FTF FREESCALE TECHNOLOGY FORUM 2014

Data Path Acceleration Architecture (DPAA) Debug

FTF-NET-F0148

Kelly Johnson | Applications Engineering

APR.2014

freescale™

External Use

Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, Qorivva, SafeAssure, the SafeAssure logo, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SMARTMOS, Tower, TurboLink, UMEMS, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2014 Freescale Semiconductor, Inc.

Session Introduction

- This session will take an in-depth look at QorIQ Data Path Acceleration Architecture (DPAA) and how each component interacts with each other and the core. Discussion will focus on debugging common issues such as packet loss and queue manager enqueue rejections.

Session Objectives

- After completing this session you will be able to:
  - Find where missing packets are being dropped or lost in receive path
  - Determine reason for Queue Manager enqueue rejection
  - Identify SDK version of images used on target

Agenda

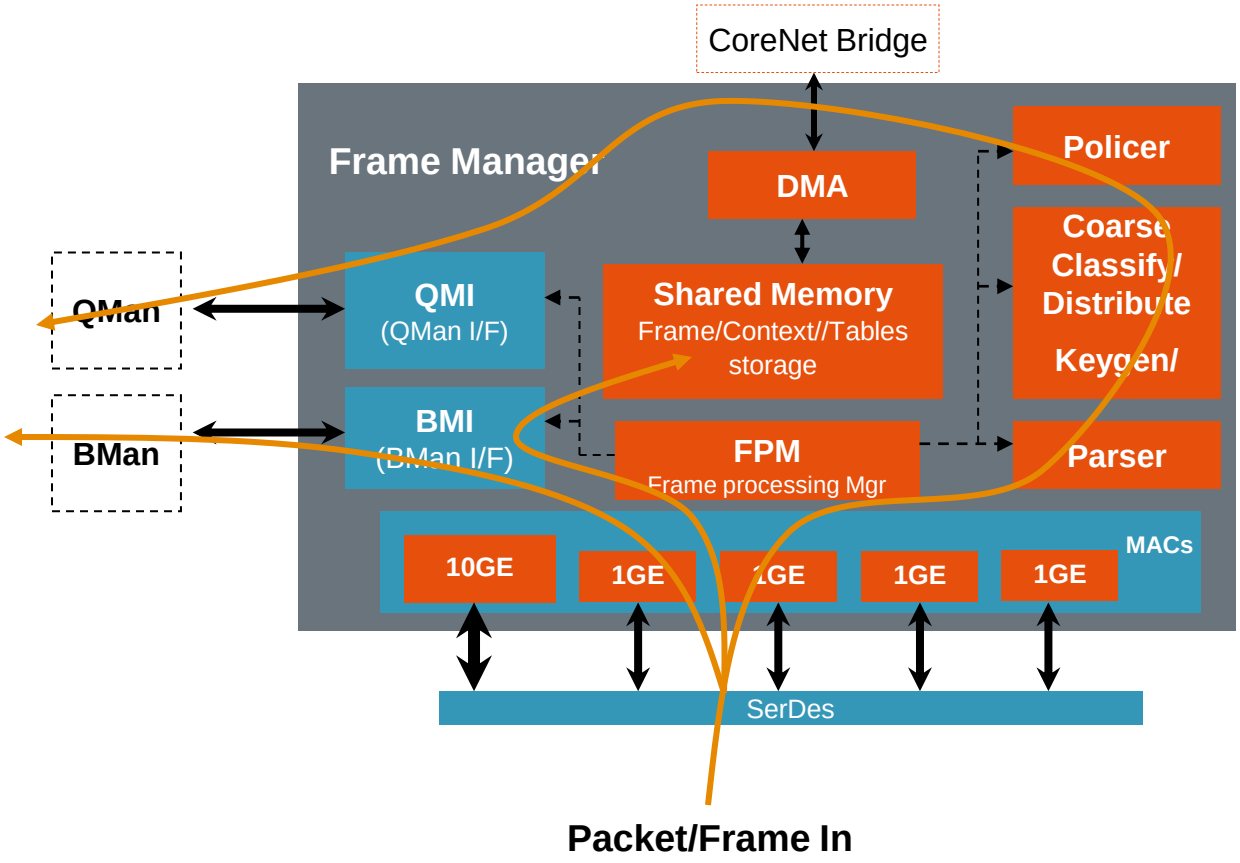
- Packet Loss
  - Frame Manager Statistics Counters
  - Linux Sysfs
- Queue Manager Enqueue Rejections
  - Frame Queue Descriptor
  - Linux Debugfs
- SDK Version Compatibility

Packet Loss

- Packets sent to the SoC but we have lost track of where they ended up. Missing some or all packets sent to the SoC
- What is Frame Manager doing with the packets it receives? Enqueueing the packets to Queue Manager? Dropping the packets?
- First check whether or not the packets made it through the MAC and all the way to a Frame Manager port
- Use Frame Manager RX port statistics counters registers
  - There's an easy way to do this when running Linux

Frame Manager (FMan)

- FMan supports a flexible pipeline of packet processing elements
- Frame data and per frame context is stored in internal memory while frame is processed
- Frame Processor Manager (FPM) “schedules” frames for processing by different elements to create appropriate pipeline
- Default pipeline configured for each port



Packet Loss

Frame Manager Buffer Manager Interface (BMI) Rx Port Statistics Counters (1/2)

- FMBM_RFRC – Receive Frame Counter
  - Total number of frames received on the Rx port
- FMBM_RBFC – Bad Frames Counter
  - Number of frames received on the Rx port with an error indication.
  - Error cause could be FCS error, MAC FIFO overflow, code error, etc.
  - These frames are discarded and not shown to receive queues, unless FMBM_RCFG[FDOVR] is set.
  - Rx FD Status Field “FPE” (frame physical error) bit will be set.
- FMBM_RFFC – Filter Frames Counter
  - Number of frames received on the Rx port that were filtered out by the parse and classify modules of the Fman.
  - See Rx FD Status Field.
  - These frames are discarded and not shown to receive queues, unless FMBM_RCFG[FDOVR] is set.

