

**Audi HCP3**

**Ethernet Network Configuration Whitepaper**

**Advanced Networking**

Draft

Gerd Zimmermann gerd.zimmermann@harman.com

Advanced\_Networking

Harman Infotainment Division

Becker Goering Str. 16

76307 Karlsbad

Germany

Person in charge:

Gerd Zimmermann gerd.zimmermann@harman.com

Publication date 2020/05/26

Draft

Copyright © 2020 Harman Infotainment Division

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher.

**Revision History**

|  |
| --- |
| *Revision 1.0 09.04.2020 Gerd Zimmermann* |
| *Initial document* |
|  |
| *Revision 1.1 09.04.2020 Gerd Zimmermann* |
| *update 3.1* |
|  |
| *Revision 1.2 04.05.2020 Gerd Zimmermann* |
| *update 3* |
|  |
| *Revision 1.3 05.05.2020 Gerd Zimmermann* |
| *added: 3.5 Expected VLAN Filter Settings; 3.6 Expected IP Filter Settings* |
|  |
| *Revision 14 14.05.2020 Gerd Zimmermann* |
| *fix: 3.4* |
|  |
| *Revision 1.5 19.05.2020 Gerd Zimmermann* |
| *chapter 2.1 + 2.2 updated to latest conclusion with Audi*  *chapter 3.5.1 + 3.5.2 invers filter settings deleted* |
|  |
| *Revision 1.6 20.05.2020 Gerd Zimmermann* |
| *updated chapter chapter 3.5.2 with missing VLAN filter setting for prio 0* |
|  |
| *Revision 1.7 26.05.2020 Gerd Zimmermann* |
| *updated chapter chapter 3.5 additional VLAN-ID routes to DMA ch0* |
|  |
| *Revision 1.8 26.05.2020 Gerd Zimmermann* |
| *deleted V1.7 changes; additionally changed: 3.1; 3.3; 3.4.1; 3.4.2; 3.5, added 3.7* |

**Table of Contents**

[1. Introduction 6](#_Toc41452527)

[2. EMAC Queue Composition 7](#_Toc41452528)

[2.1 System Partition "EthCore 1" 8](#_Toc41452529)

[2.2 Android Partition "EthCore 2" 10](#_Toc41452530)

[3. Testing 12](#_Toc41452531)

[3.1 Predefined Rx packet length 12](#_Toc41452532)

[3.2 Predefined Tx packet length 13](#_Toc41452533)

[3.3 Predefined Trace messages 14](#_Toc41452534)

[3.4 Expected register Content 15](#_Toc41452535)

[3.4.1 System Partition 15](#_Toc41452536)

[3.4.2 Android Partition 15](#_Toc41452537)

[3.5 Expected VLAN Filter Settings 16](#_Toc41452538)

[3.5.1 System Partition 16](#_Toc41452539)

[3.5.2 Android Partition 16](#_Toc41452540)

[3.6 Expected IP Filter Settings 18](#_Toc41452541)

[3.6.1 System Partition 18](#_Toc41452542)

[3.6.2 Android Partition 19](#_Toc41452543)

[3.7 Promiscuous Mode 19](#_Toc41452544)

[4. Recommended work flow 20](#_Toc41452545)

[4.1 With DMA CH0 serviced from Standard Driver 20](#_Toc41452546)

[4.1 With DMA CH0 serviced from Outside 20](#_Toc41452547)

[5. Gerds Sandbox 21](#_Toc41452548)

[Appendix 22](#_Toc41452549)

[B Ethernet Restrictions: Samsung ExynosAuto Samples 22](#_Toc41452550)

[B.1 EVT0 (Q4 2018) 22](#_Toc41452551)

[B.1 EVT1 (Q4 2019) 23](#_Toc41452552)

[C Ethernet Restrictions: EMAC FIFO size used by queues 24](#_Toc41452553)

[C.1 EVT0 (Q4 2018) 24](#_Toc41452554)

[C.2 EVT1 (Q4 2019) 24](#_Toc41452555)

[D EMAC queue and dma channel explanation 25](#_Toc41452556)

[E EMAC Device Tree Configuration options (EVT0 "stmmac") 29](#_Toc41452557)

[F Mutli TX DMA Channel usage 36](#_Toc41452558)

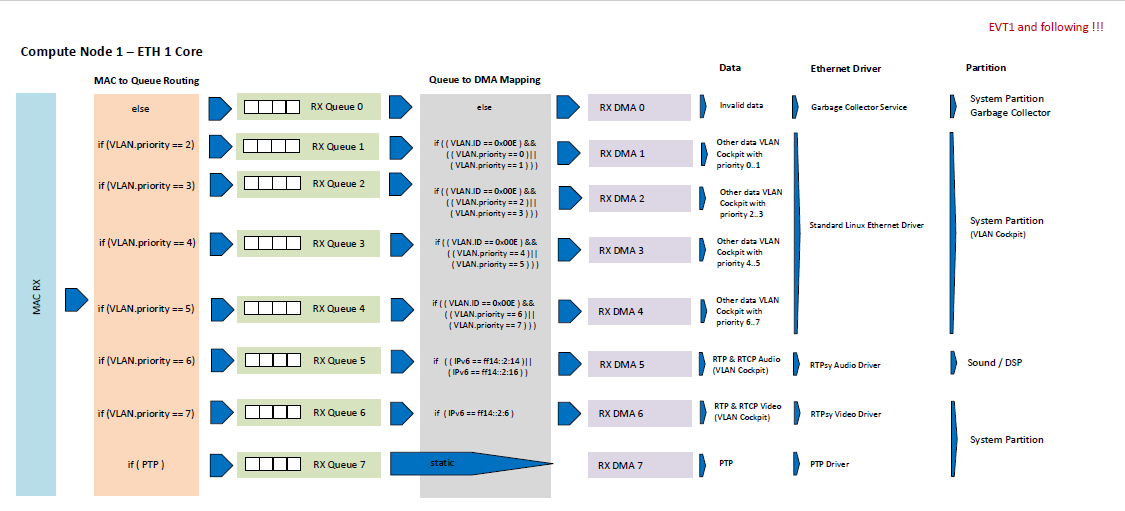
1. Introduction

This ConfigurationWhitepaper provides more details for the realization of the architectural design of the network driver for the Computing Node. It considers the HW capabilities of the Samsung ExynosAuto SoC.

