RTL9600

SINGLE-CHIP PON

NOT FOR PUBLIC RELEASE

CPU NIC

Application Note

(CONFIDENTIAL: Development Partners Only)

Rev. 1.0.1
3 May 2013

Realtek Semiconductor Corp.

No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211 Fax: +886-3-577-6047

www.realtek.com

USING THIS DOCUMENT

This document is intended for use by the system engineer when integrating with Realtek Managedd Switch Software. Though every effort has been made to assure that this document is current and accurate, more information may have become available subsequent to the production of this guide. In that event, please contact your Realtek representative for additional information that may help in the development process.

Revision History

Revision	Release Date	Summary
1.0.0	2013/04/25	First release
1.0.1	2013/05/03	Add Rx reason code

Table of Contents

1 Overview

The CPU NIC HW use to interface CPU and switch on RTL9600. This application note descripts how to control GMAC for packet TX/RX process.
For CPU NIC Rx, switch can pass addition rx information to CPU NIC via insert Realtek CPU-Tag. CPU NIC will parse CPU-TAG and translate to “rx_info” software can get this information from “rx_info”. For CPU NIC Tx, software can modify “tx_info” then pass to CPU NIC, it will insert Realtek CPU-Tag and fill tag information according to software “tx_info”.

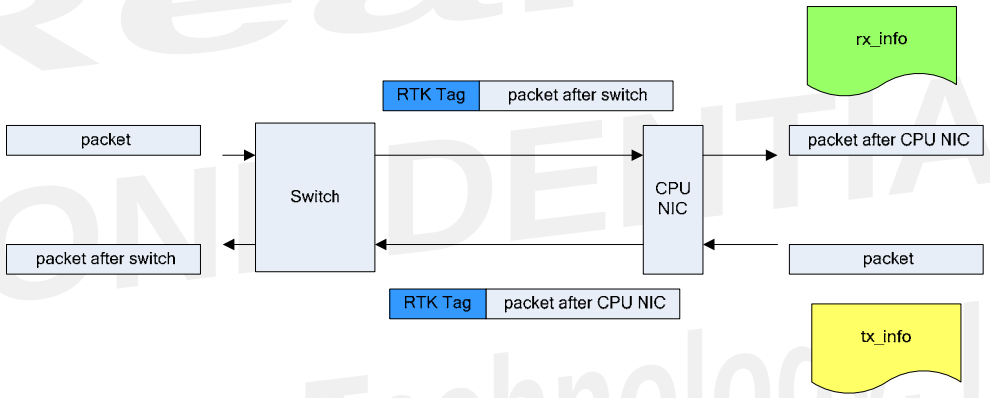

Figure 1 CPU NIC parsing flow

2 CPU NIC Driver

CPU NIC driver, which reside in re8686.c and re8686.h, are located at uClinux-dist/linux-2.6.x/drivers/net.

3 TX procedure

CPU NIC Tx will modify packet content and add CPU tag according to TX descriptor.
TX part hook to linux device driver is “re8670_start_xmit”.

```
dev->hard_start_xmit = re8670_start_xmit
```

GMAC provide some advanced features, user can give more information to TX descriptor in “tx_info”. If user need to do customize please modify tx_info in re8670_start_xmit. Customers can reference function “tx_additional_setting()”.

3.1 tx_info structure

Table 1 Tx info structure

Field Name	Bits	Description
own	1	Don’t care and do not modify
eor	1	Don’t care and do not modify
fs	1	Don’t care and do not modify
ls	1	Don’t care and do not modify
ipcs	1	Always set to 0
l4cs	1	Always set to 0
keep	1	(add information to CPU-tag) Commands switch do keep original packet do not modify packet (remarking, VLAN translation …).
blu	1	(add information to CPU-tag) valid if keep = 0 0b0: Commands switch do L34 lookup and L2 lookup 0b1: Commands switch do L2 lookup only
crc	1	If this bit is set then append CRC at the end of Ethernet frame. For short packet (<64B), this bit must set to 1.
vsel	1	(add information to CPU-tag) Reserved always set to 0
dislrn	1	(add information to CPU-tag) Command switch to disable switch l2 learning for this packet
cputag_ipcs		(add information to CPU-tag) Command switch to do IPv4 checksum offload. The IPv4 checksum will be re-calculate by ASIC.
cputag_l4sel		(add information to CPU-tag) Command switch to do L4 Checksum offload, TCP/UDP/ICMP/IGMP checksum offload.

