

June, 2010

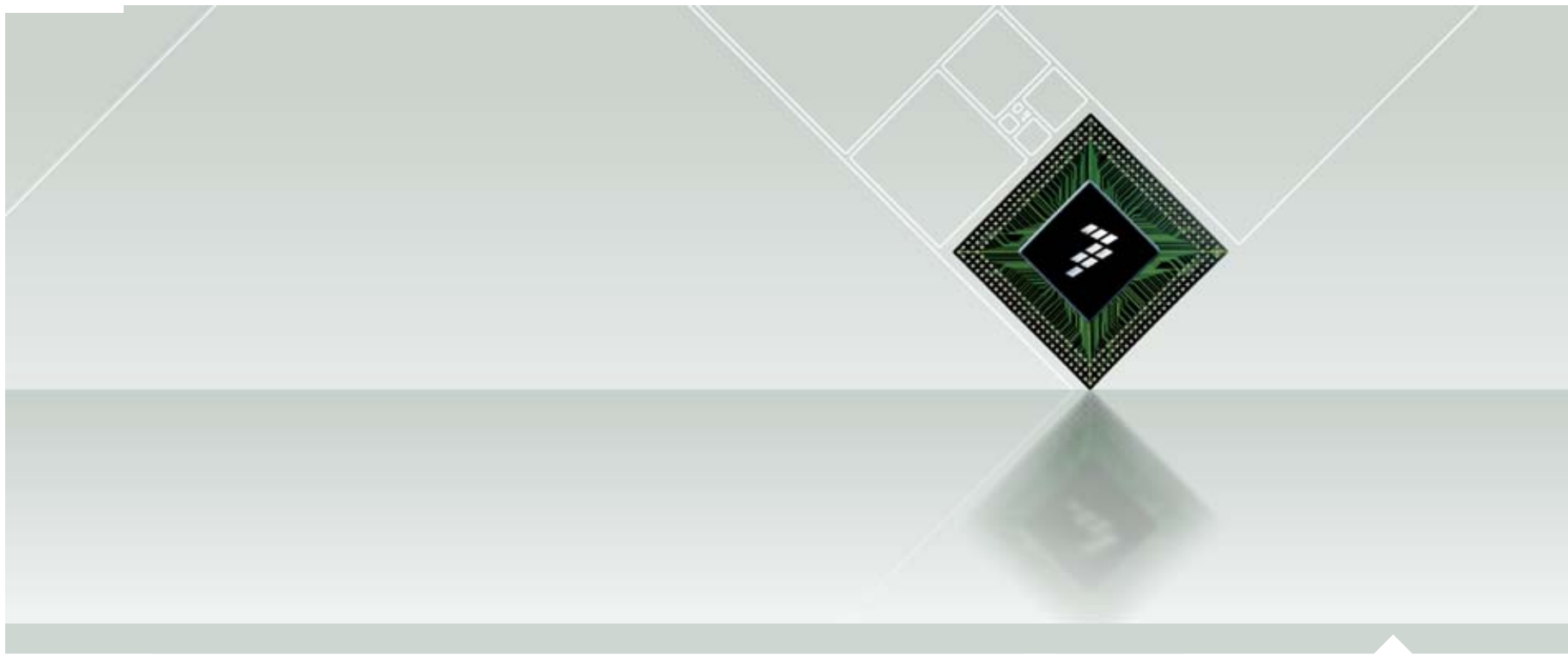
Leveraging the QorIQ Data Path Acceleration Architecture (DPAA) for Wireless Applications

FTF-NET-F0704



Stephen Cole – Senior Systems Architect
Jonas Svennebring – Senior Software FAE

- ▶ Challenges in Wireless Industry
- ▶ QorIQ Application Mapping
- ▶ Data Path Acceleration – quick recap.
- ▶ Packet I/O
- ▶ PME – traffic monitoring and management
- ▶ SEC – Ciphering and packet formatting
- ▶ Use-case: Voice packet processing
- ▶ Use-case: HSPA, mobile broadband



Challenges in Wireless Industry

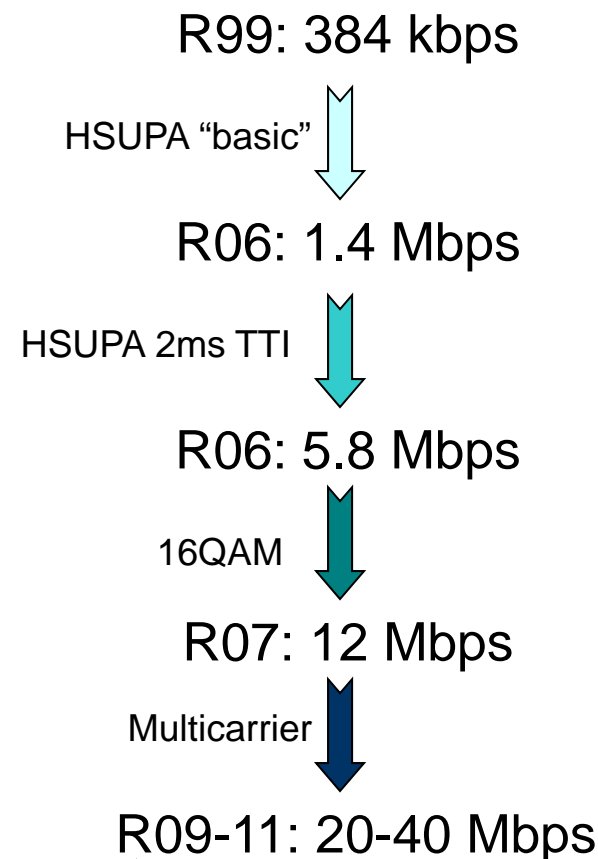
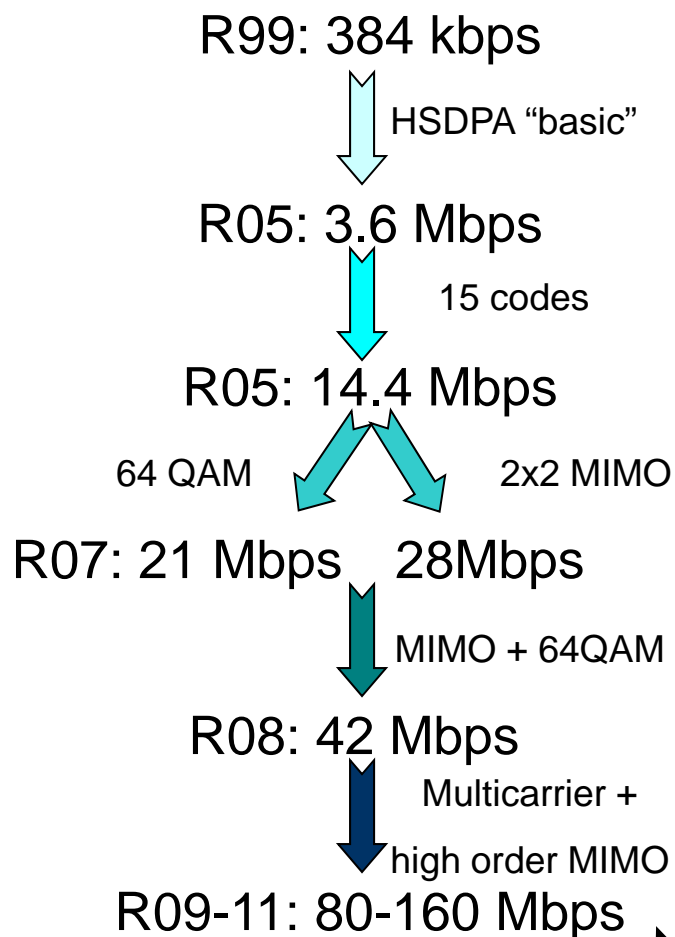
- ▶ Mobile broadband has taken off
 - High bandwidths
 - High user peak rates
 - Low network latency