2. EMAC Queue Composition

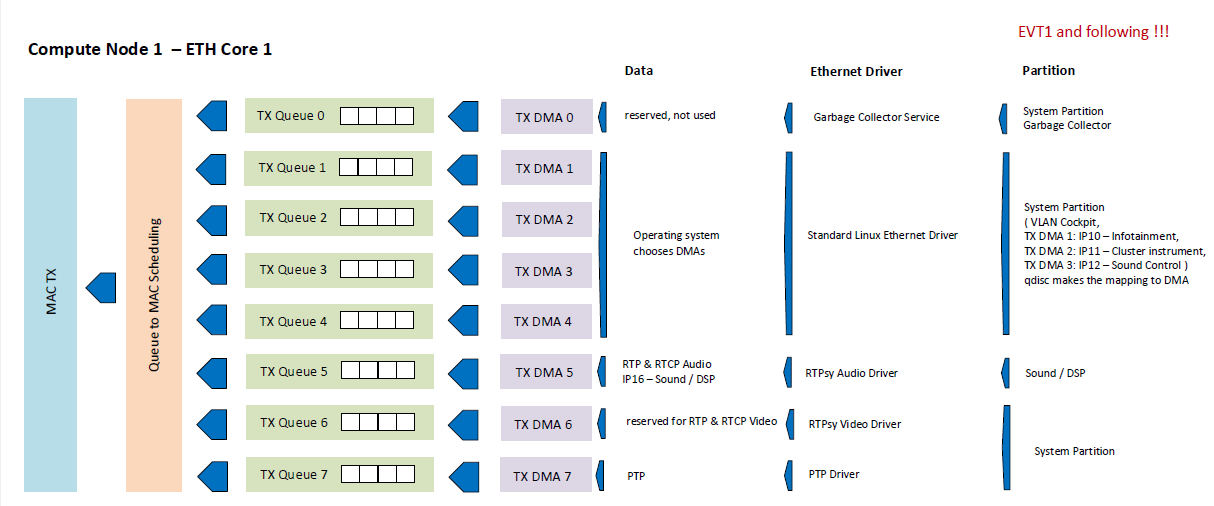
Samsung Exynos hosts 2 EMAC modules. Both are of same type and there's no interconnection between them. Each EMAC module offers 8 Rx and Tx queues. This means for Tx, that 8 independent instances could send Ethernet packets without interaction, also from different partitions or different cores. And it means for Rx, that the HW offers routing capabilities of received packets into dedicated Rx queue, e. g. based on VLAN, MAC or IPv6 address.

2.1 System Partition "EthCore 1"



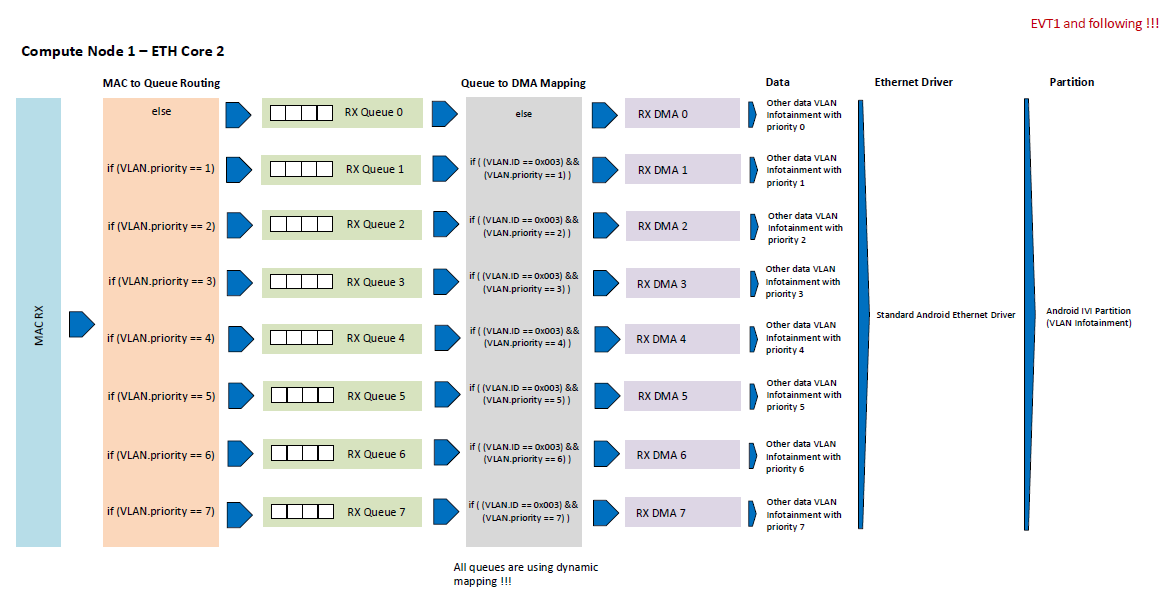
Source: Chrisitian Sander / Audi 2020/05/19

**Hint:** Gerd Zimmermann 2020/05/19: "else" path in "Queue to DMA Mapping" would need "invers vlan filter settings". But using those breaks all other vlan based filter routing - will be clarified with Audi



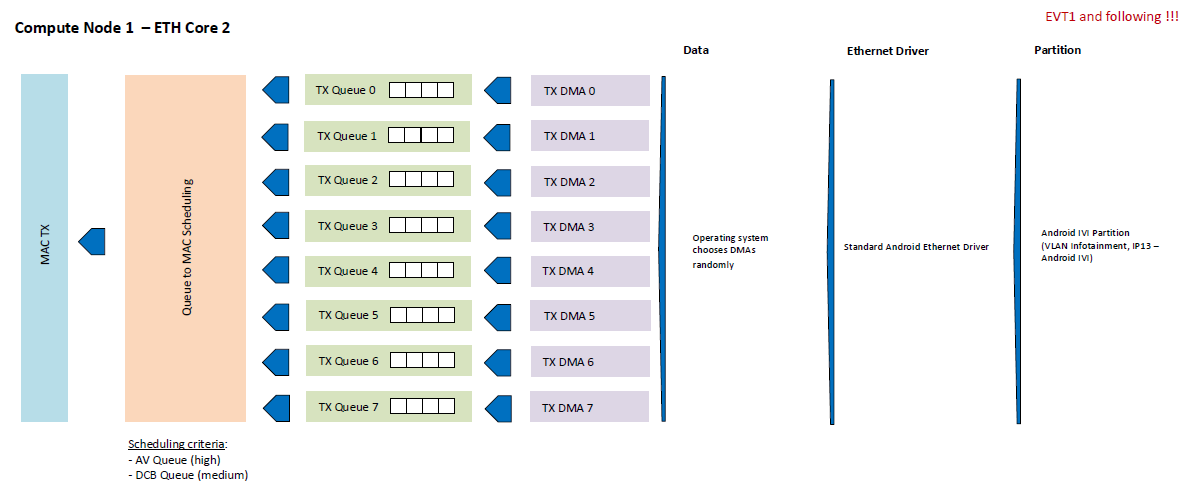
Source: Chrisitian Sander / Audi 2020/05/19

2.2 Android Partition "EthCore 2"



Source: Chrisitian Sander / Audi 2020/05/19

**Hint:** Gerd Zimmermann 2020/05/19: "else" path in "Queue to DMA Mapping" would need "invers vlan filter settings". But using those breaks all other vlan based filter routing - will be clarified with Audi



Source: Chrisitian Sander / Audi 2020/05/19

3. Testing

The verification of the configuration is a little bit tricky, because dedicated packets have to be detected during runtime This takes performance. Also looking into the packet content is very time consuming. So we'll use the length information which is part of the descriptor.