		The L4 checksum will be re-calculate by ASIC. For IP fragmented packet, SW should not set this field.
data_tength	17	Tx buffer packet size
cputag	1	Force CPU NIC generate Realteck CPU-tag Always set to 1
aspri	1	(add information to CPU-tag) Turn on CPU force priority for this packet
cputag_pri	1	(add information to CPU-tag) CPU force priority (for switch internal priority) This field valid when aspri =1
tx_pppoe_action	2	(add information to CPU-tag) Command switch Tx mac for PPPOE header action. Egress action for PPPoE header whose Ether Type = 0x8864. 00: intact, 01: Add PPPoE header, 10: remove PPPoE header, 11: remarking PPPoE This bit valid only tx_portmask!=0 This field should be set to 00 by software if ″ blu ″ = 0.
tx_pppoe_idx	3	PPPoE index pointed to an entry of the pppoe table in switch. Switch TX-MAC look PPPoE table up by this index to get PPPoE session ID. This field is valid only if ″TX PPPoE Present″ = 01 or 11. (add information to CPU-tag)
efid	1	(add information to CPU-tag) Force assign efid

enhance_fid	3	(add information to CPU-tag) Efid value, this field valid if efid field = 1
tx_vlan_action	2	CPU NIC VLAN tx action. CVLAN action for this egress frame ob00: intact, 0b01: insert VLAN header, 0b10: remove VLAN hdr, 0b11: remarking VID The command action is executed by CPU NIC module. VLAN header information is {vidl,vidh,prio,cfi} VLAN tag maybe modify again by switch decision.
vidl	8	VLAN tag vid[7:0]
vidh	4	VLAN tag vid[11:8]
prio	3	VLAN tag priority
cfi	1	VLAN tag CFI
extspa	3	(add information to CPU-tag) Indicate this packet is come from which extention virtual extension port for switch LUT learning.
tx_portmask	6	(add information to CPU-tag) Command switch force forwarding using this tx_portmask.
l34_keep	1	(add information to CPU-tag) Commands switch only handle L34 part, do not modify L2 part. (ex: do not do 1p remarking, VLAN tag modify)
cputag_psel	1	(add information to CPU-tag) For PON stream id (for GPON)/LLID index(for EPON) selections. This bit valid only tx_portmask including PON port.

		The stream id/LLID index is assign by “tx_dst_stream_id”
ptp	1	(add information to CPU-tag) For PON queue selections. This bit valid only tx_portmask including PON port. The queue id is assign by “tx_dst_stream_id”
tx_dst_stream_id	7	(add information to CPU-tag) If cputag_psel=1: PON stream id(for GPON)/LLID index(for EPON) If ptp=1: PON Queue id Priroty PTP > PSEL

4 Rx procedure

ASIC forward or trap to CPU port will add a CPU-Tag, CPU NIC parsing CPU-tag and packet content translate information into RX descriptor. Software can get some extra information from descriptor. CPU NIC will remove CVLAN tag, and keep VLAN tag information in descriptor.

4.1 Rx Hook

For packet Rx, system provided hook function, the hook API. Here list the hook API.

```
/* Purpose: Used for hook rx callback function
* Parameters:
*    portmask - this callback function want to receive from which ports
*  priority - the priority for callback fun, 0~100, 0: lowest, 100: highest
*  rx   - callback function
*/
int drv_nic_register_rxhook(int portmask,int priority,p2rfunc_t rx)
```

Customer can process packet in rx callback. Here lsit rx callback prototype.

```
typedef int (*p2rfunc_t)(struct re_private *cp, struct sk_buff *skb, struct

rx_info *pRxInfo)
```

The CPU NIC parsing info will keep in rx_info.

The return code of callback function would be follows.

RE8670_RX_STOP	packet have handled and occupied by this callback do not call other callback function
RE8670_RX_CONTINUE	continue to call other callback
RE8670_RX_STOP_SKBNOFREE	stop call other callback and do not free skb

Customer can unregister rx callback by drv_nic_unregister_rxhook() API.

4.2 RX Ring configuration

CPU NIC provide 6 RX ring, customer can configured ring size in re8686.h. Each ring had its own ring size definition, if set to 0 means do not using this ring. Please set the size in order of 2.

