

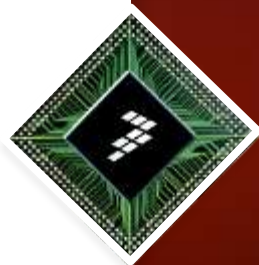


**FTF** | FREESCALE TECHNOLOGY FORUM  
POWERING INNOVATION

# QorIQ Platform DPAA Deep Dive

## FTF-NET-F0155

Sam Siu  
Systems and Applications Engineer



June 2012

Freescale, the Freescale logo, AlliVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, the SafeAssure logo, SMARTMOS, TurboLink, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2012 Freescale Semiconductor, Inc.































# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion



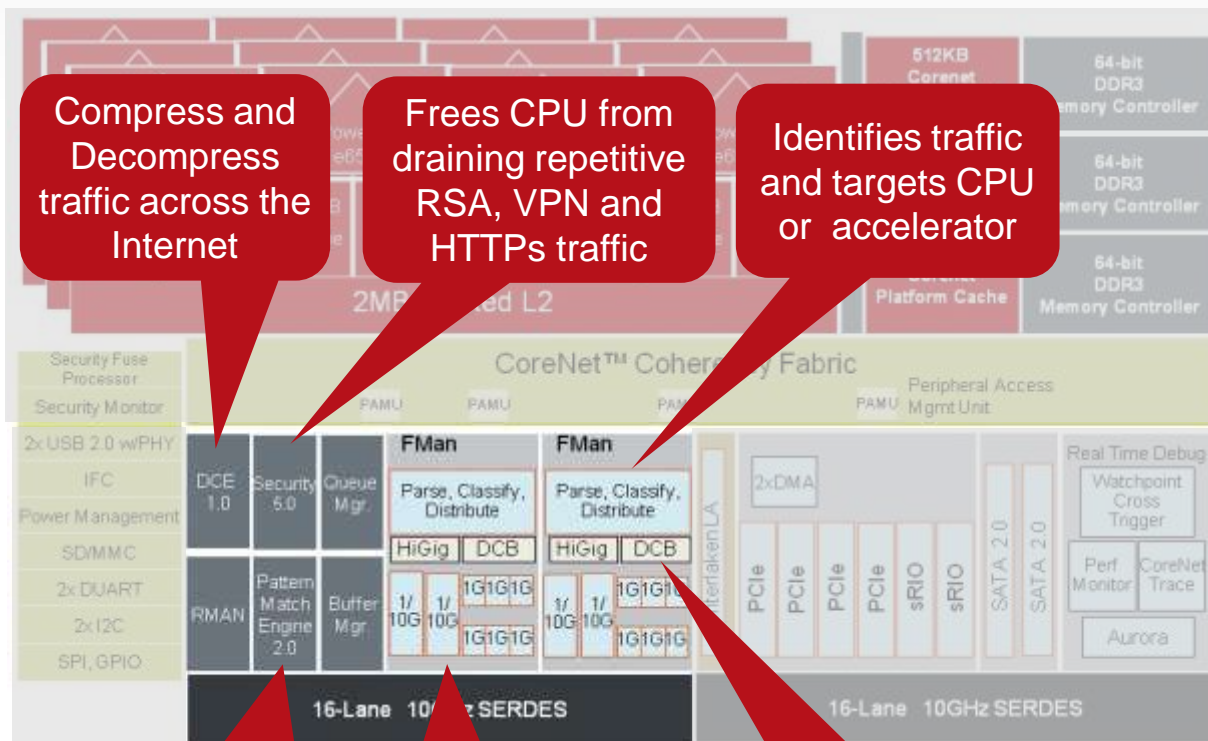
# QorIQ Processing Platforms

<p><b>QorIQ P5</b> P5020, P5010</p>	<p>64-bit High End Up to 2.2 GHz</p>	 Service Provider Routers	 Network Admission Controls	 Storage Networks	 Switching	<p><b>QorIQ T5</b></p>	 Service Provider Routers	 Storage Networks
<p><b>QorIQ P4</b> P4080, P4040</p>	<p>4 – 8 Cores Up to 1.5 GHz</p>	 Metro Carrier Edge Router	 IMS Controller	 Radio Network Control	 Serving Node Router	<p><b>QorIQ T4</b> T4240, T4160 16 – 24 cores Up to 2 GHz</p>	 Metro Carrier Edge Router	 Aerospace & Defense
<p><b>QorIQ P3</b> P3041</p>	<p>2 – 4 Cores Up to 1.5 GHz</p>	 Converged Media Gateway	 SSL, IPSec, Firewall	 Access Gateway		<p><b>QorIQ T3</b></p>	 Media Gateway	
<p><b>QorIQ P2</b> P2040, P2020, P2010</p>	<p>1 – 2 Cores Up to 1.2 GHz</p>	 Unified Threat Mgmt	 VoIP Carrier-Class Media Gateway	 Wireless Media Gateway	 Base Station	<p><b>QorIQ T2</b></p>	 Integrated Services Router	 Network Attached Storage
<p><b>QorIQ P1</b> P1010, P1011, P1012, P1013, P1014, P1015, P1016, P1017, P1020, P1021, P1022, P1023, P1024, P1025</p>	<p>1 – 2 Cores Up to 1 GHz</p>	 Integrated Services Router	 Network Attached Storage	 Home Media Hub	 Enterprise WAP			





# Enhancing Core Performance with Data Path Acceleration Architecture



Compress and Decompress traffic across the Internet

Frees CPU from draining repetitive RSA, VPN and HTTPs traffic

Identifies traffic and targets CPU or accelerator

Protects against internal and external Internet attacks

Line rate 50Gbps Networking

Quality of Service for FCoE in converged data center networking

Hardware Accelerators	
FMAN Frame Manager	50 Gbps aggregate Parse, Classify, Distribute
BMAN Buffer Manager	64 buffer pools
QMAN Queue Manager	Up to 2 <sup>24</sup> queues
RMAN Rapid IO Manager	Seamless mapping sRIO to DPAA
SEC Security	40Gbps: IPsec, SSL Public Key 25K/s 1024b RSA
PME Pattern Matching	10Gbps aggregate
DCE Data Compression	20Gbps aggregate
Saving CPU Cycles for higher value work	

**New** | **Enhanced**



Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, the SafeAssure logo, SMARTMOS, TurboLink, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2012 Freescale Semiconductor, Inc.

# DPAA Components Check List

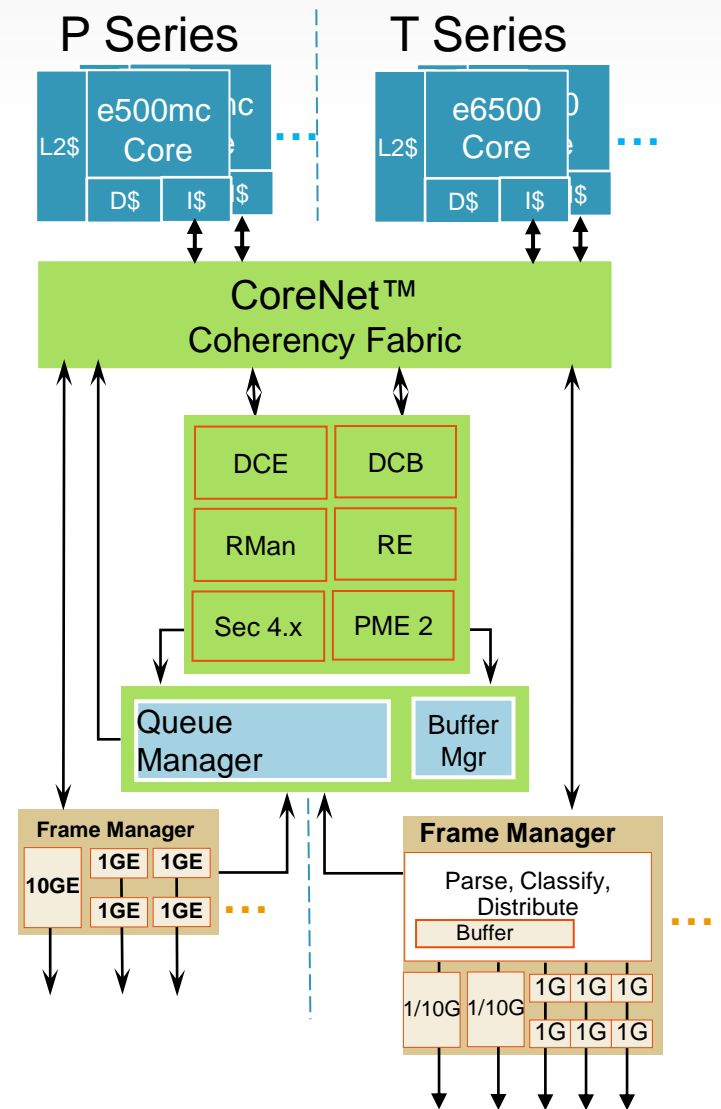
QorIQ Devices DPAA Feature List	Revision Number							
	FMan	QMan	BMan	SEC	PME	RMan	DCE	RE
<b>P1023</b>	4.0	2.0	2.0	4.2	n/a	n/a	n/a	n/a
<b>P4080 rev2</b>	2.0	1.1	1.0	4.0	2.1	n/a	n/a	n/a
<b>P2040, P2041 P3041, P4040,</b>	3.0	1.2	1.0	4.2	2.1	1.0	n/a	n/a
<b>P5020, P5010</b>	3.0	1.2	1.0	4.2	2.1	1.0	n/a	1.0
<b>T4240, T4216</b>	6.1	3.1	2.1	5.0	2.1	1.0	1.0	n/a

- QorIQ P class devices has “Datapath Three Speed Ethernet Controller (dTSEC) and 10-Gigabit Ethernet Media Access Controller (10GEC)
- QorIQ T class devices has Ethernet Media Access Controller (EMAC)



# Data Path Acceleration Architecture Philosophy

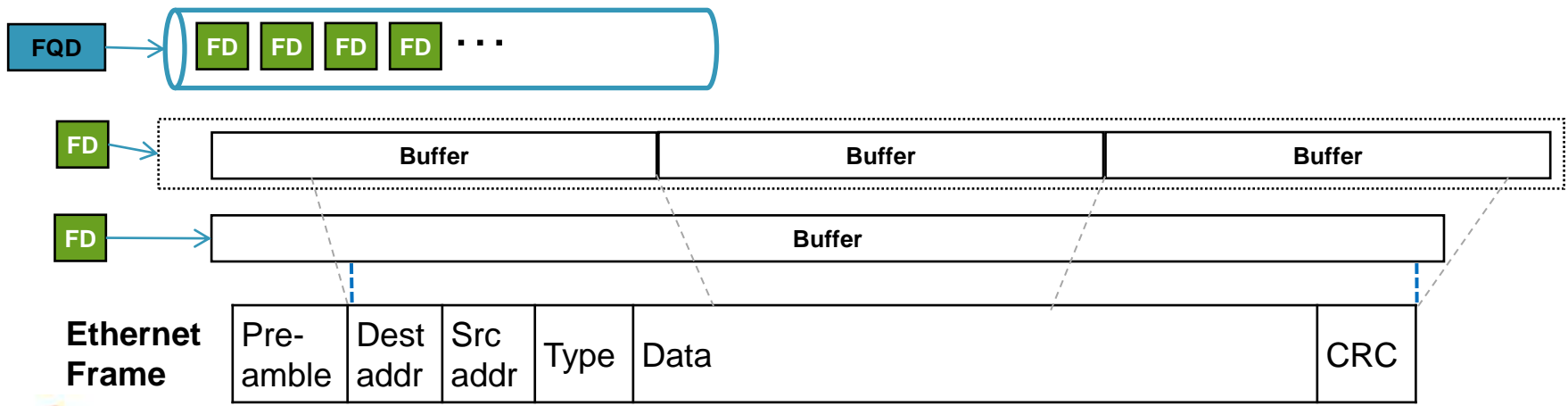
- DPAA is design to balance the performance of **Accelerators with seamless Integrations**
  - ANY packet to ANY core to ANY accelerator or network interface efficiently **WITHOUT** locks or semaphores.
- “Infrastructure” components
  - Queue Manager (QMan)
  - Buffer Manager (BMan)
- “Accelerator” Components
  - Cores
  - Frame Manager (FMan)
  - RapidIO Message Manager (RMan)
  - Cryptographic accelerator (SEC)
  - Pattern matching engine (PME)
  - Decompression/Compression Engine (DCE)
  - DCB (Data Center Bridging)
  - RAID Engine (RE)
- CoreNet
  - Provides the interconnect between the cores and the DPAA infrastructure as well as access to memory.



# DPAA Terminology

- Buffer: Unit of contiguous memory, allocated by software
- **Frame**: Buffer(s) that hold a data element (generally a packet)
  - Frames can be single buffers or multiple buffers (scatter/gather lists)
    - A “simple frame” has one delimited data element
    - A “multi buffer frame” has two or more data elements
- **Frame Descriptor (FD)**: Proxy structure used to represent frames
- **Frame Queue**:
  - FIFO of related Frames Descriptor.(e.g. TCP session)
  - The basic queuing structure supported by QMan

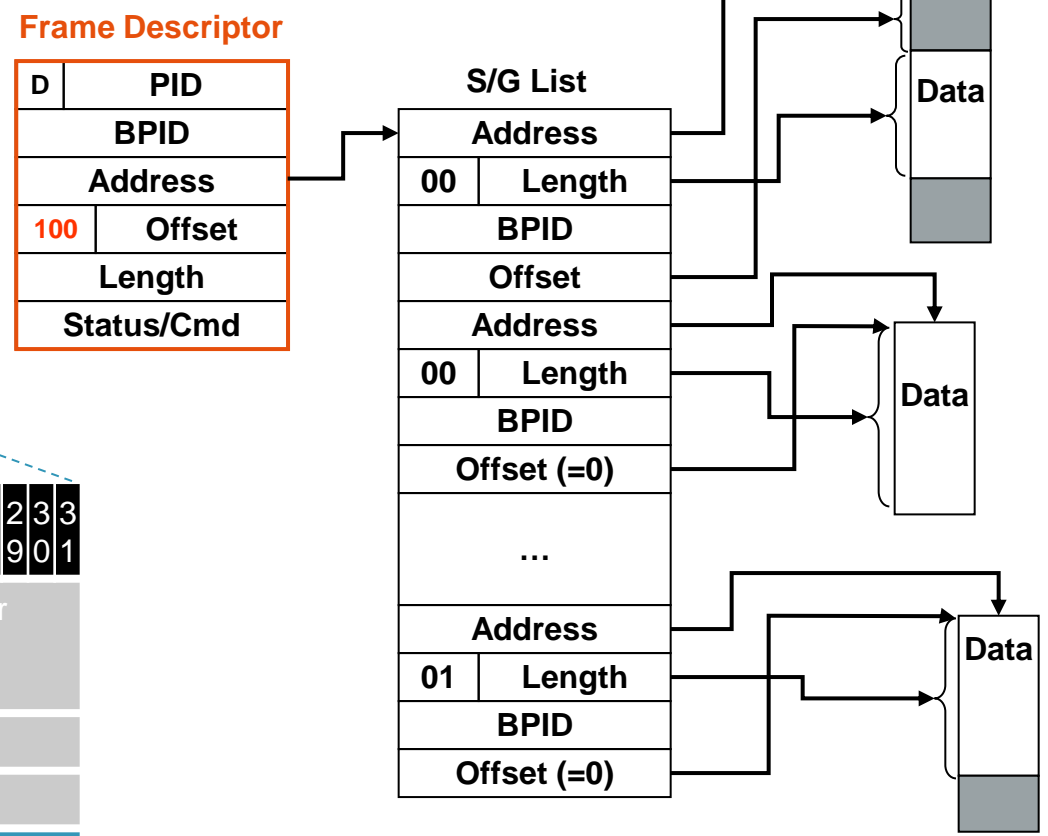
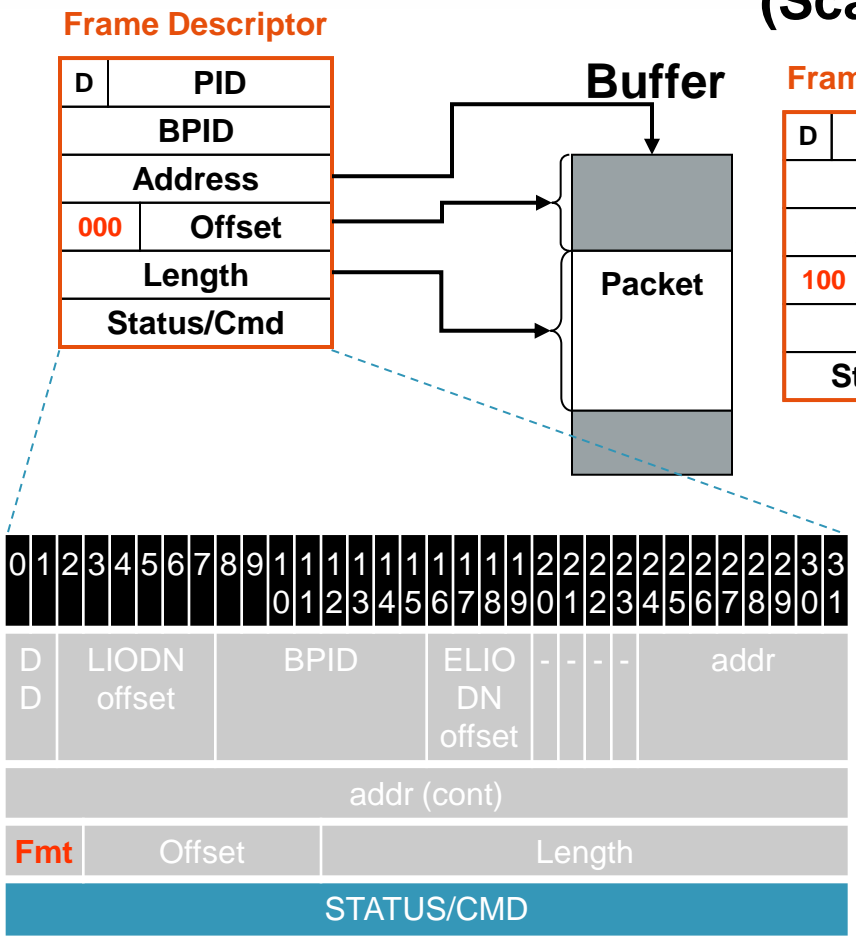
**Frame Queue Descriptor (FQD)**: Structure used to manage Frame Queues



# DPAA Building Block: Frame Descriptor (FD)

## Simple Frame

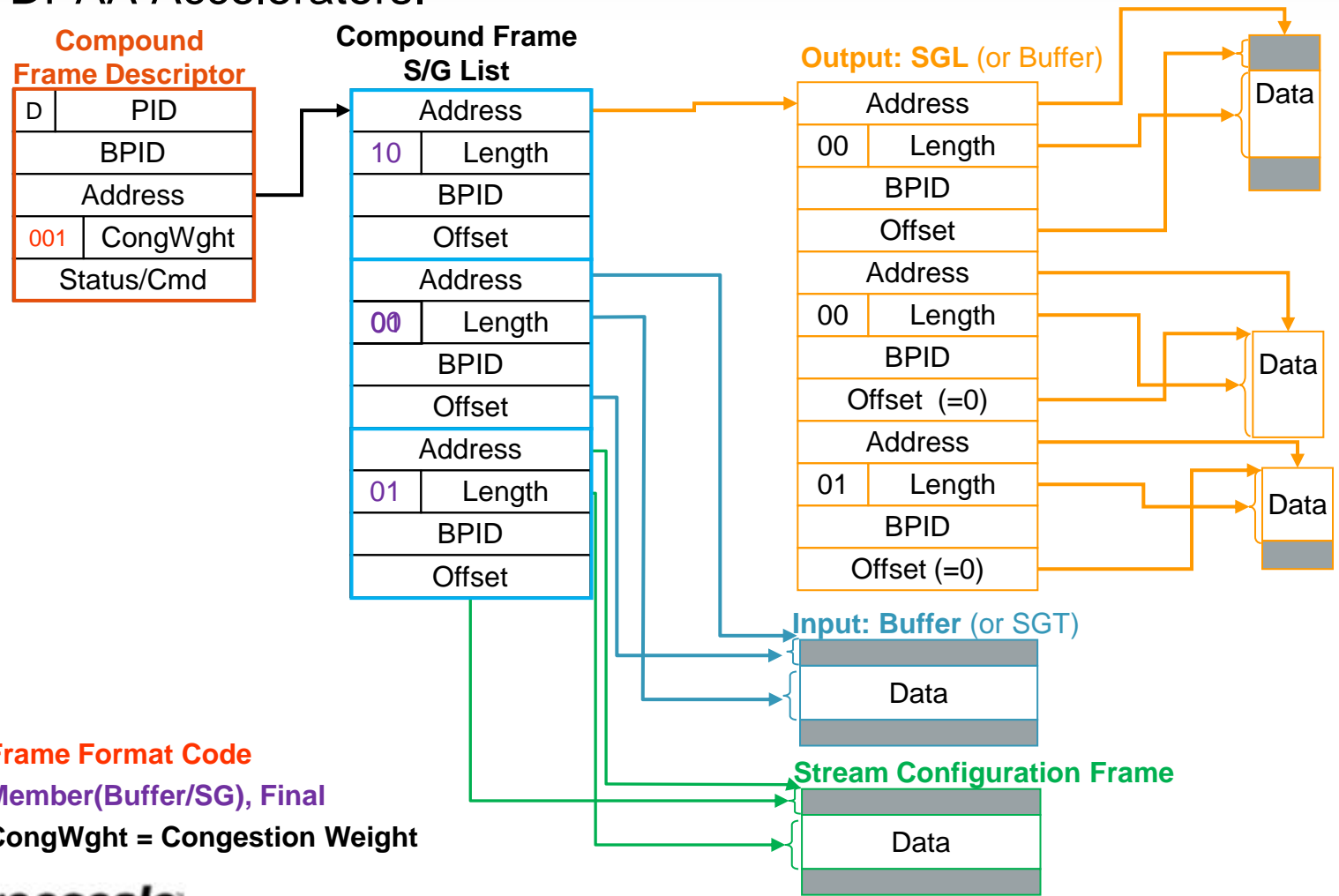
## Multi-buffer Frame (Scatter/gather)





# DPAA Interaction: Compound Frame

- Compound frames allows related data to be passed in a single unit to the DPAA Accelerators.