Frame Manager Buffer Manager Interface (BMI) Rx Port Statistics Counters (2/2)

- FMBM_RFDC – Frames Discard Counter
  - Number of frames received on the Rx port that were *not able to enter the receive queue system* due to WRED algorithm. Other reasons for enqueue reject may be tail drop, out of service FQ, etc.
- FMBM_RODC – Out of Buffers Discard Counter
  - Number of received frames that were discarded due to lack of external buffers.
- FMBM_RDBC – Buffer Deallocate Counter
  - Number of buffer deallocate operations. The counter increments whenever a buffer is returned to BMan pools.
- FMBM_RLFC – Large Frames Counter
  - Over size indication is marked when frame size exceeds the maximum configured in the corresponding MAC configuration register.

Frame Manager Queue Manager Interface (QMI)
Rx Port Register

- FMQM_PnETFC - PortID n Enqueue Total Frame Counter
  - Number of enqueue operations performed for this portID

- For further information regarding these registers, see Data Path Acceleration Architecture Reference Manual (DPAARM)

Linux Sysfs Support for Fman Rx Port Statistics

Example: FMan1 dTSEC5

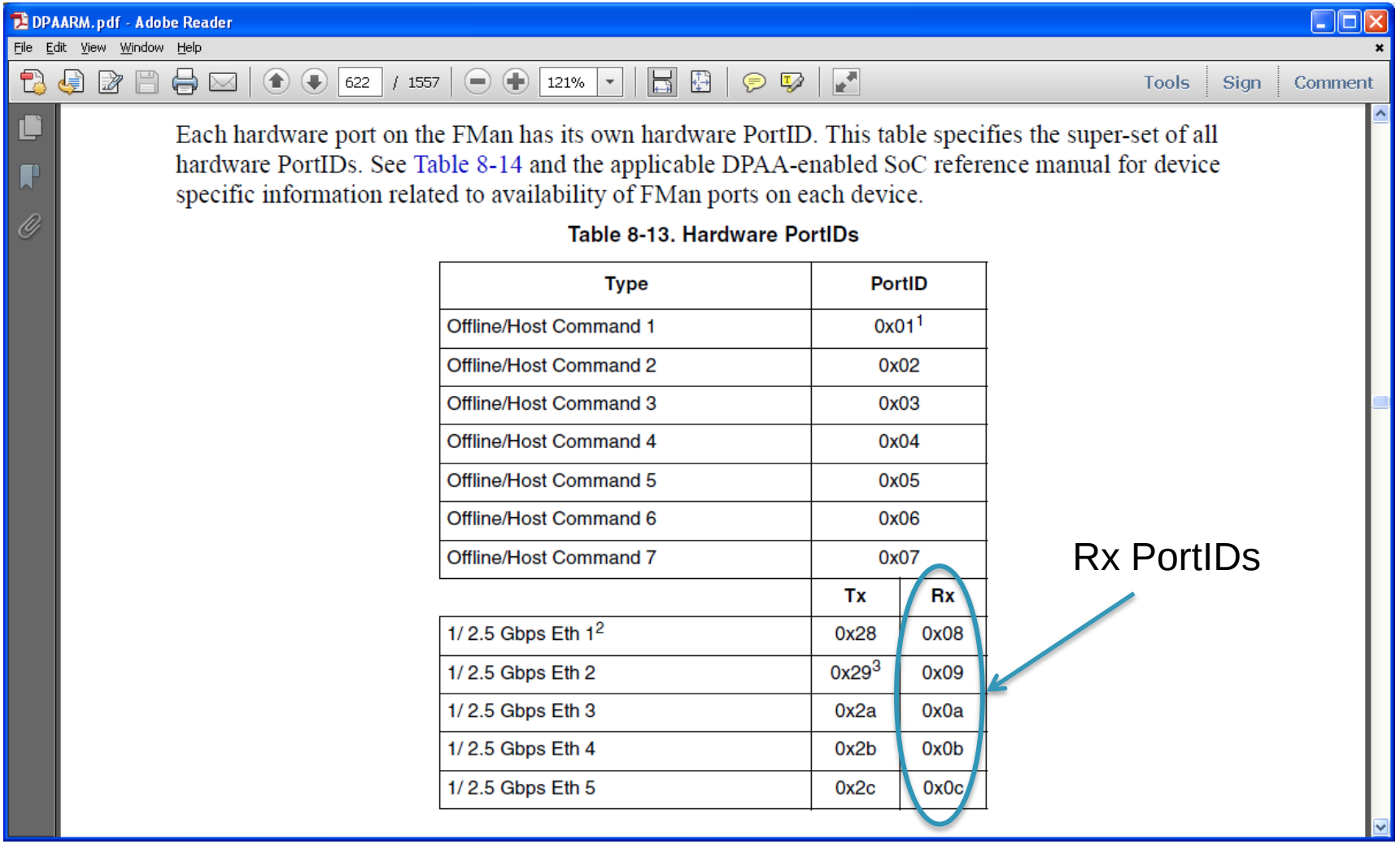
```
$ls /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/*
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_dealloc_buf
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_discard_frame
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_enq_total
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_frame
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_rx_bad_frame
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_rx_filter_frame
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_rx_large_frame
 /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/port_rx_out_of_buffers_
   discard

 $cat /sys/devices/ffe000000.soc/ffe400000.fman/ffe48c000.port/statistics/*
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
          fm0-port-rx4 counter: 0
```