Taking only the length, you can use whatever protocol you want.

It's not clear at the moment how to proof this scenario. So may be traces on this condition have to be included into target releases

3.1 Predefined Rx packet length

This is the expected length on the wire! As we use a range from 999 - 1007, the driver rx service can check this range and only print with trace message from 3.2 if a packet fits this range. Additionally, the length of 666 is used for packets normally discarded by the filters.

So, it depends if your protocol stimulator which send packets to HCP3 takes the VLAN tag into account or not!

|  |  |  |  |
| --- | --- | --- | --- |
| Length (byte) | VLAN Tag | Priority | Test Scenario Name |
| 101 | yes | 1 | Vlan\_Prio1 |
| 102 | yes | 2 | Vlan\_Prio2 |
| 103 | yes | 3 | Vlan\_Prio3 |
| 104 | yes | 4 | Vlan\_Prio4 |
| 105 | yes | 5 | Vlan\_Prio5 |
| 106 | yes | 6 | Vlan\_Prio6 |
| 107 | yes | 7 | Vlan\_Prio7 |
| 100 | yes | 0 | Vlan\_Prio0 |
| 99 | no | - | Vlan\_untagged |
| 666 | - | - | DMA\_CH0\_GarbageCollector (packet that did not pass the VLAN or L3 filter) |

3.2 Predefined Tx packet length

This is the expected length on the wire!

|  |  |  |  |
| --- | --- | --- | --- |
| Length (byte) |  | IPv6 | Test Scenario Name |
| 111 |  | IP10 |  |
| 222 |  | IP11 |  |
| 333 |  | IP12 |  |

3.3 Predefined Trace messages

**RX:**

if (((frame\_len >= 99) && (frame\_len <= 107)) || (7 == queue))

{

printk("Harman: rx\_packet(%d) DMA\_CH(%d)\n", frame\_len, queue );

}

**if** ( 666 == frame\_len )

{

**if** ( 0 == queue )

{

HARMAN\_LOG("Harman: rx\_packet(%d) DMA\_CH(%d) 'else' packet for DMA Channel 0", frame\_len, queue );

}

**else**

{

HARMAN\_LOG("Harman: ERROR: rx\_packet(%d) DMA\_CH(%d) 'else' packet for DMA Channel 0", frame\_len, queue );

}

}

**TX:**

if ((111 == frame\_len) || (222 == frame\_len) || (333 == frame\_len))

{

printk("Harman: tx\_packet(%d) DMA\_CH(%d)\n", nopaged\_len, queue );

}

3.4 Expected register Content

Here's a selection of registers and their values for required settings

3.4.1 System Partition

|  |  |  |  |
| --- | --- | --- | --- |
| **Register** | **Value** | **Address@EVT0** | **Address@EVT1** |
| MAC\_RXQ\_CTRL0 | 0x0000AAAA | 0x17D800A0 | 0x17D800A0 |
| MAC\_RXQ\_CTRL1 | 0x00000070 | 0x17D800A4 | 0x17D800A4 |
| MAC\_RXQ\_CTRL2 | 0x10080400 | 0x17D800A8 | 0x17D800A8 |
| MAC\_RXQ\_CTRL3 | 0x00804020 | 0x17D800AC | 0x17D800AC |
| MAC\_PACKET\_FILTER | 0x80110010 | 0x17D80008 | 0x17D80008 |
| MTL\_RXQ\_DMA\_MAP0 | 0x13121110 | 0x17D80C30 | 0x17D82030 |
| MTL\_RXQ\_DMA\_MAP1 | 0x07161514 | 0x17D80C34 | 0x17D82034 |
|  |  |  |  |
|  |  |  |  |

3.4.2 Android Partition

|  |  |  |  |
| --- | --- | --- | --- |
| **Register** | **Value** | **Address@EVT0** | **Address@EVT1** |
| MAC\_RXQ\_CTRL0 | 0x0000AAAA | 0x17D900A0 | 0x17DA00A0 |
| MAC\_RXQ\_CTRL1 | 0x00000000 | 0x17D900A4 | 0x17DA00A4 |
| MAC\_RXQ\_CTRL2 | 0x08040200 | 0x17D900A8 | 0x17DA00A8 |
| MAC\_RXQ\_CTRL3 | 0x80402010 | 0x17D900AC | 0x17DA00AC |
| MAC\_PACKET\_FILTER | 0x80010010 | 0x17D90008 | 0x17DA0008 |
| MTL\_RXQ\_DMA\_MAP0 | 0x13121110 | 0x17D90C30 | 0x17DA2030 |
| MTL\_RXQ\_DMA\_MAP1 | 0x17161514 | 0x17D90C34 | 0x17DA2034 |
|  |  |  |  |
|  |  |  |  |

3.5 Expected VLAN Filter Settings

That discarded packets are routed to DMA channel 0 is confiugred with the RA flag in the packet filter. RA flag is bit 31.

3.5.1 System Partition

[ 100.938498] Harman:------------------------------------------------------------

[ 100.938505] Harman: dwmac5\_set\_l3l4\_vlan\_entry(0) vlan\_tag\_data = 0x0301000e

[ 100.948149] Harman: dwmac5\_show\_l3l4\_vlan\_entry(0) vlan\_tag\_ctrl = 0x00000000

[ 100.977210] Harman:------------------------------------------------------------

[ 100.977217] Harman: dwmac5\_set\_l3l4\_vlan\_entry(1) vlan\_tag\_data = 0x0301200e

[ 100.986854] Harman: dwmac5\_show\_l3l4\_vlan\_entry(1) vlan\_tag\_ctrl = 0x00000004

[ 101.015925] Harman:------------------------------------------------------------

[ 101.015931] Harman: dwmac5\_set\_l3l4\_vlan\_entry(2) vlan\_tag\_data = 0x0501400e

[ 101.025564] Harman: dwmac5\_show\_l3l4\_vlan\_entry(2) vlan\_tag\_ctrl = 0x00000008

[ 101.054641] Harman:------------------------------------------------------------

[ 101.054647] Harman: dwmac5\_set\_l3l4\_vlan\_entry(3) vlan\_tag\_data = 0x0501600e

[ 101.064281] Harman: dwmac5\_show\_l3l4\_vlan\_entry(3) vlan\_tag\_ctrl = 0x0000000c

[ 101.093355] Harman:------------------------------------------------------------

[ 101.093362] Harman: dwmac5\_set\_l3l4\_vlan\_entry(4) vlan\_tag\_data = 0x0701800e

[ 101.102994] Harman: dwmac5\_show\_l3l4\_vlan\_entry(4) vlan\_tag\_ctrl = 0x00000010