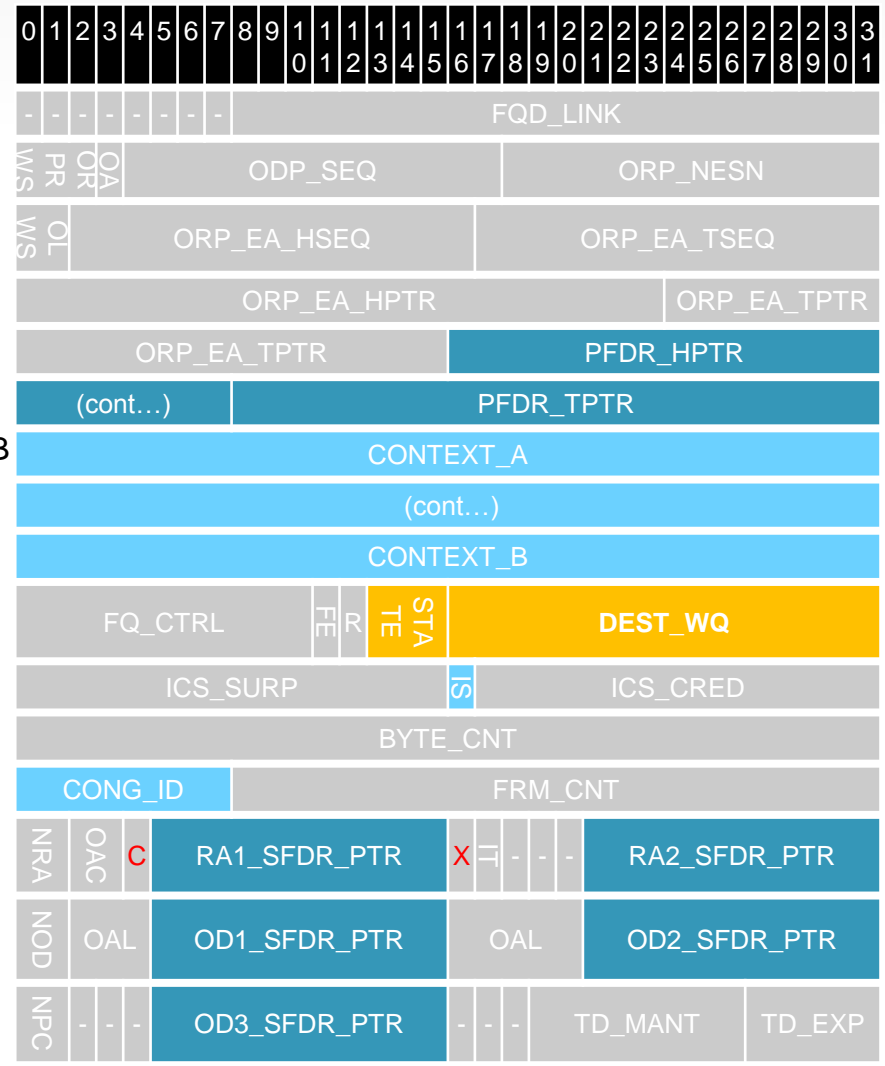


**Frame Format Code**  
**Member(Buffer/SG), Final**  
**CongWght = Congestion Weight**

# DPAA Building Block: Frame Queue Descriptor

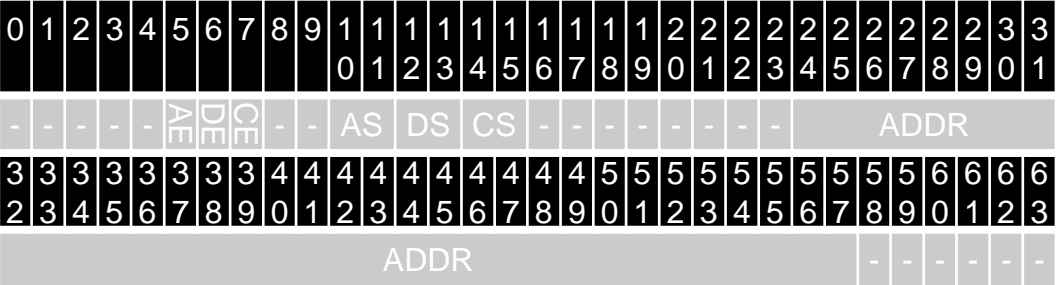
## FQD Selected Field Description:

- 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)



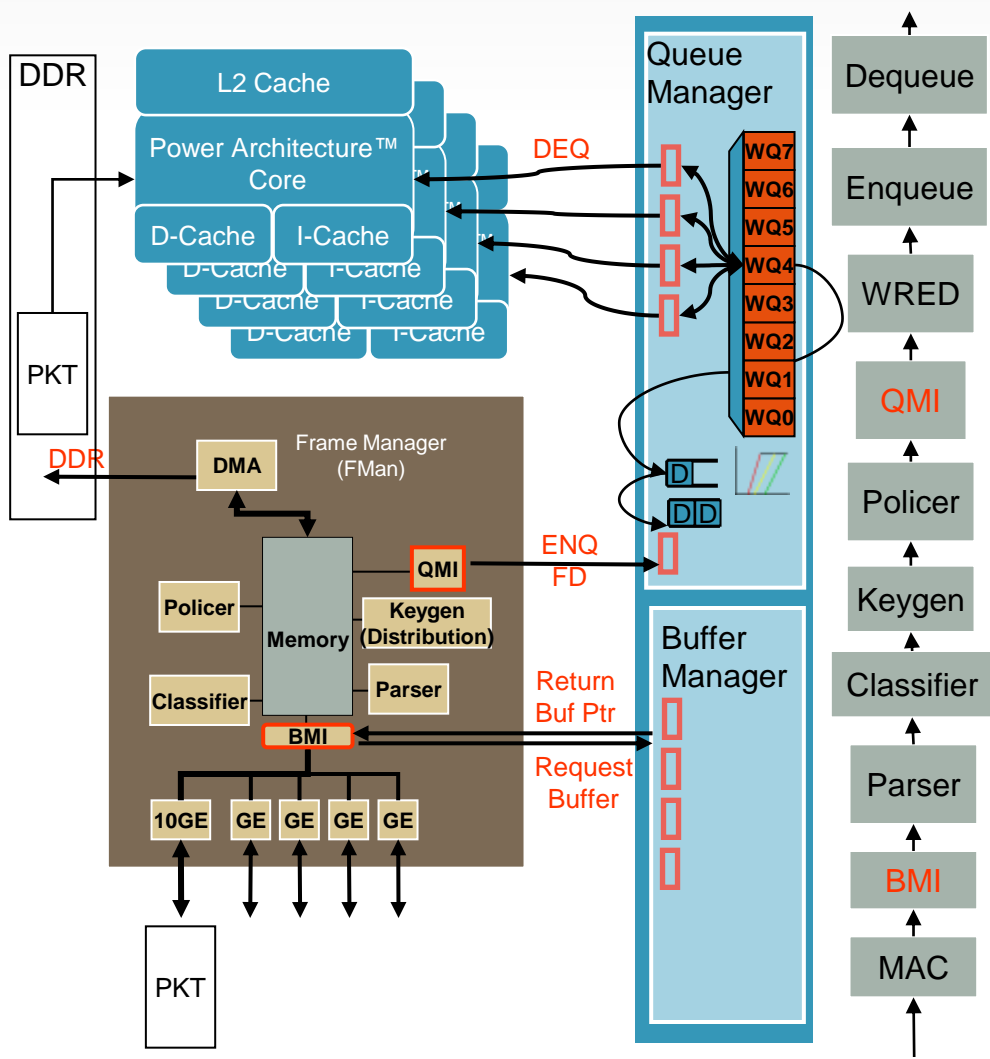
# Software Portal FQD Context\_A Usage

- AE: Frame Annotation Stash Exclusive.
- DE: Frame Data Stash Exclusive
- CE: FQ Context Stash Exclusive
  - 0: Stash transaction issued as DIRECT0. PAMU translate this to LDEC
  - 1: Stash transaction issued as DIRECT1. PAMU translate this to LDECPE/LDECPE.
- AS: Frame Annotation Stashing Size
- DS: Frame Data Stashing Size
- CS: FQ Context Stashing Size
  - Number of 64 byte coherency granules (0, 1, 2, or 3) of Frame Annotation to be stashed.
- ADDR: FQ Context Address
  - the first 64 byte coherency granule containing the FQ context information to be stashed.



# Life of an Ingress Packet

- FMan receives packets
  - allocates internal buffers
  - retrieves data from MAC
- BMI
  - acquires a buffer from BMan
  - uses DMA to store data in it
- Parse+classify+keygen select a queue and policer profile
- Policer “colors” and optionally discards frame
- QMan applies active queue management and enqueues frame
- Frame is enqueued to one of a pool of cores
- Available core dequeue FD for processing





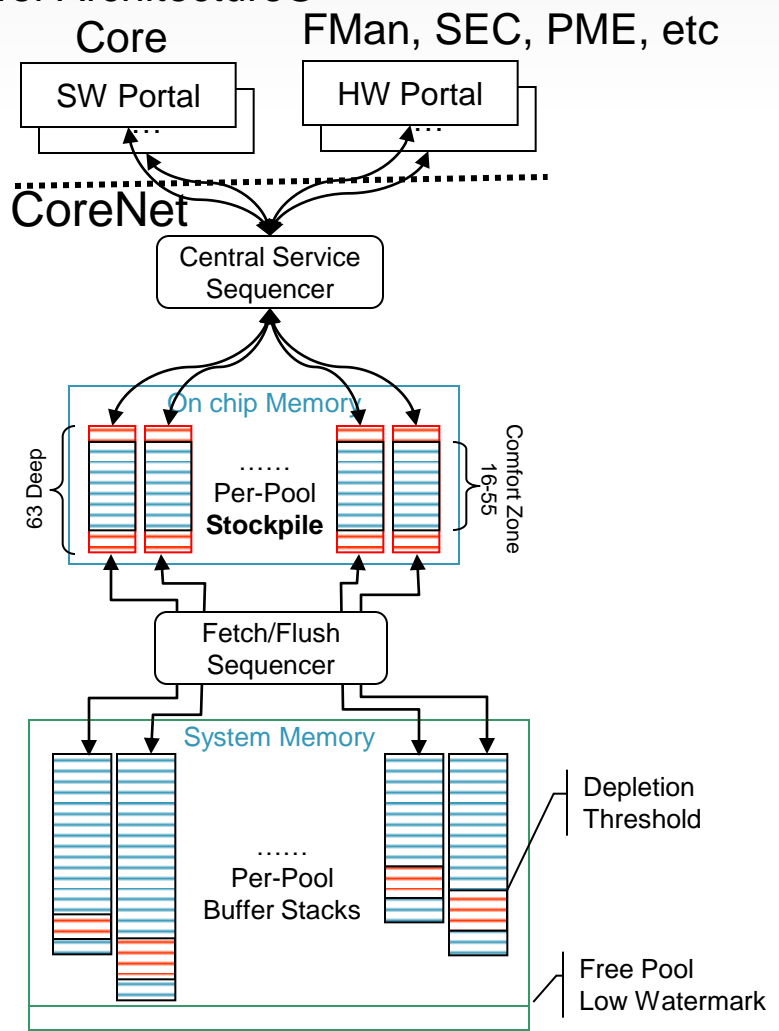
# Agenda

- DPAA
- **BMan Enablement**
- FMan Enablement
- QMan Enablement
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

# Buffer Manager Functional Blocks

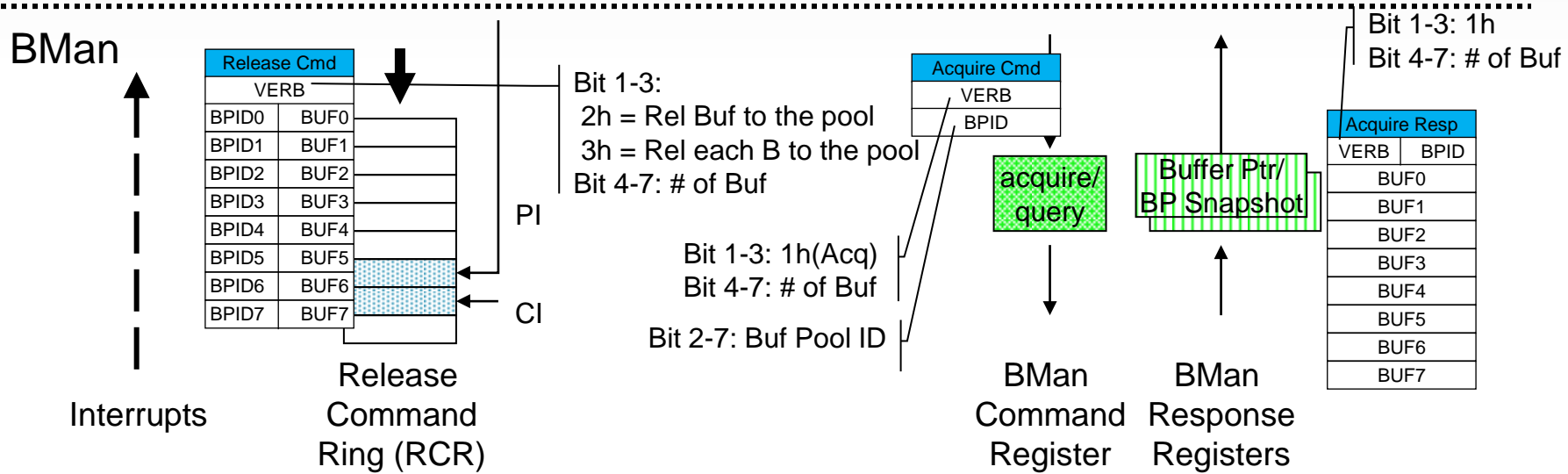
- Standardized command interface to SW and HW
  - Up to 66 Software portals for software: resolves any Multi Core race scenario
  - Up to 6 HW portal per HW block: simplified command for HW Accelerators
  - Up to 64 separate pools of free buffers
- BMan keeps a small per-pool stockpile of buffer pointers in internal memory
  - stockpile of 64 buffer pointers per pool, Maximum 2G buffer pointers
  - Absorbs bursts of acquire/release commands without external memory access
  - minimized access to memory for buffer pool management.
- Pools (buffer pointers) overflow into DRAM
- LIFO buffer allocation policy
  - A released buffer is immediately used for receiving new data, using cache lines previously allocated

## Power Architecture®



# BMan SW Portal Components

Core



- Software portals have 2 components
  - Management commands:
    - Command Registers (BCSPi\_CR) : acquire 1-8 buffers OR query availability
    - Response Registers (BCSPi\_RR0 / RR1): buffer address OR Buffer Pool Availability and Depletion state
  - Buffer Release:
    - Release Command Ring (RCR) Entry (BCSPi\_RCRj): Circular FIFO
- Interrupts can be used to signal availability of space (in RCR) and that pools are depleted and require replenishment (RCR Interrupt Threshold Register)

# BMan Command and Response

- BMan Command Type
  - BMan command registers (BCSPi\_CR) are 64B long.
  - Command Verb (1B) + Buffer Pool ID (1B)
    - Bit 1-3: Response Type. Valid encodings are:
      - 001 = Acquire buffers (Acquire)
      - 010= Release buffers to the pool identified in byte field 1 (Release)
      - 011= Release each buffer to the pool identified in byte field immediately preceding its buffer field (Release)
      - 100=Query buffer pool state, depletion and availability.
      - 110= Invalid command (Response)
      - 111= Stockpile ECC Error (Response)
    - Bit 4-7: Number of buffers associated with command type, maximum 8
      - 0h = Zero buffers. 1h = One buffer .... 8h = Eight buffers
    - Returns up to eight 48bit buffer addresses

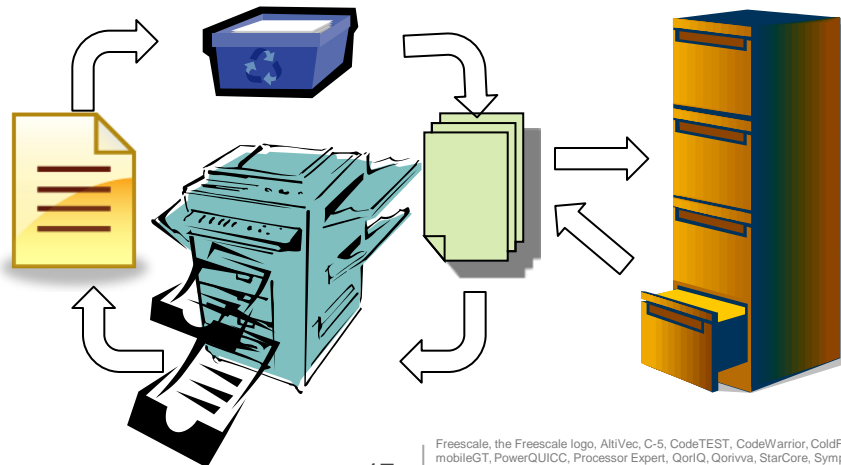
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Verb x18	BPID bp1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



# BMan Analogy

BMan functions like paper tray in a photocopier.

Photo Copier	BMan
Each tray holds same paper size	Buffer Pool holds fixed size buffer pointers
Different trays for different paper type	Different BPs for different sizes Buffers
Stock up papers that you need	Configure a contiguous memory region
Initially fill up the paper tray	Fill the stockpile with buffer pointers
Add paper when the tray runs low	Refill/remove BPs when it runs low/high
Does not detect what is in the tray 😊	BMan does not check the size of buffer





# Agenda

- DPAA
- BMan Enablement
- **FMan Enablement**
- QMan Enablement
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

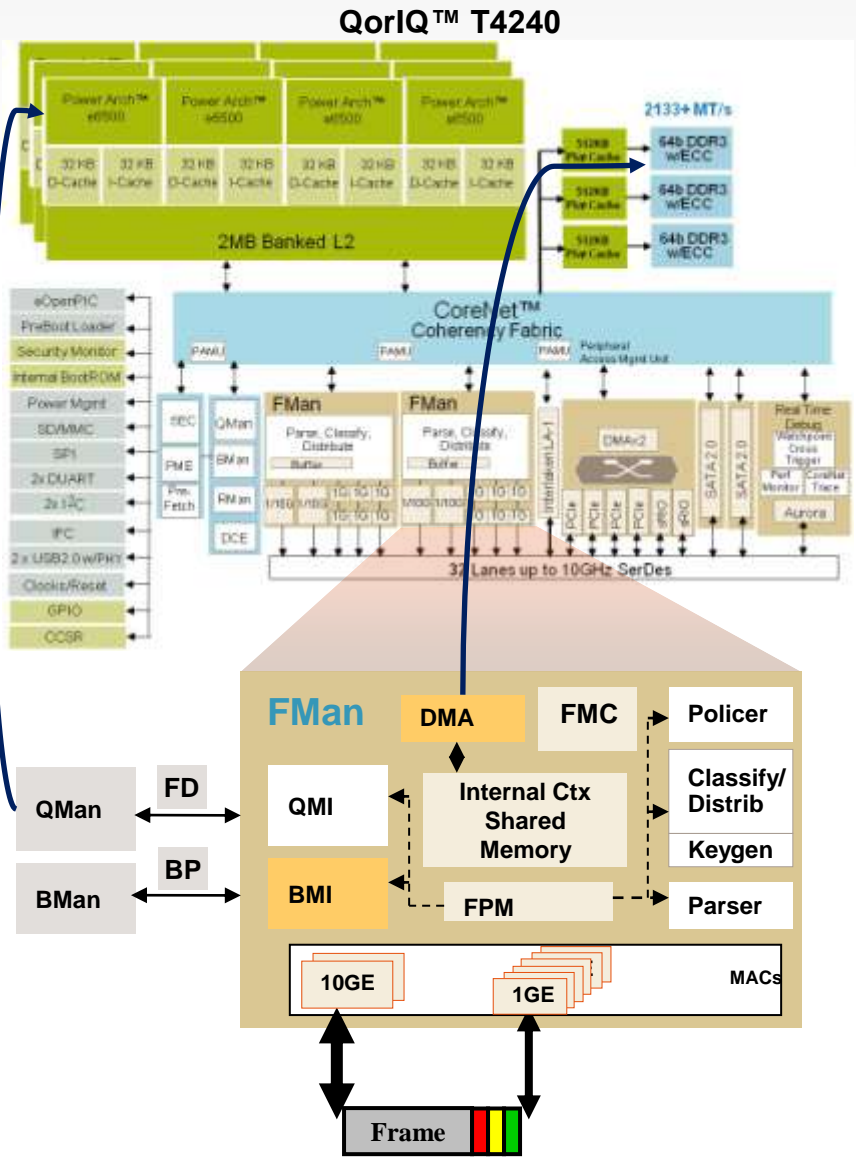
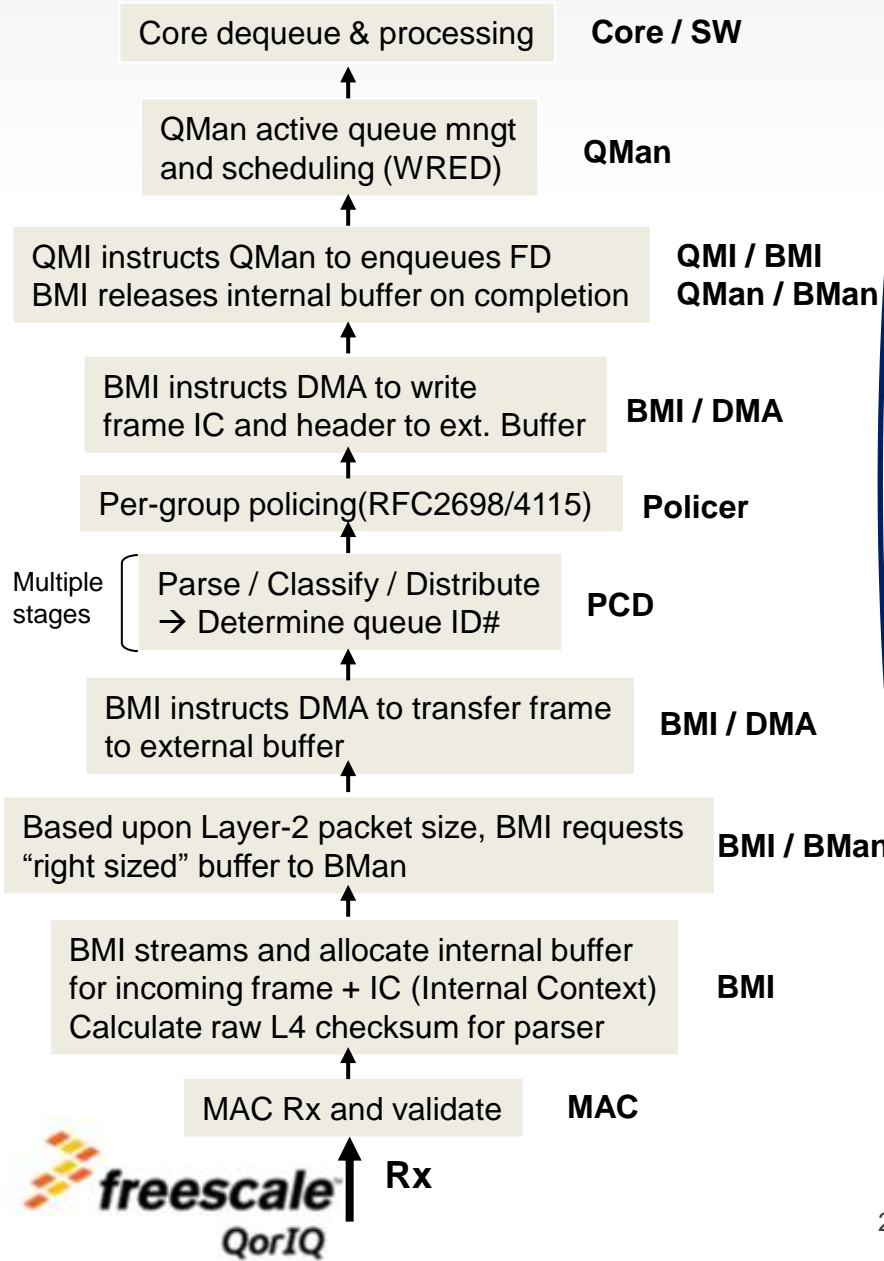
# New Frame Manager (FMan) Features

- FMan combines the Ethernet network interfaces with packet distribution logic to provide intelligent distribution and queuing decisions for incoming traffic at line rate.
- FMan key new features for QorIQ T4 processors.
  - Six 1G/2.5G multirate Ethernet MACs (mEMACs) per Frame Manager
  - Two 10G multirate Ethernet MACs (mEMACs) per Frame Manager
  - QMan interface: Supports priority based flow control message pass from Ethernet MAC to Qman
  - Comply with IEEE 803.3az (Energy efficient ethernet) and IEEE 802.1QBbb, in addition of IEEE Std 802.3®, IEEE 802.3u, IEEE 802.3x, IEEE 802.3z, IEEE 802.3ac, IEEE 802.3ab, and IEEE-1588 v2 (clock synchronization over Ethernet)
  - Port Virtualization: Virtual Storage profile (SPID) selection after classification or distribution function evaluation.
  - Rx port multicast support.
  - Egress Shaping.
  - Offline port: able to copy the frame into new buffers and enqueue back to the QMan.



