 1)
	0x0	0x9	0x4200 (Mode 2)
	0x0	0x9	0x8200 (Mode 4)

The user also can use CmdIssue-S17 to initial the chip for Mode 1.

```
or 10 20241320
## Mode 1
or 3c c01d000b
or 3c c01e0005
or 3c c0008140
or 3c c0092200
or 3c c03d000b
or 3c c03e0005
or 3c c0208140
or 3c c0292200
or 3c c05d000b
or 3c c05e0005
or 3c c0408140
or 3c c0492200
or 3c c07d000b
or 3c c07e0005
or 3c c0608140
or 3c c0692200
or 3c c09d000b
or 3c c09e0005
or 3c c0808140
or 3c c0892200
```

19.2 Loopback

Please follow the below procedures to perform loopback test by external toolkit.

(Pin 1←→Pin 3, Pin 2←→Pin 6)

The user need to configure PHY debug 0xb[15]=0 to disable hibernate mode and debug 0x11[0]=1 to enable PHY external loopback function.

	PHY Address	REG Address	Value
1	0x0	0x1d	0xb
	0x0	0x1e	0x3c40
2	0x0	0x1d	0x11
	0x0	0x1e	0x7553
3	0x0	0x0	0xa100 (100M)
	0x0	0x0	0x8100 (10M)

The user also can use CmdIssue-S17 to initial the chip.

```
or 3c c01d000b
or 3c c01e3c40
or 3c c01d0011
or 3c c01e7553
or 3c c000a100
```

19.3 How to do rambist test?

The following is the procedure to do rambist test.

Item	Address Offset	Value	Description
1	0x0040	0x80700000	Write Bit [31]=1 to begin bist test and should be cleaned to 0 by hardware after test done. Bit [22] is pattern 2 for bist test Bit [21] is pattern 1 for bist test Bit [20] is pattern 0 for bist test
2	0x0040	Read value	If Bit [30] =1, it means one data memory error is found in bist test.
3	0x0044	Read value	Bit [12:0] is the test error address of data memory.
4	0x0000	0x80000000	Write Bit [31]=1 to enable software reset.

19.4 How to do VCT (Virtual Cable Tester) test?

The following is the procedure to do rambist test.

	PHY Address	REG Address	Value	Description
1	0x0	0x16	0x0001	enable VCT MDI0 Pair
	0x0	0x1c	Read value	Bit[9:8]
2	0x0	0x16	0x0101	enable VCT MDI1 Pair
	0x0	0x1c	Read value	Bit[9:8]
3	0x0	0x16	0x0201	enable VCT MDI2 Pair
	0x0	0x1c	Read value	Bit[9:8]
4	0x0	0x16	0x0301	enable VCT MDI3 Pair
	0x0	0x1c	Read value	Bit[9:8]

00 = Valid test, normal cable (no short or open)

10 = Valid test, open in cable

01 = Valid test, short in cable

11 = Test Fail

20 Reference Circuit for SFP Transceiver

To make sure AR8327/8328 SGMII/Serdes interface can work correctly with most SFP transceiver. Please refer to HDK-031 circuit as follow.

