


Commit a8cd67fa authored 2 months ago by  [coolsnowwolf](#)

## add ASUS RT-AC58U in IPQ40xx

parent [d1f8ddaf](#)   No related merge requests found

Showing 18 changed files ▾ with 1012 additions and 61 deletions

▼  config/Config-images.in

```

...      ...      @@ -18,6 +18,7 @@ menu "Target Images"
18      18      default TARGET_INITRAMFS_COMPRESSION_LZMA if TARGET_ramips
19      19      default TARGET_INITRAMFS_COMPRESSION_LZMA if TARGET_apm821xx
20      20      default TARGET_INITRAMFS_COMPRESSION_LZMA if TARGET_mpc85xx
21      21      + default TARGET_INITRAMFS_COMPRESSION_LZMA if TARGET_ipq40xx
22      22      default TARGET_INITRAMFS_COMPRESSION_NONE
23      23      depends on TARGET_ROOTFS_INITRAMFS
24      24      help
...      ...

```

▼  target/linux/ipq40xx/base-files/etc/board.d/01 leds

```

...      ...      @@ -15,12 +15,22 @@ asus,rt-ac58u)
15      15      ucidef_set_led_wlan "wlan2g" "WLAN2G" "${boardname}:blue:wlan2g" "phy0tpt"
16      16      ucidef_set_led_wlan "wlan5g" "WLAN5G" "${boardname}:blue:wlan5g" "phy1tpt"
17      17      ucidef_set_led_usbport "usb" "USB" "${boardname}:blue:usb" "usb1-port1" "usb2-port1" "usb3-port1" "usb4-
18      18      port1"
19      19      - ucidef_set_led_netdev "wan" "WAN" "${boardname}:blue:wan" "eth1"
20      20      + ucidef_set_led_switch "wan" "WAN" "${boardname}:blue:wan" "switch0" "0x20"
21      21      ucidef_set_led_switch "lan" "LAN" "${boardname}:blue:lan" "switch0" "0x1e"
22      22      ;;
23      23      + asus,rt-acrh17)
24      24      + ucidef_set_led_default "status" "STATUS" "${boardname}:blue:status" "1"
25      25      + ucidef_set_led_wlan "wlan2g" "WLAN2G" "${boardname}:blue:wlan2g" "phy1tpt"
26      26      + ucidef_set_led_wlan "wlan5g" "WLAN5G" "${boardname}:blue:wlan5g" "phy0tpt"
27      27      + ucidef_set_led_switch "wan" "WAN(blue)" "${boardname}:blue:wan" "switch0" "0x20"
28      28      + ucidef_set_led_switch "lan1" "LAN1" "${boardname}:blue:lan1" "switch0" "0x02"
29      29      + ucidef_set_led_switch "lan2" "LAN2" "${boardname}:blue:lan2" "switch0" "0x04"
30      30      + ucidef_set_led_switch "lan3" "LAN3" "${boardname}:blue:lan3" "switch0" "0x08"
31      31      + ucidef_set_led_switch "lan4" "LAN4" "${boardname}:blue:lan4" "switch0" "0x10"
32      32      + ;;
33      33      avm,fritzbox-4040)
34      34      ucidef_set_led_wlan "wlan" "WLAN" "fritz4040:green:wlan" "phy0tpt" "phy1tpt"
35      35      - ucidef_set_led_netdev "wan" "WAN" "fritz4040:green:wan" "eth1"
36      36      + ucidef_set_led_switch "wan" "WAN" "fritz4040:green:wan" "switch0" "0x20"
37      37      ucidef_set_led_switch "lan" "LAN" "fritz4040:green:lan" "switch0" "0x1e"
38      38      ;;
39      39      glinet,gl-b1300)
...      ...

```

▼  target/linux/ipq40xx/base-files/etc/board.d/02 network

```

...      ...      @@ -16,27 +16,32 @@ asus,rt-ac58u)
16      16      CI_UBIPART=UBI_DEV
17      17      wan_mac_addr=$(mtd_get_mac_binary_ubi Factory 20486)
18      18      lan_mac_addr=$(mtd_get_mac_binary_ubi Factory 4102)
19      19      - ucidef_set_interfaces_lan_wan "eth0" "eth1"
20      20      ucidef_add_switch "switch0" \
21      21      - "0u@eth0" "1:lan" "2:lan" "3:lan" "4:lan"
22      22      + "0t@eth0" "1:lan" "2:lan" "3:lan" "4:lan" "5:wan"
23      23      ucidef_set_interface_macaddr "lan" "$lan_mac_addr"
24      24      ucidef_set_interface_macaddr "wan" "$wan_mac_addr"
25      25      ;;
26      26      - avm,fritzbox-4040)
27      27      - ucidef_set_interfaces_lan_wan "eth0" "eth1"
28      28      + asus,rt-acrh17)
29      29      + CI_UBIPART=UBI_DEV
30      30      + lan_mac_addr=$(mtd_get_mac_binary_ubi Factory 4102)
31      31      + wan_mac_addr=$(mtd_get_mac_binary_ubi Factory 36870)
...      ...

```

```

27 28         uci_def_add_switch "switch0" \
28 -             "0u@eth0" "1:lan" "2:lan" "3:lan" "4:lan"
29 +             "0t@eth0" "1:lan" "2:lan" "3:lan" "4:lan" "5:wan"
30 +         uci_def_set_interface_macaddr "lan" "$lan_mac_addr"
31 +         uci_def_set_interface_macaddr "wan" "$wan_mac_addr"
29 32         ;;
30 - compex,wpj428)
31 -         uci_def_set_interface_lan "eth0 eth1"
33 + avm,fritzbox-4040)
34 +         uci_def_add_switch "switch0" \
35 +             "0t@eth0" "1:lan" "2:lan" "3:lan" "4:lan" "5:wan"
32 36         ;;
33 37 glinet,gl-b1300)
34 -         uci_def_set_interfaces_lan_wan "eth0" "eth1"
35 38         uci_def_add_switch "switch0" \
36 -             "0u@eth0" "3:lan" "4:lan"
39 +             "0t@eth0" "3:lan" "4:lan" "5:wan"
37 40         ;;
41 + compex,wpj428|\
38 42 openmesh,a42)
39 -         uci_def_set_interfaces_lan_wan "eth1" "eth0"
43 +         uci_def_add_switch "switch0" \
44 +             "0t@eth0" "4:lan" "5:wan"
40 45         ;;
41 46
42 47 meraki,mr33)
... ..

```

#### target/linux/ipq40xx/base-files/etc/hotplug.d/firmware/11-ath10k-caldata

```

... .. @@ -112,6 +112,7 @@ case "$FIRMWARE" in
112 112         ;;
113 113 "ath10k/pre-cal-ahb-a000000.wifi.bin")
114 114         case "$board" in
115 +         asus,rt-acrh17|\
115 116         asus,rt-ac58u)
116 117             CI_UBIPART=UBI_DEV
117 118             ath10kcal_ubi_extract "Factory" 4096 12064
... .. @@ -158,6 +159,14 @@ case "$FIRMWARE" in
158 159         ;;
159 160         esac
160 161         ;;
162 + "ath10k/pre-cal-pci-0000:01:00.0.bin")
163 +         case "$board" in
164 +         asus,rt-acrh17)
165 +             CI_UBIPART=UBI_DEV
166 +             ath10kcal_ubi_extract "Factory" 36864 12064
167 +             ;;
168 +         esac
169 +         ;;
161 170 *)
162 171         exit 1
163 172         ;;
... ..