- ▶ Smart Phones (iphone etc) require intensive signaling, i.e. control processing.

- ▶ WCDMA follows the LTE track in development
 - Competitive performance with HSPA+
 - Support for flat architecture

- ▶ Home base-station, will 2010/11 be the break through?
- ▶ All IP? Investments in ATM needs smooth migration, i.e. IP over ATM.
- ▶ Content Caches in field to offload main pipe to Internet?

Challenge: Mobile Broadband

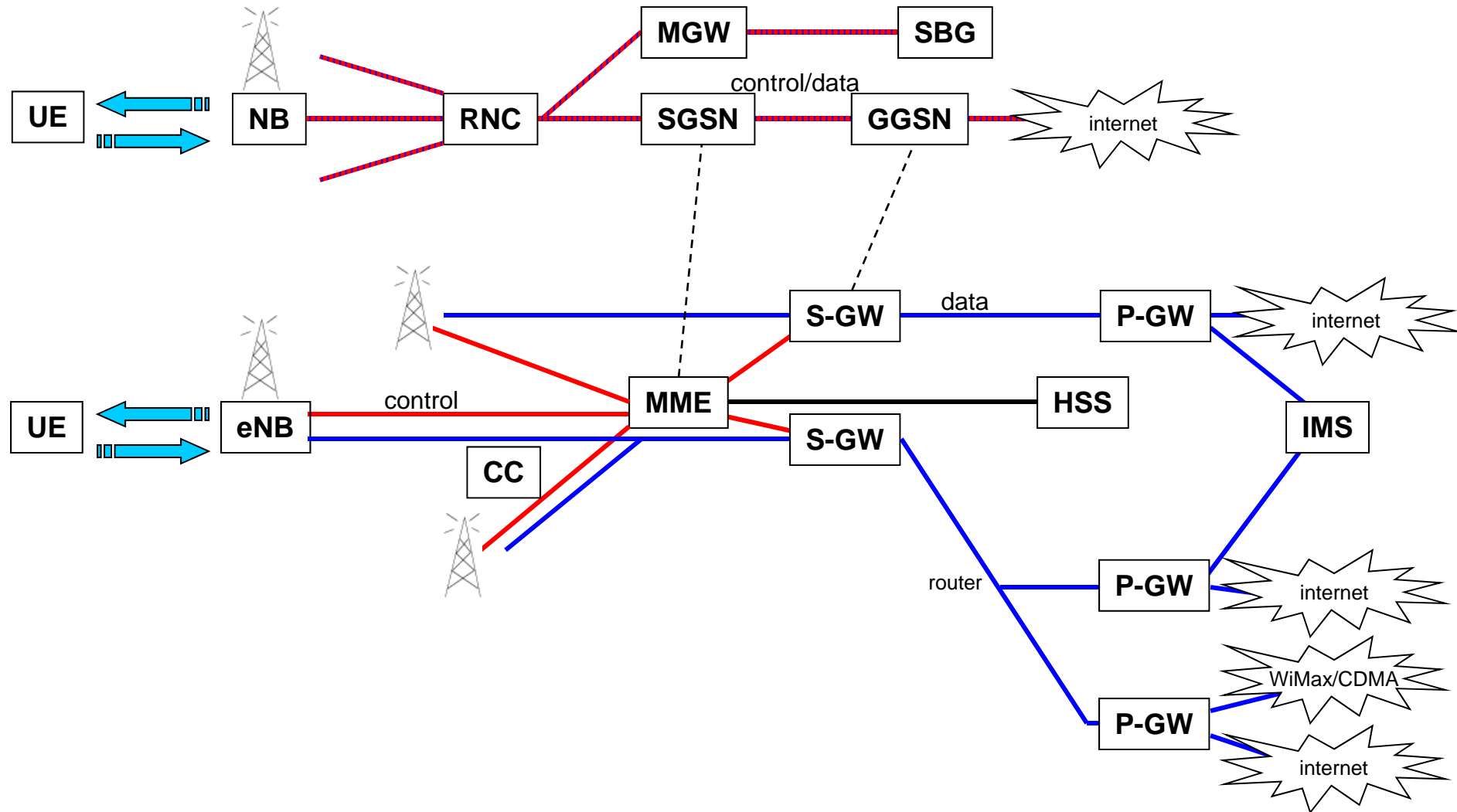


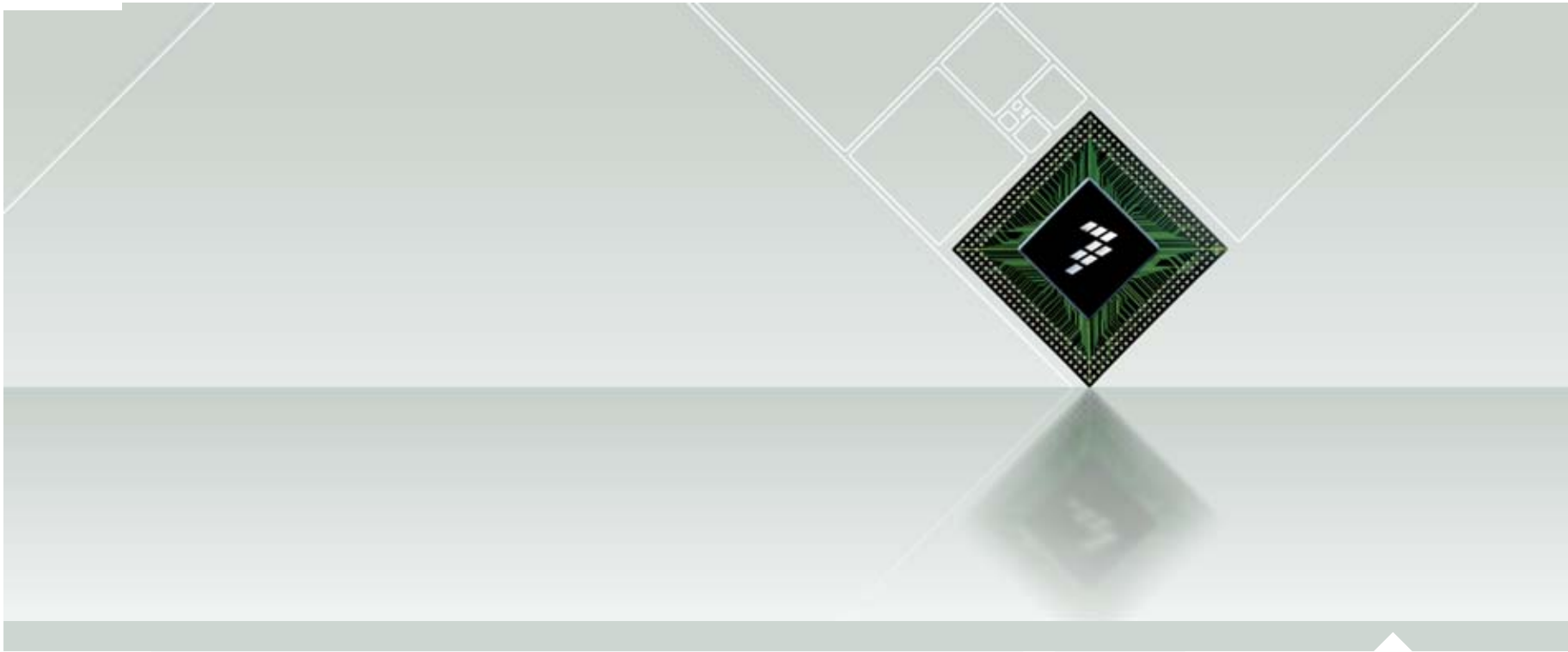
Very high peak rates!
Huge total bandwidth!

R12-- : 400+ Mbps

How can this be handled within a device?

WCDMA and LTE Core Networks





QorIQ Application Mapping

► Our 15+ year Heritage:

► 3rd Generation Data Path

- Gen-1: CPM MPC8260
- Gen-2: QUICC Engine™ MPC8360
- Gen-3: DPAA QorIQ P4080

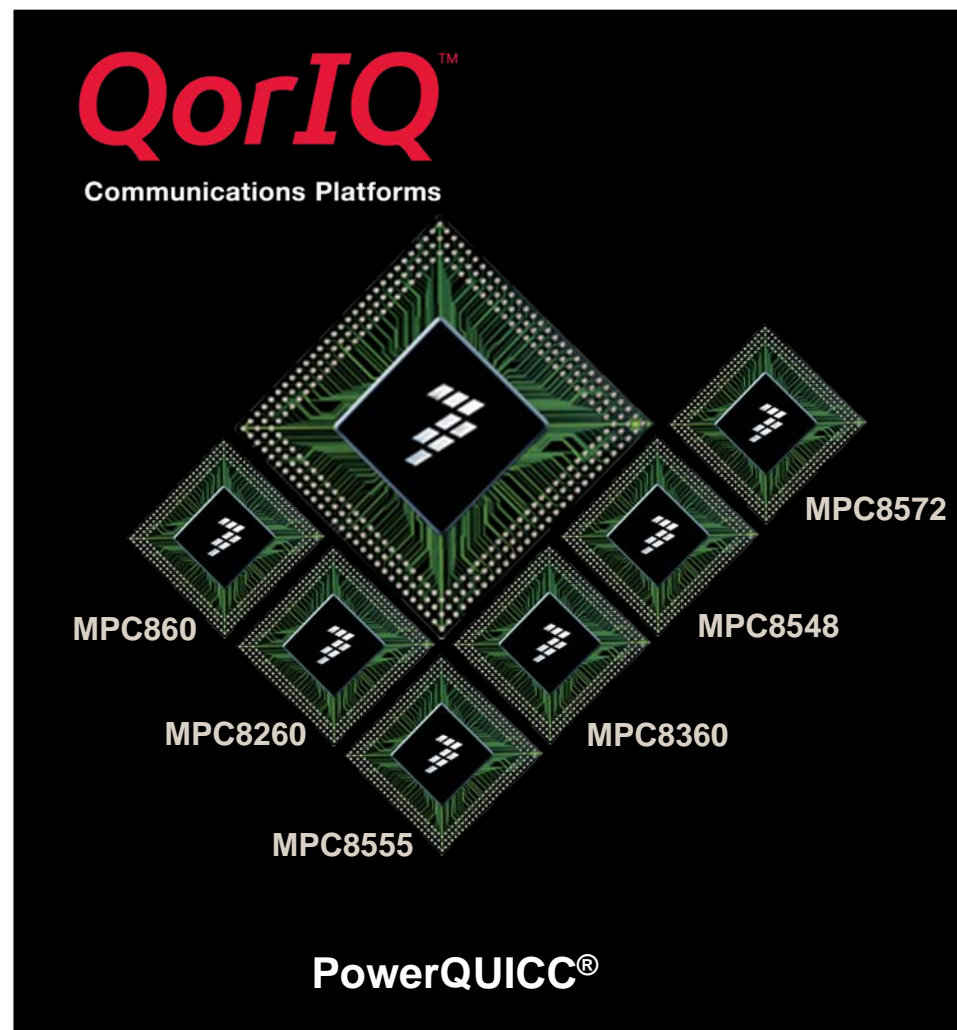
► Accelerating Connectivity

- eTSEC
- SEC 4.0
- PME 2.0
- PCIe, Serial RapidIO®, XAUI

► Power Architecture™ ISA

- e500 PowerQUICC® III
- e500 QorIQ P1, P2 platforms
- e500mc QorIQ P3, P4 platforms