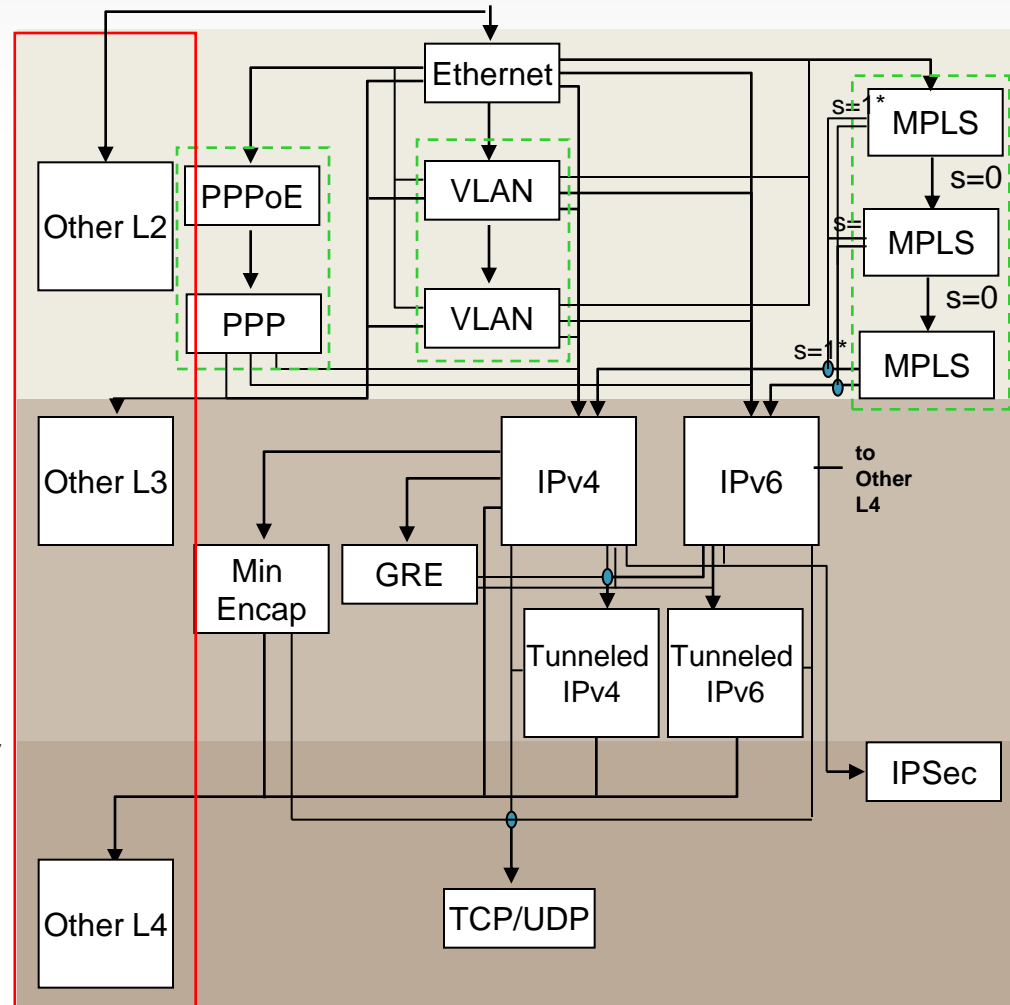


# Human Modular Architecture Processing Pipeline



# 1) Parser

- Performs parsing of common L2/L3/L4 headers, including tunneled protocols
- Can be augmented by the user to parse other standard protocols
- Can also parse proprietary, user-defined headers at any layer:
  - Self-describing, using standard fields such as proprietary Ethertype, Protocol ID, Next Header, etc.
  - Non-Self-Describing through configuration.
- Parse results, including proprietary fields, can be used by the classifier, and/or software.
- Soft parse can modify any field in parse results





# Flexible Parsing of User Defined Fields (UDF)

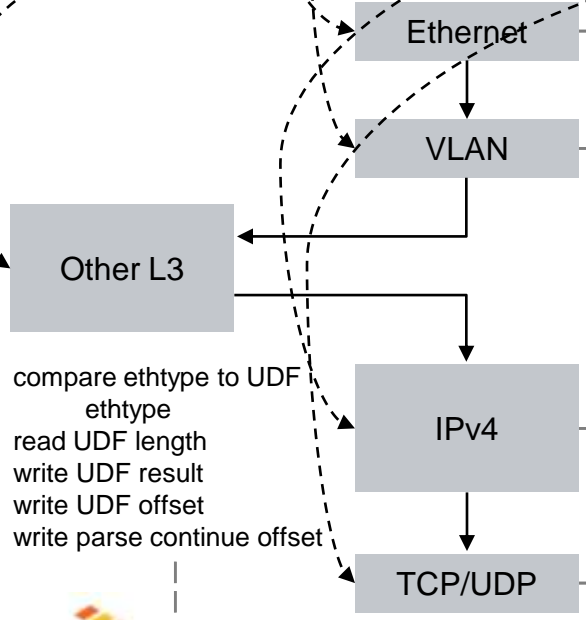
The parser stores all parsing results in the Parse Array located inside the parser.

## Incoming frame



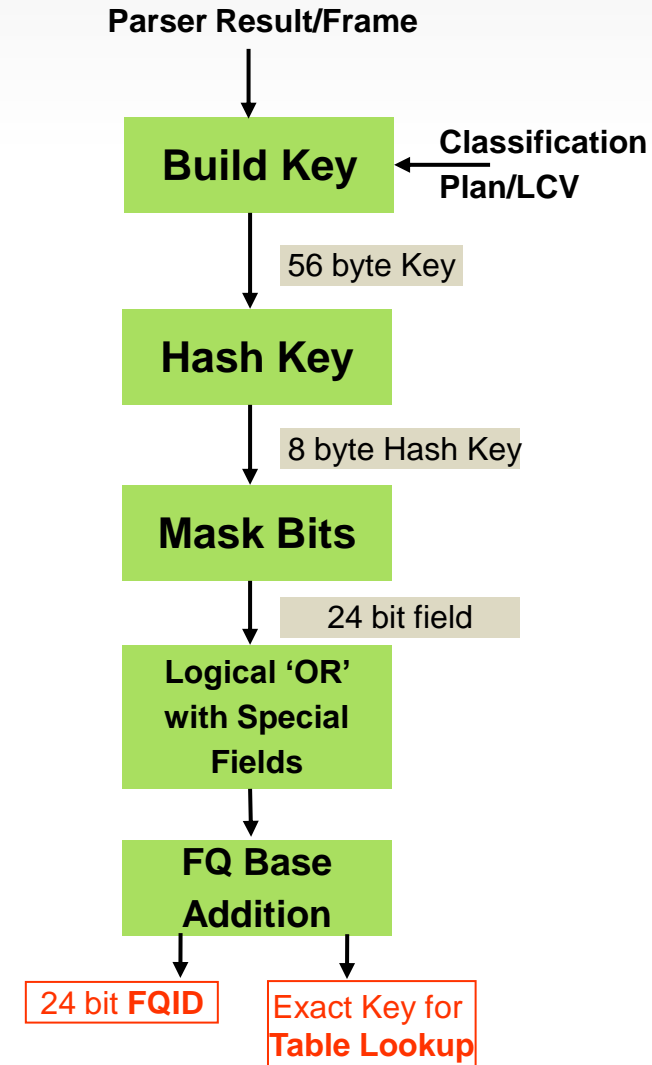
## Parse Result Array (simplified)

B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1x	Logical Port ID	ShimR	L2R	L3R	L4	Classification Plan ID	Nxt Hdr	Running Sum	Flags, Flag offset	Rout Type	RHP					
2x	Offset_1	Shim Offset_2	IP Prd offset	Eth Offset	LLC+Snap Offset	VLAN TC1 offset_1	VLAN TC1 offset_n	LastE TypeOffset	PPPoE Offset	MPLS Offset_1	MPLS Offset_n	IPOffset_1	IPOffset_n	GRE Offset	L4 Offset	NxtHdr offset



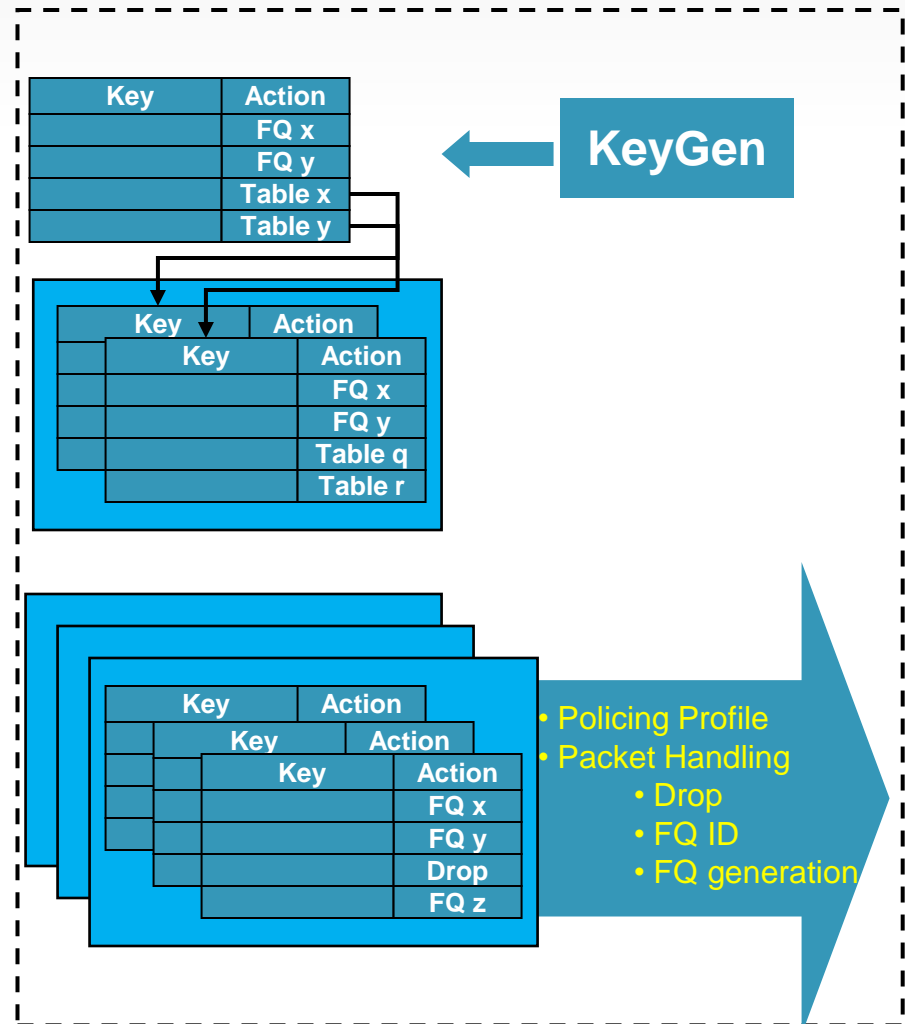
## 2) KeyGen – Key Generator

- 256 Classification Plans
  - Indicates which fields of a parsed packet are of interest for key generation
  
- 32 Key Generation schemes
  - Direct Method
    - Used for port based or post coarse classification
  - Indirect Method
    - Based on the parsed protocol stack of the frame and the source port
    - The presence (or absence) of valid headers can direct the scheme used



# 3) Classifier

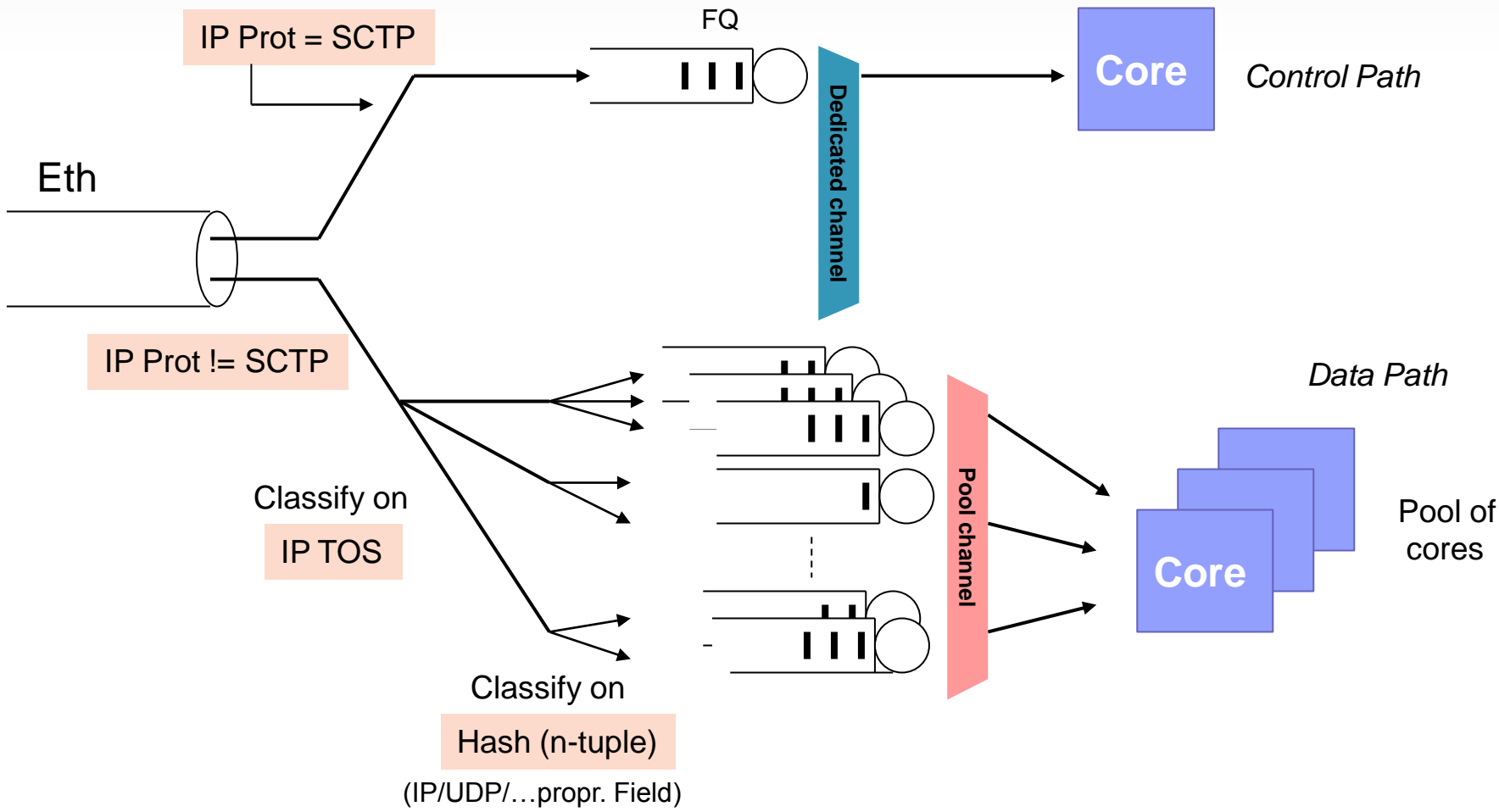
- Up to three level tree search at line rate
  - Up to 256 bytes per tree level
  - Up to 512 bytes of total key data for the three level
- Each table is an ordered list of entries, returning the first match
  - Keys for the initial table are generated from parse results/KeyGen
  - Keys for subsequent tables are generated from parse and previous lookup results
  - Keys can be generated from any fields in the frame, including proprietary UDF
  - Each entry is individually maskable.
- Each table entry has Action description
  - Queue ID, next table, hash and distribute, or Drop.
- Output of classifier
  - Single queue
  - Set of queues and a KeyGen scheme if distributing
  - Policing Profile



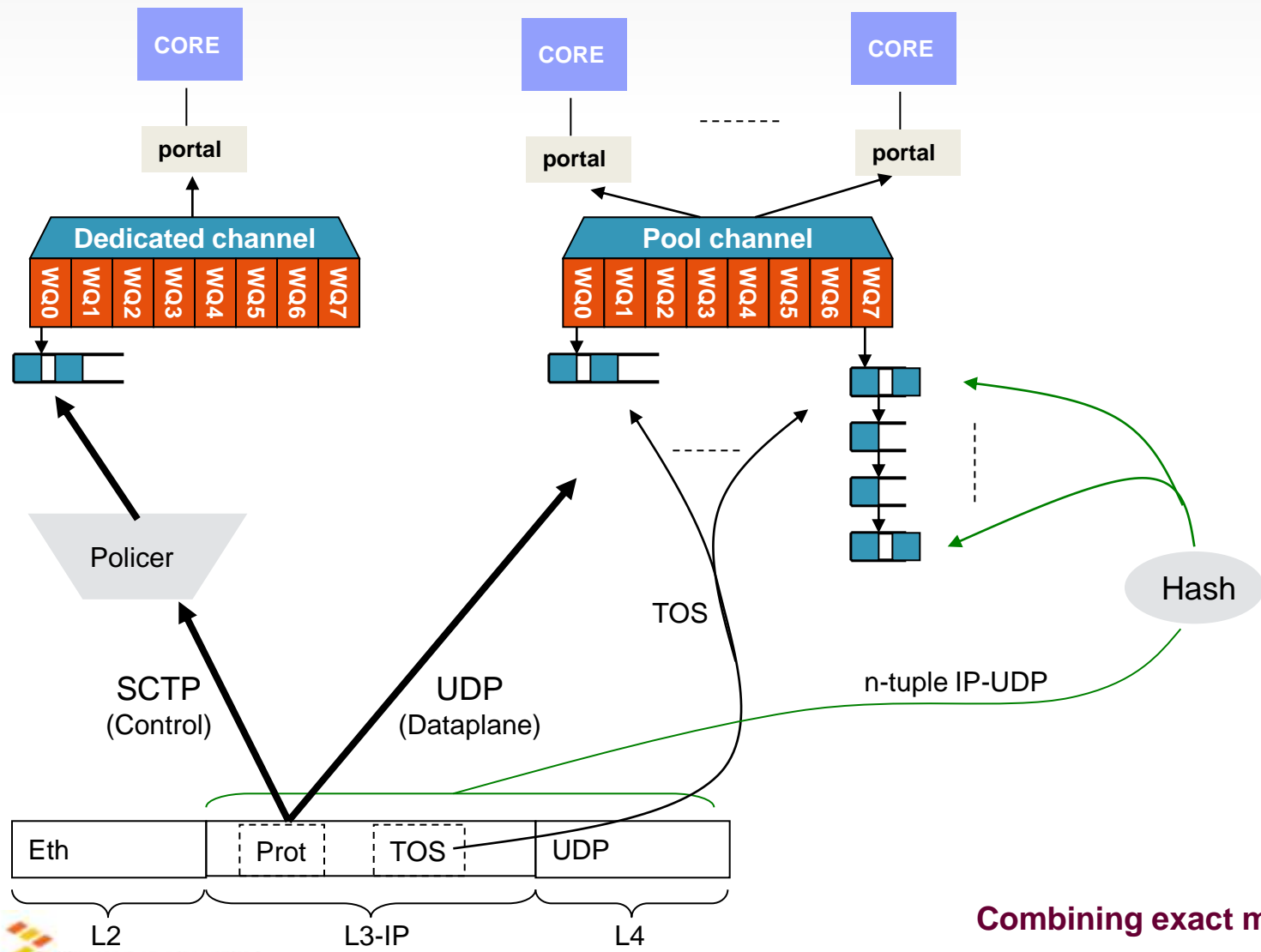




# Case Study: Spread Control and User Paths Traffic



# Case Study: High Level Mapping to DPAA



**Combining exact match + Hash**

# Case Study: Configure FMan with FMC Tool

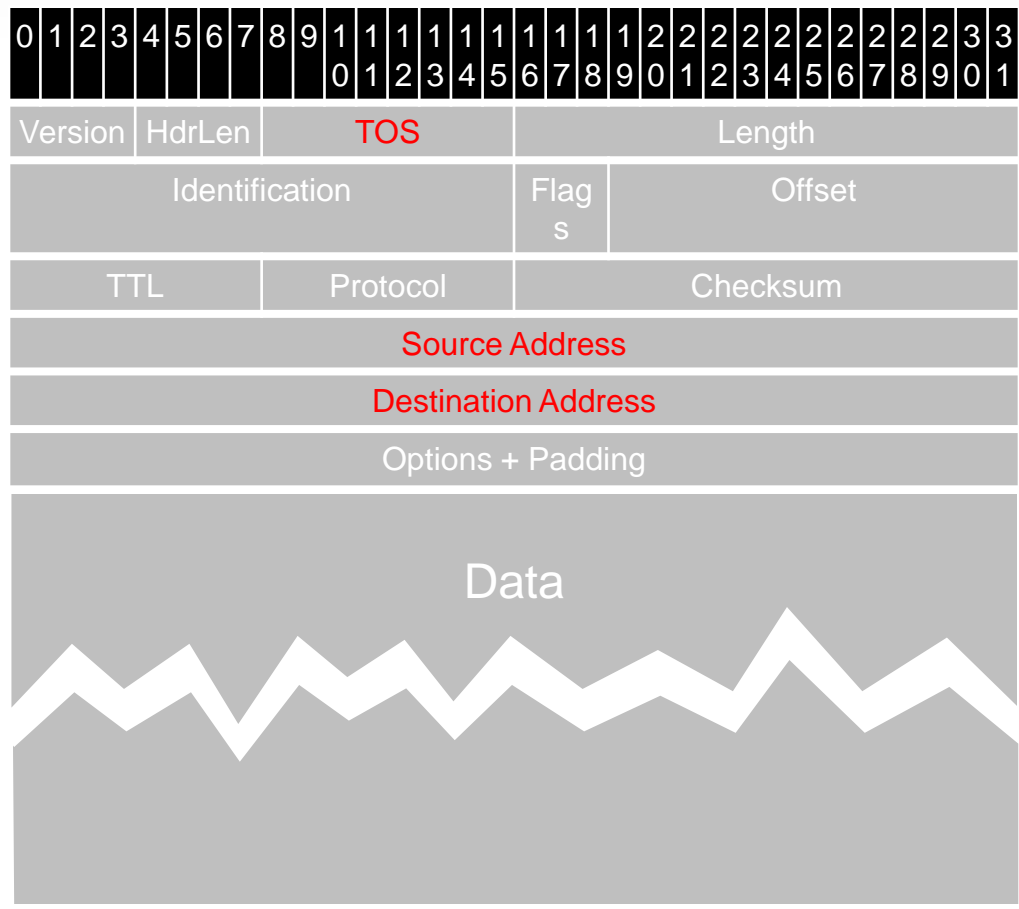
- rootfs/etc/fmc/config/8c-128fq-p.xml

```

...
<distribution name="ipv4eth0">
  <queue count="128" base="0x3800"/>
  <key>
    <fieldref name="ipv4.src"/>
    <fieldref name="ipv4.dst"/>
    <fieldref name="ipv4.tos"/>
  </key>
</distribution>

<distribution name="ipv4eth1">
  <queue count="128" base="0x3880"/>
  <key>
    <fieldref name="ipv4.src"/>
    <fieldref name="ipv4.dst"/>
    <fieldref name="ipv4.tos"/>
  </key>
</distribution>
...