```

#### target/linux/ipq40xx/base-files/etc/hotplug.d/net/16\_adjust\_network\_0 → 100644

```

1 + #!/bin/sh
2 +
3 + [ -f /lib/adjust_network.sh ] && {
4 +     . /lib/adjust_network.sh
5 +
6 +     adjust_eth_queue
7 + }

```

#### target/linux/ipq40xx/base-files/etc/init.d/adjust\_network\_0 → 100755

```

1 + #!/bin/sh /etc/rc.common
2 + # Copyright (C) 2006-2011 OpenWrt.org
3 +
4 + START=11
5 + STOP=98
6 +
7 + adjust_smp_affinity() {
8 +     test -f /lib/adjust_network.sh && {
9 +         . /lib/adjust_network.sh
10 +
11 +         adjust_edma_smp_affinity
12 +         adjust_radio_smp_affinity
13 +     }

```

```

14 + }
15 +
16 + boot() {
17 +     adjust_smp_affinity
18 + }

```

target/linux/ipq40xx/base-files/lib/adjust\_network.sh\_0 → 100644

```

1 + #!/bin/sh
2 + # this scripts is used for adjust cpu's choice of interrupts.
3 + #
4 +
5 + #####
6 + # Adjust smp_affinity of edma
7 + # Globals:
8 + #     None
9 + # Arguments:
10 + #     None
11 + # Returns:
12 + #     None
13 + # Remark:
14 + #     execute only once on start-up.
15 + #####
16 + adjust_edma_smp_affinity() {
17 +     grep -q edma_eth_ /proc/interrupts || return 0
18 +     local nr=`cat /proc/cpuinfo | grep processor | wc -l`
19 +     local cpu=0
20 +     local tx_irq_num
21 +
22 +     for tx_num in `seq 0 1 15` ; do
23 +         cpu=`printf "%x" $((1<<((tx_num/4+3)%nr)))`
24 +         tx_irq_num=`grep -m1 edma_eth_tx$tx_num /proc/interrupts | cut -d ':' -f 1 | tail -n1 | tr -d ' '`
25 +         [ -n "$tx_irq_num" ] && echo $cpu > /proc/irq/$tx_irq_num/smp_affinity
26 +     done
27 +
28 +     for rx_num in `seq 0 1 7` ; do
29 +         cpu=`printf "%x" $((1<<((rx_num/2)%nr)))`
30 +         rx_irq_num=`grep -m1 edma_eth_rx$rx_num /proc/interrupts | cut -d ':' -f 1 | tail -n1 | tr -d ' '`
31 +         [ -n "$rx_irq_num" ] && echo $cpu > /proc/irq/$rx_irq_num/smp_affinity
32 +     done
33 + }
34 +
35 + #####
36 + # Adjust smp_affinity of ath10k for 2G and 5G
37 + # Globals:
38 + #     None
39 + # Arguments:
40 + #     None
41 + # Returns:
42 + #     None
43 + # Remark:
44 + #     execute only once on start-up.
45 + #####
46 + adjust_radio_smp_affinity() {
47 +     local irqs=`grep -E 'ath10k' /proc/interrupts | cut -d ':' -f 1 | tr -d ' '`
48 +     local nr=`cat /proc/cpuinfo | grep processor | wc -l`
49 +     local idx=2
50 +
51 +     for irq in $irqs; do
52 +         cpu=`printf "%x" $((1<<((idx)%nr)))`
53 +         echo $cpu > /proc/irq/$irq/smp_affinity
54 +         idx=$((idx+1))
55 +     done
56 + }
57 +
58 + #####
59 + # Adjust queue of eth
60 + # Globals:
61 + #     None
62 + # Arguments:
63 + #     None
64 + # Returns:
65 + #     None
66 + # Remark:
67 + #     Each network reboot needs to be executed.
68 + #####
69 + adjust_eth_queue() {
70 +     local nr=`cat /proc/cpuinfo | grep processor | wc -l`
71 +     local cpu=`printf "%x" $((1<<(nr-1)))`

```

```

72 +
73 +     for epath in /sys/class/net/eth[0-9]*; do
74 +         test -e $epath || break
75 +         echo $epath | grep -q "\." && continue
76 +         eth=`basename $epath`
77 +         for exps in /sys/class/net/$eth/queues/rx-[0-9]*/rps_cpus; do
78 +             test -e $exps || break
79 +             echo $cpu > $exps
80 +             echo 256 > `dirname $exps`/rps_flow_cnt
81 +         done
82 +         which ethtool >/dev/null 2>&1 && ethtool -K $eth gro off
83 +     done
84 +
85 +     echo 1024 > /proc/sys/net/core/rps_sock_flow_entries
86 + }

```

#### target/linux/ipq40xx/base-files/lib/preinit/05\_set\_iface\_mac\_ipq40xx.sh

```

... .. @@ -4,6 +4,12 @@
4 4
5 5 preinit_set_mac_address() {
6 6     case $(board_name) in
7 7 +     asus,rt-acrh17|\
8 8 +     asus,rt-ac58u)
9 9 +         CI_UBIPART=UBI_DEV
10 10 +         mac=$(mtd_get_mac_binary_ubi Factory 4102)
11 11 +         ifconfig eth0 hw ether $mac 2>/dev/null
12 12 +         ;;
13 13     meraki,mr33)
14 14         mac_lan=$(get_mac_binary "/sys/bus/i2c/devices/0-0050/eprom" 102)
15 15         [ -n "$mac_lan" ] && ip link set dev eth0 address "$mac_lan"
... ..

```

#### target/linux/ipq40xx/base-files/lib/upgrade/platform.sh

```

... .. @@ -4,37 +4,157 @@ REQUIRE_IMAGE_METADATA=1
4 4 RAMFS_COPY_BIN='fw_printenv fw_setenv'
5 5 RAMFS_COPY_DATA='/etc/fw_env.config /var/lock/fw_printenv.lock'
6 6
7 7 - platform_check_image() {
8 8     case "$(board_name)" in
9 9     asus,rt-ac58u)
10 10         CI_UBIPART="UBI_DEV"
11 11         local ubidev=$(nand_find_ubi $CI_UBIPART)
12 12         local asus_root=$(nand_find_volume $ubidev jffs2)
13 13 + ubi_kill_if_exist() {
14 14 +     local ubidev="$( nand_find_ubi "$CI_UBIPART" )"
15 15 +     local c_ubivol="$( nand_find_volume $ubidev $1 )"
16 16 +     umount -f /dev/$c_ubivol 2>/dev/null
17 17 +     [ "$c_ubivol" ] && ubirmvol /dev/$ubidev -N $1 || true
18 18 +     echo "Partition $1 removed."
19 19 + }
20 20
21 21 - [ -n "$asus_root" ] || return 0
22 22 + # idea from @981213
23 23 + # Tar sysupgrade for ASUS RT-AC82U/RT-AC58U
24 24 + # An ubi repartition is required due to the strange partition table created by Asus.
25 25 + # We create all the factory partitions to make sure that the U-boot tftp recovery still works.
26 26 + # The reserved kernel partition size should be enough to put the factory image in.
27 27 + asus_nand_upgrade_tar() {
28 28 +     local kpart_size="$1"
29 29 +     local tar_file="$2"
30 30
31 31 -     cat << EOF
32 32 -     jffs2 partition is still present.
33 33 -     There's probably no space left
34 34 -     to install the filesystem.
35 35 +     local board_dir=$(tar tf $tar_file | grep -m 1 '^sysupgrade-./$')
36 36 +     board_dir=${board_dir%/}
37 37
38 38 -     You need to delete the jffs2 partition first:
39 39 -     # ubirmvol /dev/ubi0 --name=jffs2
40 40 +     local kernel_length=$(tar xf $tar_file ${board_dir}/kernel -O | wc -c) 2> /dev/null`
41 41 +     local rootfs_length=$(tar xf $tar_file ${board_dir}/root -O | wc -c) 2> /dev/null`
42 42
43 43 -     Once this is done. Retry.
44 44 -     EOF
45 45 +     local mtdnum="$( find_mtd_index "$CI_UBIPART" )"
46 46 +     if [ ! "$mtdnum" ]; then
47 47 +         echo "cannot find ubi mtd partition $CI_UBIPART"