[ 101.132070] Harman:------------------------------------------------------------

[ 101.132076] Harman: dwmac5\_set\_l3l4\_vlan\_entry(5) vlan\_tag\_data = 0x0701a00e

[ 101.141709] Harman: dwmac5\_show\_l3l4\_vlan\_entry(5) vlan\_tag\_ctrl = 0x00000014

[ 101.170785] Harman:------------------------------------------------------------

[ 101.170791] Harman: dwmac5\_set\_l3l4\_vlan\_entry(6) vlan\_tag\_data = 0x0901c00e

[ 101.180424] Harman: dwmac5\_show\_l3l4\_vlan\_entry(6) vlan\_tag\_ctrl = 0x00000018

[ 101.209500] Harman:------------------------------------------------------------

[ 101.209506] Harman: dwmac5\_set\_l3l4\_vlan\_entry(7) vlan\_tag\_data = 0x0901e00e

[ 101.219138] Harman: dwmac5\_show\_l3l4\_vlan\_entry(7) vlan\_tag\_ctrl = 0x0000001c

3.5.2 Android Partition

[ 151.076101] Harman:------------------------------------------------------------

[ 151.076109] Harman: dwmac5\_set\_l3l4\_vlan\_entry(0) vlan\_tag\_data = 0x03010003

[ 151.085750] Harman: dwmac5\_show\_l3l4\_vlan\_entry(0) vlan\_tag\_ctrl = 0x00000000

[ 151.076101] Harman:------------------------------------------------------------

[ 151.076109] Harman: dwmac5\_set\_l3l4\_vlan\_entry(1) vlan\_tag\_data = 0x03012003

[ 151.085750] Harman: dwmac5\_show\_l3l4\_vlan\_entry(1) vlan\_tag\_ctrl = 0x00000000

[ 151.114815] Harman:------------------------------------------------------------

[ 151.114820] Harman: dwmac5\_set\_l3l4\_vlan\_entry(2) vlan\_tag\_data = 0x05014003

[ 151.124457] Harman: dwmac5\_show\_l3l4\_vlan\_entry(2) vlan\_tag\_ctrl = 0x00000004

[ 151.153531] Harman:------------------------------------------------------------

[ 151.153536] Harman: dwmac5\_set\_l3l4\_vlan\_entry(3) vlan\_tag\_data = 0x07016003

[ 151.163173] Harman: dwmac5\_show\_l3l4\_vlan\_entry(3) vlan\_tag\_ctrl = 0x00000008

[ 151.192245] Harman:------------------------------------------------------------

[ 151.192250] Harman: dwmac5\_set\_l3l4\_vlan\_entry(4) vlan\_tag\_data = 0x09018003

[ 151.201887] Harman: dwmac5\_show\_l3l4\_vlan\_entry(4) vlan\_tag\_ctrl = 0x0000000c

[ 151.230959] Harman:------------------------------------------------------------

[ 151.230963] Harman: dwmac5\_set\_l3l4\_vlan\_entry(5) vlan\_tag\_data = 0x0b01a003

[ 151.240599] Harman: dwmac5\_show\_l3l4\_vlan\_entry(5) vlan\_tag\_ctrl = 0x00000010

[ 151.269673] Harman:------------------------------------------------------------

[ 151.269677] Harman: dwmac5\_set\_l3l4\_vlan\_entry(6) vlan\_tag\_data = 0x0d01c003

[ 151.279316] Harman: dwmac5\_show\_l3l4\_vlan\_entry(6) vlan\_tag\_ctrl = 0x00000014

[ 151.308388] Harman:------------------------------------------------------------

[ 151.308393] Harman: dwmac5\_set\_l3l4\_vlan\_entry(7) vlan\_tag\_data = 0x0f01e003

[ 151.318031] Harman: dwmac5\_show\_l3l4\_vlan\_entry(7) vlan\_tag\_ctrl = 0x00000018

3.6 Expected IP Filter Settings

3.6.1 System Partition

[ 101.272432] Harman:------------------------------------------------------------

[ 101.272440] Harman: readl 0 GMAC\_L3\_L4\_CONTROL = 0x15000011

[ 101.290837] Harman: readl 0 GMAC\_L3\_IPV6\_31\_0 = 0xFF140000

[ 101.299606] Harman: readl 0 GMAC\_L3\_IPV6\_63\_32 = 0x00000000

[ 101.308371] Harman: readl 0 GMAC\_L3\_IPV6\_95\_64 = 0x00000000

[ 101.317138] Harman: readl 0 GMAC\_L3\_IPV6\_127\_96 = 0x00020014

[ 101.325904] Harman:------------------------------------------------------------

[ 101.325911] Harman: readl 1 GMAC\_L3\_L4\_CONTROL = 0x15000011

[ 101.342484] Harman: readl 1 GMAC\_L3\_IPV6\_31\_0 = 0xFF140000

[ 101.349430] Harman: readl 1 GMAC\_L3\_IPV6\_63\_32 = 0x00000000

[ 101.356373] Harman: readl 1 GMAC\_L3\_IPV6\_95\_64 = 0x00000000

[ 101.363319] Harman: readl 1 GMAC\_L3\_IPV6\_127\_96 = 0x00020016

[ 101.370261] Harman:------------------------------------------------------------

[ 101.370269] Harman: readl 2 GMAC\_L3\_L4\_CONTROL = 0x16000011

[ 101.386842] Harman: readl 2 GMAC\_L3\_IPV6\_31\_0 = 0xFF140000

[ 101.393786] Harman: readl 2 GMAC\_L3\_IPV6\_63\_32 = 0x00000000

[ 101.400730] Harman: readl 2 GMAC\_L3\_IPV6\_95\_64 = 0x00000000

[ 101.407676] Harman: readl 2 GMAC\_L3\_IPV6\_127\_96 = 0x00020006

[ 101.414617] Harman:------------------------------------------------------------

[ 101.414624] Harman: readl 3 GMAC\_L3\_L4\_CONTROL = 0x00000031

[ 101.431198] Harman: readl 3 GMAC\_L3\_IPV6\_31\_0 = 0xFF140000

[ 101.438143] Harman: readl 3 GMAC\_L3\_IPV6\_63\_32 = 0x00000000

[ 101.445088] Harman: readl 3 GMAC\_L3\_IPV6\_95\_64 = 0x00000000

[ 101.452032] Harman: readl 3 GMAC\_L3\_IPV6\_127\_96 = 0x00020014

[ 101.458976] Harman:------------------------------------------------------------

[ 101.458983] Harman: readl 4 GMAC\_L3\_L4\_CONTROL = 0x00000031

[ 101.475556] Harman: readl 4 GMAC\_L3\_IPV6\_31\_0 = 0xFF140000

[ 101.482504] Harman: readl 4 GMAC\_L3\_IPV6\_63\_32 = 0x00000000

[ 101.489444] Harman: readl 4 GMAC\_L3\_IPV6\_95\_64 = 0x00000000

[ 101.496390] Harman: readl 4 GMAC\_L3\_IPV6\_127\_96 = 0x00020016

[ 101.503332] Harman:------------------------------------------------------------