```





# Agenda

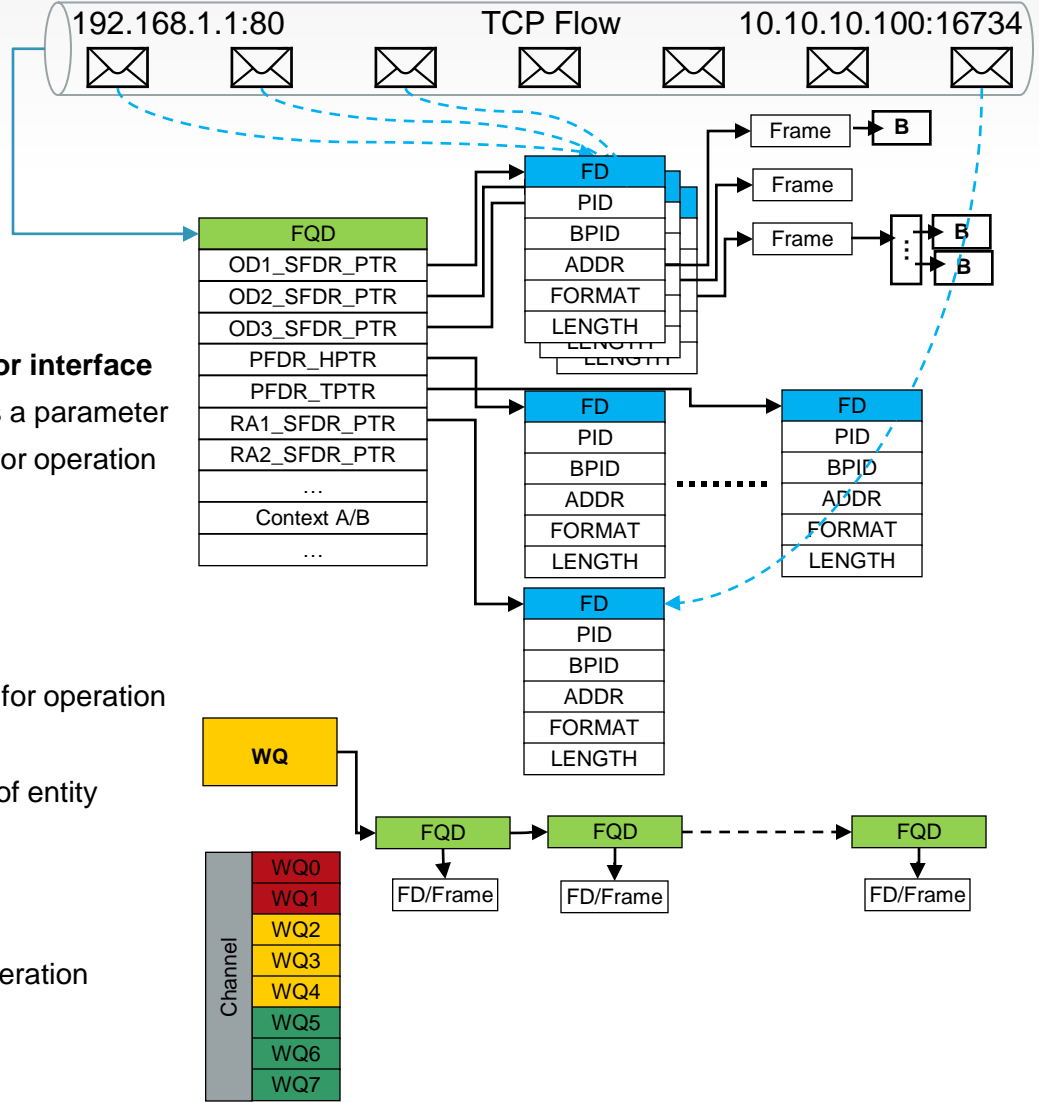
- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
  - QMan Building Blocks
  - QMan Functions
  - QMan Scheduling
  - Order Restoration, Order Preservation and Atomicity
  - Congestion Management and Avoidance
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

# New Features for T4240 Queue Manager (QMan)

- T4xxx has a total of 50 Software Portals (SP), increase from 10 SP found in the P class processors.
- Supports Customer Edge Egress Traffic Management (CEETM) that provides hierarchical class based scheduling and traffic shaping:
  - Available as an alternate to FQ/WQ scheduling mode on the egress side of specific direct connect portals
  - Enhanced class based scheduling supporting 16 class queues per channel
  - Token bucket based dual rate shaping representing Committed Rate (CR) and Excess Rate (ER)
  - Congestion avoidance mechanism equivalent to that provided by FQ congestion groups
- A total of 48 algorithmic sequencers are provided, allowing multiple enqueue/dequeue operations to execute simultaneously.
- Support up to 295M enqueue/dequeue operations per second.

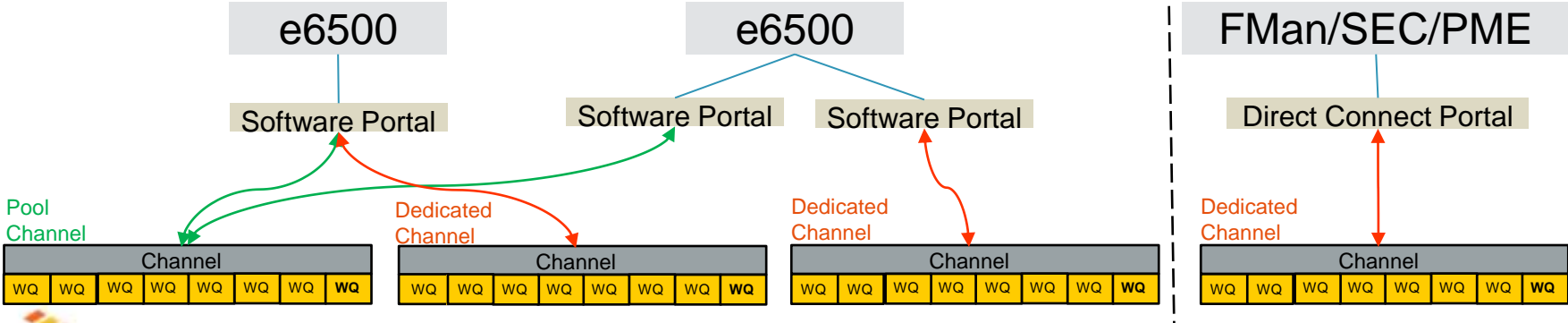
# Queuing Structure

- Frame Descriptor (FD)**
  - The basic queue Element that describe a frame
  - Usually a single IO packet will use a single frame
  - Other scenarios: commands with no buffer
- Frame Queue Descriptor (FQD)**
  - A linked list of FD's
  - Usually a frame queue is associated with a **flow or interface**
  - Enqueue operation must include the target FQ as a parameter
  - Dequeue operation may use FQ as a parameter for operation
  - Head of frame queue can be associated to ODP
- Work Queue Structure**
  - Linked list of FQD
  - Hold flows of the same priority and designation
  - Dequeue operation may use WQ as a parameter for operation
- Channel**
  - Set of eight WQ channel served by a single type of entity
  - Dequeue from channel can be configured to be:
    - strict priority
    - round robin (Simple, Weighted or Deficit)
  - Dequeue may use channel as a parameter for operation



# QMan Communicates with Portals

- Portals are the interface between QMan and the accelerator which use them
  - Direct Connect Portals has direct connect signals to Dedicated Channel
    - **Dedicated Channels** are always serviced by a single entity, e.g. FMan, etc.
  - Software Portals use CoreNet as the physical interconnect to the processor core.
    - Each Software Portal serves a **Dedicated Channel**, and optionally services **Pool Channels**.
    - Software and QMan interact by “reading” and “writing” data across CoreNet
- Each channel consists of 8 WQs, and thus there are 8 possible priorities.
- QMan contains a total of 110 channels for T4240
  - 16 Dedicated Channels (Sub Portals) per Frame Manager’s Direct Connect Portal
  - 1 for SEC (8 SPs), 1 for PME, 1 for DCE, 1 for RMan
  - Up to 50 CoreNet dedicated channels for software portals
  - 15 for CoreNet pool channels which are shared by all software portals





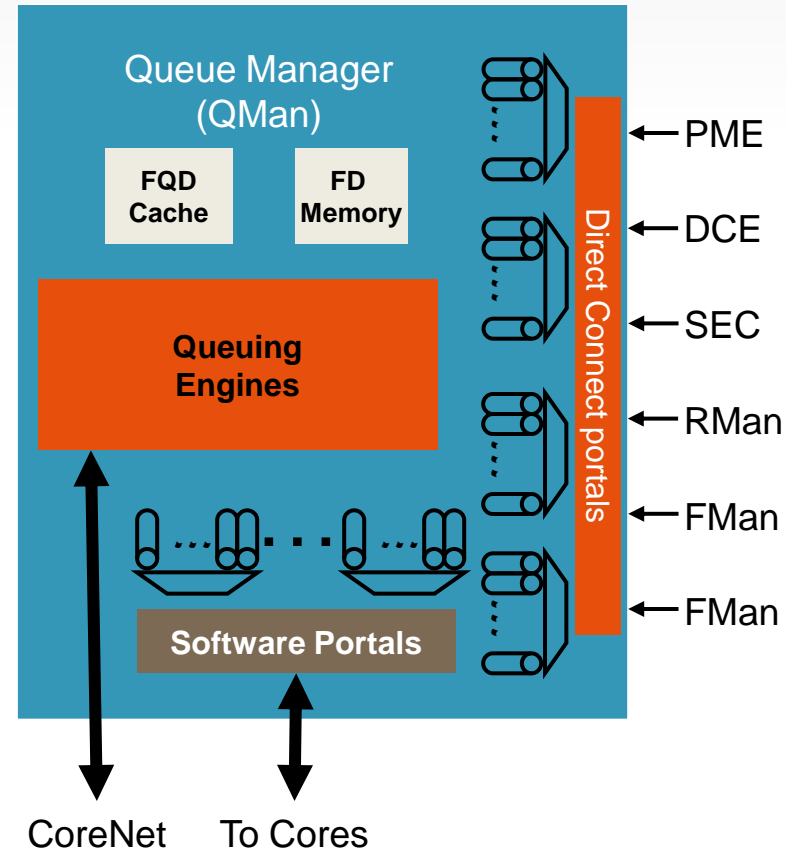
# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
  - QMan Building Blocks
  - **QMan Functions**
  - QMan Scheduling
  - Order Restoration, Order Preservation and Atomicity
  - Congestion Management and Avoidance
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

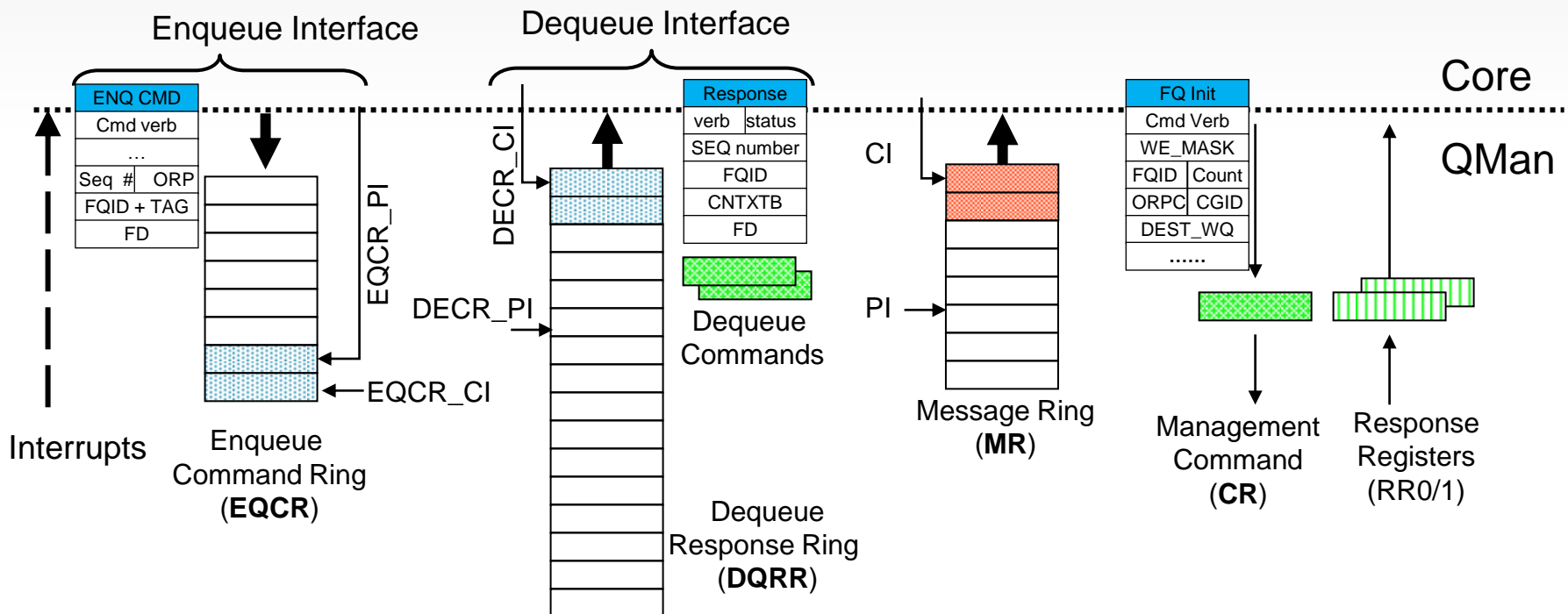


# Queue Management

- QMan provides a way to inter-connect DPAA components
  - Cores (including IPC)
  - Hardware offload accelerators
  - Network interfaces – Frame Manager
- Queue management
  - High performance interfaces (“portals”) for enqueue/dequeue
  - Internal buffering of queue/frame data to enhance performance
- Congestion avoidance and management
  - RED/WRED
  - Tail drop for single queues and aggregates of queues
  - Congestion notification for “loss-less” flow control
- Load spreading across processing engines (cores, HW accelerators)
  - Order restoration
  - Order preservation/atomicity
- Delivery to cache/HW accelerators of per queue context information with the data (Frames)
  - This is an important offload for software using hardware accelerators

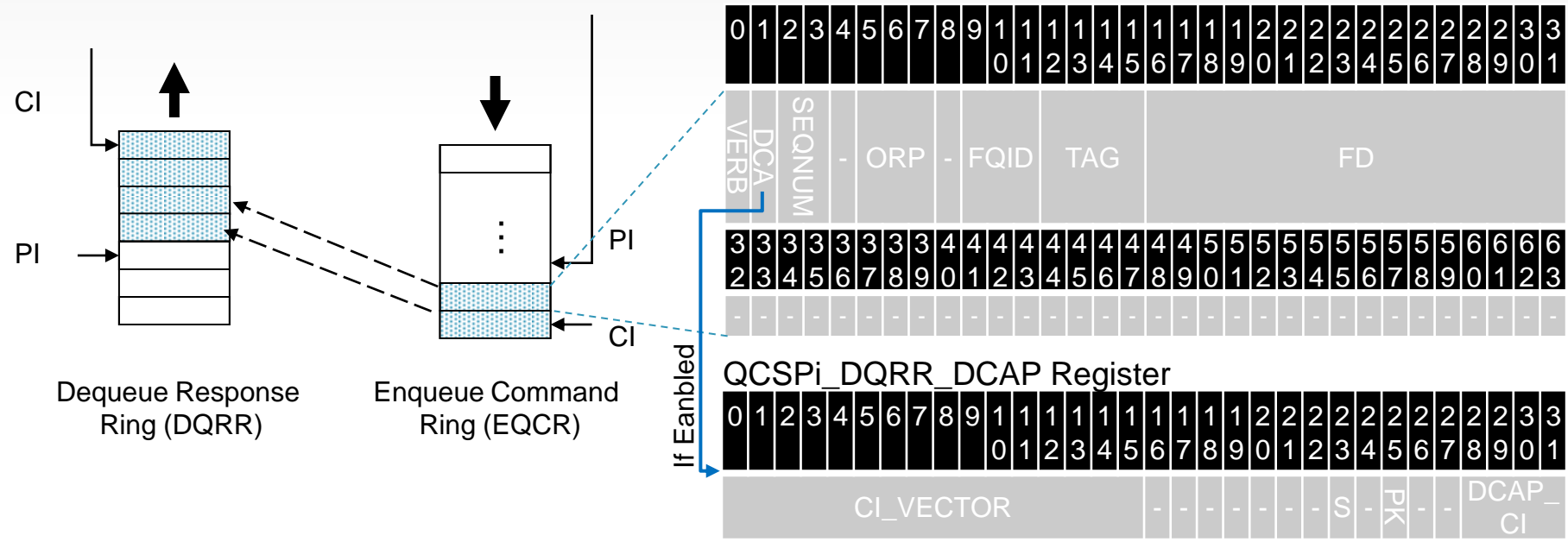


# QMan Software Portal Components



- Software portals have 4 components
  - Enqueue: EQCR
  - Dequeue: Command registers + DQRR
  - Messages: MR (e.g. enqueue rejections)
  - Management commands: command/response registers
- Interrupts can be used to signal availability of data or space (in EQCR)
- Rings provide finite size FIFOs
  - Up to 16 entries for DQRR, 8 entries for EQCR and MR
- Portal components are implemented inside QMan to reduce access latency
  - Unlike traditional BD rings which are in “memory” and “registers”
- QMan can “push” (stash) DQRR entries across CoreNet into the appropriate core’s cache
- PI and CI are the basic mechanisms used with rings but other forms of notification of data availability and data consumption are supported
- When these other mechanisms are used QMan maintains PI/CI

# Discrete Consumption Acknowledgement



- DCA is a form of “indirect” Consumer Index (CI) update
- DCA is targeted at forwarding applications where most frames which are received are then transmitted
- Each entry in the TX ring (EQCR) can contain a DCA acknowledge for the corresponding entry on the RX (DQRR) ring via the [QCSPi\\_DQRR\\_DCAP register](#), or by adding an embedded DCA to enqueue commands issued in the EQCR

# QMan Cache Warming

- In addition to stashing DQRR entries into cache, QMan's software portals can also “warm” a core's (L1 or L2) cache with frame and queue related data
  - Actual frame data for single buffer frames
  - Scatter gather list for multi-buffer frames
  - Frame “annotations”
    - Data between Address and Offset at start of frame
    - Used to pass additional information about the frame which is not “frame data” e.g. FM parse results
  - Data referenced by FQ Context
- Stashing options can be configured on a per FQ basis
- Cache warming are performed at the time that the frame is dequeued (i.e., DQRR entry is created)

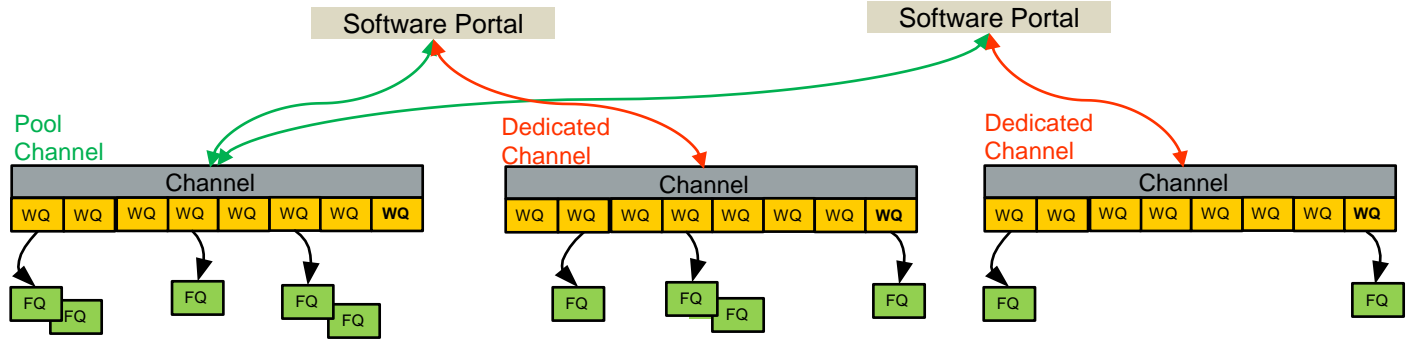


# Agenda

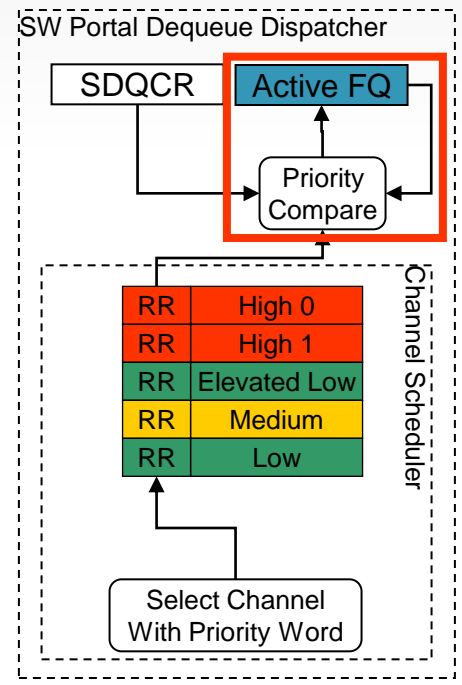
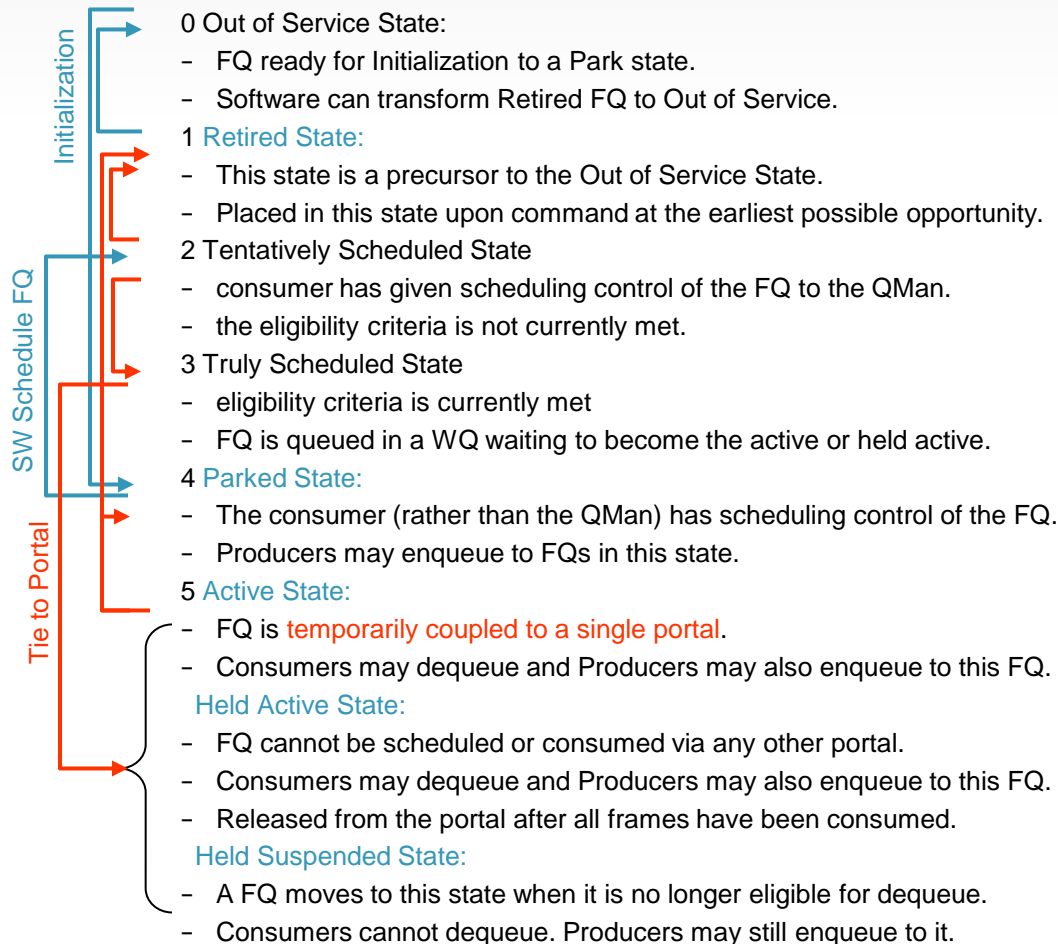
- DPAA Overview
- BMan Enablement
- FMan Enablement
- QMan Enablement
  - QMan Building Blocks
  - QMan Functions
  - **QMan Scheduling**
  - Order Restoration, Order Preservation and Atomicity
  - Congestion Management and Avoidance
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