```

```

26         return 1
27     ;;
28     esac
34     fi
35     +
36     local ubidev="$( nand_find_ubi "$CI_UBIPART" )"
37     if [ ! "$ubidev" ]; then
38         ubiattach -m "$mtdnum"
39         sync
40         ubidev="$( nand_find_ubi "$CI_UBIPART" )"
41     fi
42     +
43     if [ ! "$ubidev" ]; then
44         echo "cannot find ubi device $CI_UBIPART"
45         return 1
46     fi
47     +
48     local root_ubivol="$( nand_find_volume $ubidev rootfs )"
49     # remove ubiblock device of rootfs
50     local root_ubiblk="ubiblock${root_ubivol:-3}"
51     if [ "$root_ubivol" -a -e "/dev/$root_ubiblk" ]; then
52         echo "removing $root_ubiblk"
53         if ! ubiblock -r /dev/$root_ubivol; then
54             echo "cannot remove $root_ubiblk"
55             return 1;
56         fi
57     fi
58     +
59     ubi_kill_if_exist linux
60     ubi_kill_if_exist linux2
61     ubi_kill_if_exist rootfs
62     ubi_kill_if_exist rootfs_data
63     ubi_kill_if_exist jffs2
64     +
65     ubimkvol /dev/$ubidev -N linux -s $kpart_size
66     ubimkvol /dev/$ubidev -N linux2 -s $kpart_size
67     ubimkvol /dev/$ubidev -N jffs2 -s 2539520
68     ubimkvol /dev/$ubidev -N rootfs -s $rootfs_length
69     ubimkvol /dev/$ubidev -N rootfs_data -m
70     +
71     local kern_ubivol="$(nand_find_volume $ubidev $CI_KERNELPART)"
72     echo "Kernel at $kern_ubivol.Writing..."
73     tar xf $tar_file ${board_dir}/kernel -O | \
74         ubiupdatevol /dev/$kern_ubivol -s $kernel_length -
75     echo "Done."
76     +
77     local root_ubivol="$(nand_find_volume $ubidev rootfs)"
78     echo "Rootfs at $root_ubivol.Writing..."
79     tar xf $tar_file ${board_dir}/root -O | \
80         ubiupdatevol /dev/$root_ubivol -s $rootfs_length -
81     echo "Done."
82     +
83     nand_do_upgrade_success
84 + }
85 +
86 + # idea from @981213
87 + # Factory image sysupgrade for ASUS RT-AC82U/RT-AC58U
88 + # Delete all the partitions we created before, create "linux" partition and write factory image in.
89 + # Skip the first 64bytes which is an uImage header to verify the firmware.
90 + # The kernel partition size should be the original one.
91 + asus_nand_upgrade_factory() {
92 +     local kpart_size="$1"
93 +     local fw_file="$2"
94     +
95     local mtdnum="$( find_mtd_index "$CI_UBIPART" )"
96     if [ ! "$mtdnum" ]; then
97         echo "cannot find ubi mtd partition $CI_UBIPART"
98         return 1
99     fi
100 +
101     local ubidev="$( nand_find_ubi "$CI_UBIPART" )"
102     if [ ! "$ubidev" ]; then
103         ubiattach -m "$mtdnum"
104         sync
105         ubidev="$( nand_find_ubi "$CI_UBIPART" )"
106     fi
107     +
108     if [ ! "$ubidev" ]; then
109         echo "cannot find ubi device $CI_UBIPART"

```

```

110 +         return 1
111 +     fi
112 +
113 +     local root_ubivol="$( nand_find_volume $ubidev rootfs )"
114 +     # remove ubiblock device of rootfs
115 +     local root_ubiblk="ubiblock${root_ubivol:-3}"
116 +     if [ "$root_ubivol" -a -e "/dev/$root_ubiblk" ]; then
117 +         echo "removing $root_ubiblk"
118 +         if ! ubiblock -r /dev/$root_ubivol; then
119 +             echo "cannot remove $root_ubiblk"
120 +             return 1;
121 +         fi
122 +     fi
123 +
124 +     ubi_kill_if_exist linux
125 +     ubi_kill_if_exist linux2
126 +     ubi_kill_if_exist rootfs
127 +     ubi_kill_if_exist rootfs_data
128 +     ubi_kill_if_exist jffs2
129 +
130 +     ubimkvol /dev/$ubidev -N linux -s $kpart_size
131 +
132 +     local kern_ubivol="$(nand_find_volume $ubidev $CI_KERNELPART)"
133 +     echo "Asus linux at $kern_ubivol.Writing..."
134 +     ubiupdatevol /dev/$kern_ubivol --skip=64 $fw_file
135 +     echo "Done."
136 +
137 +     umount -a
138 +     reboot -f
139 + }
140 +
141 + platform_check_image() {
29 142     return 0;
30 143 }
31 144
32 145 platform_do_upgrade() {
33 146     case "$(board_name)" in
147 +     asus,rt-acrh17|\
34 148     asus,rt-ac58u)
149 +         local magic=$(get_magic_long "$1")
35 150         CI_UBIPART="UBI_DEV"
36 151         CI_KERNELPART="linux"
37 152         nand_do_upgrade "$1"
153 +         if [ "$magic" == "27051956" ]; then
154 +             echo "Got Asus factory image."
155 +             asus_nand_upgrade_factory 50409472 "$1"
156 +         else
157 +             asus_nand_upgrade_tar 20951040 "$1"
158 +         fi
38 158         ;;
39 159     openmesh,a42)
40 160         PART_NAME="inactive"
... 161     @@ -52,10 +172,6 @@ platform_do_upgrade() {
52 172
53 173 platform_nand_pre_upgrade() {
54 174     case "$(board_name)" in
55 175     asus,rt-ac58u)
56 176         CI_UBIPART="UBI_DEV"
57 177         CI_KERNELPART="linux"
58 178         ;;
59 179     meraki,mr33)
60 180         CI_KERNELPART="part.safe"
61 181         ;;
... 182     ...

```

#### target/linux/ipq40xx/files-4.14/arch/arm/boot/dts/qcom-ipq4018-a42.dts

```

... 207     @@ -207,19 +207,7 @@
208 208     };
209 209     &gmac0 {
210 210         - qcom,phy_mdio_addr = <4>;
211 211         - qcom,poll_required = <1>;
212 212         - qcom,forced_speed = <1000>;
213 213         - qcom,forced_duplex = <1>;
214 214         - vlan_tag = <2 0x20>;
215 215     - };
216 216     -
217 217     - &gmac1 {

```

```

218 -     qcom,phy_mdio_addr = <3>;
219 -     qcom,poll_required = <1>;
220 -     qcom,forced_speed = <1000>;
221 -     qcom,forced_duplex = <1>;
222 -     vlan_tag = <1 0x10>;
210 +     vlan_tag = <1 0x31>;
223 211 };
224 212
225 213 &usb2_hs_phy {
...

```

▼  [target/linux/ipq40xx/files-4.14/arch/arm/boot/dts/qcom-ipq4019-rt-acrh17.dts](#) 0 → 100644

```

1 + /* Copyright (c) 2015, The Linux Foundation. All rights reserved.
2 + *
3 + * Permission to use, copy, modify, and/or distribute this software for any
4 + * purpose with or without fee is hereby granted, provided that the above
5 + * copyright notice and this permission notice appear in all copies.
6 + *
7 + * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
8 + * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
9 + * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
10 + * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
11 + * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
12 + * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
13 + * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
14 + *
15 + */
16 +
17 + #include "qcom-ipq4019-ap.dk04.1.dtsi"
18 + #include "qcom-ipq4019-bus.dtsi"
19 + #include <dt-bindings/gpio/gpio.h>
20 + #include <dt-bindings/input/input.h>
21 + #include <dt-bindings/soc/qcom,tcsr.h>
22 +
23 + / {
24 +     model = "ASUS RT-ACRH17";
25 +     compatible = "asus,rt-acrh17", "qcom,ipq4019";
26 +
27 +     memory {
28 +         device_type = "memory";
29 +         reg = <0x80000000 0x10000000>;
30 +     };
31 +
32 +     aliases {
33 +         led-boot = &power;
34 +         led-failsafe = &power;
35 +         led-running = &power;
36 +         led-upgrade = &power;
37 +     };
38 +
39 +     reserved-memory {
40 +         #address-cells = <0x1>;
41 +         #size-cells = <0x1>;
42 +         ranges;
43 +
44 +         rsvd1@87E00000 {
45 +             reg = <0x87e00000 0x200000>;
46 +             no-map;
47 +         };
48 +     };
49 +
50 +     soc {
51 +         spi_0: spi@78b5000 {
52 +             status = "disabled";
53 +         };
54 +
55 +         tcsr@194b000 {
56 +             /* select hostmode */
57 +             compatible = "qcom,tcsr";
58 +             reg = <0x194b000 0x100>;
59 +             qcom,usb-hsphy-mode-select = <TCSR_USB_HSPHY_HOST_MODE>;
60 +             status = "ok";
61 +         };
62 +
63 +         ess_tcsr@1953000 {
64 +             compatible = "qcom,tcsr";
65 +             reg = <0x1953000 0x1000>;
66 +             qcom,ess-interface-select = <TCSR_ESS_PSGMII>;