CCSRBAR + FMan1 Base Address + HW Ports Base Address + HW Port ID

0xF_FE00_0000 + 0x40_0000 + 0x8_0000 + 0x0_C000 = 0xF_FE48_C000

Frame Manager Hardware PortIDs



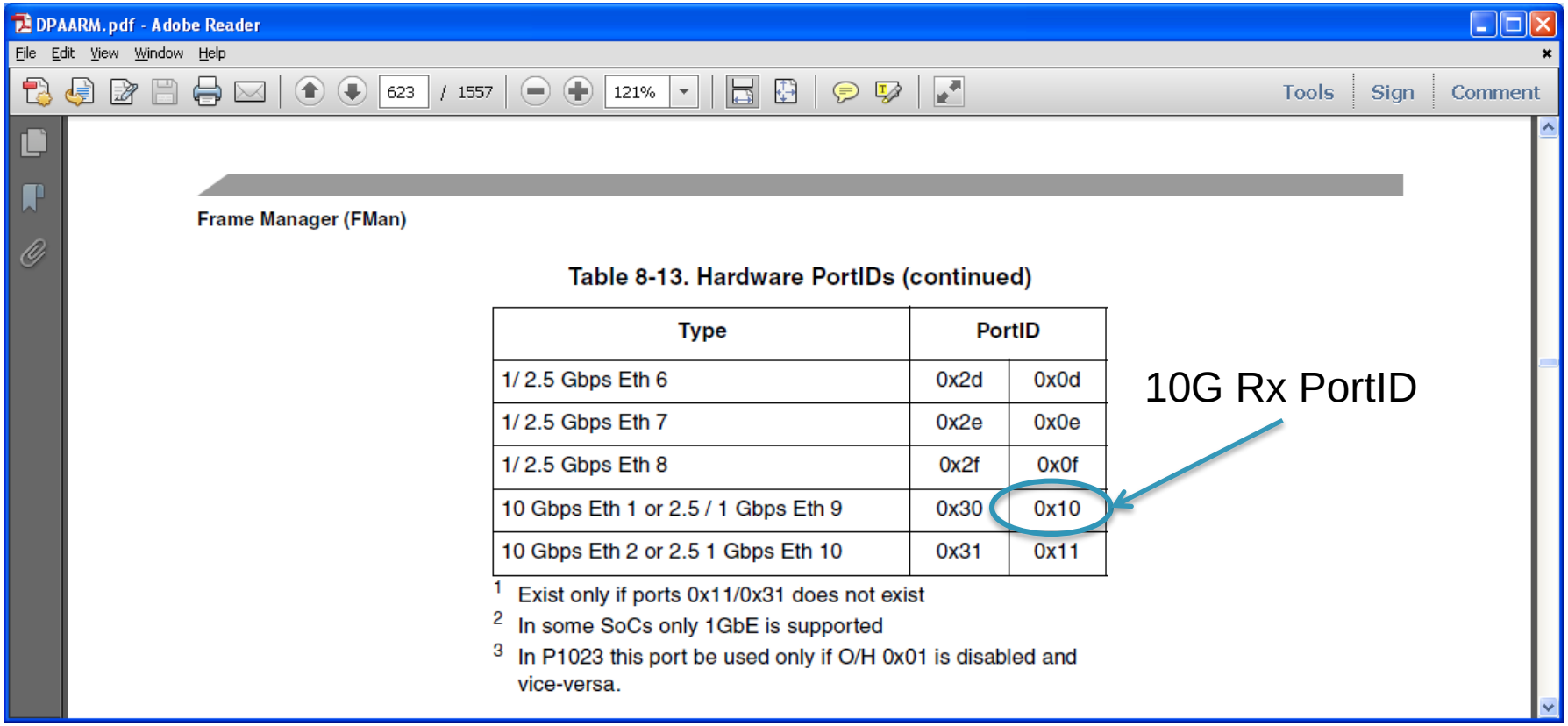
Each hardware port on the FMan has its own hardware PortID. This table specifies the super-set of all hardware PortIDs. See Table 8-14 and the applicable DPAA-enabled SoC reference manual for device specific information related to availability of FMan ports on each device.