[ 101.503339] Harman: readl 5 GMAC\_L3\_L4\_CONTROL = 0x00000031

[ 101.519912] Harman: readl 5 GMAC\_L3\_IPV6\_31\_0 = 0xFF140000

[ 101.526857] Harman: readl 5 GMAC\_L3\_IPV6\_63\_32 = 0x00000000

[ 101.533801] Harman: readl 5 GMAC\_L3\_IPV6\_95\_64 = 0x00000000

[ 101.540747] Harman: readl 5 GMAC\_L3\_IPV6\_127\_96 = 0x00020006

3.6.2 Android Partition

not needed

3.7 Promiscuous Mode

All tests have to be repeated 2 times, first with promiscuous mode disbaled (default), second with promiscuous mode enabled. The enable has to be done by running tcpdump in the background. Setting the PR flag in the register is not allowed!

4. Recommended work flow

4.1 With DMA CH0 serviced from Standard Driver

That DMA channel 0 is serviced from outside has minor priority! Highest priority have the correct settings of queues and DMA channels according to the final architecture.

1. Create your test stimulation according to 3.1
2. Update driver receive function with logs from 3.2
3. test without harman-synopsys.ko (delete it on the target). So all 8 queues/DMA channels can be used
4. start with the Android setting. Test it on ystem partition if this is easier for you!
5. test queue settings first. "MAC to queue routing"
6. Then test "Queue to DMA mapping" with the features from Appendix D
7. Queue settings System Partition ( all 8 ch, without harman-synopsys.ko)
8. DMA mapping System parttion( all 8 ch, without harman-synopsys.ko)
9. Final System setting with harman-synopsys.ko

4.1 With DMA CH0 serviced from Outside

Garbage Collector has to be inplemented from ADV\_NET. integrated in ptpDiagnosis app. Only service of DMA RX channel and print parts of received packets like done in the test feature for harman.synopsys.

5. Gerds Sandbox

Latest test code from Gerd is attahced in the email as perforce access is not possible at the moment for me.

The provide code my not be up to the latest changes of stmmac!

Look into harman.h/ harmanHal.h for advanced debug settings and trace macros!

**HINT:**

all my changes are encapsulated with "HARMAN" like define or macro

To use all 8 queues set "VARIANT\_EVT1" in harman.h (now default)

Explore at which code point chapter 3.2 is reliazed

**HINT:**

The topic with queue to DMA mapping is not solved in this code.

use nicsSynopsystest -S81 and -S84 to display the settings.

Should be for 0x17d800a8 --> 0x08040201

Should be for 0x17d800aA --> 0x80402010

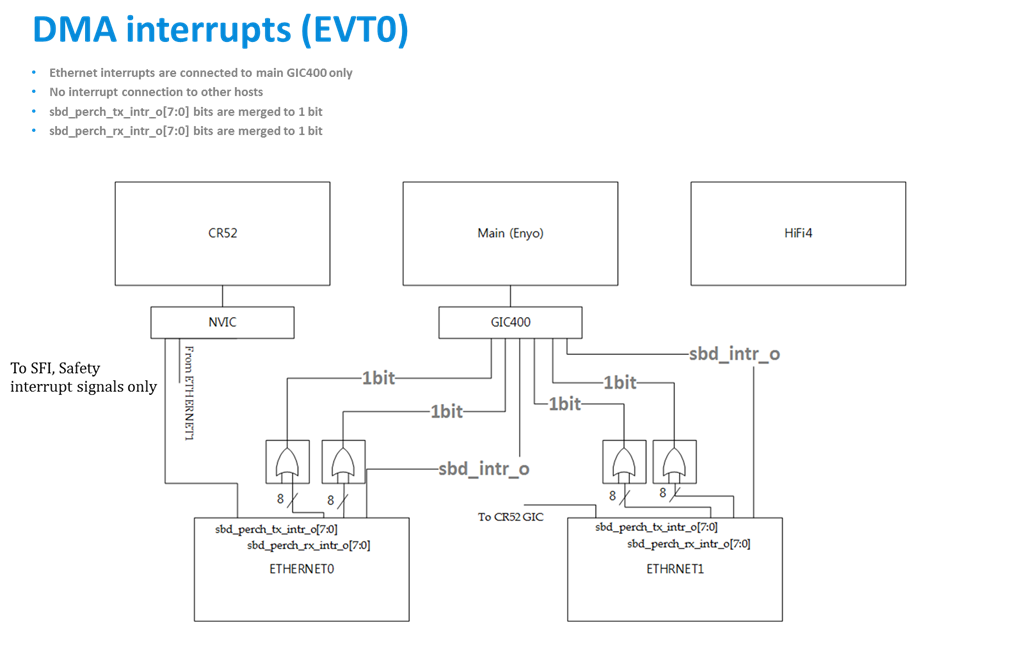
--> you can set it during runtime with devmem!

Appendix

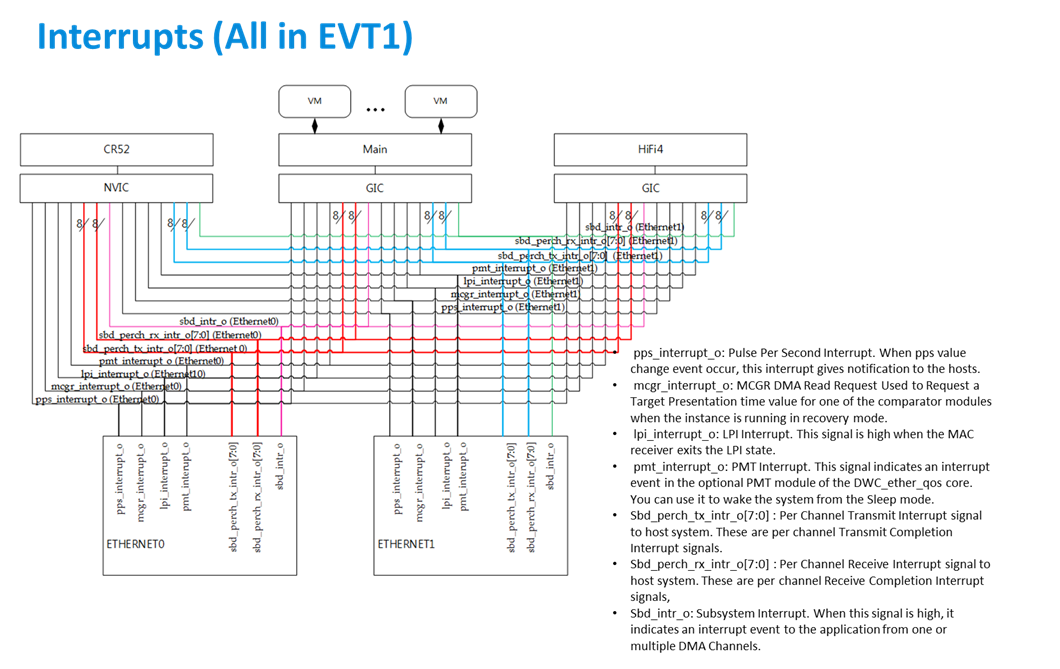
B Ethernet Restrictions: Samsung ExynosAuto Samples

In the Samsung EVT0 sample the interrupt lines of the EMAC module are available only rudimentally. This has consequences to the complete Ethernet Driver Architecture.