```

```

67 +     };
68 +
69 +     tcsr@1949000 {
70 +         compatible = "qcom,tcsr";
71 +         reg = <0x1949000 0x100>;
72 +         qcom,wifi_glb_cfg = <TCSR_WIFI_GLB_CFG>;
73 +     };
74 +
75 +     tcsr@1957000 {
76 +         compatible = "qcom,tcsr";
77 +         reg = <0x1957000 0x100>;
78 +         qcom,wifi_noc_memtype_m0_m2 = <TCSR_WIFI_NOC_MEMTYPE_M0_M2>;
79 +     };
80 +
81 +     mdio@90000 {
82 +         status = "okay";
83 +     };
84 +
85 +     ess-switch@c000000 {
86 +         status = "okay";
87 +     };
88 +
89 +     ess-psgmii@98000 {
90 +         status = "okay";
91 +     };
92 +
93 +     edma@c080000 {
94 +         status = "okay";
95 +     };
96 +
97 +     wifi0: wifi@a000000 {
98 +         status = "ok";
99 +         core-id = <0x0>;
100 +         qca,msi_addr = <0x0b006040>;
101 +         qca,msi_base = <0x40>;
102 +         wifi_led_num = <2>;          /* Wifi 2G */
103 +         wifi_led_source = <0>;      /* source id 0 */
104 +         qcom,mtd-name = "0:ART";
105 +         qcom,cal-offset = <0x1000>;
106 +         qcom,cal-len = <12064>;
107 +     };
108 +
109 +     wifi1: wifi@a800000 {
110 +         status = "disabled";
111 +     };
112 + };
113 +
114 + gpio-keys {
115 +     compatible = "gpio-keys";
116 +
117 +     reset {
118 +         label = "reset";
119 +         gpios = <&t1mm 18 GPIO_ACTIVE_LOW>;
120 +         linux,code = <KEY_RESTART>;
121 +     };
122 +
123 +     wps {
124 +         label = "wps";
125 +         gpios = <&t1mm 11 GPIO_ACTIVE_LOW>;
126 +         linux,code = <KEY_WPS_BUTTON>;
127 +     };
128 + };
129 +
130 + gpio-leds {
131 +     compatible = "gpio-leds";
132 +
133 +     power: status {
134 +         label = "rt-acrh17:blue:status";
135 +         gpios = <&t1mm 40 GPIO_ACTIVE_LOW>;
136 +     };
137 +
138 +     lan1 {
139 +         label = "rt-acrh17:blue:lan1";
140 +         gpios = <&t1mm 45 GPIO_ACTIVE_LOW>;
141 +     };
142 +
143 +     lan2 {
144 +         label = "rt-acrh17:blue:lan2";
145 +         gpios = <&t1mm 43 GPIO_ACTIVE_LOW>;

```



```

146 +         };
147 +
148 +         lan3 {
149 +             label = "rt-acrh17:blue:lan3";
150 +             gpios = <&t1mm 42 GPIO_ACTIVE_LOW>;
151 +         };
152 +
153 +         lan4 {
154 +             label = "rt-acrh17:blue:lan4";
155 +             gpios = <&t1mm 49 GPIO_ACTIVE_LOW>;
156 +         };
157 +
158 +         wan_blue {
159 +             label = "rt-acrh17:blue:wan";
160 +             gpios = <&t1mm 61 GPIO_ACTIVE_HIGH>;
161 +         };
162 +
163 +         wan_red {
164 +             label = "rt-acrh17:red:wan";
165 +             gpios = <&t1mm 68 GPIO_ACTIVE_HIGH>;
166 +         };
167 +
168 +         wlan2g {
169 +             label = "rt-acrh17:blue:wlan2g";
170 +             gpios = <&t1mm 52 GPIO_ACTIVE_LOW>;
171 +         };
172 +
173 +         wlan5g {
174 +             label = "rt-acrh17:blue:wlan5g";
175 +             gpios = <&t1mm 54 GPIO_ACTIVE_LOW>;
176 +         };
177 +     };
178 + };
179 +
180 + &nand {
181 +     pinctrl-0 = <&nand_pins>;
182 +     pinctrl-names = "default";
183 +     status = "okay";
184 +
185 +     nand@0 {
186 +         partitions {
187 +             compatible = "fixed-partitions";
188 +             #address-cells = <1>;
189 +             #size-cells = <1>;
190 +
191 +             partition@0 {
192 +                 label = "Bootloader";
193 +                 reg = <0x00000000000000 0x00000004000000>;
194 +                 read-only;
195 +             };
196 +             partition@1 {
197 +                 label = "UBI_DEV";
198 +                 reg = <0x00000004000000 0x0000007C000000>;
199 +             };
200 +         };
201 +     };
202 + };
203 +
204 + &pcie0 {
205 +     status = "okay";
206 +     perst-gpio = <&t1mm 38 GPIO_ACTIVE_LOW>;
207 +     wake-gpio = <&t1mm 50 GPIO_ACTIVE_LOW>;
208 +     clkreq-gpio = <&t1mm 39 GPIO_ACTIVE_LOW>;
209 + };
210 +
211 + &qpic_bam {
212 +     status = "okay";
213 + };
214 +
215 + &nand_pins {
216 +     pullups {
217 +         pins = "gpio53", "gpio58",
218 +             "gpio59";
219 +         function = "qpic";
220 +         bias-pull-up;
221 +     };
222 +
223 +     pulldowns {
224 +         pins = "gpio55", "gpio56",

```

```

225 +         "gpio57", "gpio60",
226 +         "gpio62", "gpio63", "gpio64",
227 +         "gpio65", "gpio66", "gpio67",
228 +         "gpio69";
229 +         function = "qpic";
230 +         bias-pull-down;
231 +     };
232 + };
233 +
234 + &i2c_0_pins {
235 +     pinmux {
236 +         function = "blsp_i2c0";
237 +         pins = "gpio10";
238 +     };
239 +     pinconf {
240 +         pins = "gpio10";
241 +         drive-strength = <16>;
242 +         bias-disable;
243 +     };
244 + };

```

#### target/linux/ipq40xx/files-4.14/arch/arm/boot/dts/qcom-ipq4028-wpj428.dts

```

...     @@ -255,19 +255,7 @@
255     };
256
257     &gmac0 {
258 -         qcom,phy_mdio_addr = <4>;
259 -         qcom,poll_required = <1>;
260 -         qcom,forced_speed = <1000>;
261 -         qcom,forced_duplex = <1>;
262 -         vlan_tag = <2 0x20>;
263 -     };
264 -
265     - &gmac1 {
266 -         qcom,phy_mdio_addr = <3>;
267 -         qcom,poll_required = <1>;
268 -         qcom,forced_speed = <1000>;
269 -         qcom,forced_duplex = <1>;
270 -         vlan_tag = <1 0x10>;
258 +         vlan_tag = <1 0x31>;
271     };
272
273     &usb3_ss_phy {
...

```

#### target/linux/ipq40xx/files-4.14/arch/arm/boot/dts/qcom-ipq4029-gl-b1300.dts

```

...     @@ -289,6 +289,10 @@
289     };
290
291     };
292 + &gmac0 {
293 +     vlan_tag = <1 0x39>;
294 + };
295 +
292     &usb2_hs_phy {
293     status = "okay";
294     };
...