Here list an example set ring size of ring1 to 512, ring2~ring6 set to 8.

```
#define RE8670_RX_RING1_SIZE 512
#define RE8670_RX_RING2_SIZE 8
#define RE8670_RX_RING3_SIZE 8
#define RE8670_RX_RING4_SIZE 8
#define RE8670_RX_RING5_SIZE 8

#define RE8670_RX_RING6_SIZE 8
```

The packet internal priority and RX_RING mapping would be configured by RRING_ROUTING1 register.

RRING_ROUTING1 [3:0]	ring assignment for internal priority 0
RRING_ROUTING1 [7:4]	ring assignment for internal priority 1
RRING_ROUTING1 [11:8]	ring assignment for internal priority 2
RRING_ROUTING1 [15:12]	ring assignment for internal priority 3
RRING_ROUTING1 [19:16]	ring assignment for internal priority 4
RRING_ROUTING1 [23:20]	ring assignment for internal priority 5
RRING_ROUTING1 [27:24]	ring assignment for internal priority 6

RRING_ROUTING1 [31:28]	ring assignment for internal priority 7

The RX_RING value 0, 1 and 7 are means mapping to RING1
2 mapping RING2
3 mapping RING3
4 mapping RING4
5 mapping RING5
6 mapping RING6
For example mapping internal priority 7 to RX_RING6, priority 6 to RX_RING5, priority 5 to RX_RING4, priority 4 to RX_RING3, priority 3 to RX_RING2, priority 2~2 to RX_RING1 we can set RRING_ROUTING1 to 0x65432000.
RRING_ROUTING1= 0x65432000

4.3 rx_info structure

Table 2 rx_info structure

Field Name	Bits	Description
own	1	Don't care and do not modify
eor	1	Don't care and do not modify
fs	1	Don't care and do not modify
ls	1	Don't care and do not modify
crcerr	1	CRC error. When set, indicates that a CRC error has occurred on the received packet.
pkttype	4	frame type indication, 0=Ethernet: Not IPv4 nor IPv6. 1=IPv4. 2=IPv4/PPTP, 3=IPv4/ICMP, 4=IPv4/IGMP, 5=IPv4/TCP, 6=IPv4/UDP, 7=IPv6, 8=ICMPv6,

		9=IPv6/TCP, a=IPv6/UDP
ipv4csf	1	When set, indicates L3 checksum failure in IPv4 packet . This field is valid for PktType is equal to 1,2,3,4,5 and 6. This field is 1 for PktType is equal to 0,7,8,9 and a.
l4csf	1	When set, indicates UDP/TCP/ICMP/IGMP checksum failure in IPv4/IPv6 packet . This field is valid for PktType is equal to 3,4,5,6,8,9 and a. This field is 1 for pkttype is equal to 0,1,2 and 7.
rcdf	1	Indicates rx close dma fail.
ipfrag	1	Indicates this is a IP fragment packet
pppoetag	1	Have PPPoE tag
rwt	1	When set, indicates that the received packet length exceeds 1536(0x600) bytes, and stop receive engine.
l3routing	1	This frame is an intended routed frames.
origformat	1	The received frame by NIC is the original packet's format before switch lookup, switch do not do any modified
pctrl	1	PON control frame indication, indicate this packet is PON control packet(OMCI for GPON, OAM for EPON)
data_length	1	This indicates the number of bytes of data on the page pointed by the descriptor. The content of the page should start with no reserve at the start of the page.
cputag	1	Don't care, this bit will always be 1.
ptp_in_cpu_tag_exist	1	Ethenet AV information exists in CPU tag header, GMAC will not remove CPU tag in this case
svlan_tag_exist	1	It indicates the frame format received by NIC. 1= SLAN header exists 0= SVLAN header is inexistent SVLAN TPID current is 0x88a8
pon_stream_id	7	When cputag.spa = r_spa_pon, this field is

		cputag.pon_stream_id. when PTP in cputage exist = 1, this field is PTP sequencedID.
ctagva	1	Tag Available. When set, the received packet is an IEEE802.1Q VLAN TAG (0x8100) available packet.
cvlan_tag	1	If the packet 's TAG (etherType field) is 0x8100, The Ethernet module's MAC extracts four bytes from after source ID, sets TAVA bit to1, and moves the TAG value to this field in Rx descriptor. VIDH: The high 4 bits of a 12-bit VLAN ID. VIDL: The low 8 bits of a 12-bit VLAN ID. PRIO: 3-bit 8-level priority. CFI: Canoethernetal Format Indicator.
src_port_num	5	Receive source port number
dst_port_mask	6	Switch lookup destination port mask. includes 5 extension ports and 1 CPU bit0 : CPU , bit1~5: extension port 0~4
reason	8	Trap Reason
internal_priority	3	NIC get this field from CPU tag header and uses it to make a RX ring# mapping.
ext_port_ttl_1	5	For IP multicast indicate software which extension port need do TTL-1 procedure