B.1 EVT0 (Q4 2018)



B.1 EVT1 (Q4 2019)



C Ethernet Restrictions: EMAC FIFO size used by queues

C.1 EVT0 (Q4 2018)

Each queues needs a part of the FIFO memory to be assigned. The size should host at least one maximum packet size. Recommended size is 2048 byte.

EVT0 has only 8KByte of FIFO. If all 8 queues would be used, it's only possible to assign 1KB. Therefore in the project only 4 queues are used with 2KB assigned.

If the packet does not fit into assigned queue fifo size, some features like checksum offloading do not work correctly. But, as said, dependent on the received packet length.

**HINT:**

For testing purpose all 8 queues and DMA channels could be used if test packets are below 1 KB.

Additionally, you can disable cheksum offloading, then the packets will be received and forwarded in any case!

C.2 EVT1 (Q4 2019)

EVT1 has 16KByte of FIFO. So all 8 queues can be used if 2KB are assigned.

D EMAC queue and dma channel explanation

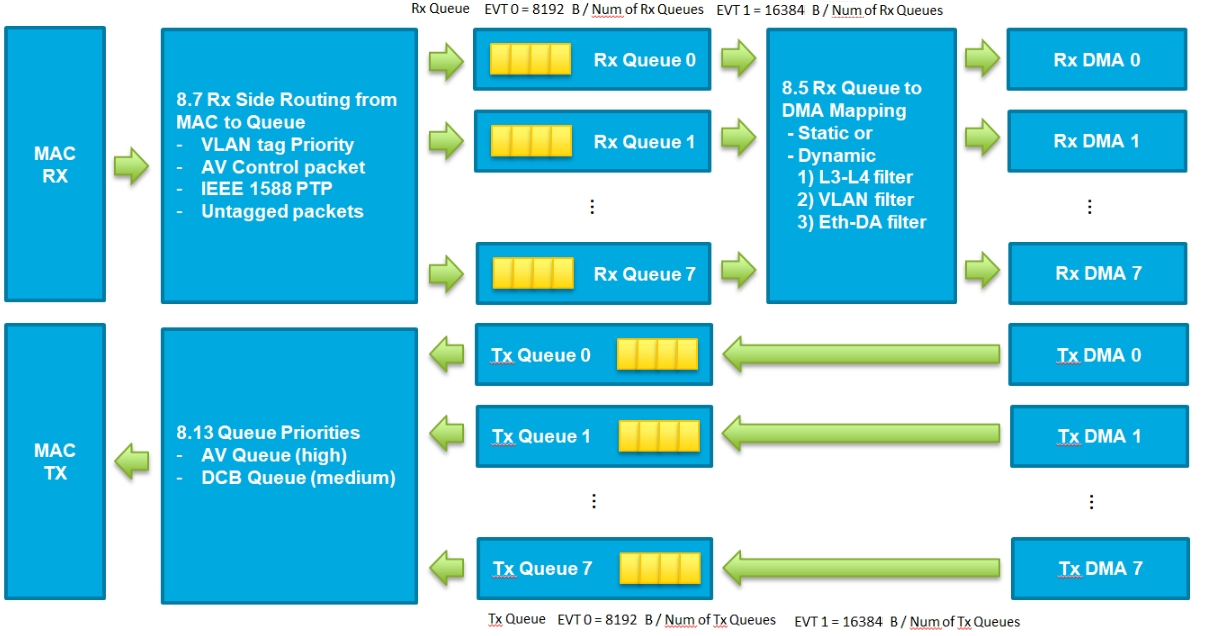
Jung-Sik Lee, Samsung, 2018/12/14

**[Samsung, 20181214]**

**RX / Tx flow concept on Synopsys Ethernet IP (MAC) in EA9**

**1. Rx: Static or Dynamic Mapping is supported between Rx Queues and DMA CHs.**

**2. Tx:  Queue Priority algorithm can be selected.**



**[Samsung, 20181214]  You can select Static mapping or Dynamic mapping.**

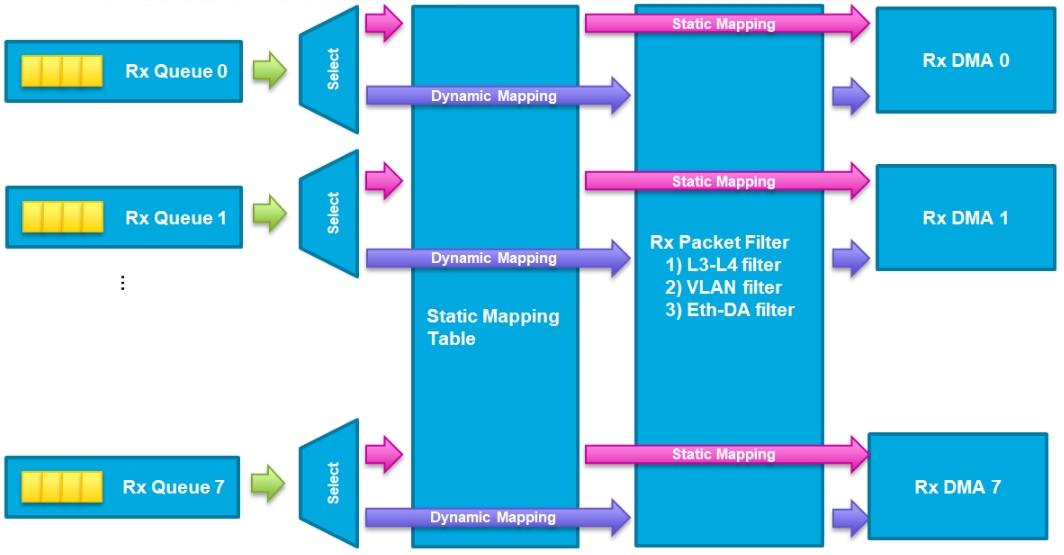
**If you select Dynamic mapping, you can use Rx Packet filter to select DMA CH.**

**- Eth-DA (MAC-DA) filter: MAC address is used. (You can make this filter with maximum 32 MAC addresses.)**

**- VLAN filter: You can make this filter with maximum 32 VLAN Tags.**

**- L3-L4 filter: IP address or port is used. (** **You can make this filter with maximum 8. For example,  you set up 4 IP addresses and 4 port numbers, or 8 IP addresses, or 8 port numbers, or etc.)**