```

#### target/linux/ipq40xx/image/Makefile

```

...     @@ -47,11 +47,30 @@ define Device/asus_rt-ac58u
47     #         to add a version... or we are very careful not to add '\0' into that
48     #         string and call it a day.... Yeah, we do the latter!
49     IMAGE_NAME:=$(shell echo -e '\03\01\01\01RT-AC58U')
50 +     KERNEL_INITRAMFS := $$ (KERNEL) | uImage none
51 +     KERNEL_INITRAMFS_SUFFIX := -factory.trx
50     IMAGES := sysupgrade.bin
51     DEVICE_PACKAGES := ipq-wifi-asus_rt-ac58u kmod-usb-ledtrig-usbport
52     endif
53     TARGET_DEVICES += asus_rt-ac58u
54
57 + define Device/asus_rt-acrh17
58 +     $(call Device/fitImageLzma)
59 +     DEVICE_DTS := qcom-ipq4019-rt-acrh17
60 +     BLOCKSIZE := 128k
61 +     PAGESIZE := 2048
62 +     DTB_SIZE := 65536

```

```

63 +     DEVICE_TITLE := Asus RT-ACRH17
64 +     IMAGE_SIZE := 20439364
65 +     FILESYSTEMS := squashfs
66 +     UIMAGE_NAME:=$(shell echo -e '\03\01\01\01RT-AC82U')
67 +     KERNEL_INITRAMFS := $$$(KERNEL) | uImage none
68 +     KERNEL_INITRAMFS_SUFFIX := -factory.trx
69 +     IMAGES := sysupgrade.bin
70 +     DEVICE_PACKAGES := ath10k-firmware-qca4019 ath10k-firmware-qca9984
71 + endif
72 + TARGET_DEVICES += asus_rt-acrh17
73 +
55 74     define Device/avm_fritzbox-4040
56 75         $(call Device/FitImageLzma)
57 76         DEVICE_DTS := qcom-ipq4018-fritz4040
... ..

```

#### target/linux/ipq40xx/patches-4.14/069-arm-boot-add-dts-files.patch

```

... .. @@ -10,7 +10,7 @@ Signed-off-by: John Crispin <john@phrozen.org>
10 10
11 11 --- a/arch/arm/boot/dts/Makefile
12 12 +++ b/arch/arm/boot/dts/Makefile
13 - @@ -697,7 +697,14 @@ dtb-$(CONFIG_ARCH_QCOM) += \
13 + @@ -697,7 +697,15 @@ dtb-$(CONFIG_ARCH_QCOM) += \
14 14     qcom-apq8074-dragonboard.dtb \
15 15     qcom-apq8084-ifc6540.dtb \
16 16     qcom-apq8084-mtp.dtb \
... .. @@ -19,6 +19,7 @@ Signed-off-by: John Crispin <john@phrozen.org>
19 19 +     qcom-ipq4018-rt-ac58u.dtb \
20 20     qcom-ipq4019-ap.dk01.1-c1.dtb \
21 21 +     qcom-ipq4019-ap.dk04.1-c1.dtb \
22 22 ++     qcom-ipq4019-rt-acrh17.dtb \
22 23 +     qcom-ipq4028-wpj428.dtb \
23 24 +     qcom-ipq4029-gl-b1300.dtb \
24 25 +     qcom-ipq4029-mr33.dtb \
... ..

```

#### target/linux/ipq40xx/patches-4.14/713-0001-essedma-fixup-ethernet-driver-rx-bug.patch 0 → 100644

```

1 + From 0bcf3e3c613d6ed8044404bc1cc3c29ff961d89c Mon Sep 17 00:00:00 2001
2 + From: Chen Minqiang <ptpt52@gmail.com>
3 + Date: Thu, 15 Mar 2018 04:59:57 +0800
4 + Subject: [PATCH 1/2] essedma: fixup ethernet driver rx bug
5 +
6 + - modify the error rx ring full conditions
7 + - in rare cases, out of memory allocation failure causes the receive queues stop
8 + we use the timer to re-alloc rx rings under these circumstances
9 +
10 + Signed-off-by: Chen Minqiang <ptpt52@gmail.com>
11 + ---
12 + drivers/net/ethernet/qualcomm/essedma/edma.c | 51 ++++++
13 + drivers/net/ethernet/qualcomm/essedma/edma.h | 3 ++
14 + drivers/net/ethernet/qualcomm/essedma/edma_axi.c | 8 +++
15 + 3 files changed, 58 insertions(+), 4 deletions(-)
16 +
17 + diff --git a/drivers/net/ethernet/qualcomm/essedma/edma.c b/drivers/net/ethernet/qualcomm/essedma/edma.c
18 + index fecc0ba..3f1da93 100644
19 + --- a/drivers/net/ethernet/qualcomm/essedma/edma.c
20 + +++ b/drivers/net/ethernet/qualcomm/essedma/edma.c
21 + @@ -530,6 +530,47 @@ static int edma_rx_complete_paged(struct sk_buff *skb, u16 num_rfds, u16 length,
22 +     return sw_next_to_clean;
23 + }
24 +
25 + +static int edma_rfd_desc_unused(struct edma_rfd_desc_ring *erdr)
26 + +{
27 + +     if (erdr->sw_next_to_clean > erdr->sw_next_to_fill)
28 + +         return erdr->sw_next_to_clean - erdr->sw_next_to_fill - 1;
29 + +     return erdr->count + erdr->sw_next_to_clean - erdr->sw_next_to_fill - 1;
30 + +}
31 + +
32 + +void edma_rx_realloc(unsigned long data)
33 + +{
34 + +     struct edma_per_cpu_queues_info *edma_percpu_info = (struct edma_per_cpu_queues_info *)data;
35 + +     struct edma_common_info *edma_cinfo = edma_percpu_info->edma_cinfo;
36 + +     s32 status = edma_percpu_info->rx_realloc_status;
37 + +
38 + +     while (status) {
39 + +         int queue_id;
40 + +         int ret_count;
41 + +         struct edma_rfd_desc_ring *erdr;