Table 8-13. Hardware PortIDs

Type	PortID	
Offline/Host Command 1	0x01[1]	
Offline/Host Command 2	0x02	
Offline/Host Command 3	0x03	
Offline/Host Command 4	0x04	
Offline/Host Command 5	0x05	
Offline/Host Command 6	0x06	
Offline/Host Command 7	0x07	
	Tx	Rx
1/ 2.5 Gbps Eth 1[2]	0x28	0x08
1/ 2.5 Gbps Eth 2	0x29[3]	0x09
1/ 2.5 Gbps Eth 3	0x2a	0x0a
1/ 2.5 Gbps Eth 4	0x2b	0x0b
1/ 2.5 Gbps Eth 5	0x2c	0x0c

Rx PortIDs

Frame Manager Hardware PortIDs



Type	PortID	
1/ 2.5 Gbps Eth 6	0x2d	0x0d
1/ 2.5 Gbps Eth 7	0x2e	0x0e
1/ 2.5 Gbps Eth 8	0x2f	0x0f
10 Gbps Eth 1 or 2.5 / 1 Gbps Eth 9	0x30	0x10
10 Gbps Eth 2 or 2.5 1 Gbps Eth 10	0x31	0x11

1 Exist only if ports 0x11/0x31 does not exist
2 In some SoCs only 1GbE is supported
3 In P1023 this port be used only if O/H 0x01 is disabled and vice-versa.

FMan1 TGEC:
cat /sys/devices/ffe000000.soc/ffe400000.fman/ffe490000.port/statistics/*

Example: FMAN Rx Port Statistics – Discarded Frames

```
# cat /sys/devices/ffe000000.soc/ffe400000.fman/ffe488000.port/statistics/*
        FM 0 Port 0 counter: 71      port_dealloc_buf
        FM 0 Port 0 counter: 71      port_discard_frame
        FM 0 Port 0 counter: 495255  port_enq_total
        FM 0 Port 0 counter: 495255  port_frame
        FM 0 Port 0 counter: 0       port_rx_bad_frame
        FM 0 Port 0 counter: 0       port_rx_filter_frame
        FM 0 Port 0 counter: 0       port_rx_large_frame
        FM 0 Port 0 counter: 0       port_rx_out_of_buffers_discard
```

- FMBM_RFDC – Frames Discard Counter
  - Number of frames received on the Rx port that were not able to enter the receive queue system due to WRED algorithm. Other reasons for enqueue reject may be tail drop, out of service FQ, etc.

Example: FMAN Rx Port Statistics – Out of Buffers

```
cat /sys/devices/ffe000000.soc/ffe500000.fman/ffe590000.port/statistics/*
          fm1-port-rx6 counter: 0
          fm1-port-rx6 counter: 0
          fm1-port-rx6 counter: 71836626    port_enq_total
          fm1-port-rx6 counter: 71862438    port_frame
          fm1-port-rx6 counter: 0
          fm1-port-rx6 counter: 0
          fm1-port-rx6 counter: 0
          fm1-port-rx6 counter: 25628       port_rx_out_of_buffers_discard
```

- Running out of buffers points to congestion in the system

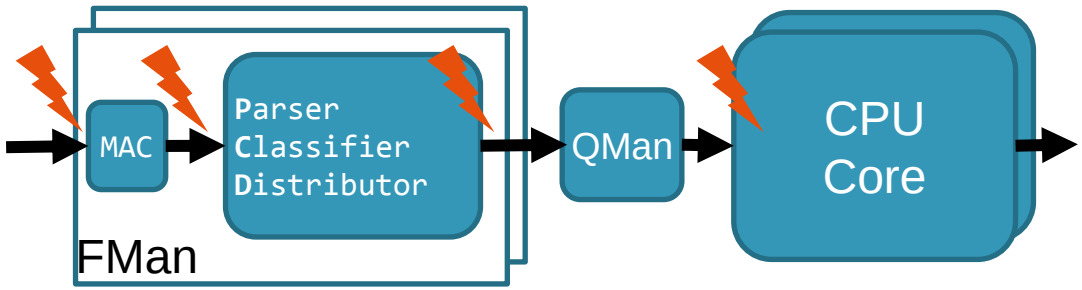
Example: FMAN Rx Port Statistics – No Dropped Packets

```
# cat /sys/devices/ffe000000.soc/ffe400000.fman/ffe489000.port/statistics/*
        FM 0 Port 1 counter: 0
        FM 0 Port 1 counter: 0
        FM 0 Port 1 counter: 714541   port_enq_total
        FM 0 Port 1 counter: 714541   port_frame
        FM 0 Port 1 counter: 0
        FM 0 Port 1 counter: 0
        FM 0 Port 1 counter: 0
        FM 0 Port 1 counter: 0
```

- All packets received on this Rx Port were enqueued to Queue Manager.
- After checking FMan Port statistics, if packets are still unaccounted for, then check MAC counter registers.
- We've discussed Rx Counters, there's also a set of FMan Tx Port Statistics registers.

MAC Layer Receive Counters and Offsets - mEMAC

- 0x100 REOCTL Receive Ethernet Octets
- 0x108 ROCT Receive Octets
- 0x110 RALN Receive Alignment Error
- 0x118 RXPF Receive Valid Pause Frame
- 0x120 RFRM Receive Frame
- 0x128 RFCS Receive FCS Error
- 0x130 RVLAN Receive VLAN Frame
- 0x138 RERR Receive Frame Error
- 0x140 RUCA Receive Unicast Frame
- 0x148 RMCA Receive Multicast Frame
- 0x150 RBCA Receive Broadcast Frame
- 0x158 RDRP Receive Dropped Packets
- 0x160 RPKT Receive Packets
- 0x168 RUND Receive Undersized Packet
- 0x170 R64 Receive 64 Octet Packet
- 0x178 R127 Receive 65-127 Octet Packet
- 0x180 R255 Receive 128-255 Octet Packet
- 0x188 R511 Receive 256-511 Octet Packet
- 0x190 R1023 Receive 512-1023 Octet Packet
- 0x198 R1518 Receive 1024-1518 Octet Packet
- 0x1A0 R1519X Receive 1519-Max Octet Packet
- 0x1A8 ROVR Receive Oversized Packet
- 0x1B0 RJBR Receive Jabber Packet
- 0x1B8 RFRG Receive Fragment Packet
- 0x1C0 RCNP Receive Control Packet
- 0x1C8 RDRNTP Receive Dropped Not Truncated