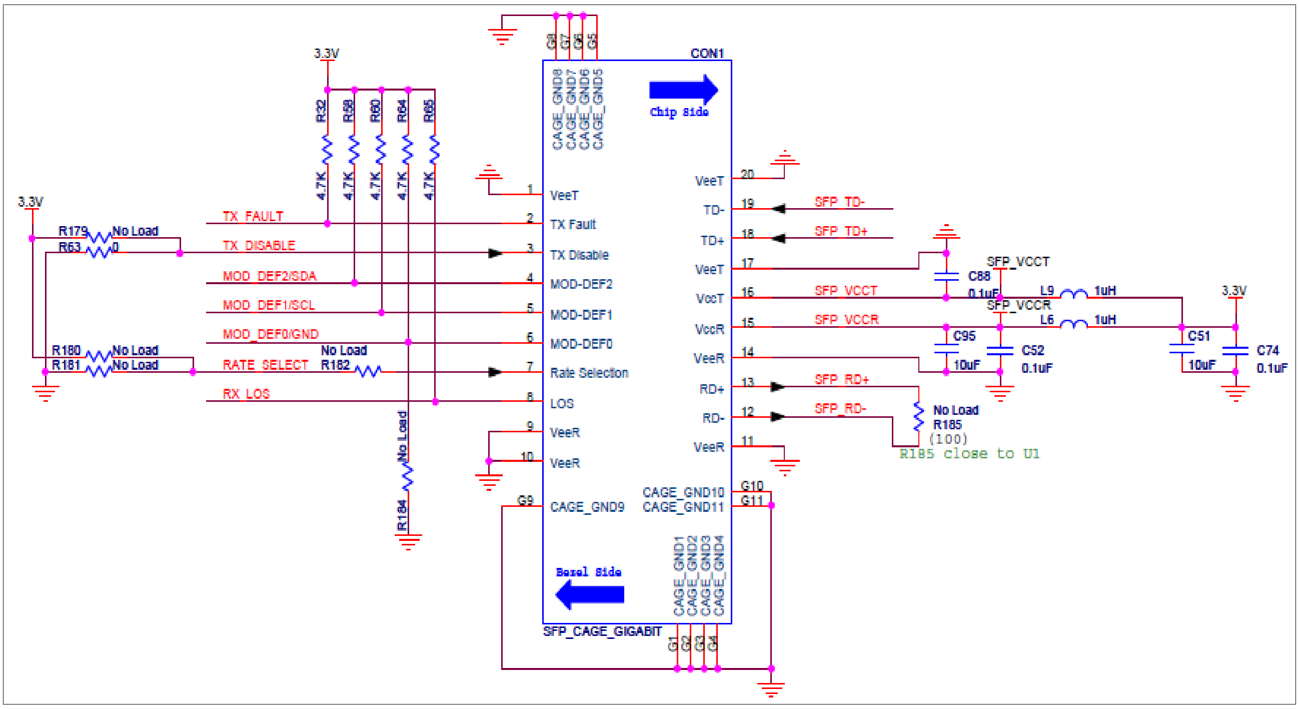


Fig 20-1: Connect SFP Pin3 to GND to enable Transmitter

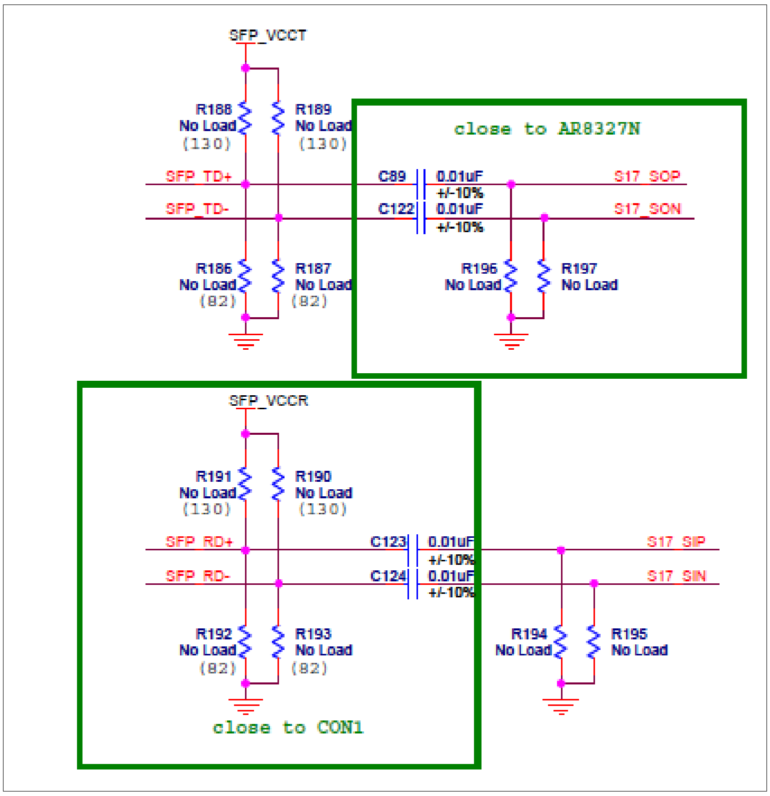


Fig 20-2: Use 0.01uF capacitor to isolate the DC bias voltage