► Long life cycle management for industry



QorIQ™ Platform Levels matching Wireless Applications

45nm PLATFORMS / PRODUCTS

QorIQ P5

Highest-performing embedded processors
PRODUCTS: P5010 / P5020



MME



SGSN

QorIQ P4

Octal and Quad core family
PRODUCTS: P4080
P4040



IS Gateway



Radio Network
Control



Access
Gateway

QorIQ P3

Quad core family – entry level MC
PRODUCTS: P3040 / P3041



Content Caches



Basestation

QorIQ P2

High performance Dual and Single Core
PRODUCTS:
P2020 and P2010



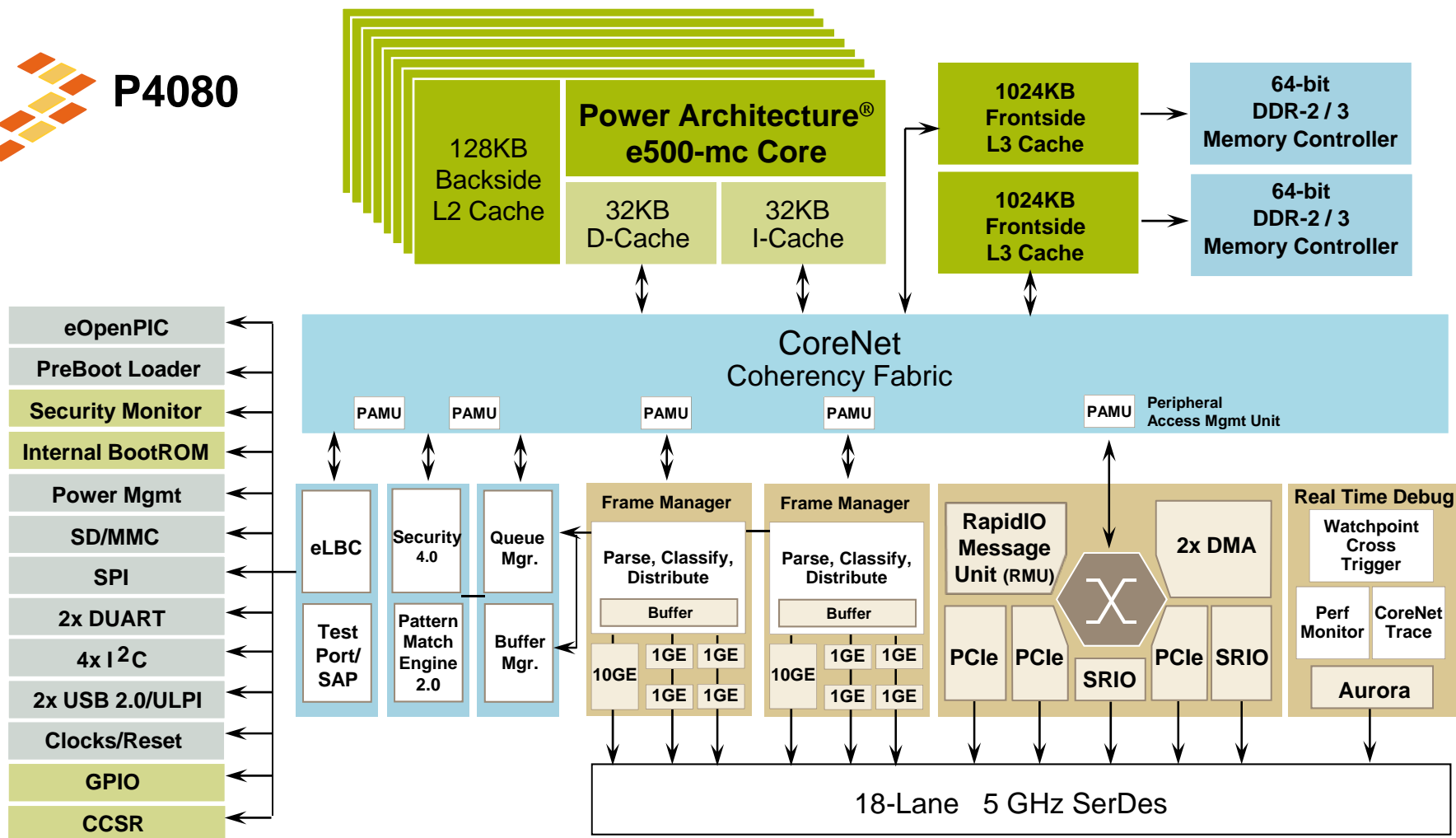
Network Interface
And Line cards

QorIQ P1

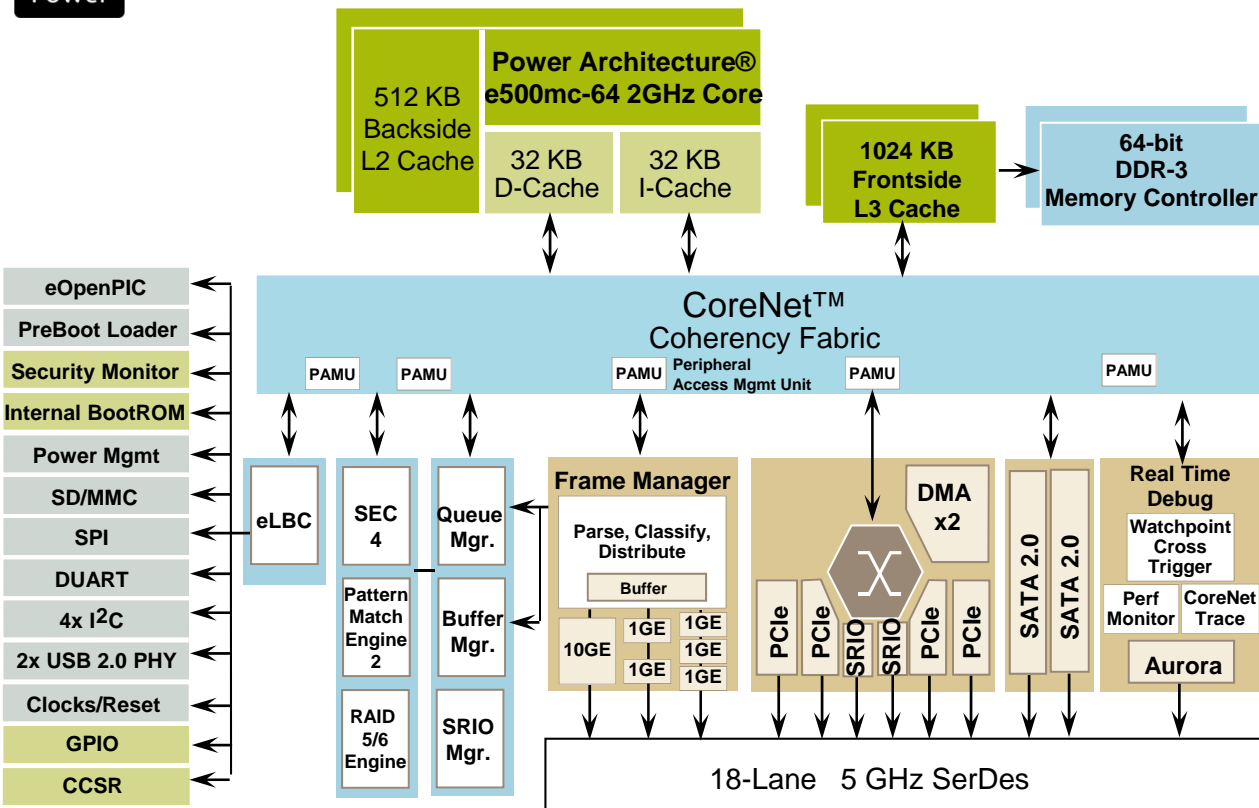
Low Power Dual and Single Core
PRODUCTS: P1020 , P1011, P1010