```

```

42 ++
43 ++     queue_id = ffs(status) - 1;
44 ++     erdr = edma_cinfo->rfd_ring[queue_id];
45 ++     ret_count = edma_alloc_rx_buf(edma_cinfo, erdr, edma_rfd_desc_unused(erdr), queue_id);
46 ++     if (ret_count == 0) {
47 ++         edma_percpu_info->rx_realloc_status &= ~(1 << queue_id);
48 ++     }
49 ++     status &= ~(1 << queue_id);
50 ++ }
51 ++
52 ++     if (edma_percpu_info->rx_realloc_status) {
53 ++         mod_timer(&edma_percpu_info->rx_realloc_timer, jiffies + HZ);
54 ++     }
55 ++}
56 ++
57 ++static inline void edma_realloc_timer_start(struct napi_struct *napi, int queue_id)
58 ++{
59 ++     struct edma_per_cpu_queues_info *edma_percpu_info = container_of(napi,
60 ++         struct edma_per_cpu_queues_info, napi);
61 ++
62 ++     edma_percpu_info->rx_realloc_status |= (1 << queue_id);
63 ++     mod_timer(&edma_percpu_info->rx_realloc_timer, jiffies + 5 * HZ); /* restart alloc in 5 secs */
64 ++}
65 ++
66 ++ /*
67 ++  * edma_rx_complete()
68 ++  *   Main api called from the poll function to process rx packets.
69 ++  *@@ -754,10 +795,12 @@ static void edma_rx_complete(struct edma_common_info *edma_cinfo,
70 ++     erdr->sw_next_to_clean = sw_next_to_clean;
71 ++
72 ++     /* Refill here in case refill threshold wasn't reached */
73 ++     if (likely(cleaned_count)) {
74 ++         ret_count = edma_alloc_rx_buf(edma_cinfo, erdr, cleaned_count, queue_id);
75 ++         if (ret_count)
76 ++             if (edma_rfd_desc_unused(erdr)) {
77 ++                 ret_count = edma_alloc_rx_buf(edma_cinfo, erdr, edma_rfd_desc_unused(erdr), queue_id);
78 ++                 if (ret_count) {
79 ++                     dev_dbg(&pdev->dev, "Not all buffers was reallocated");
80 ++                     edma_realloc_timer_start(napi, queue_id);
81 ++                 }
82 ++                 edma_write_reg(EDMA_REG_RX_SW_CONS_IDX_Q(queue_id),
83 ++                     erdr->sw_next_to_clean);
84 ++             }
85 ++  *@@ -1801,7 +1844,7 @@ int edma_configure(struct edma_common_info *edma_cinfo)
86 ++     /* Allocate the RX buffer */
87 ++     for (i = 0, j = 0; i < edma_cinfo->num_rx_queues; i++) {
88 ++         struct edma_rfd_desc_ring *ring = edma_cinfo->rfd_ring[j];
89 ++         ret_count = edma_alloc_rx_buf(edma_cinfo, ring, ring->count, j);
90 ++         ret_count = edma_alloc_rx_buf(edma_cinfo, ring, edma_rfd_desc_unused(ring), j);
91 ++         if (ret_count) {
92 ++             dev_dbg(&edma_cinfo->pdev->dev, "not all rx buffers allocated\n");
93 ++         }
94 ++ diff --git a/drivers/net/ethernet/qualcomm/essedma/edma.h b/drivers/net/ethernet/qualcomm/essedma/edma.h
95 ++ index 5d6dc73..29c8379 100644
96 ++ --- a/drivers/net/ethernet/qualcomm/essedma/edma.h
97 ++ +++ b/drivers/net/ethernet/qualcomm/essedma/edma.h
98 ++  *@@ -304,6 +304,8 @@ struct edma_per_cpu_queues_info {
99 ++     u32 tx_start; /* tx queue start */
100 ++     u32 rx_start; /* rx queue start */
101 ++     struct edma_common_info *edma_cinfo; /* edma common info */
102 ++     u32 rx_realloc_status;
103 ++     struct timer_list rx_realloc_timer;
104 ++ };
105 ++
106 ++ /* edma specific common info */
107 ++  *@@ -448,6 +450,7 @@ void edma_change_tx_coalesce(int usecs);
108 ++ void edma_change_rx_coalesce(int usecs);
109 ++ void edma_get_tx_rx_coalesce(u32 *reg_val);
110 ++ void edma_clear_irq_status(void);
111 ++ +void edma_rx_realloc(unsigned long data);
112 ++ void ess_set_port_status_speed(struct edma_common_info *edma_cinfo,
113 ++     struct phy_device *phydev, uint8_t port_id);
114 ++ #endif /* _EDMA_H */
115 ++ diff --git a/drivers/net/ethernet/qualcomm/essedma/edma_axi.c b/drivers/net/ethernet/qualcomm/essedma/edma_axi.c
116 ++ index 81fc1e1..d9f8b52 100644
117 ++ --- a/drivers/net/ethernet/qualcomm/essedma/edma_axi.c
118 ++ +++ b/drivers/net/ethernet/qualcomm/essedma/edma_axi.c
119 ++  *@@ -1131,6 +1131,11 @@ static int edma_axi_probe(struct platform_device *pdev)
120 ++     edma_cinfo->edma_percpu_info[i].rx_status = 0;

```

```

121 +         edma_cinfo->edma_percpu_info[i].edma_cinfo = edma_cinfo;
122 +
123 + +         edma_cinfo->edma_percpu_info[i].rx_realloc_status = 0;
124 + +         init_timer(&edma_cinfo->edma_percpu_info[i].rx_realloc_timer);
125 + +         edma_cinfo->edma_percpu_info[i].rx_realloc_timer.function = edma_rx_realloc;
126 + +         edma_cinfo->edma_percpu_info[i].rx_realloc_timer.data = (unsigned long)&edma_cinfo-
>edma_percpu_info[i];
127 + +
128 +         /* Request irq per core */
129 +         for (j = edma_cinfo->edma_percpu_info[i].tx_start;
130 +             j < tx_start[i] + 4; j++) {
131 + @@ -1259,7 +1264,10 @@ err_configure:
132 + err_rmap_add_fail:
133 +         edma_free_irqs(adapter[0]);
134 +         for (i = 0; i < CONFIG_NR_CPUS; i++)
135 +         {
136 +             napi_disable(&edma_cinfo->edma_percpu_info[i].napi);
137 + +             del_timer_sync(&edma_cinfo->edma_percpu_info[i].rx_realloc_timer);
138 +         }
139 + err_reset:
140 + err_unregister_sysctl_tbl:
141 + err_rmap_alloc_fail:
142 + --
143 + 2.7.4
144 +

```

▼ [target/linux/ipq40xx/patches-4.14/713-0002-essedma-refine-txq-to-be-adaptive-of-cpus-and-netdev.patch](#) 0 → 100644

```

1 + From 7be0cb35513b07bf74d93d052d57b12e2c654b43 Mon Sep 17 00:00:00 2001
2 + From: Chen Minqiang <ptpt52@gmail.com>
3 + Date: Thu, 15 Mar 2018 05:04:37 +0800
4 + Subject: [PATCH 2/2] essedma: refine txq to be adaptive of cpus and netdev
5 +
6 + - use 4 queue for each cpu if only 1 netdev
7 + - use all 16 txqueue if only 1 netdev
8 +
9 + Signed-off-by: Chen Minqiang <ptpt52@gmail.com>
10 + ---
11 + drivers/net/ethernet/qualcomm/essedma/edma.c | 22 +++++-----
12 + drivers/net/ethernet/qualcomm/essedma/edma.h | 5 +--
13 + drivers/net/ethernet/qualcomm/essedma/edma_axi.c | 40 ++++++++-----
14 + 3 files changed, 35 insertions(+), 32 deletions(-)
15 +
16 + diff --git a/drivers/net/ethernet/qualcomm/essedma/edma.c b/drivers/net/ethernet/qualcomm/essedma/edma.c
17 + index 3f1da93..05f9ce9 100644
18 + --- a/drivers/net/ethernet/qualcomm/essedma/edma.c
19 + +++ b/drivers/net/ethernet/qualcomm/essedma/edma.c
20 + @@ -22,14 +22,6 @@ extern struct net_device *edma_netdev[EDMA_MAX_PORTID_SUPPORTED];
21 + bool edma_stp_rstp;
22 + u16 edma_ath_eth_type;
23 +
24 + -/* edma_skb_priority_offset()
25 + - * get edma skb priority
26 + - */
27 + -static unsigned int edma_skb_priority_offset(struct sk_buff *skb)
28 + -{
29 + -     return (skb->priority >> 2) & 1;
30 + -}
31 + -
32 + /* edma_alloc_tx_ring()
33 + * Allocate Tx descriptors ring
34 + */
35 + @@ -1042,13 +1034,14 @@ static inline u16 edma_tpd_available(struct edma_common_info *edma_cinfo,
36 + /* edma_tx_queue_get()
37 + * Get the starting number of the queue
38 + */
39 + -static inline int edma_tx_queue_get(struct edma_adapter *adapter,
40 + +static inline int edma_tx_queue_get(struct edma_common_info *edma_cinfo, struct edma_adapter *adapter,
41 + +                                     struct sk_buff *skb, int txq_id)
42 + + {
43 + +     /* skb->priority is used as an index to skb priority table
44 + +      * and based on packet priority, correspong queue is assigned.
45 + +      * FIXME we just simple use jiffies for time base balance
46 + +      */
47 + +     return adapter->tx_start_offset[txq_id] + edma_skb_priority_offset(skb);
48 + +     return adapter->tx_start_offset[txq_id] + (jiffies % edma_cinfo->num_txq_per_core_netdev);
49 + + }
50 + +
51 + /* edma_tx_update_hw_idx()