# Core Load Distribution

- Static flow-based distribution (**Dedicated Channel**)
  - Set of WQs with different FQs directed statically to different cores
  - Distribution of frames (selection of FQ) is based on hash keys, ensuring that packets from the same traffic flow will always go to the same cores
  - Static not dynamic, doesn't react to core load, assign work to the cores in a static or fixed manner
- Adaptive load balancing (**Pool Channel**)
  - Load spread the packets (or the Frame Queue) to the cores based on actual core availability/readiness
  - QMan provides two mechanisms to deal with out of order packets:
    - Order preservation: ensure that related packets are processed in order (and typically one at a time); can also provide “atomicity” – atomic access to data
    - Order restoration: allows frames to be processed out of order and then restores their order later on before they are transmitted



# Frame Queue Life Cycle



# Dequeue Modes

- QMan supports 2 modes on software portals:
  - Push Mode:
    - Qman continues to push entries into DQRR in attempt to keep it “full”
    - QMan provides 2 command registers
      - One register is “static” and QMan repeatedly executes this command
      - One register is “volatile” and QMan executes that command a limited number of times
    - Push mode is “just like” a BD ring
  - Pull Mode:
    - QMan provides a single command register
    - Software must issue a new command for each dequeue operation
- Push mode is the most common mode
- Pull mode offers more control to applications

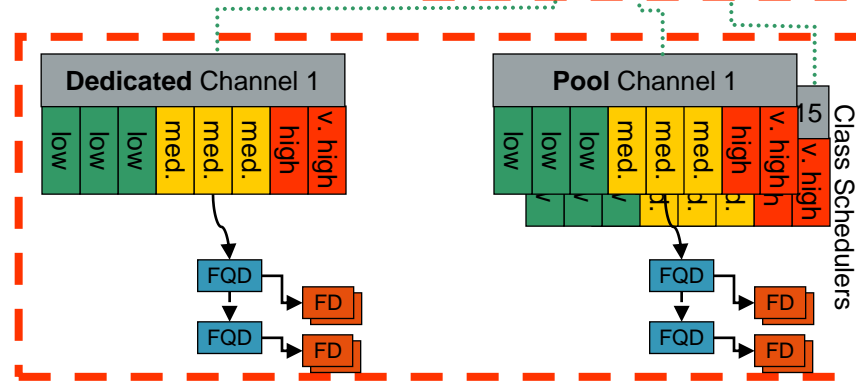
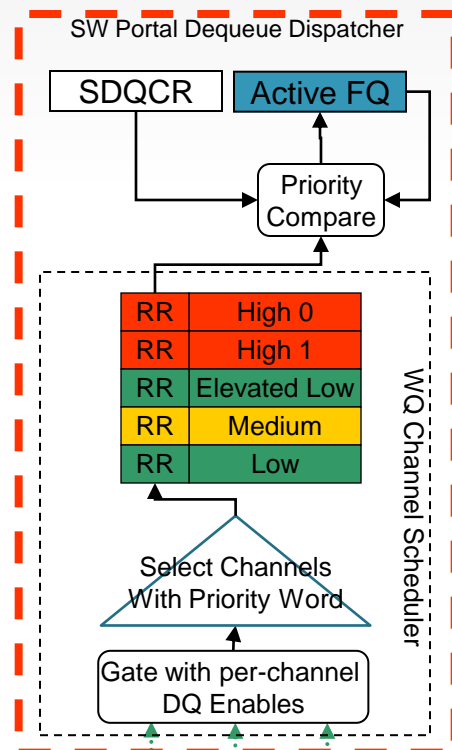


# Dequeue Scheduling

- Class Scheduler schedules **WQ**
  - A Class scheduler per channel
  - Two levels of scheduling:
    - Use Weighted Interleaved Round Robin to schedule within the medium priority queues (2 to 4) and low priority queues (5 to 7)
    - Strict priority of all 8 WQs with programmable elevation (CS\_ELEV) of the low priority tier over the medium priority tier
  - Maintain active FQ states transition
    - Keeps track of the last RR winner, selection counters, elevation pending counter
- Intra-Class scheduling schedules **FQ**
  - Schedule a frame queue within a work queue
  - Use Modified Deficit Round Robin with ICS\_CRED + ICS\_SURP

15 bit Credit	16 bit Surplus
---------------	----------------

- First Dequeue **surplus = surplus + credit**
- Dequeue 1-3 frames,
- subtract frame length(s) bytes from surplus
- If surplus > 0, dequeue 1-3 frames more
- If surplus <=0, reschedule Frame Queue



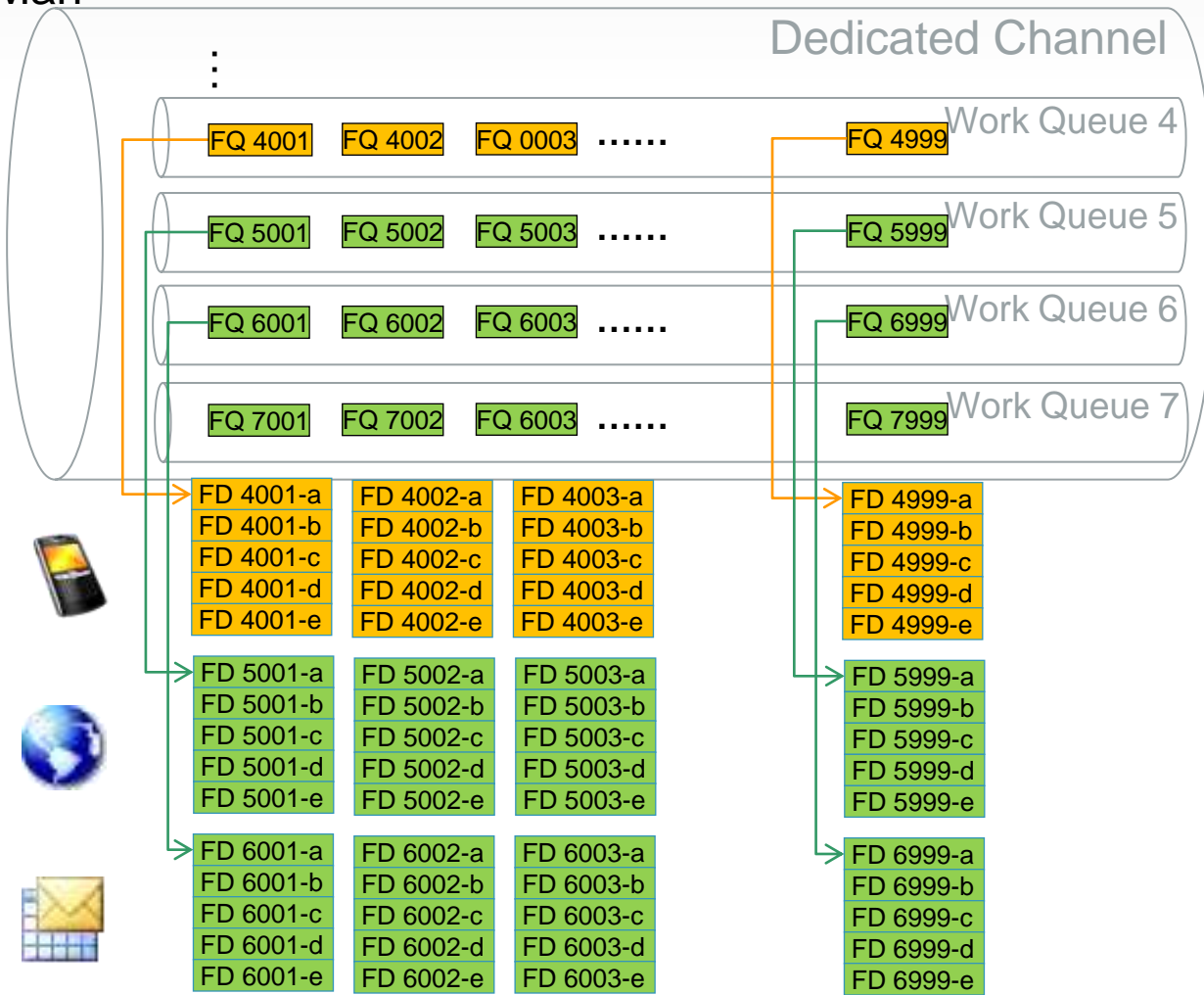
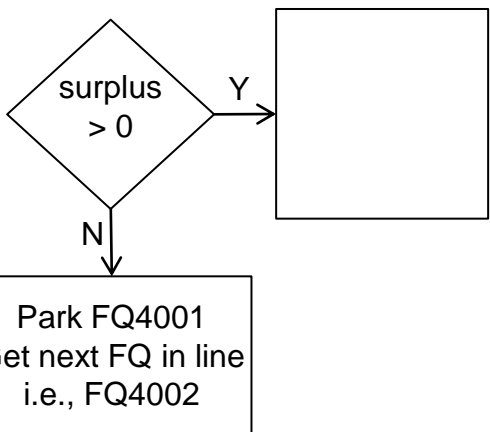
# Static Dequeue: Who Is on First?



e500 core X QMan

Push Mode

SDQCR						
-	S	F	D	-	-	DCT
	S	C	P			
Token						
DQ SRC						
DQ SRC (cont)						



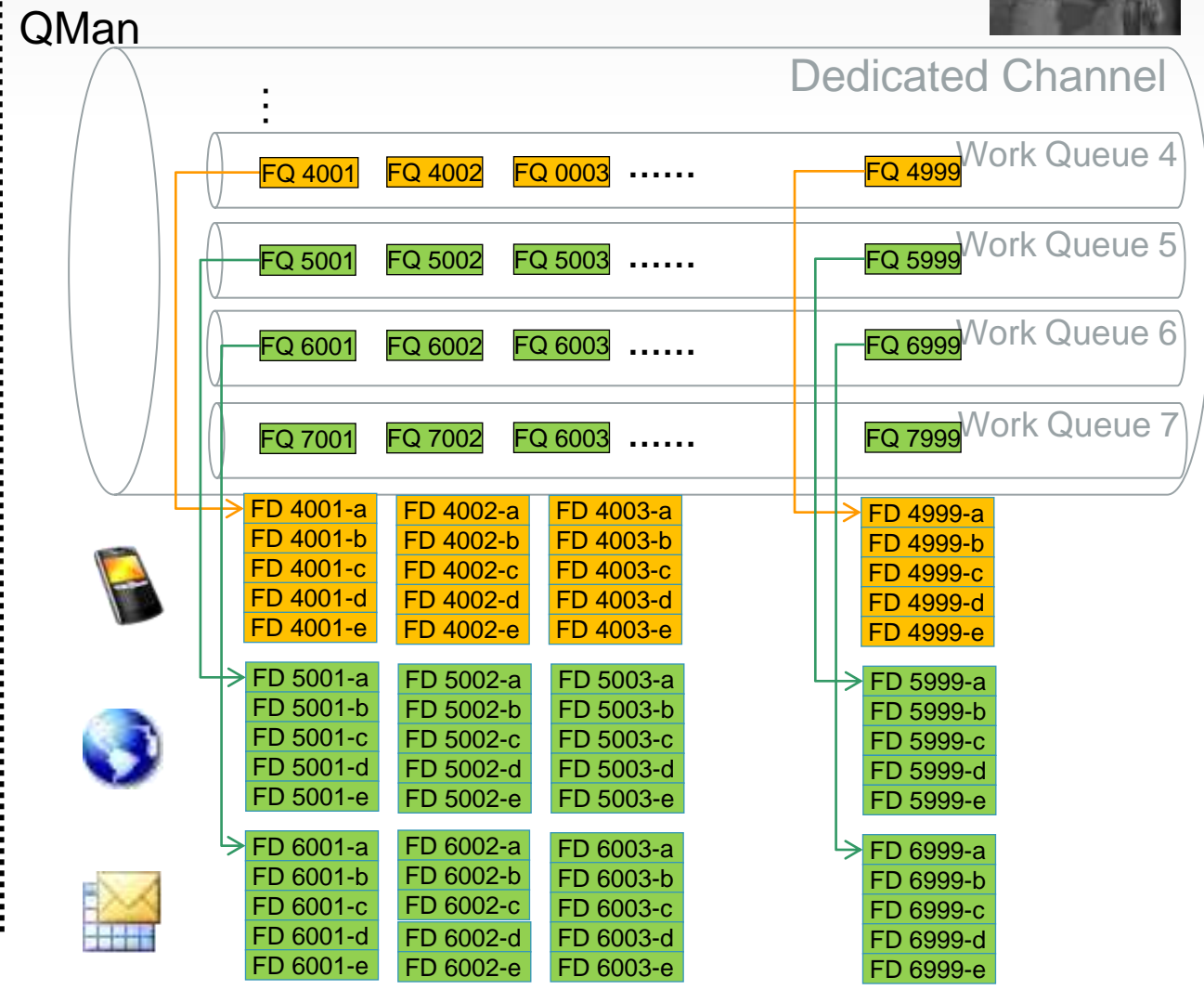
# Volatile Dequeue: Who Is on First?



e500 core X QMan

Push Mode

VDQCR		
P	E	Number of Frames
FQID=FQ6002		
.....		
.....		

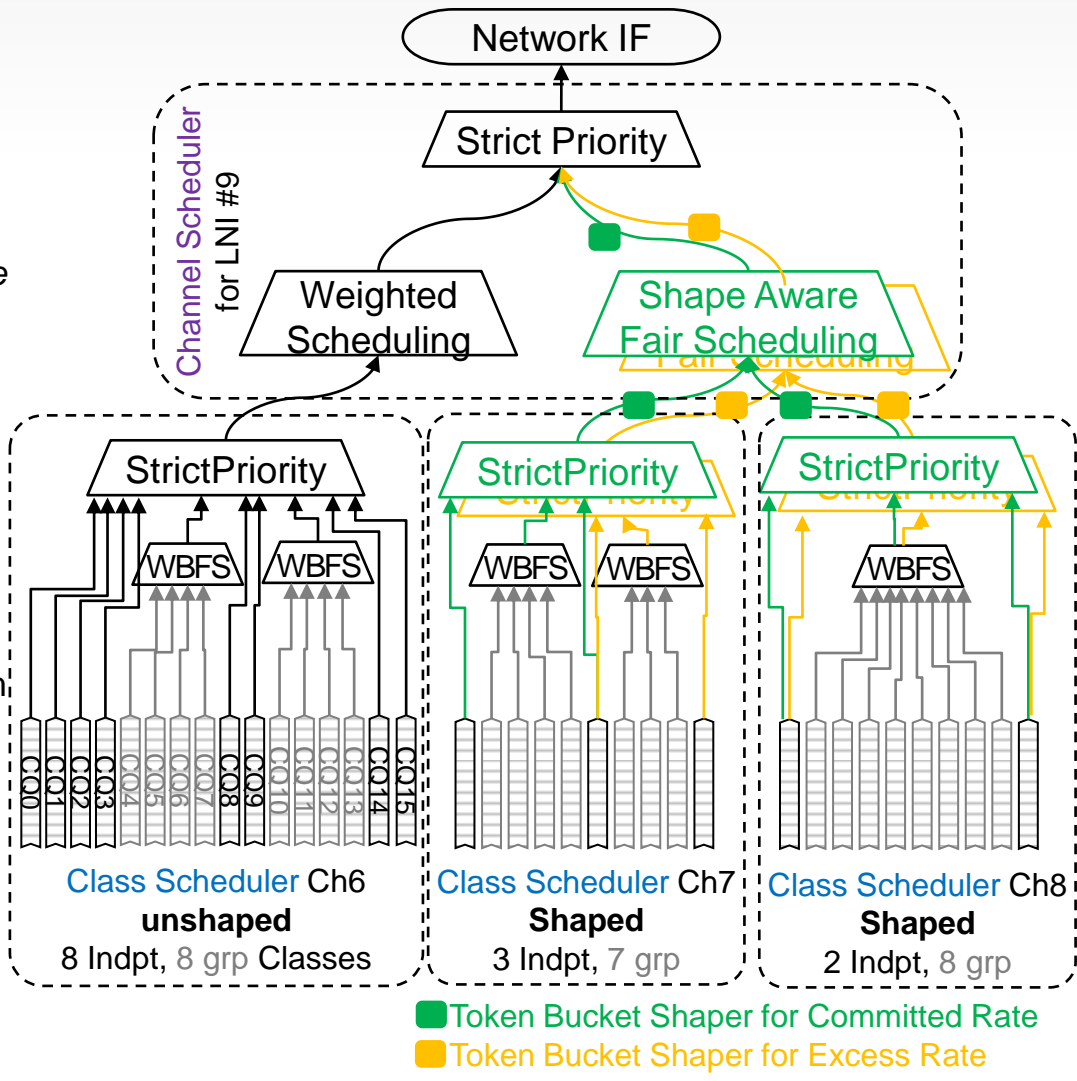


# Customer Edge Egress Traffic Management (CEETM)

- QMan 1.2 (i.e. QorIQ T42xx) supports egress traffic management by provides hierarchical class based scheduling and traffic shaping.
- On a specific QMan Direct Connect Portals (DCPs), each sub-portal that supports CEETM can be configured to use either the regular FQ/WQ scheduling mode OR CEETM scheduling mode.
  - A given sub-portal can switch between FQ/WQ and CEETM scheduling mode.
  - CEETM supports up to 8 logical network interfaces (LNI) that can each be mapped to a DCP sub-portal, whereas a DCP can support up to 16 sub-portals.
  - CEETM is supported on only a subset of the DCP portals, not on all DCP portals.
  - A single instance of the CEETM is associated with a single DCP and therefore a single egress I/O module.
- CEETM maintains the following functionality equivalent of regular FQ/WQ scheduling mode:
  - Congestion management capabilities including WRED
  - Dequeued frame context (Context\_A and Context\_B)
  - Priority or traffic class flow control
- FQID xF00000 – xFFFFFFF are reserved for CEETM as Logical FQIDs (LFQID).
  - xF00000 – xF00FFF (4k) for DCP portal 0 (i.e. FMAN0)
  - xF10000 – xF10FFF (4k) for DCP portal 1 (i.e. FMAN1)

# CEETM Scheduling Hierarchy (QMAN 1.2)

- Logics
  - Green** denotes logic units and signal paths that relate to the request and fulfillment of *Committed Rate (CR)* packet transmission opportunities.
  - Yellow** denotes the same for *Excess Rate (ER)*.
  - Black** denotes logic units and signal paths that are used for unshaped opportunities or that operate consistently whether used for CR or ER opportunities.
- Scheduler
  - Channel Scheduler**: channels are selected to send frame from Class Queues.
  - Class scheduler**: frames are selected from Class Queues. Class 0 has highest priority.
- Algorithm
  - Strict Priority (SP)
  - Weighted Scheduling
  - Shaped Aware Fair Scheduling (SAFS)
  - Weighted Bandwidth Fair Scheduling (WBFS)



# Weighted Bandwidth Fair Scheduling (WBFS)

- Weighted Bandwidth Fair Scheduling (WBFS) is used to schedule packets from queues within a priority group (A or B group) such that each gets a “fair” amount of bandwidth made available to that priority group.
- The premises for fairness for algorithm is:
  - available bandwidth is divided and offered equally to all classes.
  - offered bandwidth in excess of a class’s demand is to be re-offered equally to classes with unmet demand.