QorIQ P4 Series P4080 Block Diagram

P4080



QorIQ P5 Series P5020 Block Diagram

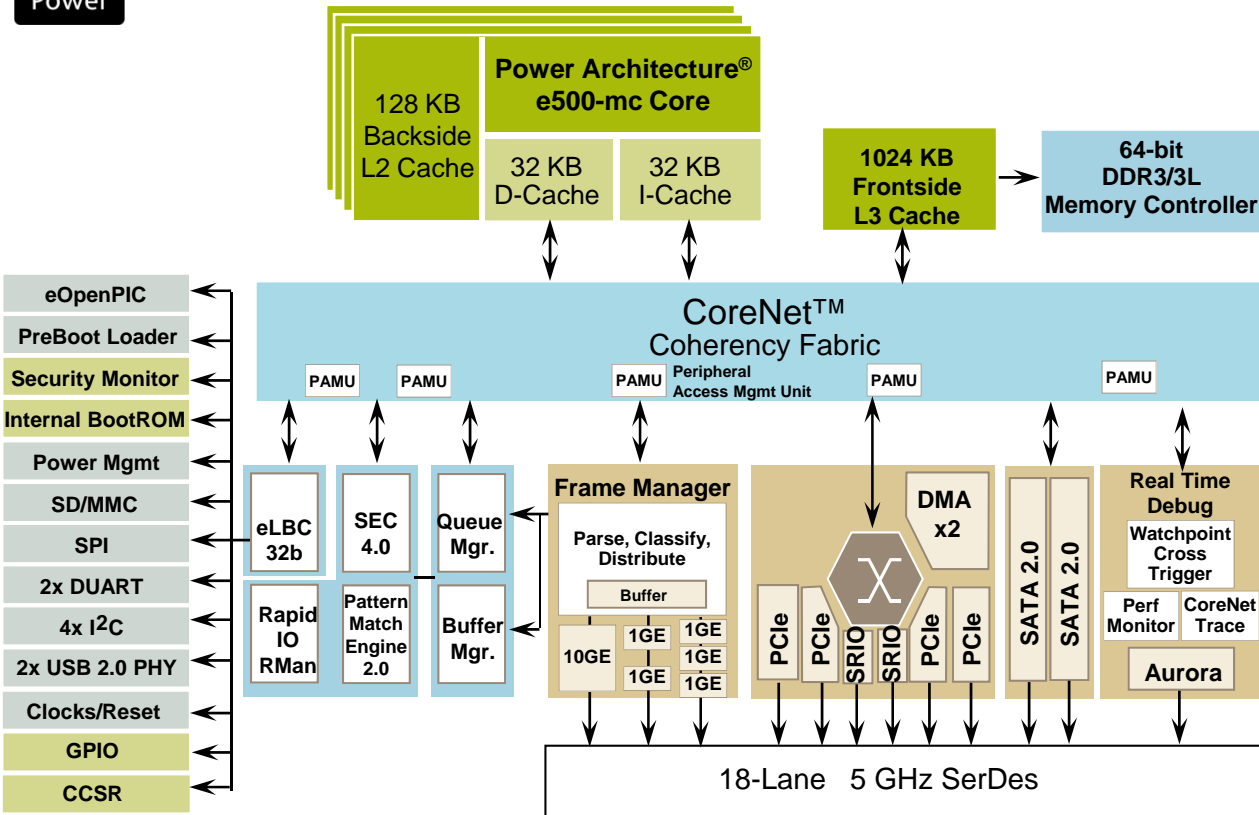


- ▶ **2x e500mc-64 core, built on Power Architecture® technology**
 - 2x 64-bit cores (up to 2+ GHz) with 512 KB backside L2 cache
 - Dual 1MB Shared L3 cache w/ECC
- ▶ **Memory Controller**
 - Dual 32/64-bit DDR3/3L w/ECC up to 1.3 GHz
- ▶ **Ethernet**
 - 5 x 10/100/1000 Ethernet controllers
 - 1 x 10GE controller (XAUI)
- ▶ **High Speed Interconnect**
 - 4 PCI Express® 2.0 controllers
 - 2 Serial RapidIO® 1.3 + 2.0 controllers
 - 2 SATA 3Gb/s
 - 2 USB 2.0 with PHY
- ▶ **CoreNet Switch Fabric**
- ▶ **Trusted Architecture**
- ▶ **Data Path Acceleration Architecture**
 - Security Engine (SEC)
 - Pattern Matching Engine (PME)
 - RAID 5/6 Engine
 - Enhanced RapidIO Messaging (Rman)

- ▶ **Pin Compatible to P4080, P4040 & P3041**
- ▶ **45nm SOI Process**



QorIQ P3 Series P3041 Block Diagram



Quad e500mc Power Architecture®

- ▶ 4 cores (up to 1.5GHz) with 128KB backside L2 cache
- ▶ 1MB Shared L3 Cache w/ECC

Memory Controller

- ▶ 32/64bit DDR3/3L w/ECC up to 1.3 GHz

Ethernet

- ▶ 5 x 10/100/1000 Ethernet Controllers