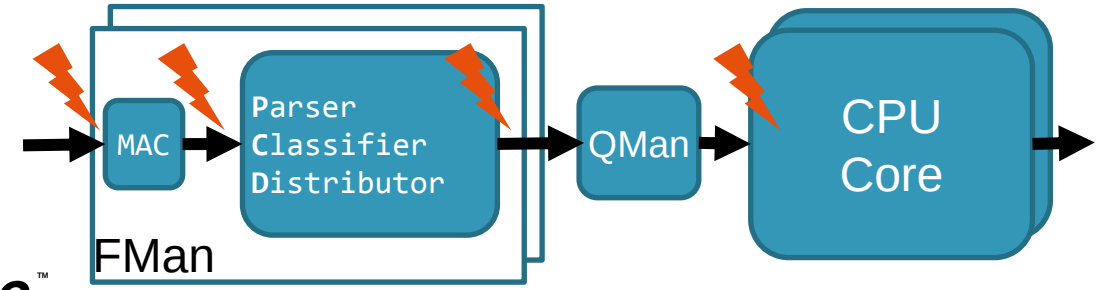
There's also a set of MAC Transmit Counters

MAC Layer Receive Counters and Offsets:

There's also a set of MAC Transmit Counters

- dTSEC:
  - 0x21C RBYT Receive byte
  - 0x220 RPKT Receive packet
  - 0x224 RFCS Receive FCS error
  - 0x228 RMCA Receive multicast packet
  - 0x22C RBCA Receive broadcast packet
  - 0x230 RXCF Receive control frame packet
  - 0x234 RXPF Receive PAUSE frame packet
  - 0x238 RXUO Receive unknown OP code
  - 0x23C RALN Receive alignment error
  - 0x240 RFLR Receive frame length error
  - 0x244 RCDE Receive code error
  - 0x248 RCSE Receive carrier sense error
  - 0x24C RUND Receive undersize packet
  - 0x250 ROVR Receive oversize packet
  - 0x254 RFRG Receive fragments
  - 0x258 RJBR Receive jabber
  - 0x25C RDRP Receive drop

- 10GEC:
  - 0x88 RFRM Receive Frame
  - 0x90 RFCS Receive Frame Check Sequence Error
  - 0x98 RALN Receive Alignment Error
  - 0xA8 RXPF Receive Valid Pause Frame
  - 0xB0 RLONG Receive Frame Too Long Error
  - 0xB8 RFLR Receive Frame Length Error
  - 0xC8 RVLAN Receive VLAN Frame
  - 0xD8 ROCT Receive Octets
  - 0xE0 RUCA Receive Unicast Frame
  - 0xE8 RMCA Receive Multicast Frame
  - 0xF0 RBCA Receive Broadcast Frame
  - 0x120 RDRP Receive Dropped Packets
  - 0x128 REOCT Receive Ethernet Octets
  - 0x130 RPKT Receive Packet Counter



Linux Ethernet Driver for DPAA - Sysfs Support

- DPAA Ethernet Driver exports a series of information in Sysfs such as buffer pool IDs and frame queue IDs used by the interface and MAC registers. Example from SDK 1.5:

root@t4240qds:/sys/devices/fsl,dpaa.18/ethernet.20/net/fm1-mac9# ls

addr_assign_type	dev_id	flags	mac_regs	statistics
addr_len	device	fqids	mtu	subsystem
address	device_addr	ifalias	netdev_group	tx_queue_len
bpids	device_type	ifindex	operstate	type
broadcast	dormant	iflink	queues	uevent
carrier	duplex	link_mode	speed	

root@t4240qds:/sys/devices/fsl,dpaa.18/ethernet.20/net/fm1-mac9/statistics# ls

collisions	rx_errors	rx_packets	tx_errors
multicast	rx_fifo_errors	tx_aborted_errors	tx_fifo_errors
rx_bytes	rx_frame_errors	tx_bytes	tx_heartbeat_errors
rx_compressed	rx_length_errors	tx_carrier_errors	tx_packets
rx_crc_errors	rx_missed_errors	tx_compressed	tx_window_errors
rx_dropped	rx_over_errors	tx_dropped	

Use Sysfs to Check FQIDs and BPIDs

- root@t4240qds:/sys/devices/fsl,dpaa.18/ethernet.20/net/fm1-mac9# cat fqids
  Rx error: 338
  Rx default: 339
  Rx PCD: 15360 - 15487
  Tx confirmation (mq): 340 - 363
  Tx(recycling): 364 - 387
  Tx error: 388
  Tx default confirmation: 389
  Tx: 390 - 413

0x3c00 – 0x3c7f

- root@t4240qds:/sys/devices/fsl,dpaa.18/ethernet.20/net/fm1-mac9# cat bpids
  32

Queue Manager (Qman) Enqueue Rejections

QMan Enqueue Rejections

- Reasons for an Enqueue Rejection
  - Congestion Group tail drop threshold exceeded
  - WRED congestion avoidance
  - FQ tail drop threshold exceeded
  - Error condition, indicated in QMAN_ERR_ISR
  - Order Restoration enabled and frame arrived in late or early rejection window. Plus a couple others related to order restoration