	Initial Distribution		First ReDistribution		Second Redistribution		Total BW Attained
BW available	10G		1.5G		.2G		0G
Number of classes with unmet demand	5		3		2		
Bandwidth to be offer to each class	2G		.5G		.1G		
	Demand	Offered & Retained	Unmet Demand	Offered & Retained	Unmet Demand	Offered & Retained	
Class 0	.5G	.5G	0				.5G
Class 1	2G	2G	0				2G
Class 2	2.3G	2G	.3G	.3G	0		2.3G
Class 3	3G	2G	1G	.5G	.5G	.1G	2.6G
Class 4	4G	2G	2G	.5G	1.5G	.1G	2.6G
<b>Total Consumption</b>	<b>11.8G</b>	<b>8.5G</b>		<b>1.3G</b>		<b>.2G</b>	<b>10G</b>



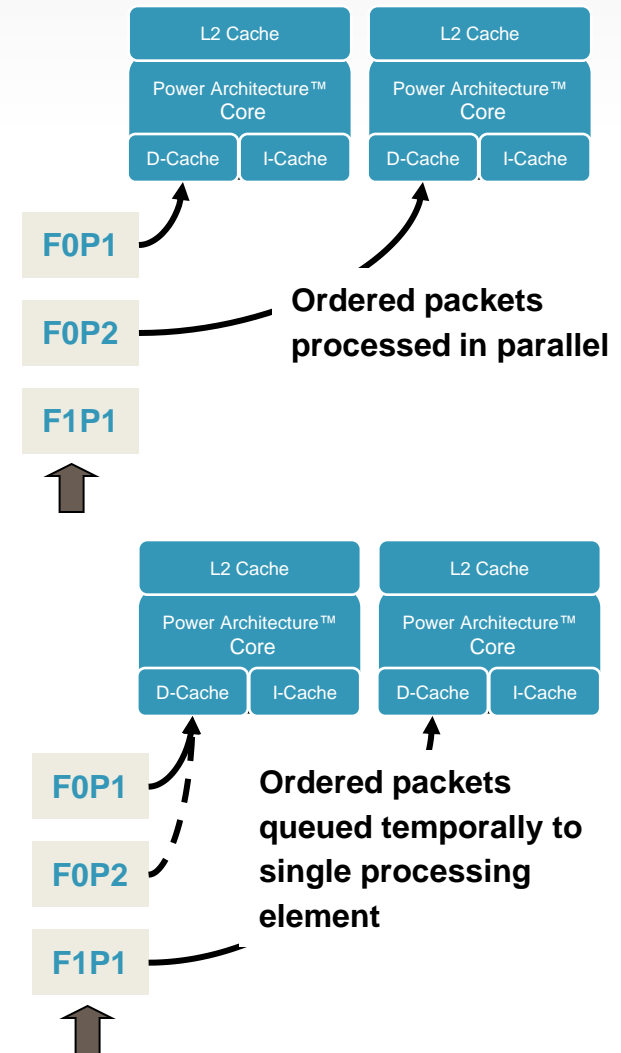


# Agenda

- DPAA Overview
- BMan Enablement
- FMan Enablement
- QMan Enablement
  - QMan Building Blocks
  - QMan Functions
  - QMan Scheduling
  - **Order Restoration, Order Preservation and Atomicity**
  - Congestion Management and Avoidance
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

# Addressing Ordering Requirements

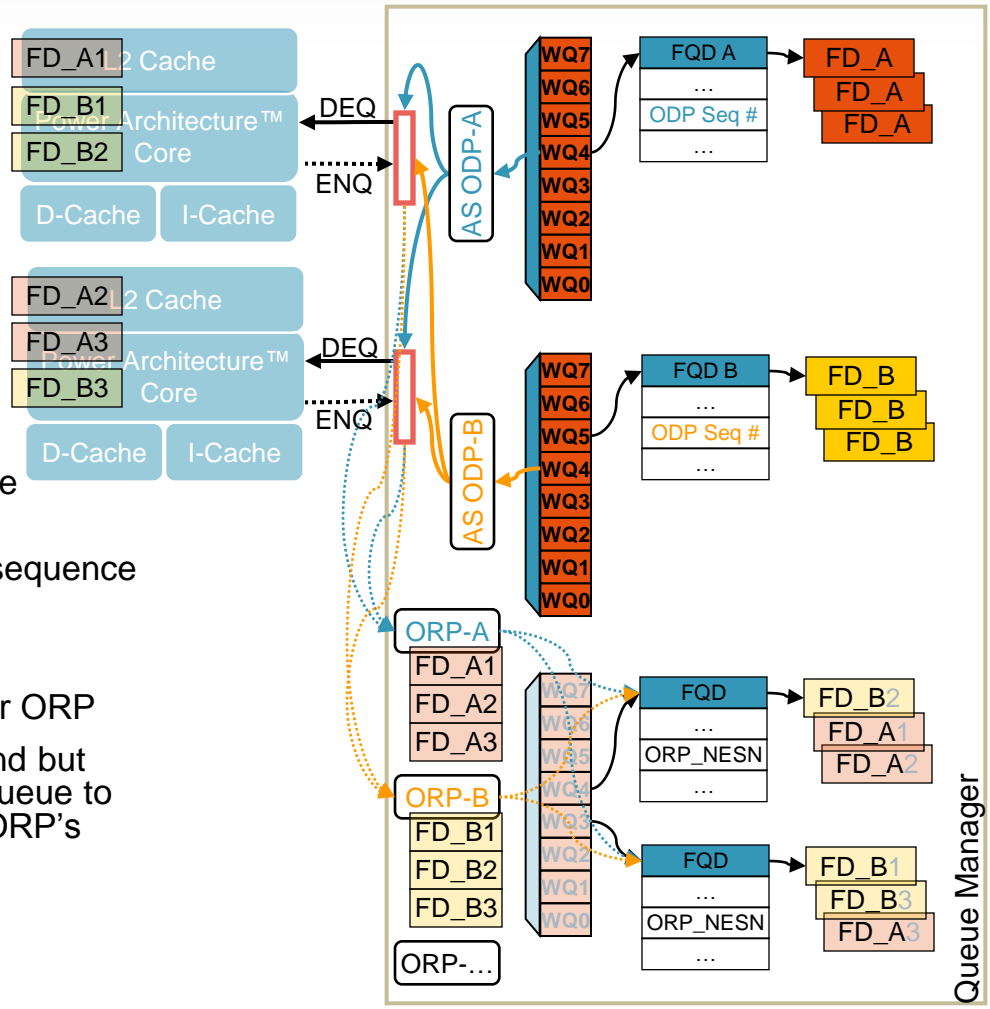
- There are two basic approaches to addressing this requirement:
  - Order restoration
    - Take note of the correct order (or sequence) of packets before processing starts and restore the packets to that order before they are transmitted
  - Order preservation
    - Ensure that related packets are processed in order (and typically one at a time)
    - Order preservation can also provide “atomicity” – atomic access to data used in processing the frame
- QMan requires that related frames (which must be transmitted in order) be placed on the same frame queue for both of these approaches
  - This does not mean that only related frames are placed on a given FQ
  - Many sets of related frames can be placed on an FQ
  - Frame Manager is responsible for achieving this





# QMan Order Restoration

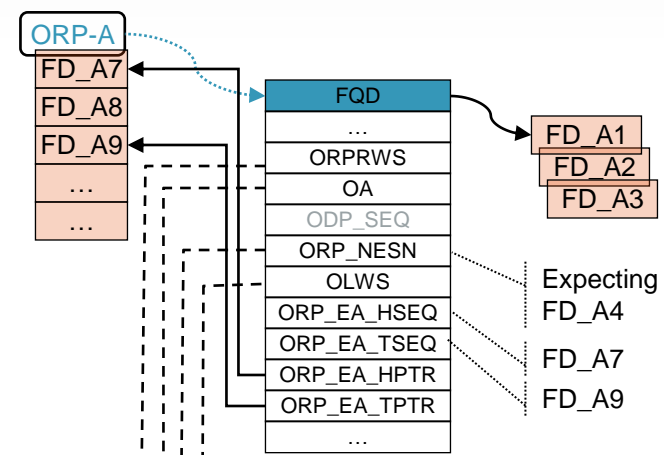
- QMan's order restoration support has two components:
  - Order Definition Point (ODP)
    - A point defined relative sequence to each Frames' pass
    - ODP id is associated to FQ-ID
    - assigning a monotonically increasing 14 bits sequence number to a series of frames
    - QM supports single ODP on head of queue
    - ODP can be made anywhere in the system i.e. SW can be an ODP
  - Order Restoration Point (ORP)
    - A point relative sequence associated to single ODP is restored
    - Allows frames insertion into the flow (single sequence number with more/last indication)
- Behavior highlights
  - Configurable number of "in-flight" packets per ORP
  - resection of ORP is part of enqueue command but Queue tail is not associated to ORP i.e., enqueue to single destination queue can respect many ORP's
  - Note: Frames are not "marked"



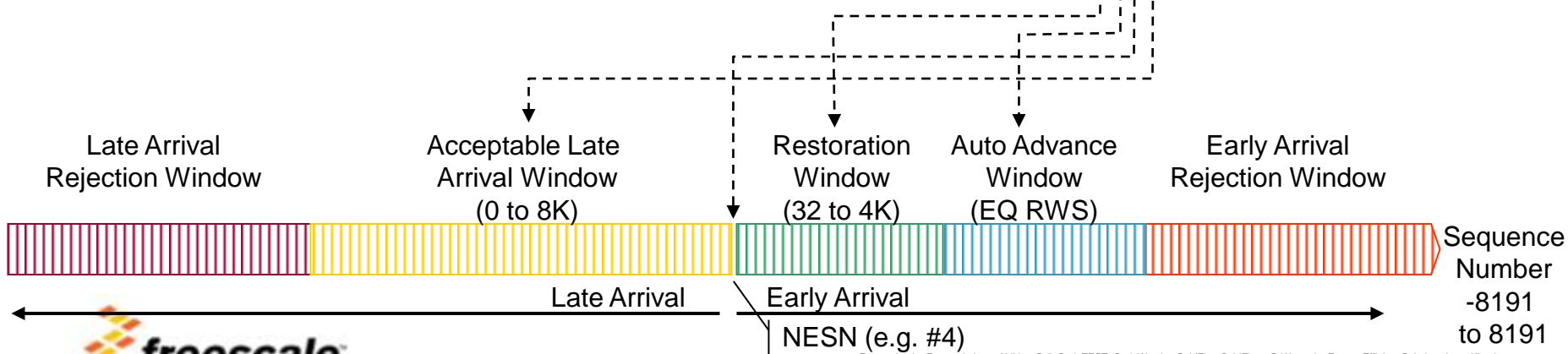
ODP-A  
ODP-B

# Order Restoration Configuration

- Treatment of a FD is determined by:
  - ORP Restoration Window Size (ORPRWS)
  - ORP Auto Advance NESN Window Toggle (OA)
  - ODP Sequence Number (SEQ)
  - ORP Next Expected Sequence Number (NESN)
  - ORP Acceptable Late Arrival Window Size (OLWS)
  - ORP Early Arrival Head Sequence Number (EA\_HSEQ)
  - ORP Early Arrival Tail Sequence Number (EA\_TSEQ)
  - ORP Early Arrival Head Pointer (EA\_HPTR)
  - ORP Early Arrival Tail Pointer (EA\_TPTR)



Note: SEQ is 14 bit wide. Create a sliding (8k - 1) sliding windows before and after NESN



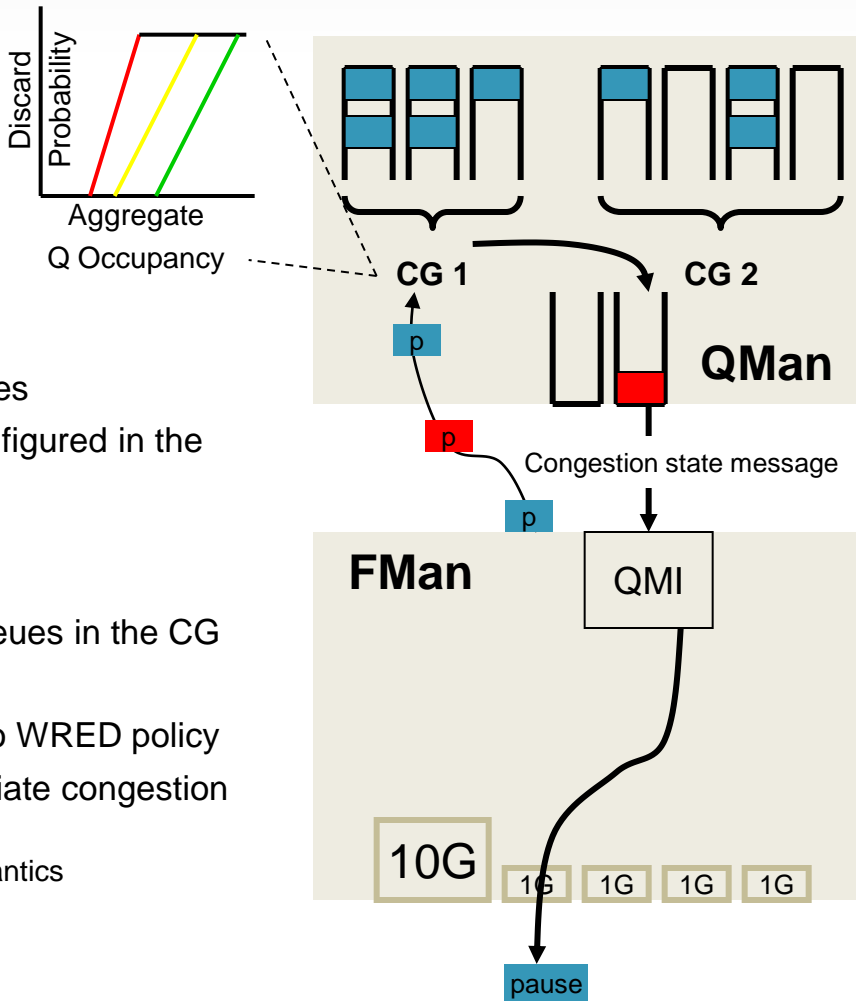


# Agenda

- DPAA Overview
- BMan Enablement
- FMan Enablement
- QMan Enablement
  - QMan Building Blocks
  - QMan Functions
  - QMan Scheduling
  - Order Restoration, Order Preservation and Atomicity
  - Congestion Management and Avoidance
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

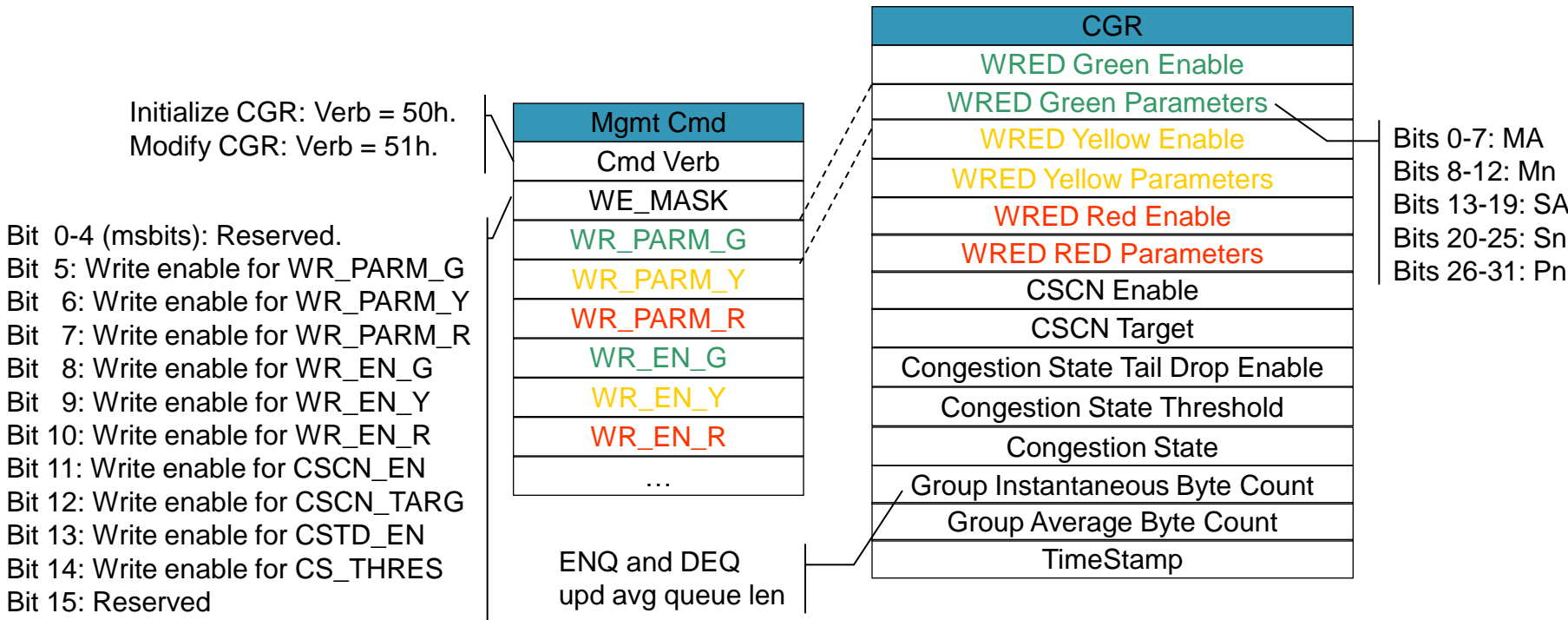
# Congestion Management and Avoidance

- Both FM and QMan are involved in supporting congestion management and avoidance
- QMan provides the following support
  - Congestion management (**loss-less flow control, threshold/tail drop**)
  - Congestion avoidance (**RED/WRED**)
- Congestion Groups (CG) define granularity
  - Every frame queue has a tail drop threshold and configured to a congestion group
  - Congestion calculations are done on groups of queues
  - Congestion avoidance/management calculations configured in the CG
- Congestion Group Details
  - 256 Congestion Groups
  - Time aware weighted average queue depth of all queues in the CG
  - 3 color configurable WRED curves
  - Enqueued packet may be rejected (discarded) due to WRED policy
  - Instantaneous CG depth +/- hysteresis value can initiate congestion state messages to enqueue sources
    - Lossless flow control on interfaces supporting PAUSE semantics



# Setup Congestion Group Record

- On enqueue:
  - The color for the frame being enqueued is used to select a probability curve
  - The frame may be selected for random discard

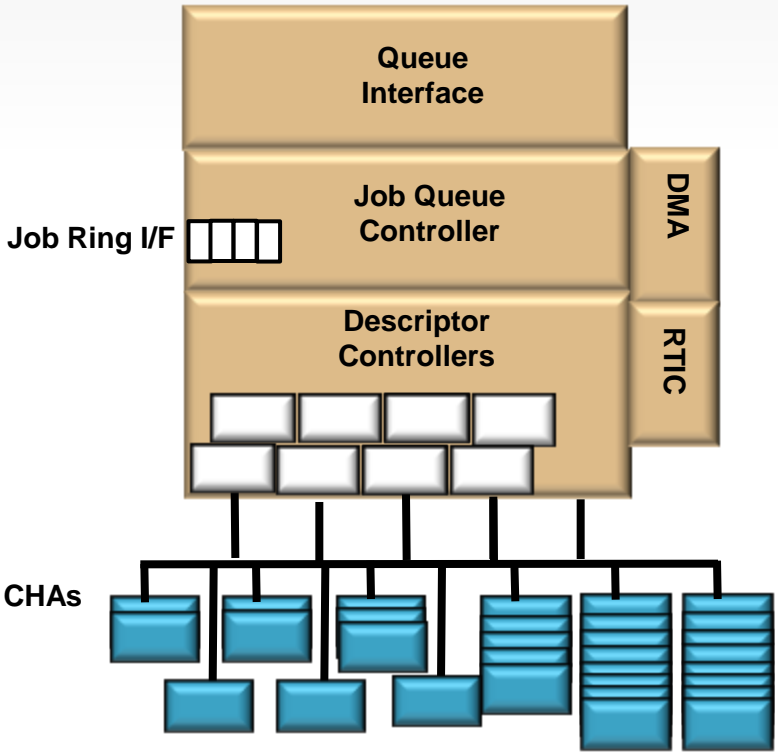




# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- **SEC Enablement**
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

# Security Block - Version 5.0



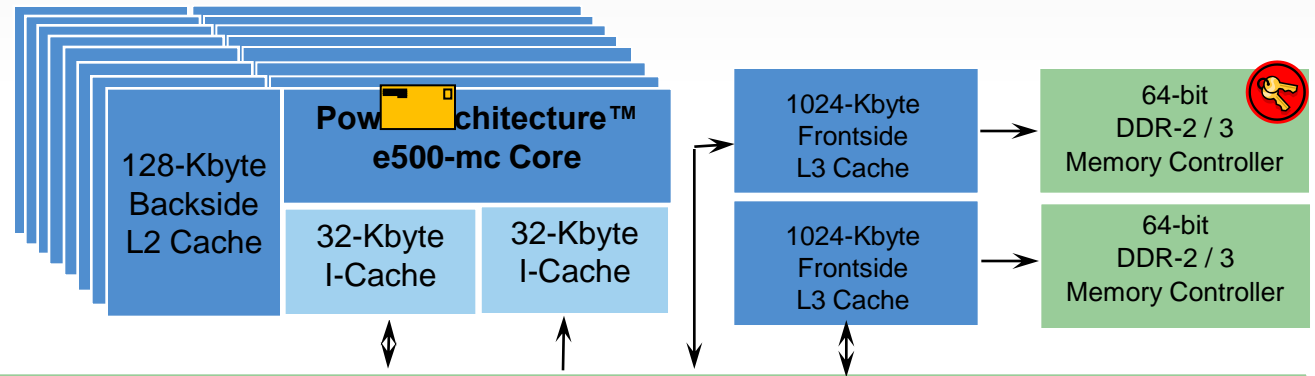
Supports protocol processing for the following:

- IPsec
- 802.1ae (MACSEC)
- SSL/TLS/DTLS
- 3GPP RLC
- LTE PDCP
- SRTP
- 802.11i (WiFi)
- 802.16e (WiMax)

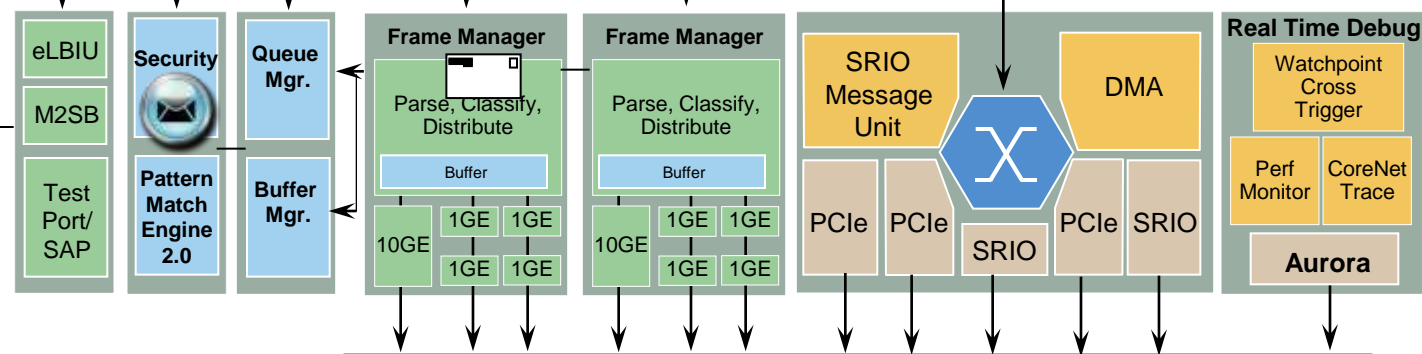
- Public Key Hardware Accelerators (PKHA)
  - RSA and Diffie-Hellman (to 4096b)
  - Elliptic curve cryptography (1023b)
- Data Encryption Standard Accelerators (DESA)
  - DES, 3DES (2K, 3K)
  - ECB, CBC, OFB modes
- Advanced Encryption Standard Accelerators (AESA)
  - Key lengths of 128-, 192-, and 256-bit
  - ECB, CBC, CTR, CCM, GCM, CMAC, OFB, CFB, and XTS
- ARC Four Hardware Accelerators (AFHA)
  - Compatible with RC4 algorithm
- Message Digest Hardware Accelerators (MDHA)
  - SHA-1, SHA-2 256,384,512-bit digests
  - MD5 128-bit digest
  - HMAC with all algorithms
- Kasumi/F8 Hardware Accelerators (KFHA)
  - F8 , F9 as required for 3GPP
  - A5/3 for GSM and EDGE
  - GEA-3 for GPRS
- Snow 3G Hardware Accelerators (STHA)
  - Implements Snow 3.0
- ZUC Hardware Accelerators (ZHA)
  - Implements 128-EEA3 & 128-EIA3
- CRC Unit
  - Standard and user defined polynomials
- Random Number Generator, random IV generation