```

```

52 + @@ -1417,8 +1410,9 @@ netdev_tx_t edma_xmit(struct sk_buff *skb,
53 +     }
54 +
55 +     /* this will be one of the 4 TX queues exposed to linux kernel */
56 + - txq_id = skb_get_queue_mapping(skb);
57 + - queue_id = edma_tx_queue_get(adapter, skb, txq_id);
58 + + /* XXX what if num_online_cpus() > EDMA_CPU_CORES_SUPPORTED */
59 + + txq_id = smp_processor_id() % EDMA_CPU_CORES_SUPPORTED;
60 + + queue_id = edma_tx_queue_get(edma_cinfo, adapter, skb, txq_id);
61 +     etdr = edma_cinfo->tpd_ring[queue_id];
62 +     nq = netdev_get_tx_queue(net_dev, txq_id);
63 +
64 + @@ -1899,8 +1893,8 @@ void edma_free_irqs(struct edma_adapter *adapter)
65 +     int i, j;
66 +     int k = ((edma_cinfo->num_rx_queues == 4) ? 1 : 2);
67 +
68 + -     for (i = 0; i < CONFIG_NR_CPUS; i++) {
69 + -         for (j = edma_cinfo->edma_percpu_info[i].tx_start; j < (edma_cinfo->edma_percpu_info[i].tx_start +
70 + 4); j++)
71 + +         for (i = 0; i < num_online_cpus() && i < EDMA_CPU_CORES_SUPPORTED; i++) {
72 + +             for (j = edma_cinfo->edma_percpu_info[i].tx_start; j < (edma_cinfo->edma_percpu_info[i].tx_start +
73 + edma_cinfo->num_txq_per_core); j++)
74 +                 free_irq(edma_cinfo->tx_irq[j], &edma_cinfo->edma_percpu_info[i]);
75 +
76 +             for (j = edma_cinfo->edma_percpu_info[i].rx_start; j < (edma_cinfo->edma_percpu_info[i].rx_start +
77 + k); j++)
78 + + diff --git a/drivers/net/ethernet/qualcomm/essedma/edma.h b/drivers/net/ethernet/qualcomm/essedma/edma.h
79 + index 29c8379..2ba43e0 100644
80 + --- a/drivers/net/ethernet/qualcomm/essedma/edma.h
81 + +++ b/drivers/net/ethernet/qualcomm/essedma/edma.h
82 + @@ -325,6 +325,7 @@ struct edma_common_info {
83 +     u32 from_cpu; /* from CPU TPD field */
84 +     u32 num_rxq_per_core; /* Rx queues per core */
85 +     u32 num_txq_per_core; /* Tx queues per core */
86 + +     u32 num_txq_per_core_netdev; /* Tx queues per core per netdev */
87 +     u16 tx_ring_count; /* Tx ring count */
88 +     u16 rx_ring_count; /* Rx ring*/
89 +     u16 rx_head_buffer_len; /* rx buffer length */
90 + @@ -332,7 +333,7 @@ struct edma_common_info {
91 +     u32 page_mode; /* Jumbo frame supported flag */
92 +     u32 fraglist_mode; /* fraglist supported flag */
93 +     struct edma_hw hw; /* edma hw specific structure */
94 + -     struct edma_per_cpu_queues_info edma_percpu_info[CONFIG_NR_CPUS]; /* per cpu information */
95 + +     struct edma_per_cpu_queues_info edma_percpu_info[EDMA_CPU_CORES_SUPPORTED]; /* per cpu information */
96 +     spinlock_t stats_lock; /* protect edma stats area for updation */
97 +
98 +     bool is_single_phy;
99 + @@ -401,7 +402,7 @@ struct edma_adapter {
100 +     u32 link_state; /* phy link state */
101 +     u32 phy_mdio_addr; /* PHY device address on MII interface */
102 +     u32 poll_required; /* check if link polling is required */
103 + -     u32 tx_start_offset[CONFIG_NR_CPUS]; /* tx queue start */
104 + +     u32 tx_start_offset[EDMA_CPU_CORES_SUPPORTED]; /* tx queue start */
105 +     u32 default_vlan_tag; /* vlan tag */
106 +     u32 dp_bitmap;
107 +     uint8_t phy_id[MII_BUS_ID_SIZE + 3];
108 + + diff --git a/drivers/net/ethernet/qualcomm/essedma/edma_axi.c b/drivers/net/ethernet/qualcomm/essedma/edma_axi.c
109 + index d9f8b52..5824680 100644
110 + --- a/drivers/net/ethernet/qualcomm/essedma/edma_axi.c
111 + +++ b/drivers/net/ethernet/qualcomm/essedma/edma_axi.c
112 + @@ -721,11 +721,7 @@ static int edma_axi_probe(struct platform_device *pdev)
113 +     int i, j, k, err = 0;
114 +     int portid_bmp;
115 +     int idx = 0, idx_mac = 0;
116 + -
117 + -     if (CONFIG_NR_CPUS != EDMA_CPU_CORES_SUPPORTED) {
118 + -         dev_err(&pdev->dev, "Invalid CPU Cores\n");
119 + -         return -EINVAL;
120 + -     }
121 + +     int netdev_group = 2;
122 +
123 +     if ((num_rxq != 4) && (num_rxq != 8)) {
124 +         dev_err(&pdev->dev, "Invalid RX queue, edma probe failed\n");
125 + @@ -749,7 +745,7 @@ static int edma_axi_probe(struct platform_device *pdev)
126 +     /* Initialize the netdev array before allocation
127 +      * to avoid double free
128 +      */
129 +     for (i = 0 ; i < edma_cinfo->num_gmac ; i++)
130 +     for (i = 0 ; i < EDMA_MAX_PORTID_SUPPORTED; i++)

```

```

128 +         edma_netdev[i] = NULL;
129 +
130 +         for (i = 0 ; i < edma_cinfo->num_gmac ; i++) {
131 + @@ -770,8 +766,11 @@ static int edma_axi_probe(struct platform_device *pdev)
132 +
133 +         /* Fill ring details */
134 +         edma_cinfo->num_tx_queues = EDMA_MAX_TRANSMIT_QUEUE;
135 + -         edma_cinfo->num_txq_per_core = (EDMA_MAX_TRANSMIT_QUEUE / 4);
136 + +         edma_cinfo->num_txq_per_core = (EDMA_MAX_TRANSMIT_QUEUE / num_online_cpus());
137 + +         edma_cinfo->num_txq_per_core_netdev = (EDMA_MAX_TRANSMIT_QUEUE / netdev_group / num_online_cpus());
138 +         edma_cinfo->tx_ring_count = EDMA_TX_RING_SIZE;
139 + +         if (edma_cinfo->num_txq_per_core == 0)
140 + +             edma_cinfo->num_txq_per_core = 1;
141 +
142 +         /* Update num rx queues based on module parameter */
143 +         edma_cinfo->num_rx_queues = num_rxq;
144 + @@ -941,6 +940,13 @@ static int edma_axi_probe(struct platform_device *pdev)
145 +             idx_mac++;
146 +         }
147 +
148 + +         if (edma_cinfo->num_gmac == 1) {
149 + +             netdev_group = 1;
150 + +             edma_cinfo->num_txq_per_core_netdev = (EDMA_MAX_TRANSMIT_QUEUE / netdev_group / num_online_cpus());
151 + +         }
152 + +         if (edma_cinfo->num_txq_per_core_netdev == 0)
153 + +             edma_cinfo->num_txq_per_core_netdev = 1;
154 + +
155 +         /* Populate the adapter structure register the netdevice */
156 +         for (i = 0; i < edma_cinfo->num_gmac; i++) {
157 +             int k, m;
158 + @@ -948,17 +954,16 @@ static int edma_axi_probe(struct platform_device *pdev)
159 +             adapter[i] = netdev_priv(edma_netdev[i]);
160 +             adapter[i]->netdev = edma_netdev[i];
161 +             adapter[i]->pdev = pdev;
162 + -             for (j = 0; j < CONFIG_NR_CPUS; j++) {
163 + -                 m = i % 2;
164 + -                 adapter[i]->tx_start_offset[j] =
165 + -                     ((j << EDMA_TX_CPU_START_SHIFT) + (m << 1));
166 + +             for (j = 0; j < num_online_cpus() && j < EDMA_CPU_CORES_SUPPORTED; j++) {
167 + +                 m = i % netdev_group;
168 + +                 adapter[i]->tx_start_offset[j] = j * edma_cinfo->num_txq_per_core + m * edma_cinfo->
169 + +                 num_txq_per_core_netdev;
170 + +                 /* Share the queues with available net-devices.
171 + +                 * For instance , with 5 net-devices
172 + +                 * eth0/eth2/eth4 will share q0,q1,q4,q5,q8,q9,q12,q13
173 + +                 * and eth1/eth3 will get the remaining.
174 + +                 */
175 + +                 for (k = adapter[i]->tx_start_offset[j]; k <
176 + +                     (adapter[i]->tx_start_offset[j] + 2); k++) {
177 + +                     (adapter[i]->tx_start_offset[j] + edma_cinfo->num_txq_per_core_netdev); k++) {
178 + +                         if (edma_fill_netdev(edma_cinfo, k, i, j)) {
179 + +                             pr_err("Netdev overflow Error\n");
180 + +                             goto err_register;
181 + + @@ -1111,9 +1116,12 @@ static int edma_axi_probe(struct platform_device *pdev)
182 + +                 /* populate per_core_info, do a napi_Add, request 16 TX irqs,
183 + +                 * 8 RX irqs, do a napi enable
184 + +                 */
185 + +                 for (i = 0; i < CONFIG_NR_CPUS; i++) {
186 + +                     for (i = 0; i < num_online_cpus() && i < EDMA_MAX_TRANSMIT_QUEUE; i++) {
187 + +                         u8 rx_start;
188 + +
189 + +                         tx_mask[i] = (0xFFFF >> (16 - edma_cinfo->num_txq_per_core)) << (i * edma_cinfo->num_txq_per_core);
190 + +                         tx_start[i] = i * edma_cinfo->num_txq_per_core;
191 + +
192 + +                         edma_cinfo->edma_percpu_info[i].napi.state = 0;
193 + +
194 + +                         netif_napi_add(edma_netdev[0],
195 + + @@ -1138,7 +1146,7 @@ static int edma_axi_probe(struct platform_device *pdev)
196 + +
197 + +                         /* Request irq per core */
198 + +                         for (j = edma_cinfo->edma_percpu_info[i].tx_start;
199 + +                             j < tx_start[i] + 4; j++) {
200 + +                             j < tx_start[i] + edma_cinfo->num_txq_per_core; j++) {
201 + +                                 sprintf(&edma_tx_irq[j][0], "edma_eth_tx%d", j);
202 + +                                 err = request_irq(edma_cinfo->tx_irq[j],
203 + +                                     edma_interrupt,
204 + + @@ -1263,7 +1271,7 @@ err_configure:
205 + + #endif
206 + + err_rmap_add_fail:

```

```

206 +     edma_free_irqs(adapter[0]);
207 + -     for (i = 0; i < CONFIG_NR_CPUS; i++)
208 + +     for (i = 0; i < num_online_cpus() && i < EDMA_CPU_CORES_SUPPORTED; i++)
209 +     {
210 +         napi_disable(&edma_cinfo->edma_percpu_info[i].napi);
211 +         del_timer_sync(&edma_cinfo->edma_percpu_info[i].rx_realloc_timer);
212 + @@ -1314,7 +1322,7 @@ static int edma_axi_remove(struct platform_device *pdev)
213 +         unregister_netdev(edma_netdev[i]);
214 +
215 +         edma_stop_rx_tx(hw);
216 + -     for (i = 0; i < CONFIG_NR_CPUS; i++)
217 + +     for (i = 0; i < num_online_cpus() && i < EDMA_CPU_CORES_SUPPORTED; i++)
218 +         napi_disable(&edma_cinfo->edma_percpu_info[i].napi);
219 +
220 +     edma_irq_disable(edma_cinfo);
221 + --
222 + 2.7.4
223 +

```

▼  [target/linux/ipq40xx/patches-4.14/901-essedma-disable-default-vlan-tagging.patch](#) 0 → 100644

```

1 + From 42b508d91b7f51b054f383e3aa42089ccab9300d Mon Sep 17 00:00:00 2001
2 + From: Chen Minqiang <ptpt52@gmail.com>
3 + Date: Thu, 15 Mar 2018 05:33:46 +0800
4 + Subject: [PATCH] essedma: disable default vlan tagging
5 +
6 + The essedma driver has its own unique take on VLAN management
7 + and its configuration. In the original SDK, each VLAN is
8 + assigned one virtual ethernet netdev.
9 +
10 + However, this is non-standard. So, this patch does away
11 + with the default_vlan_tag property the driver is using
12 + and therefore forces the user to use the kernel's vlan
13 + feature.
14 +
15 + This patch also removes the "qcom,poll_required = <1>," from
16 + the essedma node.
17 +
18 + Signed-off-by: Christian Lamparter <chunkeey@googlemail.com>
19 + Signed-off-by: Chen Minqiang <ptpt52@gmail.com>
20 + ---
21 + arch/arm/boot/dts/qcom-ipq4019.dtsi | 5 ++---
22 + drivers/net/ethernet/qualcomm/essedma/edma.c | 14 +++++-----
23 + 2 files changed, 7 insertions(+), 12 deletions(-)
24 +
25 + diff --git a/arch/arm/boot/dts/qcom-ipq4019.dtsi b/arch/arm/boot/dts/qcom-ipq4019.dtsi
26 + index 3c4617f..7c3af8e 100644
27 + --- a/arch/arm/boot/dts/qcom-ipq4019.dtsi
28 + +++ b/arch/arm/boot/dts/qcom-ipq4019.dtsi
29 + @@ -598,8 +598,7 @@
30 +         qcom,page-mode = <0>;
31 +         qcom,rx_head_buf_size = <1540>;
32 +         qcom,mdio_supported;
33 + -         qcom,poll_required = <1>;
34 + -         qcom,num_gmac = <2>;
35 + +         qcom,num_gmac = <1>;
36 +         interrupts = <0 65 IRQ_TYPE_EDGE_RISING
37 +             0 66 IRQ_TYPE_EDGE_RISING
38 +             0 67 IRQ_TYPE_EDGE_RISING
39 + @@ -637,7 +636,7 @@
40 +
41 +         gmac0: gmac0 {
42 +             local-mac-address = [00 00 00 00 00 00];
43 + -             vlan_tag = <1 0x1f>;
44 + +             vlan_tag = <1 0x3f>;
45 +         };
46 +
47 +         gmac1: gmac1 {
48 + diff --git a/drivers/net/ethernet/qualcomm/essedma/edma.c b/drivers/net/ethernet/qualcomm/essedma/edma.c
49 + index 05f9ce9..a3c0d66 100644
50 + --- a/drivers/net/ethernet/qualcomm/essedma/edma.c
51 + +++ b/drivers/net/ethernet/qualcomm/essedma/edma.c
52 + @@ -750,13 +750,11 @@ static void edma_rx_complete(struct edma_common_info *edma_cinfo,
53 +         edma_receive_checksum(rd, skb);
54 +
55 +         /* Process VLAN HW acceleration indication provided by HW */
56 + +         if (unlikely(adapter->default_vlan_tag != rd->rrd4)) {
57 + -             vlan = rd->rrd4;
58 + -             if (likely(rd->rrd7 & EDMA_RRD_CVLAN))

```



```
59 + -         __vlan_hwaccel_put_tag(skb, htons(ETH_P_8021Q), vlan);
60 + -     else if (rd->rrd1 & EDMA_RRD_SVLAN)
61 + -         __vlan_hwaccel_put_tag(skb, htons(ETH_P_8021AD), vlan);
62 + -     }
63 + +     vlan = rd->rrd4;
64 + +     if (likely(rd->rrd7 & EDMA_RRD_CVLAN))
65 + +         __vlan_hwaccel_put_tag(skb, htons(ETH_P_8021Q), vlan);
66 + +     else if (rd->rrd1 & EDMA_RRD_SVLAN)
67 + +         __vlan_hwaccel_put_tag(skb, htons(ETH_P_8021AD), vlan);
68 +
69 +         /* Update rx statistics */
70 +         adapter->stats.rx_packets++;
71 + @@ -1434,8 +1432,6 @@ netdev_tx_t edma_xmit(struct sk_buff *skb,
72 +     if (!adapter->edma_cinfo->is_single_phy) {
73 +         if (unlikely(skb_vlan_tag_present(skb)))
74 +             flags_transmit |= EDMA_VLAN_TX_TAG_INSERT_FLAG;
75 + -         else if (adapter->default_vlan_tag)
76 + -             flags_transmit |= EDMA_VLAN_TX_TAG_INSERT_DEFAULT_FLAG;
77 +     }
78 +
79 +     /* Check and mark checksum offload */
80 + --
81 + 2.7.4
82 +
```

Please [register](#) or [sign in](#) to comment