**- A packet is failed to pass any filter, the packet will be discard.**



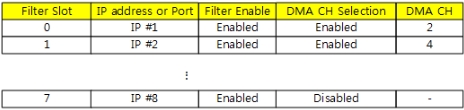
**[Samsung, 20181214] Rx DMA CH Selection concpet. (The steps are opposite from filter's steps.)**

**1. If A packet info matches L3-L4's filters and L3-L4's Filters' DMA Selection is enabled, the matched packet will go to defined Rx DMA CH by MAC\_L3\_L4\_Control(#i) register.**

**- L3-L4 Filter: IPv6 Addresses (Dest or Src addr), L4 Filter: port number of Dest or Src**

**- Maximum filter slots for L3-L4 is 8.**

**- Each filter has a DMA selection register. So each IP address can select a Rx DMA CH.**

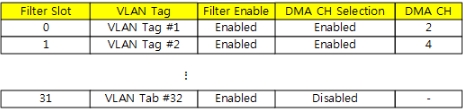
****

**2. VLAN filters info is using, if DMA selection is not enabled in L3-L4's filters. Same as #1.**

**- maximum filter slots for is 32.**

**- You can use  both VLAN tag ID (12bit) or VLAN tag (16bit) in filter.**

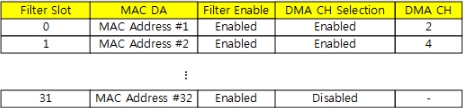
**- Each filter has a DMA selection register. So each VLAN can select a Rx DMA CH.**

****

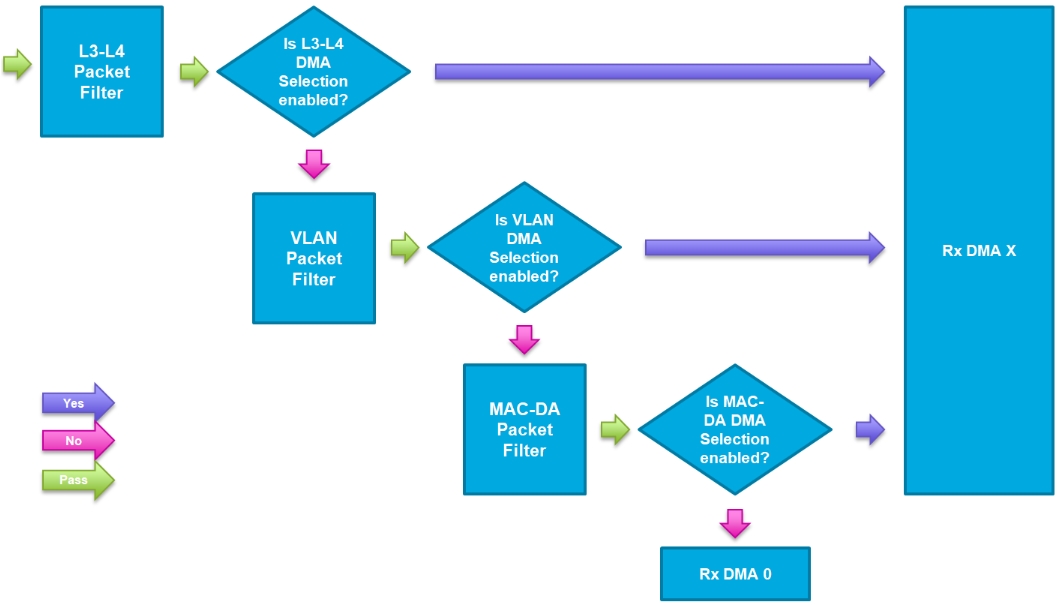
**3. MAC-DA(Eth-DA) filters info is using, if DMA selection is not enabled in VLAN filters. Same as #1.**

**- maximum filter slots for is 32.**

**- Each filter has a DMA selection register. So each MAC DA can select a Rx DMA CH.**

****

**4. If last filter's DMA selection is disabled, the packet will go to Rx DMA CH 0.**



E EMAC Device Tree Configuration options (EVT0 "stmmac")

\* STMicroelectronics 10/100/1000 Ethernet driver (GMAC)

Required properties:

- compatible: Should be "snps,dwmac-<ip\_version>", "snps,dwmac"

For backwards compatibility: "st,spear600-gmac" is also supported.

- reg: Address and length of the register set for the device

- interrupt-parent: Should be the phandle for the interrupt controller

that services interrupts for this device

- interrupts: Should contain the STMMAC interrupts

- interrupt-names: Should contain a list of interrupt names corresponding to

the interrupts in the interrupts property, if available.

Valid interrupt names are:

- "macirq" (combined signal for various interrupt events)

- "eth\_wake\_irq" (the interrupt to manage the remote wake-up packet detection)

- "eth\_lpi" (the interrupt that occurs when Tx or Rx enters/exits LPI state)

- phy-mode: See ethernet.txt file in the same directory.

- snps,reset-gpio gpio number for phy reset.

- snps,reset-active-low boolean flag to indicate if phy reset is active low.

- snps,reset-delays-us is triplet of delays

The 1st cell is reset pre-delay in micro seconds.

The 2nd cell is reset pulse in micro seconds.

The 3rd cell is reset post-delay in micro seconds.

Optional properties:

- resets: Should contain a phandle to the STMMAC reset signal, if any

- reset-names: Should contain the reset signal name "stmmaceth", if a

reset phandle is given

- max-frame-size: See ethernet.txt file in the same directory

- clocks: If present, the first clock should be the GMAC main clock and

the second clock should be peripheral's register interface clock. Further

clocks may be specified in derived bindings.

- clock-names: One name for each entry in the clocks property, the

first one should be "stmmaceth" and the second one should be "pclk".

- ptp\_ref: this is the PTP reference clock; in case of the PTP is available

this clock is used for programming the Timestamp Addend Register. If not

passed then the system clock will be used and this is fine on some

platforms.

- tx-fifo-depth: See ethernet.txt file in the same directory

- rx-fifo-depth: See ethernet.txt file in the same directory

- snps,pbl Programmable Burst Length (tx and rx)

- snps,txpbl Tx Programmable Burst Length. Only for GMAC and newer.

If set, DMA tx will use this value rather than snps,pbl.

- snps,rxpbl Rx Programmable Burst Length. Only for GMAC and newer.

If set, DMA rx will use this value rather than snps,pbl.

- snps,no-pbl-x8 Don't multiply the pbl/txpbl/rxpbl values by 8.

For core rev < 3.50, don't multiply the values by 4.

- snps,aal Address-Aligned Beats

- snps,fixed-burst Program the DMA to use the fixed burst mode

- snps,mixed-burst Program the DMA to use the mixed burst mode

- snps,force\_thresh\_dma\_mode Force DMA to use the threshold mode for

both tx and rx

- snps,force\_sf\_dma\_mode Force DMA to use the Store and Forward

mode for both tx and rx. This flag is

ignored if force\_thresh\_dma\_mode is set.