4.4 Rx reason code

Reason	item	Sub item	Comment
0	Normal		
1~59	NAT		NAT reason
60	NAT	PPPOE bind/Multicast with TTL<=1	Binding packet with PPPOE and NON-IP packet(trap)/IP multicast routing packet with TTL <=1(drop)

61	Reserved		
62	NAT	LLC	NAT L2TRAN=1 but packet with SNAP header
63	NAT	L2TABLE	L2 entry not valid from ALE34's index
64~127	ACL		Trap by acl rule n, then the reason will be 64+n, drop by ACL is 64
128	DoS	daeqsa	
129		land_attack	
130		blat_attack	
131		synfin_scan	
132		xmas_scan	
133		null_scan	
134		syn1024	
135		tcp_shorthdr	
136		tcpfragerror	
137		icmpfragment	
138		pingofdeath	
139		udpbomb	
140		synwithdata	
141		synflood	
142		finflood	
143		icmpflood	
144	CVLAN	policing	
145		eg mask	
146		ig drop	
147		type_check	
148	SVLAN	Untag	
149		Unmatch	
150		drop	
151		eg mask	
152	rlpp		
153	rldp		
154	lldp		
155	other_rldp		
156	force/dsl trap		
157	pktlen		
158	spanning tree	TX	
159		RX	
160	RMA	IEEE-00	Bridge Group Address

161		IEEE-01	IEEE Std 802.3, 1988 Edition, Full Duplex PAUSE operation
162		IEEE-02	IEEE Std 802.3ad Slow Protocols- Multicast address
163		IEEE-03	IEEE Std 802.1X PAE address
164		IEEE-04	Reserved(01-80-C2-00-00-04~07 | 01-80-C2-00-00-09~0C | 01-80-C2-00-00-0F)
165		IEEE-08	Provider Bridge Group Address
166		IEEE-0D	Provider Bridge GVRP Address
167		IEEE-0E	IEEE Std. 802.1AB Link Layer Discovery Protocol multicast address
168		IEEE-10	All LANs Bridge Management Group Address
169		IEEE-11	Load Server Generic Address
170		IEEE-12	Loadable Device Generic Address
171		IEEE-13	Reserved(01-80-C2-00-00-13~17 | 01-80-C2-00-00-19 | 01-80-C2-00-00-1B~1F)
172		IEEE-18	Generic Address for All Manager Stations
173		IEEE-1A	Generic Address for All Agent Stations
174		IEEE-20	GMRP Address
175		IEEE-21	GVRP address
176		IEEE-22	Undefined GARP address(01-80-C2-00-00-22 ~ 01-80-C2-00-00-2F)
177		CISCO-CC	CDP(Cisco Discovery Protocol)
178		CISCO-CD	Cisco Shared Spanning Tree Protocol
179	Reserved		
~			
190			
191	CPU	drop by extension	drop when destination is to cpu and extension port is empty
192	12	learning limit per port	
193		learning limit system	
194	802.1x	trap/drop	
195		egress pm	
196	unkn_sa		
197	unma_sa		
198	link		

199	port_isolation		
200	storm control	bcast	
201		kn_mcast	
202		unkn_ucast	
203		unkn_mcast	
204	unknown da	uc	
205		l2mc	
206		ipv4mc	
207		ipv6mc	
208	mpcp		
209	ds_omci		
210	classification		
211	oam		
212	sa block		
213	da block		
214	flood		
215	igmp		
216	mc_data		
217	Reserved		
218	mirr_iso		
219	egress_drop		
220	src_blk		
221	tx_mir		
222	rx_mir		
223	l2 fwd		
224	mtu_exceed	bind L2	
225		IPMC Route/Bridge	
226	wan_drop		drop because wan to port(ext)/port(ext) to wan is not allowed
227	snap direct tx		direct tx snap packet with pppoe action
228	Reserved		
~			
239			
240	ptp		PTP trap
241			PTP rx mirror
242	PTP Reserved		
~			

252			
253	ptp		ptp latch egress timestamp and tx mirror to CPU
254	PTP Reserved		
255			

Realtek Semiconductor Corp.

Headquarters

No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211 Fax: +886-3-577-6047

www.realtek.com