- ▶ 1 x 10GE Controllers

High Speed Interconnect

- ▶ 4 PCIe 2.0 Controllers
- ▶ 2 SRIO 1.3 + 2.0 Controllers
- ▶ 2 SATA 2.0
- ▶ 2 USB 2.0 w/PHY

CoreNet Switch Fabric

Trusted Architecture

Datapath Acceleration Architecture

- ▶ Security Engine (SEC)
- ▶ Pattern Matching Engine (PME)
- ▶ Enhanced RapidIO Messaging (Rman)

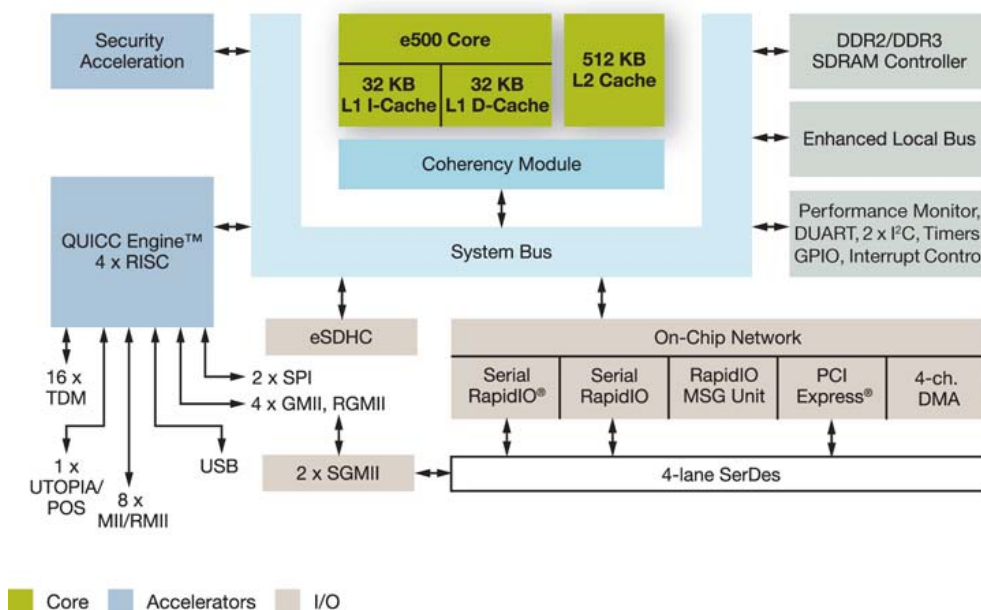
- ▶ Pin Compatible to P4080, P4040, P5020 & P5010
- ▶ 45nm SOI Process



MPC8569 PowerQUICC III

Bridging the Gap to the All-IP Network

MPC8569E Block Diagram



► e500 PowerPC from 800MHz to 1.33 GHz

- 512KB L2 Cache w/ ECC
- 36bit physical addressing
- Double Precision Floating Point

► System Interfaces

- 64b or 2x32-bit DDR2/3 w/ ECC
- 800 Mbps/pin data rate
- 16-bit Local Bus for SRAM/Flash
- Timers, DUART, 2xI²C, GPIO, SPI
- USB 2.0 Full Speed

► High Speed Serial Interfaces

- Dual SGMII
- Dual x1 Serial RapidIO or PCI-Ex

► QUICC Engine

- 4 RISCs up to 667 MHz
- Maximum of 8 Ethernet interfaces, one per UCC:
 - 4 x Gigabit Eth (up to 2 w/SGMII)
 - Up to 8 x 10/100 Ethernet
- Multi-PHY UTOPIA/POS-PHY L2 (16-bit)
- IEEE1588 Support v2
- 16 x T1/E1 (512 x 64kbps channels)