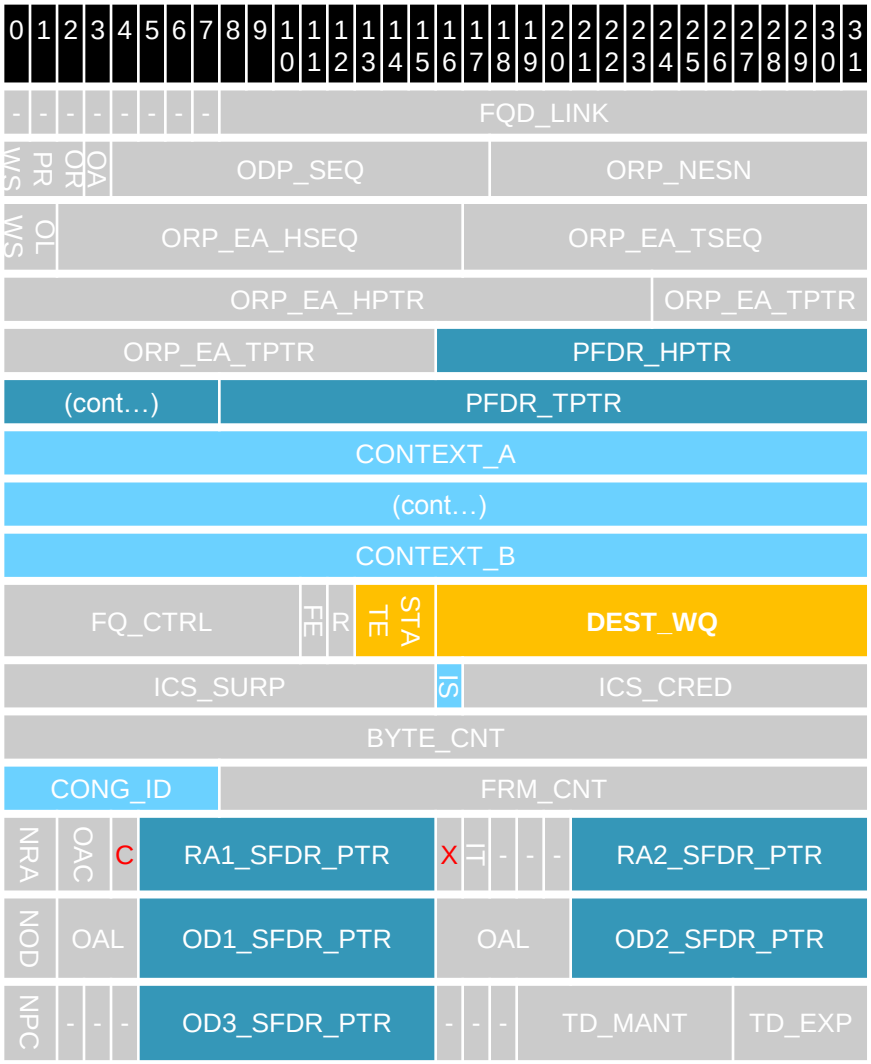
- Software can check Enqueue Rejection Notice (ERN) message
  - ERN[RC] – rejection code

- QMAN_ERR_ISR[IESI]
  - Invalid Enqueue State
  - FQ is retired or out-of-service (perhaps not initialized)
  - The portal in which the error was detected, and some related error information, are captured when the first such error occurs
    - QMAN_ECSR, QMAN_ECIR[FQID], QMAN_ECIR2[T], QMAN_ECIR2[PORTAL]

Frame Queue Descriptor

FQD Selected Field Description:

- FQD_LINK: Link to the next FQD in a queue of FQDs, used for Work Queues
- ORPRWS: ORP Restoration Window Size
- OA: ORP Auto Advance NESN Window Size
- ODP_SEQ: ODP Sequence Number
- ORP_NESN: ORP Next Expected Sequence Number.
- ORP_EA_HPTR, ORP_EA_TPTR: ORP Early Arrival Head and Tail Pointer
- PFDR_HPTR, PFDR_TPTR : PFDR Head and Tail Pointer
- **CONTEXT_A, CONTEXT_B**: Frame Queue Context A and B
- **STATE**: FQ State
- **DEST_WQ**: Destination Work Queue
- ICS_SURP: Intra-Class Scheduling Surplus or Deficit.
- IS: Intra-Class Scheduling Surplus or Deficit identifier
- ICS_CRED: Intra-Class Scheduling Credit
- CONG_ID: Congestion Group ID
- RA[1-2]_SFDR_PTR: SFDR Pointer for Recently Arrived frame # 1 and 2
- TD_MANT, TD_EXP : Tail Drop threshold Exponent and Mantissa
- C: FQD in external memory or in cache (Qman 1.1)
- X: XON or XOFF for flow control command (Qman1.1)



QMan Debugfs – query_fq_fields (Uninitialized FQ)

```
root@p2041rdb:~# cd /sys/kernel/debugfs/qman

root@p2041rdb: /sys/kernel/debugfs/qman/fqd #  echo 1 > query_fq_fields

root@p2041rdb: /sys/kernel/debugfs/qman/fqd #  cat query_fq_fields

Query FQ Programmable Fields Result fqid 0x1
  orprws: 0
  oa: 0
  olws: 0
  cgid: 0
  fq_ctrl: None
  dest_channel: 0
  dest_wq: 0
  ics_cred: 0
  td_mant: 0
  td_exp: 0
  ctx_b: 0
  ctx_a: 0x0
  ctx_a_stash_exclusive: None
  ctx_a_stash_annotation_cl: 0
  ctx_a_stash_data_cl: 0
  ctx_a_stash_context_cl: 0
```