- snps,en-tx-lpi-clockgating Enable gating of the MAC TX clock during

TX low-power mode

- snps,multicast-filter-bins: Number of multicast filter hash bins

supported by this device instance

- snps,perfect-filter-entries: Number of perfect filter entries supported

by this device instance

- snps,ps-speed: port selection speed that can be passed to the core when

PCS is supported. For example, this is used in case of SGMII

and MAC2MAC connection.

- snps,tso: this enables the TSO feature otherwise it will be managed by

MAC HW capability register. Only for GMAC4 and newer.

- AXI BUS Mode parameters: below the list of all the parameters to program the

AXI register inside the DMA module:

- snps,lpi\_en: enable Low Power Interface

- snps,xit\_frm: unlock on WoL

- snps,wr\_osr\_lmt: max write outstanding req. limit

- snps,rd\_osr\_lmt: max read outstanding req. limit

- snps,kbbe: do not cross 1KiB boundary.

- snps,blen: this is a vector of supported burst length.

- snps,fb: fixed-burst

- snps,mb: mixed-burst

- snps,rb: rebuild INCRx Burst

- mdio: with compatible = "snps,dwmac-mdio", create and register mdio bus.

- Multiple RX Queues parameters: below the list of all the parameters to

configure the multiple RX queues:

- snps,rx-queues-to-use: number of RX queues to be used in the driver

- Choose one of these RX scheduling algorithms:

- snps,rx-sched-sp: Strict priority

- snps,rx-sched-wsp: Weighted Strict priority

- For each RX queue

- Choose one of these modes:

- snps,dcb-algorithm: Queue to be enabled as DCB

- snps,avb-algorithm: Queue to be enabled as AVB

- snps,map-to-dma-channel: Channel to map

**HINT:** GZ: this is for fixed queue to channel mapping. In the register itself each queue has it's only flag. E. g. queue x routes to DMA Ch 0 and 7: 0x81

So the assigned value has to be according this flag scheme.

**HINT:** GZ: the older driver versions had a bug for queue 4 .. 7: check fucntion " **dwmac4\_rx\_queue\_priority()**" if register GMAC\_RXQ\_CTRL2 + 3 is not set properly

- Specifiy specific packet routing:

- snps,route-avcp: AV Untagged Control packets

- snps,route-ptp: PTP Packets

- snps,route-dcbcp: DCB Control Packets

- snps,route-up: Untagged Packets

- snps,route-multi-broad: Multicast & Broadcast Packets

- snps,priority: RX queue priority (Range: 0x0 to 0xF)

- Multiple TX Queues parameters: below the list of all the parameters to

configure the multiple TX queues:

- snps,tx-queues-to-use: number of TX queues to be used in the driver

- Choose one of these TX scheduling algorithms:

- snps,tx-sched-wrr: Weighted Round Robin

- snps,tx-sched-wfq: Weighted Fair Queuing

- snps,tx-sched-dwrr: Deficit Weighted Round Robin

- snps,tx-sched-sp: Strict priority

- For each TX queue

- snps,weight: TX queue weight (if using a DCB weight algorithm)

- Choose one of these modes:

- snps,dcb-algorithm: TX queue will be working in DCB

- snps,avb-algorithm: TX queue will be working in AVB

[Attention] Queue 0 is reserved for legacy traffic

and so no AVB is available in this queue.

- Configure Credit Base Shaper (if AVB Mode selected):

- snps,send\_slope: Send Slope Credit value

- snps,idle\_slope: Idle Slope Credit value

- snps,high\_credit: High Credit value

- snps,low\_credit: Low Credit value

- snps,priority: TX queue priority (Range: 0x0 to 0xF)

Examples:

stmmac\_axi\_setup: stmmac-axi-config {

snps,wr\_osr\_lmt = <0xf>;

snps,rd\_osr\_lmt = <0xf>;

snps,blen = <256 128 64 32 0 0 0>;

};

mtl\_rx\_setup: rx-queues-config {

snps,rx-queues-to-use = <1>;

snps,rx-sched-sp;

queue0 {

snps,dcb-algorithm;

snps,map-to-dma-channel = <0x0>;

snps,priority = <0x0>;

};

};

mtl\_tx\_setup: tx-queues-config {

snps,tx-queues-to-use = <2>;

snps,tx-sched-wrr;

queue0 {

snps,weight = <0x10>;

snps,dcb-algorithm;

snps,priority = <0x0>;

};

queue1 {

snps,avb-algorithm;

/\*

\* Example AVB parameters based on:

\* Allocated Bandwidth: 40%

\* Maximum Frame size: 1000 bytes

\* Maximum Interference size: 1500 bytes

\* Port Transmit Rate: 8

\* Scaling Factor: 1024

\*/

snps,idle\_slope = <0xCCC>;

snps,send\_slope = <0x1333>;

snps,high\_credit = <0x4B0000>;

snps,low\_credit = <0xFFB50000>;

snps,priority = <0x1>;

};

};

gmac0: ethernet@e0800000 {

compatible = "st,spear600-gmac";

reg = <0xe0800000 0x8000>;

interrupt-parent = <&vic1>;

interrupts = <24 23 22>;

interrupt-names = "macirq", "eth\_wake\_irq", "eth\_lpi";

mac-address = [000000000000]; /\* Filled in by U-Boot \*/

max-frame-size = <3800>;

phy-mode = "gmii";

snps,multicast-filter-bins = <256>;

snps,perfect-filter-entries = <128>;

rx-fifo-depth = <16384>;

tx-fifo-depth = <16384>;

clocks = <&clock>;

clock-names = "stmmaceth";

snps,axi-config = <&stmmac\_axi\_setup>;

mdio0 {

#address-cells = <1>;

#size-cells = <0>;

compatible = "snps,dwmac-mdio";

phy1: ethernet-phy@0 {

};

};

snps,mtl-rx-config = <&mtl\_rx\_setup>;

snps,mtl-tx-config = <&mtl\_tx\_setup>;

};

F Mutli TX DMA Channel usage

Samsung reply:

>jongho7.park@samsung.com added a comment - 2020-04-23T13:47:53.000+0900

|  |  |
| --- | --- |
| AUTOMOTIVE-1233 | Seqos Dws ethernet controller Queue & DMA mapping requirement |

2. seqos driver creates NAPI for each dma channel. For tx Kernel selects NAPI for each socket randomly. The selected NAPI number is addressed at socket buffer. if you want to specify a queue number (NAPI), qdisc is needed.

Using qdisc, you can filter traffic and edit queue mapping. (ex. tc filter add dev eth0 parent 1: protocol all flower dst\_mac 91:e0:f0:00:fe:00 action skbedit queue\_mapping 3).