# Life of a Crypto Packet



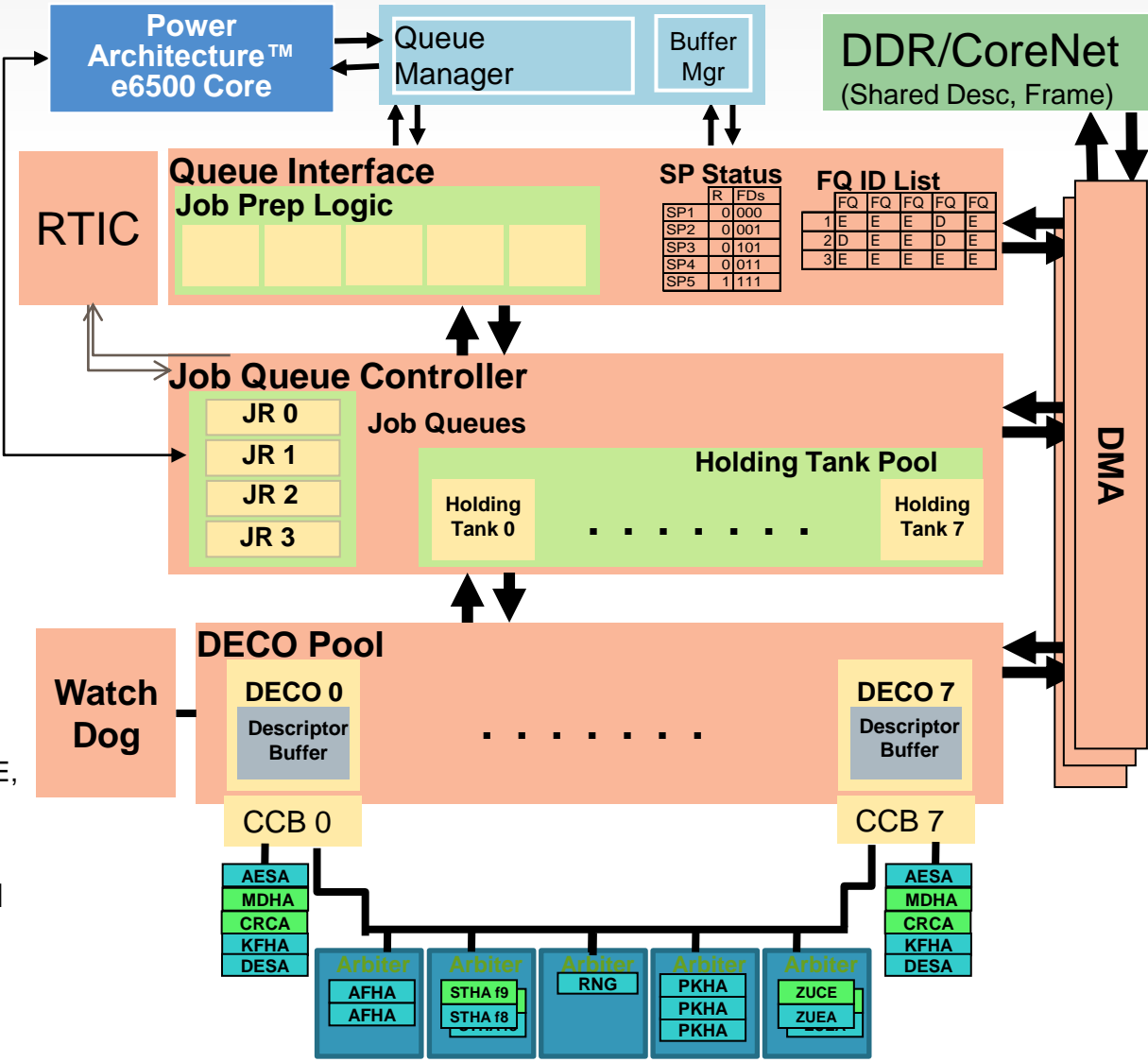
- eOpenPIC
- PreBoot Loader
- Security Monitor
- Internal BootROM
- Power Mgmt
- SD/MMC
- SPI
- DUART
- 2x I<sup>2</sup>C
- 2x USB 2.0/ULPI
- Clocks/Reset
- GPIO
- CCSR





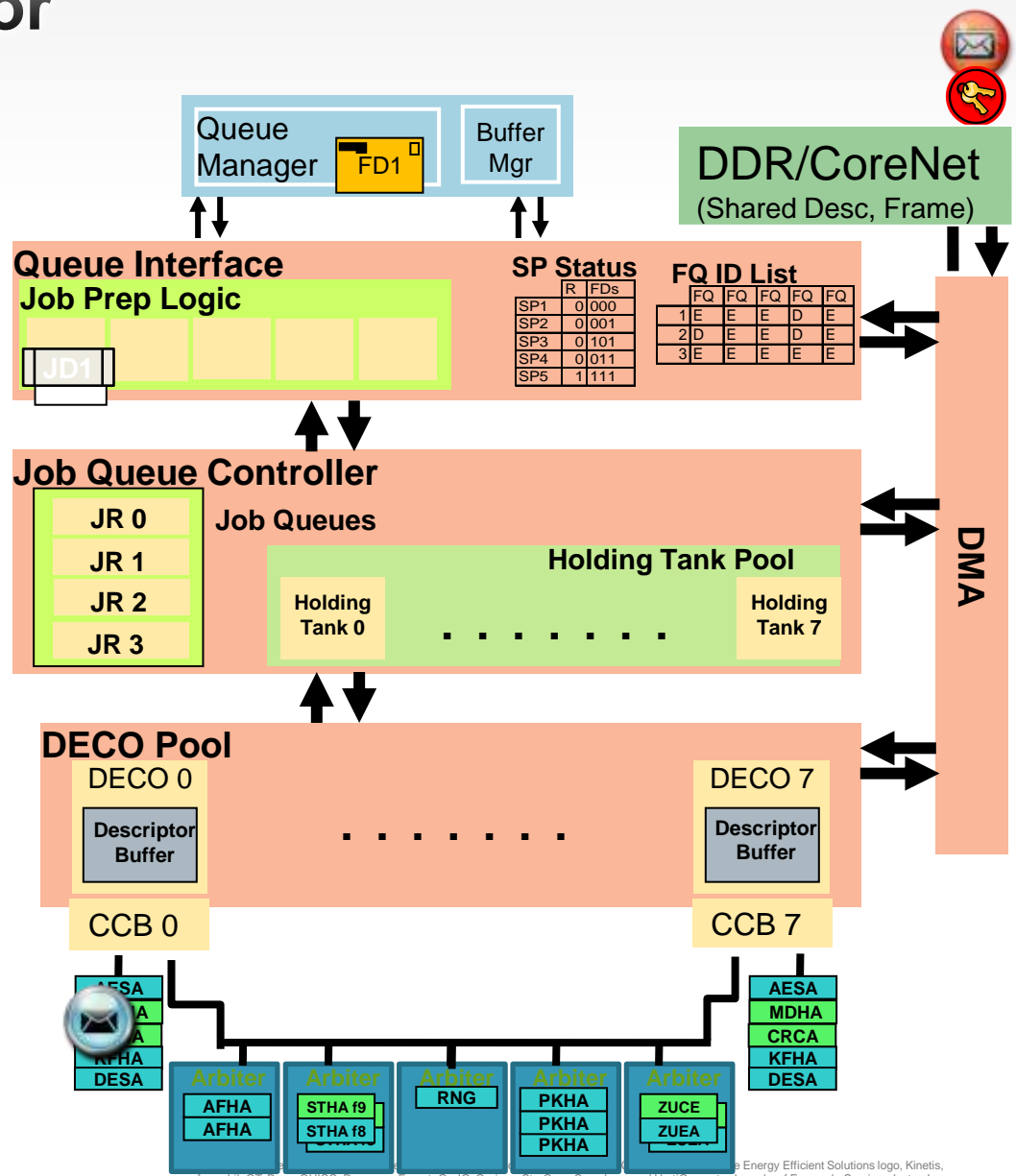
# SEC 5.x Logical Block Diagram

- JQ Controller take inputs from:
  - JR (Direct Mode)
  - QI (DPAA Mode)
  - RTIC
- DESCRIPTOR Controller
  - 8x T4240
  - 5x P4080
  - 3x P3041/P2040
- CHA Control Block
- Crypto Hardware Accelerator (CHA)
  - Dedicated CHAs
    - 8x AESA, MDHA, CRCA, KFHA, DESA
  - Pool CHAs
    - RNG, AFHA, PKHA, STHA, ZUCE, ZUCA
- Watch Dog Timer
  - Monitors DECOs for prolonged inactivity



# Life of a Job Descriptor

- QI has room for more work, issues dequeue request.
- Qman selects FQ and provides 1 FD along with FQ Information
- QI creates [internal] Job Descriptor
- QI transfers completed Job Descriptor into one of the Holding Tanks
- Job Queue Controller finds an available DECO, transfers JD1 to it
- DECO initiates DMA of SD and places it in Descriptor Buffer with JD from Holding Tank
- DECO executes descriptor commands, loading registers and FIFOs in its CCB
- CCB obtains and controls CHA(s) to process the data per DECO commands
- DECO commands DMA to store results and any updated context to system memory
- As input buffers are being emptied, DECO tells QI, which may release them back to BMan
- Upon completion of all processing through CCB, DECO resets CCB
- DECO informs QI that JD1 has completed with status code X, data of length Y has been written to address Z
- QI creates outbound FD, enqueues to Qman using FQID from Ctx B field



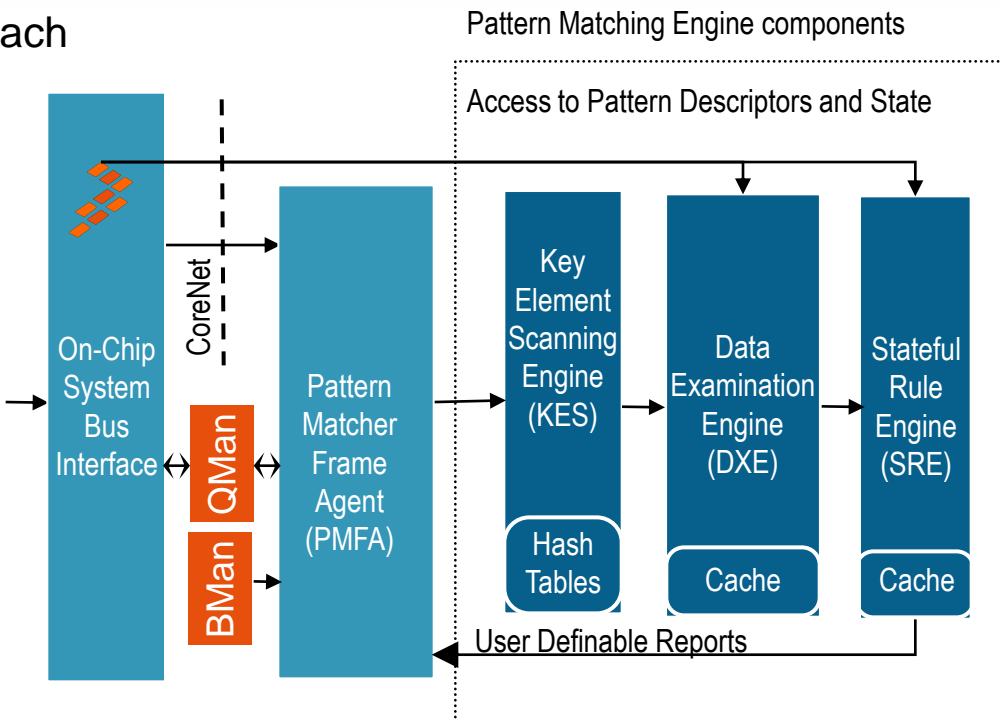


# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- SEC Enablement
- **PME Enablement**
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

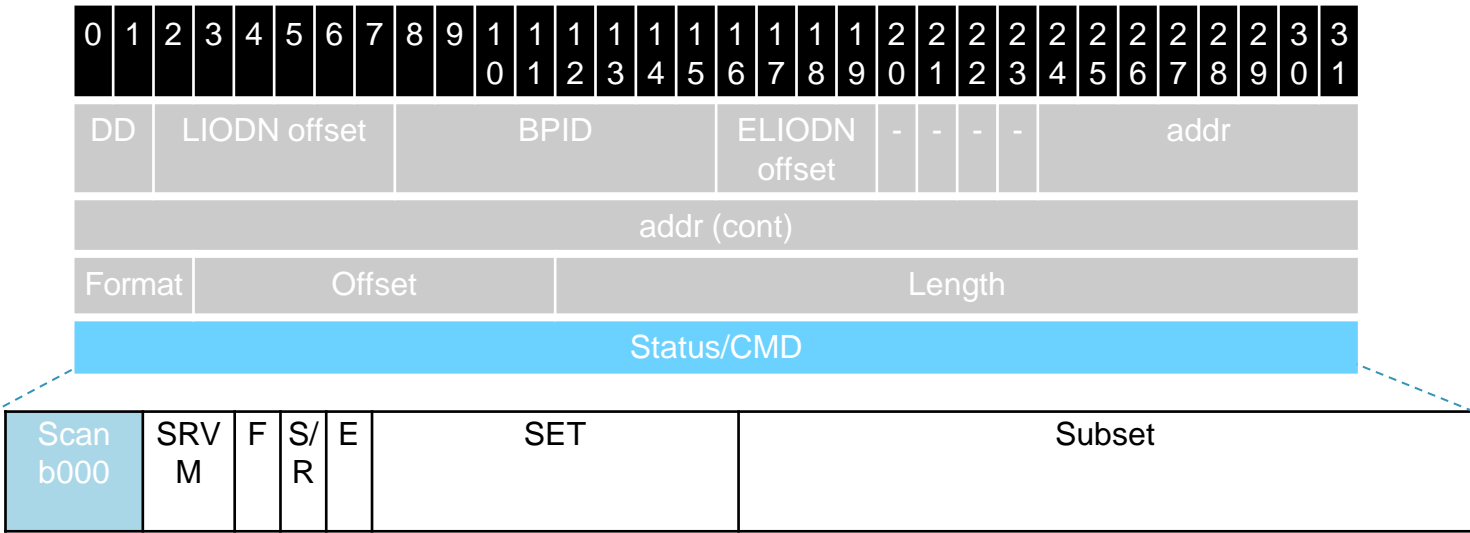
# Pattern Matching Engine (PME) 2.X Overview

- Regex support plus significant extensions:
  - Patterns can be split into 256 sets each of which can contain 16 subsets
  - 32K patterns of up to 128B length
  - 9.6 Gbps raw performance
- Combined hash/NFA technology
  - No “explosion” in number of patterns due to wildcards
  - Low system memory utilization
  - Fast pattern database compiles and incremental updates
- Matching across “work units”
  - Finds patterns in streamed data
- Pipeline of processing
  - PME offers pipeline of filtering, matching, and behavior base engine for complete pattern matching solution



# Frame Descriptor: STATUS/CMD Treatment

- PME Frame Descriptor Commands
  - b111 NOP                      NOP Command
  - b101 FCR                      Flow Context Read Command
  - b100 FCW                      Flow Context Write Command
  - b001 PMTCC                  Table Configuration Command
  - b000 **SCAN**                Scan Command



# Life of a Packet in PME

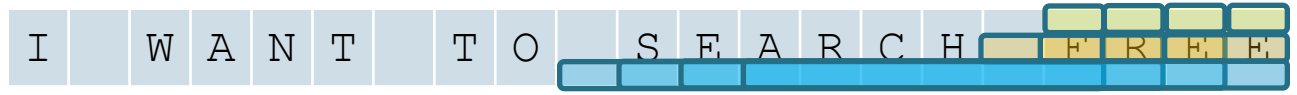


<b>FD1</b>	192.168.1.1:80	TCP	10.10.10.100:16734
	192.168.1.1:25	TCP	10.10.10.100:17784
	192.168.1.1:1863	TCP	10.10.10.100:16855

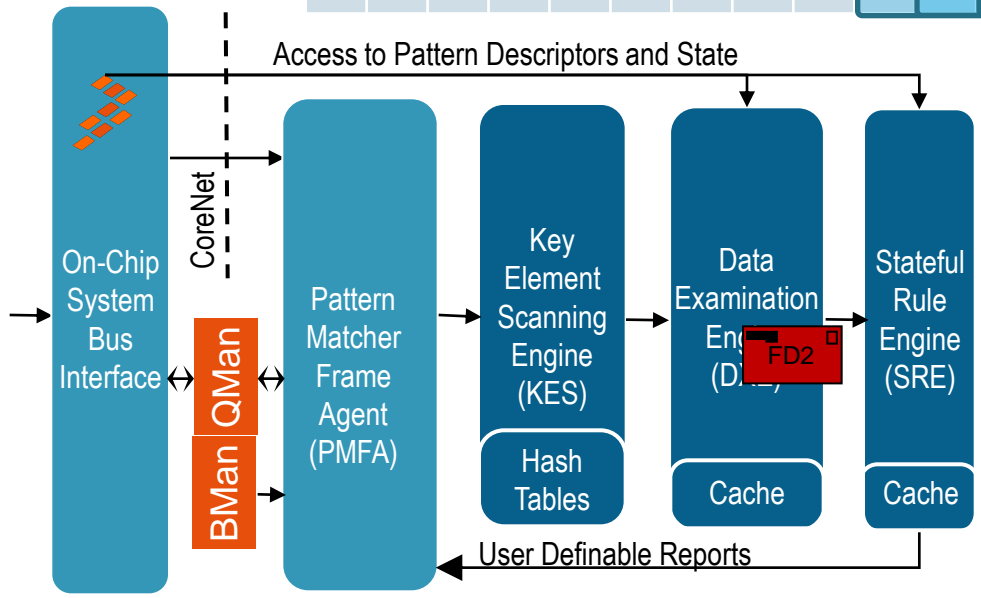


## Frame Queue: A

- flowA:FD1: 192.168.1.1:80->10.10.10.100:16734 "I want to search free"
- flowA:FD2: 192.168.1.1:80->10.10.10.100:16734 "scale FTF 2011 event schedule"



DDP  
**Patt1 /free/  
 tag=0x0001**



- Patterns
  - Patt1 /free/ tag=0x0001
  - Patt2 /freescale/ tag=0x0002
- KES
  - Compare hash value of incoming data(frames) against all patterns
- DXE
  - Retrieve the pattern with matched hash value for a final comparison.
- SRE
  - Optionally post process match result before sending the report to the CPU

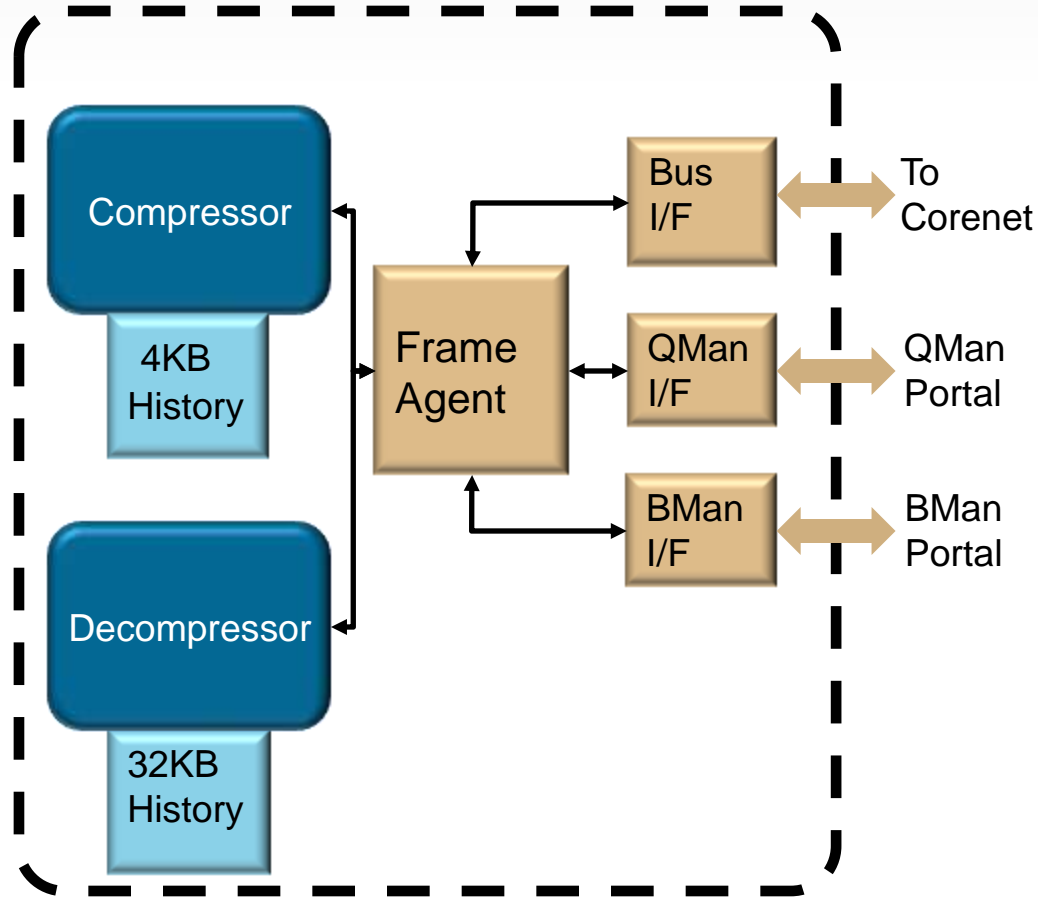


# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- SEC Enablement
- PME Enablement
- **DCE Enablement** (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- Conclusion

# DCE Logical Block Diagram

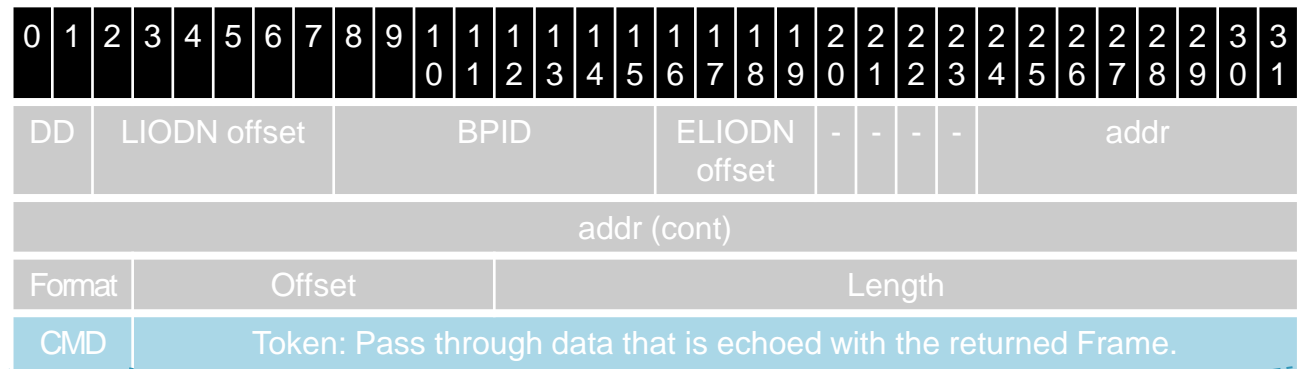
- Deflate
  - As specified as in RFC1951
- GZIP
  - As specified in RFC1952
- Zlib
  - As specified in RFC1950
  - Interoperable with the zlib 1.2.5 compression library
- Encoding
  - supports Base 64 encoding and decoding (RFC4648).
- Operate up to 600Mhz
  - 10Gbps Compress
  - 10Gbps Decompress
  - 20Gbps Aggregate



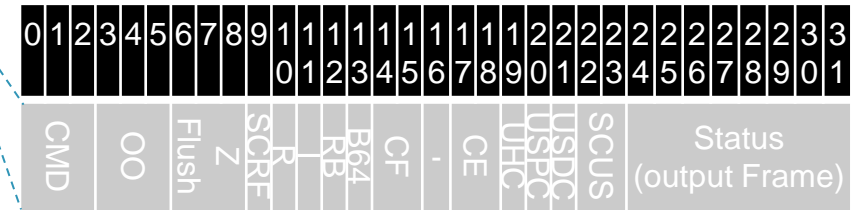


# DPAA Interaction: Frame Descriptor Status/CMD

- The Status/Command word in the dequeued FD allows software to modify the processing of individual frames while retaining the performance advantages of enqueueing to a FQ for flow based processing.
- The three most significant bits of the Command /Status field of the Frame Descriptor have the following meaning:

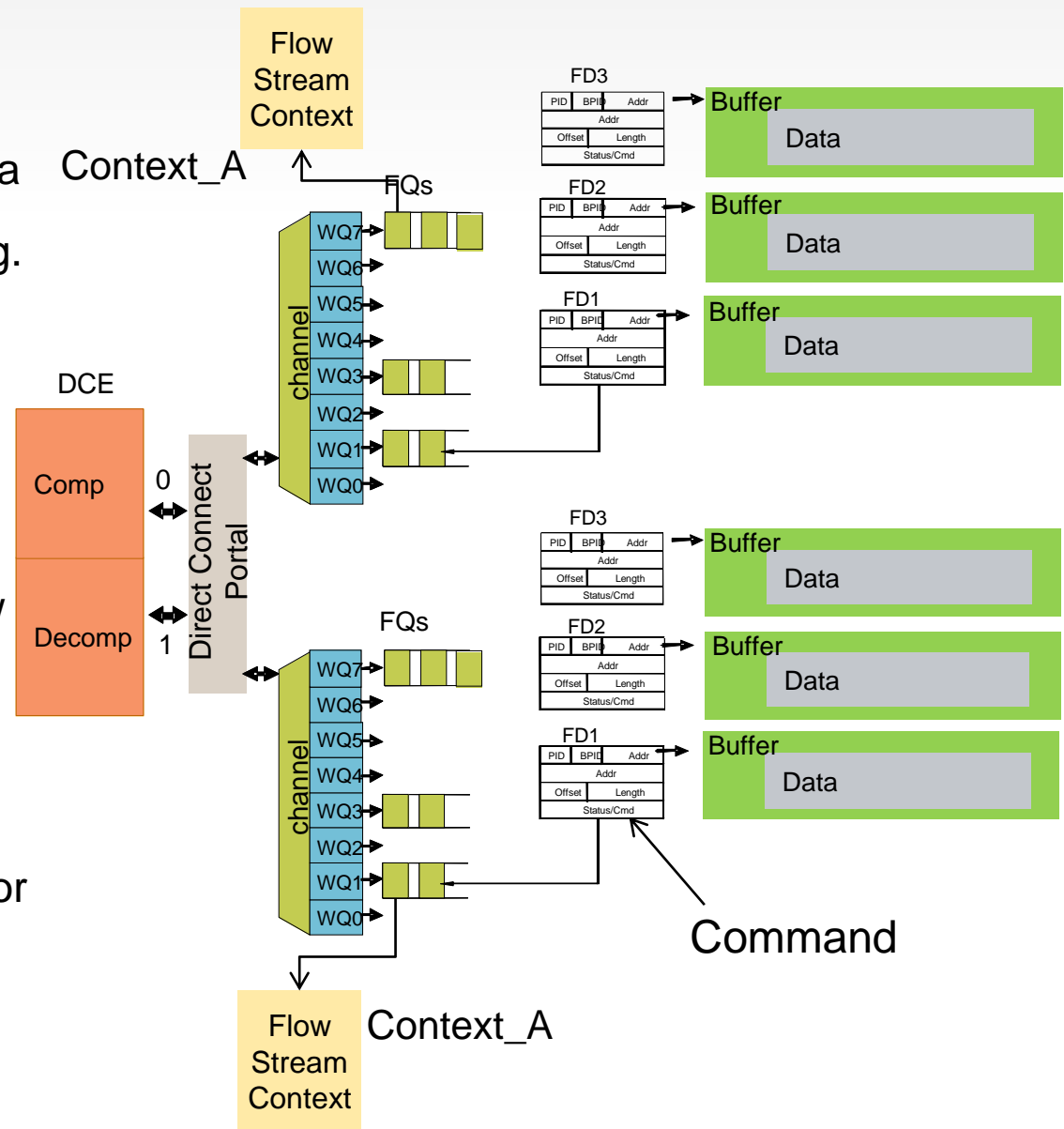


3 MSB	Description	
000	Process Command	Command Encoding
001	Reserved	
010	Reserved	
011	Reserved	
100	Context Invalidate Command	Token
101	Reserved	
110	Reserved	
111	NOP Command	Token



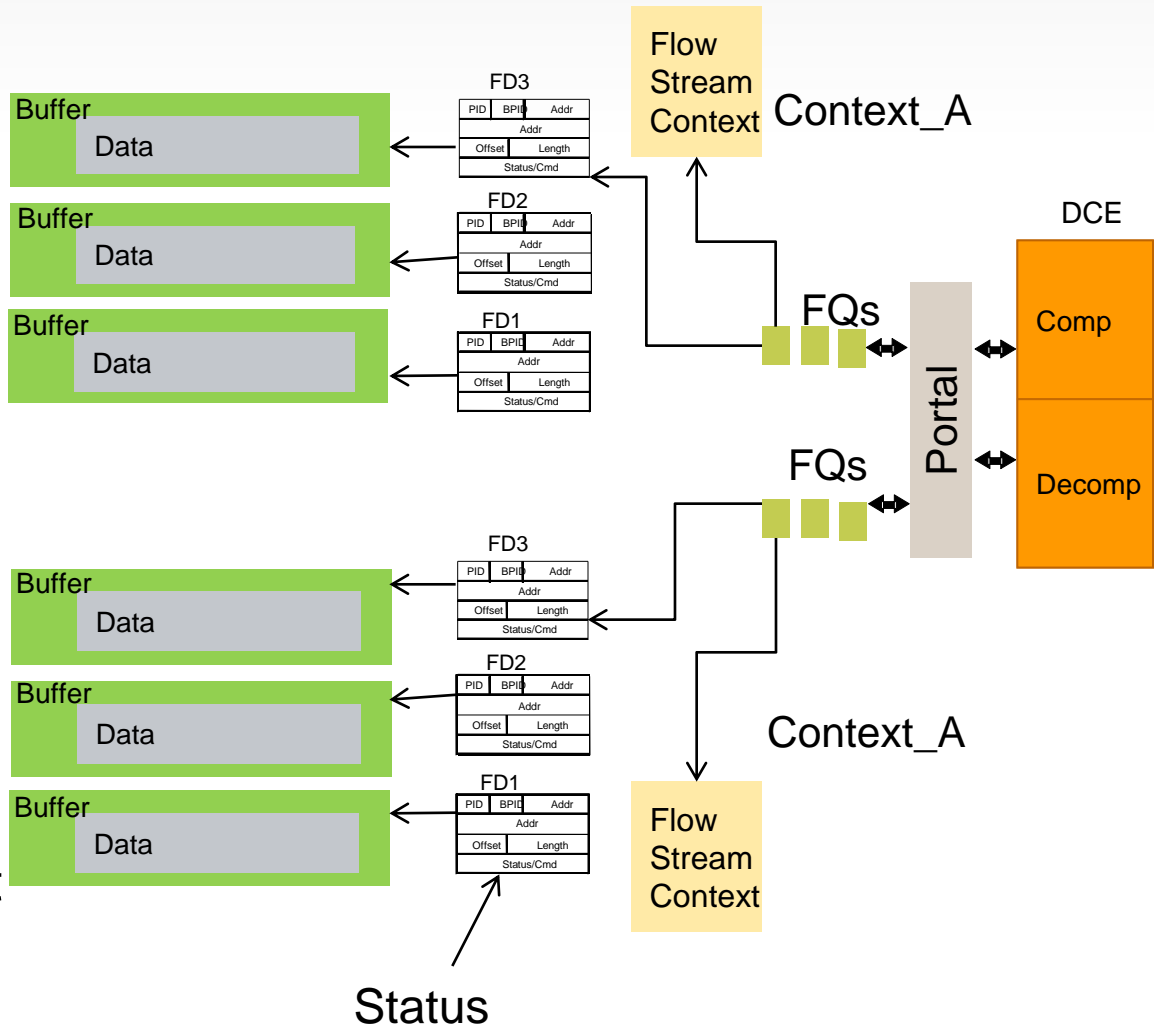
# DCE Inputs

- SW enqueues work to DCE via Frame Queues. FQs define the flow for stateful processing.
- FQ initialization creates a location for the DCE to use when storing flow stream context.
- Each work item within the flow is defined by a Frame Descriptor, which includes length, pointer, offsets, and commands.
- DCE has separate channels for compress and decompress.



# DCE Outputs

- DCE enqueues results to SW via Frame Queues as defined by FQ Context\_B field. When buffers obtained from BMan, buffer pool ID defined by Output FQ.
- Each result is defined by a Frame Descriptor, which includes a Status field.
- DCE updates flow stream context located at Context\_A as needed.





# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- **RMan Enablement**
- DCB Enablement (T series only)
- Conclusion

# RapidIO Messaging Unit Comparison

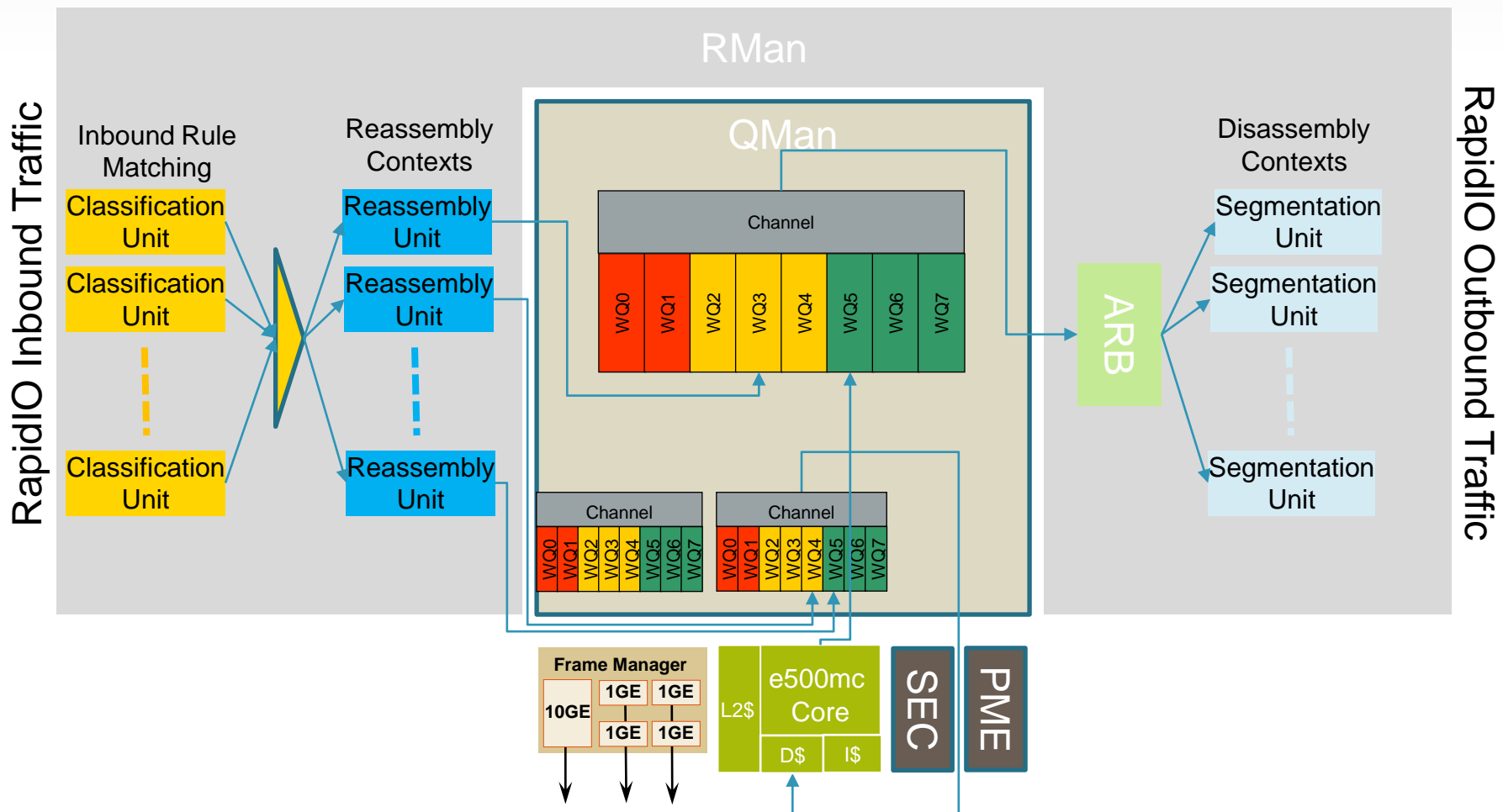
	P4080	P2040, P3, P5	
<b>Outbound</b>			
Transactions Supported	Type 10 Doorbells Type 11 Messaging	Type 5 NWRITE Type 6 SWRITE Type 8 Port-Write	Type 9 Data Streaming Type 10 Doorbells Type 11 Messaging
Queues	1 Type 10 Doorbell 2 Type 11 Messaging	Thousands of queues supporting Type 5,6,8-11	
Queue Arbitration	Round Robin	Data Path Acceleration Architecture • 3+3+1 SP+WRR	
Segmentation Resources	2 Segmentation Units	4 Segmentation Units	
Multicast Support	Type 11 256B PDU to 16 Destinations	Type 11 256B PDU to 32 Destinations	
<b>Inbound</b>			
Transactions Supported	Type 8 Port-Write Type 10 Doorbells Type 11 Messaging	Type 8 Port-Write Type 9 Data Streaming	Type 10 Doorbells Type 11 Messaging
Queues	1 Type 8 Port-Write 1 Type 10 Doorbell 2 Type 11 Messaging	1 Type 8 Port-Write 1000s Type 9-11	
Classification	<u>2 Rules (Fixed)</u> Type 11: [mbox]	64 Rules (Exact or Wildcards) or Map selected header fields to queue ID	
Simultaneous Reassembly Contexts	2 Type 11	16 Type 9, 11	
<b>Additional Features</b>			
Traffic Management	N/A	Type 9: End-to-end XON/XOFF Per-Queue Flow Control	



Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, the SafeAssure logo, SMARTMOS, TurboLink, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2012 Freescale Semiconductor, Inc.



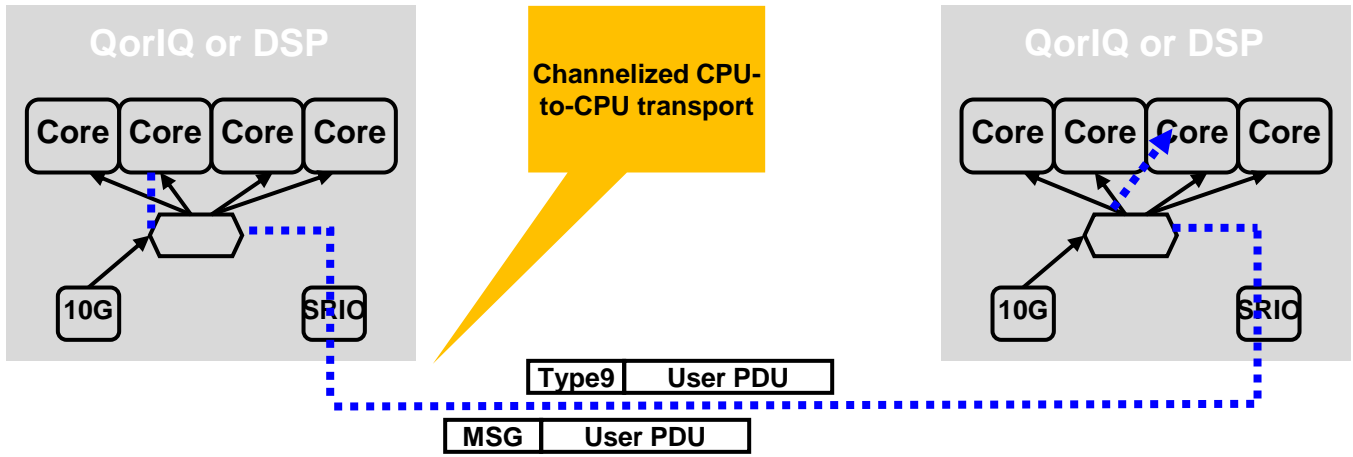
# RapidIO Message Manager (RMan)



# RMan: Greater Performance and Functionality

- Many queues allow multiple inbound/outbound queues per core
  - Hardware queue management via QorIQ Data Path Architecture (DPAA)
- Supports all messaging-style transaction types
  - Type 11 Messaging
  - Type 10 Doorbells
  - Type 9 Data Streaming
- Enables low overhead direct core-to-core communication

Device-to-Device Transport





# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- **DCB Enablement** (T series only)
- Conclusion

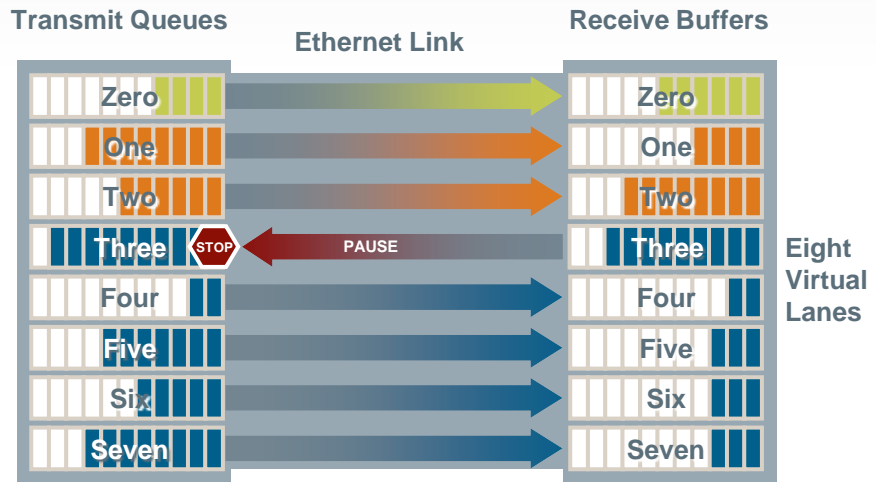


# Data Center Bridging (DCB) Overview

- QMan 1.2 (e.g. QorIQ T42xx) supports Data Center Bridging (DCB).
- DCB refers to a series of inter-related IEEE specifications collectively designed to enhance Ethernet LAN traffic prioritization and congestion management.
- DCB can be used in:
  - Between data center network nodes:
    - LAN/network traffic
    - Storage Area Network (SAN) (e.g. Fiber Channel (loss sensitive )) and,
    - IPC traffic (e.g. Infiniband (low latency))
- The DPAA is compliant with the following DCB specifications (traffic management related) :
  - IEEE Std. 802.1Qbb: Priority-based flow control (PFC)
    - To avoid frame loss, a PFC Pause frames can be sent autonomously by HW.
  - IEEE Std. 802.1Qaz: Enhanced transmission selection (ETS)
    - Support Weighted bandwidth fairness.
  - IEEE 802.1Qau: Quantized Congestion Notification (QCN)
    - end-to end congestion control mechanism.

# Supports for Priority-based Flow Control(802.1Qbb)

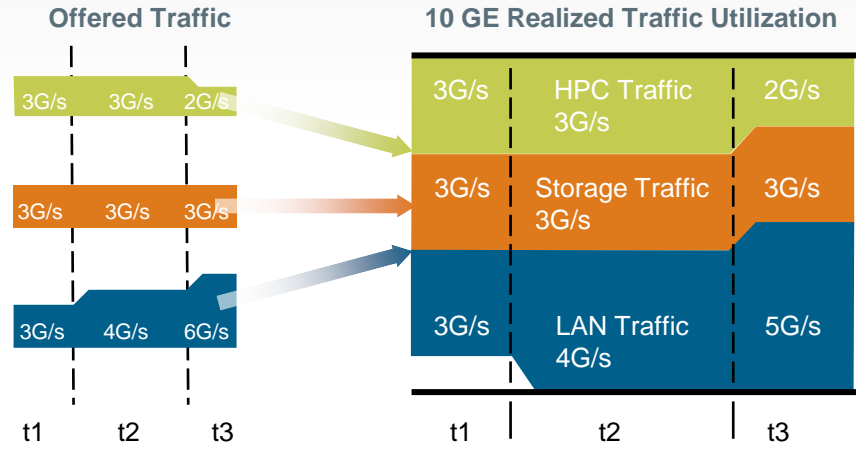
- Enables lossless behavior for each class of service
- PAUSE sent per virtual lane when buffers limit exceeded
  - FQ congestion groups state (on/off) from QMan;
    - Priority vector (8 bits) is assigned to each FQ congestion group
    - FQ congestion group(s) are assigned to each port;
    - Upon receipt of a congestion group state “on” message, for each Rx port associated with this congestion group, a PFC Pause frame is transmitted with priority level(s) configured for that group.
  - Buffer pool depletion
    - Priority level configured on per port (shared by all buffer pools used on that port)
  - Near FMan Rx FIFO full
    - There is a single Rx FIFO per port for all priorities, the PFC Pause frame is sent on all priorities.



- PFC Pause frame reception
  - QMan provides the ability to flow control 8 different traffic classes; in CEETM each of the 16 class queues within a class queue channel can be mapped to one of the 8 traffic classes & this mapping applies to all channels assigned to the link

# Supports for Enhanced Tx Selection (802.1Qaz)

- Enables Intelligent sharing of bandwidth between traffic classes control of bandwidth
- Best supported through QMan CEETM
  - 8 independent classes and 8 grouped classes
  - Strict priority scheduling of the 8 independent classes
  - Weighted bandwidth fairness within 8 grouped classes
  - Priority of the class group can be independently configured to be immediately below any of the independent classes
- Meets performance requirement for ETS: bandwidth granularity of 1% and +/-10% accuracy



- Supports 32 channels available for allocation across a single FMan;
  - e.g. for two 10G links, could allocate 16 channels (virtual links) per link
  - Supports weighted bandwidth fairness amongst channels
  - Shaping is supporting on per channel basis

# Supports for Quantized Congestion Notification

- **Reaction Point**
  - Reaction point state machine and congestion notification message (CNM) receipt and processing must be implemented in software
  - Can make use of CEETM channel shaping function to implement the rate limiter function;
    - One class queue per channel; limited scalability especially if virtualization is to be supported (total of 32 channels per FMan instance)
    - Shaped queue cannot be serviced using strict priority
- **Congestion Point**
  - No hardware support



# Agenda

- DPAA
- BMan Enablement
- FMan Enablement
- QMan Enablement
- SEC Enablement
- PME Enablement
- DCE Enablement (T series only)
- RMan Enablement
- DCB Enablement (T series only)
- **Conclusion**





# Conclusion

- QorIQ is all about Accelerators and Integration.
- The Data Path Acceleration Architecture **Accelerators** included:
  - Queue Manager
  - Buffer Manager
  - Frame Manager
  - Hardware accelerators such as SEC, PME, DCE, and RMan
  - Power Architecture Cores
- Seamless **Integration** of these components address multicore requirements:
  - Load spreading
  - Packet ordering
  - Device virtualization
  - Inter-core communication
  - HW buffer management

[Facebook.com/Freescale](https://www.facebook.com/Freescale)

Tag yourself in photos and upload your own!



Twisting?

Please use hashtag #FTF2012



**Session materials will be posted @ [www.freescale.com/FTF](http://www.freescale.com/FTF)**

Look for announcements in the FTF Group on LinkedIn or follow Freescale on Twitter