QMan Debugfs – query_fq_fields (Initialized FQ)

FQID

```
root@t4240qds:/sys/kernel/debug/qman# echo 0x3c00 > query_fq_fields
root@t4240qds:/sys/kernel/debug/qman# cat query_fq_fields
Query FQ Programmable Fields Result fqid 0x3c00
   orprws: 0
   oa: 0
   olws: 0
   cgid: 3
   fq_ctrl:
    Prefer in cache
    Avoid Blocking
    Context-A stashing
    Congestion Group Enable
   dest_channel: 49
   dest_wq: 3
   ics_cred: 0
   td_mant: 0
   td_exp: 0
   ctx_b: 0xd5
   ctx_a: 0x175d70018
   ctx_a_stash_exclusive:
    FQ Ctx Stash
    Frame Annotation Stash
   ctx_a_stash_annotation_cl: 1
   ctx_a_stash_data_cl: 2
   ctx_a_stash_context_cl: 2
```

Check DPAARM Work Queue Channel Assignments
-Dedicated Channel serviced by software portal 49
-Work Queue number 3

QMan Debugfs – query_fq_np_fields

```
root@t4240qds:/sys/kernel/debug/qman# echo 0x3c00 > query_fq_np_fields
root@t4240qds:/sys/kernel/debug/qman# cat query_fq_np_fields
  Query FQ Non Programmable Fields Result fqid 0x3c00
   force eligible pending: no
   retirement pending: no
   state: Tentatively Scheduled
   fq_link: 0x0
   odp_seq: 0
   orp_nesn: 0
   orp_ea_hseq: 0
   orp_ea_tseq: 0
   orp_ea_hptr: 0x0
   orp_ea_tptr: 0x0
   pfdr_hptr: 0x0
   pfdr_tptr: 0x0
   is: ics_surp contains a surplus
   ics_surp: 0
   byte_cnt: 0
   frm_cnt: 0
   ra1_sfdr: 0x800
   ra2_sfdr: 0x0
   od1_sfdr: 0x0
   od2_sfdr: 0x0
   od3_sfdr: 0x0
```

FQ in tentatively scheduled state

Snap shot shows that FQ is empty

Buffer Manager (BMan) Debugfs

```
root@t4240qds:/sys/kernel/debug/bman# cat query_bp_state
  bp_id  free_buffers_avail  bp_depleted
    0             no               no
    1             no               no
    2             no               no
    3             no               no
    4             no               no
    5             no               no
    …

    19            no               no
    20            no               no
    21            no               no
    22            no               no
    23            no               no
    24            no               no
    25            no               no
    26            no               no
    27            no               no
    28            no               no
    29            no               no
    30            no               no
    31            no               no
    32            yes              no
    33            no               no
    34            no               no
    …

    61            no               no
    62            no               no
    63            no               no
```

Note that on earlier slide, sysfs showed that BPID 32 was being used

SDK Version Compatibility

Unexplained Errors May Be Due to Using Images from Different SDK Version

- Use images from the same SDK version

- Example: images from SDK 1.5 for T4240QDS
  - fsl_fman_ucode_t4240_r2.0_106_4_10.bin
  - rcw_1_1_6_6_1666MHz_rev2.bin
  - uImage-t4240qds.bin
  - fsl-image-core-t4240qds.ext2.gz.u-boot
  - u-boot-T4240QDS.bin
  - uImage-t4240ds.dtb or uImage-t4240qds-usdpaa.dtb