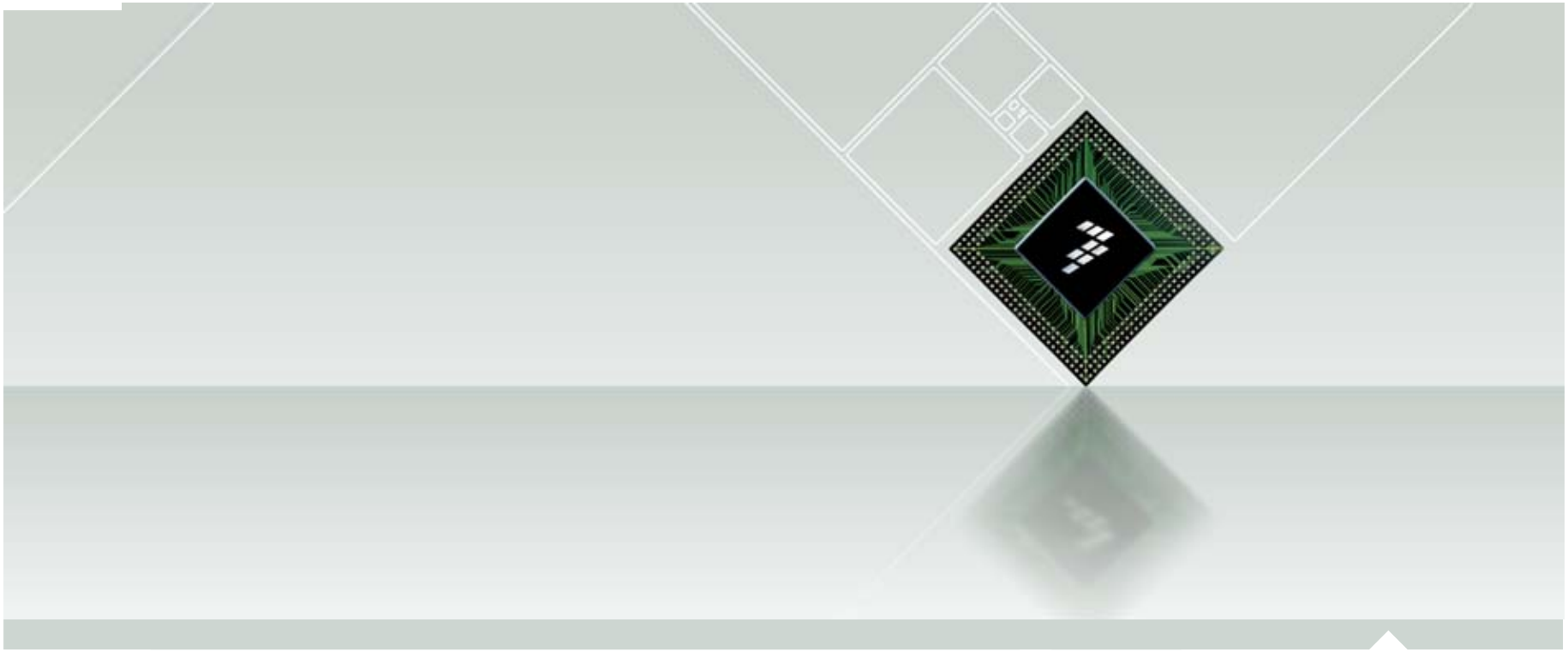
► Security Engine (SEC3.0)

- ARC4, 3DES, AES, RSA/ECC, RNG, XOR, Single pass SSL/TLS, Kasumi, SNOW

► Four-channel DMA

► 45nm SOI process technology

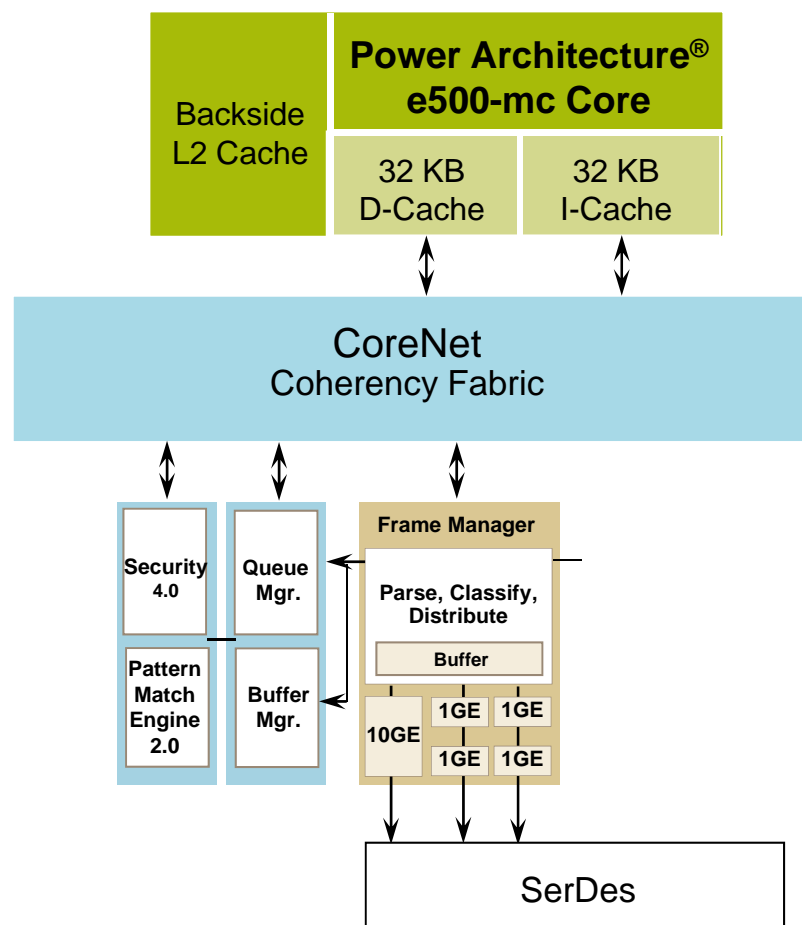
► Target <7W Power (@ 800MHz e500)