- Feature supported in most recent SDK release may not be supported in previous version

```
U-Boot 2013.01QorIQ-SDK-V1.5 (Dec 16 2013 - 11:20:16)

CPU0:  T4240E, Version: 2.0, (0x82480020)
Core:  E6500, Version: 2.0, (0x80400120)
Clock Configuration:
       CPU0:1666.667 MHz, CPU1:1666.667 MHz, CPU2:1666.667 MHz, CPU3:1666.667 MHz,
       CPU4:1666.667 MHz, CPU5:1666.667 MHz, CPU6:1666.667 MHz, CPU7:1666.667 MHz,
       CPU8:1666.667 MHz, CPU9:1666.667 MHz, CPU10:1666.667 MHz, CPU11:1666.667 MHz,
       CCB:733.333 MHz,
       DDR:933.333 MHz (1866.667 MT/s data rate) (Asynchronous), IFC:183.333 MHz
       FMAN1: 733.333 MHz
       FMAN2: 733.333 MHz
       QMAN:  366.667 MHz
       PME:   533.333 MHz
L1:    D-cache 32 kB enabled
       I-cache 32 kB enabled
Reset Configuration Word (RCW):
       00000000: 16070019 18101916 00000000 00000000
       00000010: 04023030 00558c00 ec020000 f5000000
       00000020: 00000000 ee0000ee 00000000 000307fc
       00000030: 00000000 00000000 00000000 00000038
Board: T4240QDS, Sys ID: 0x1e, Sys Ver: 0x22, vBank: 4
FPGA: v3 (T4240QDS_2012_1113_1114), build 438 on Tue Nov 13 17:14:23 2012
SERDES Reference Clocks: SERDES1=125MHz SERDES2=125MHz SERDES3=100MHz SERDES4=100MHz
I2C:   ready
SPI:   ready
DRAM:  Initializing....using SPD
Detected UDIMM 9JSF25672AZ-2G1K1
Detected UDIMM 9JSF25672AZ-2G1K1
There is no rank on CS0 for controller 1.
Not all controllers have compatible interleaving mode. All disabled.
2 GiB left unmapped
    DDR: 4 GiB (DDR3, 64-bit, CL=13, ECC on)
Flash: 128 MiB
L2:    2048 KB enabled
enable l2 for cluster 1 fec60000
enable l2 for cluster 2 feca0000
Corenet Platform Cache: 1536 KB enabled
Using SERDES1 Protocol: 1 (0x1)
Using SERDES2 Protocol: 1 (0x1)
Using SERDES3 Protocol: 6 (0x6)
Using SERDES4 Protocol: 6 (0x6)
SRIO1: enabled
SRIO2: enabled
NAND:  512 MiB
MMC:  FSL_SDHC: 0
EEPROM: NXID v1
PCIe1: Root Complex, no link, regs @ 0xfe240000
PCIe1: Bus 00 - 00
PCIe2: disabled
PCIe3: Root Complex, x1, regs @ 0xfe260000
  02:00.0     - 8086:107d - Network controller
PCIe3: Bus 01 - 02
PCIe4: disabled
In:    serial
Out:   serial
Err:   serial
Net:   Fman1: Uploading microcode version 106.4.10
Failed to connect
Failed to connect
Fman2: Uploading microcode version 106.4.10
PHY reset timed out
e1000: 00:15:17:80:ad:8c
       FM1@DTSEC5, FM1@TGEC1, FM2@DTSEC5, FM2@TGEC1, FM2@TGEC2, e1000#0 [PRIME]
```

From U-Boot log, you can check:

- U-Boot version
- RCW (copy and paste into QCS PBL Tool)
- FMan ucode version

When you move to a more recent kernel version, be sure to update u-boot, rcw, ucode.

Check SDK Version from Linux

- Starting in SDK v1.4 use the following command:

```
root@p4080ds:~# cat /etc/sdk-version
QorIQ-SDK-V1.4
root@t4240qds:~# cat /etc/sdk-version
QorIQ-SDK-V1.5
```

- Also can check kernel version:

```
root@p4080ds:~# cat /proc/version
Linux version 3.8.10-rt6-QorIQ-SDK-V1.4 (gcc
version 4.7.2 (GCC) ) #1 SMP Tue May 21 06:38:27
CDT 2013
root@t4240qds:~# cat /proc/version
Linux version 3.8.13-rt9-QorIQ-SDK-V1.5
(jenkins@sbuilder137.localdomain) (gcc version
4.7.3 (GCC) ) #1 SMP Mon Dec 16 11:25:06 CST 2013
```

Session Summary

- This session focused on Data Path Acceleration Architecture and debug methods for issues such as packet loss and queue manager enqueue rejections.

- QorIQ SDK documentation:
  www.freescale.com/infocenter

- QorIQ Packet Analysis Tool:
  FTF-SDS-F0004 QorIQ Optimization Suite (QOS) Packet Analysis Tool

Introducing The QorIQ LS2 Family

Breakthrough, software-defined approach to advance the world's new virtualized networks

New, high-performance architecture built with ease-of-use in mind

Groundbreaking, flexible architecture that abstracts hardware complexity and enables customers to focus their resources on innovation at the application level

Optimized for software-defined networking applications

Balanced integration of CPU performance with network I/O and C-programmable datapath acceleration that is right-sized (power/performance/cost) to deliver advanced SoC technology for the SDN era

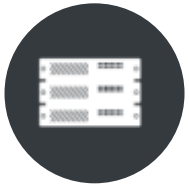
Extending the industry's broadest portfolio of 64-bit multicore SoCs

Built on the ARM® Cortex®-A57 architecture with integrated L2 switch enabling interconnect and peripherals to provide a complete system-on-chip solution

QorIQ LS2 Family

Key Features

SDN/NFV Switching



Data Center

Wireless Access

Unprecedented performance and ease of use for smarter, more capable networks

High performance cores with leading interconnect and memory bandwidth

- 8x ARM Cortex-A57 cores, 2.0GHz, 4MB L2 cache, w Neon SIMD
- 1MB L3 platform cache w/ECC
- 2x 64b DDR4 up to 2.4GT/s

A high performance datapath designed with software developers in mind

- New datapath hardware and abstracted acceleration that is called via standard Linux objects
- 40 Gbps Packet processing performance with 20Gbps acceleration (crypto, Pattern Match/RegEx, Data Compression)
- Management complex provides all init/setup/teardown tasks

Leading network I/O integration

- 8x1/10GbE + 8x1G, MACSec on up to 4x 1/10GbE
- Integrated L2 switching capability for cost savings
- 4 PCIe Gen3 controllers, 1 with SR-IOV support
- 2 x SATA 3.0, 2 x USB 3.0 with PHY

See the LS2 Family First in the Tech Lab!

4 new demos built on QorIQ LS2 processors:

- Performance Analysis Made Easy
- Leave the Packet Processing To Us
- Combining Ease of Use with Performance
- Tools for Every Step of Your Design

www.Freescale.com

© 2014 Freescale Semiconductor, Inc. | ***External Use***