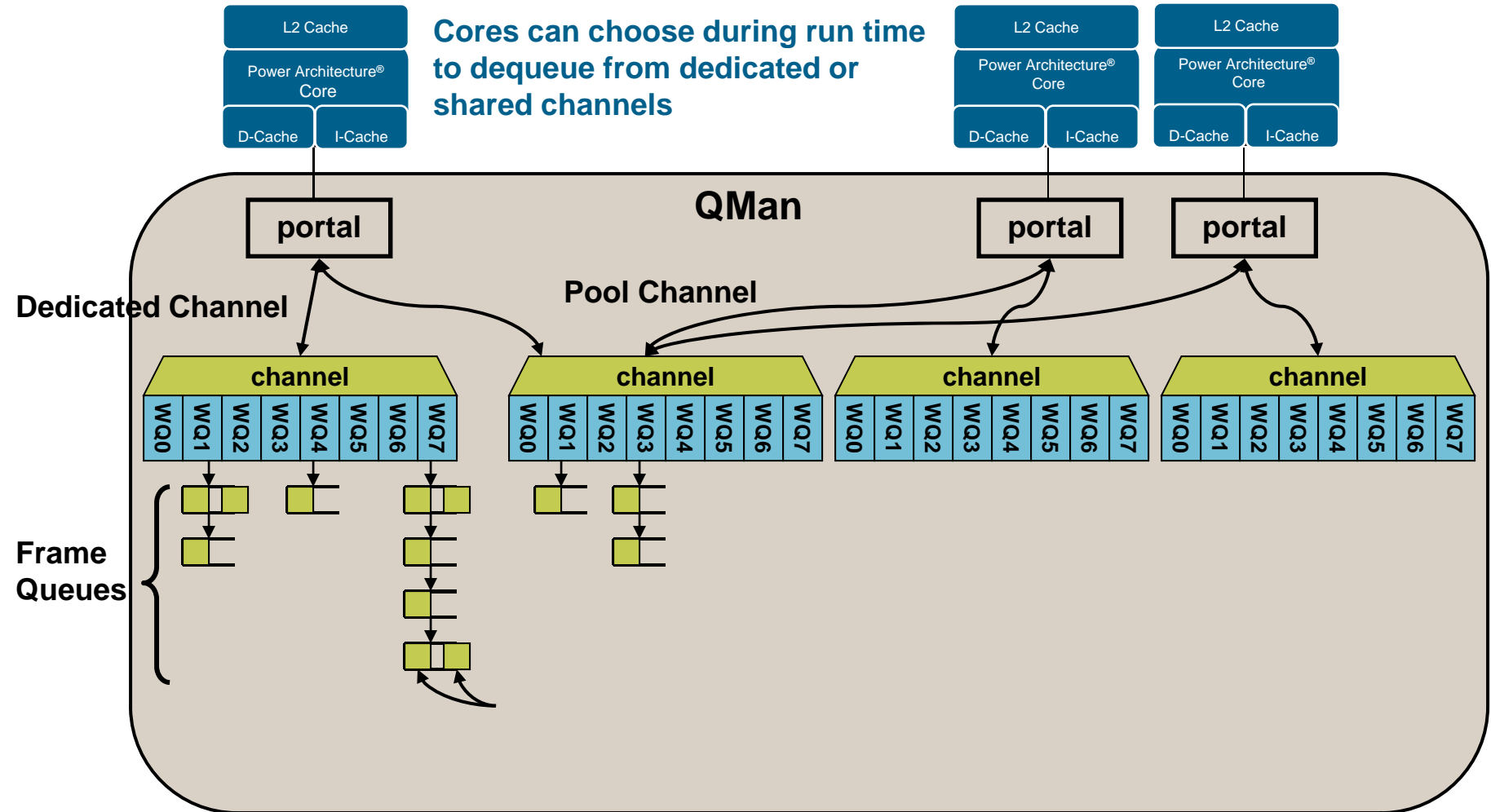


Data Path Acceleration: Quick Recap

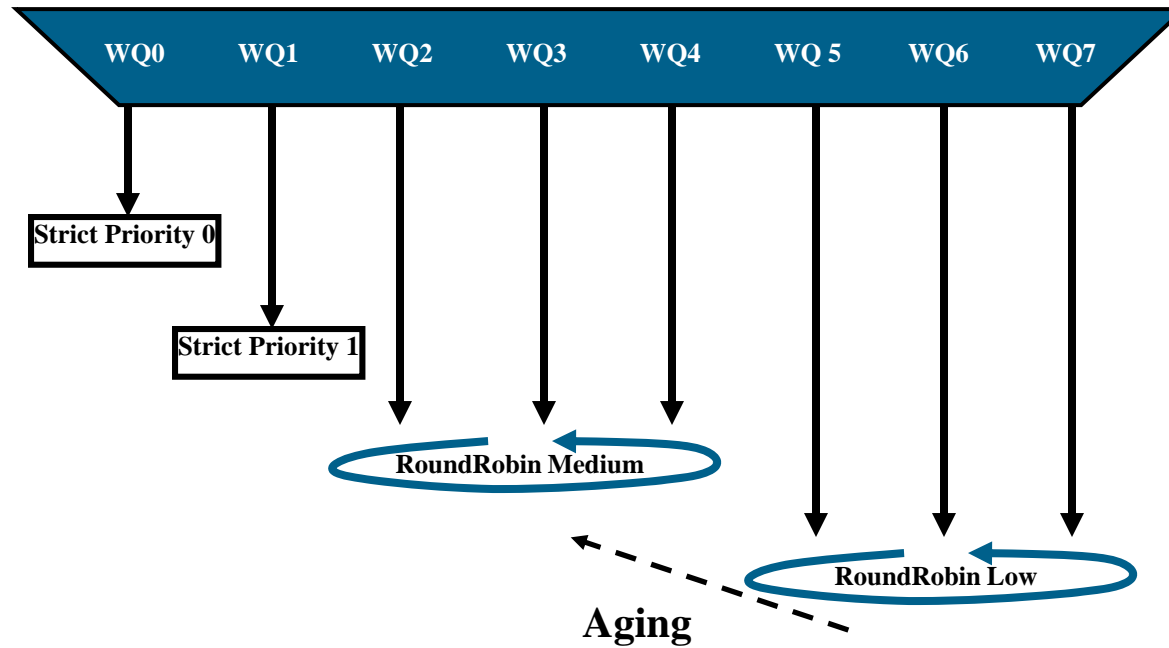
DPAA – Data Path Acceleration Architecture

- ▶ **Frame Manager:**
- ▶ Parse header information
- ▶ Classify to what destination the packet is targeted
ex. MAC, IP, UDP destination
- ▶ Distribute packets to allow load balancing between cores
- ▶ Police flows to avoid DoS attacks

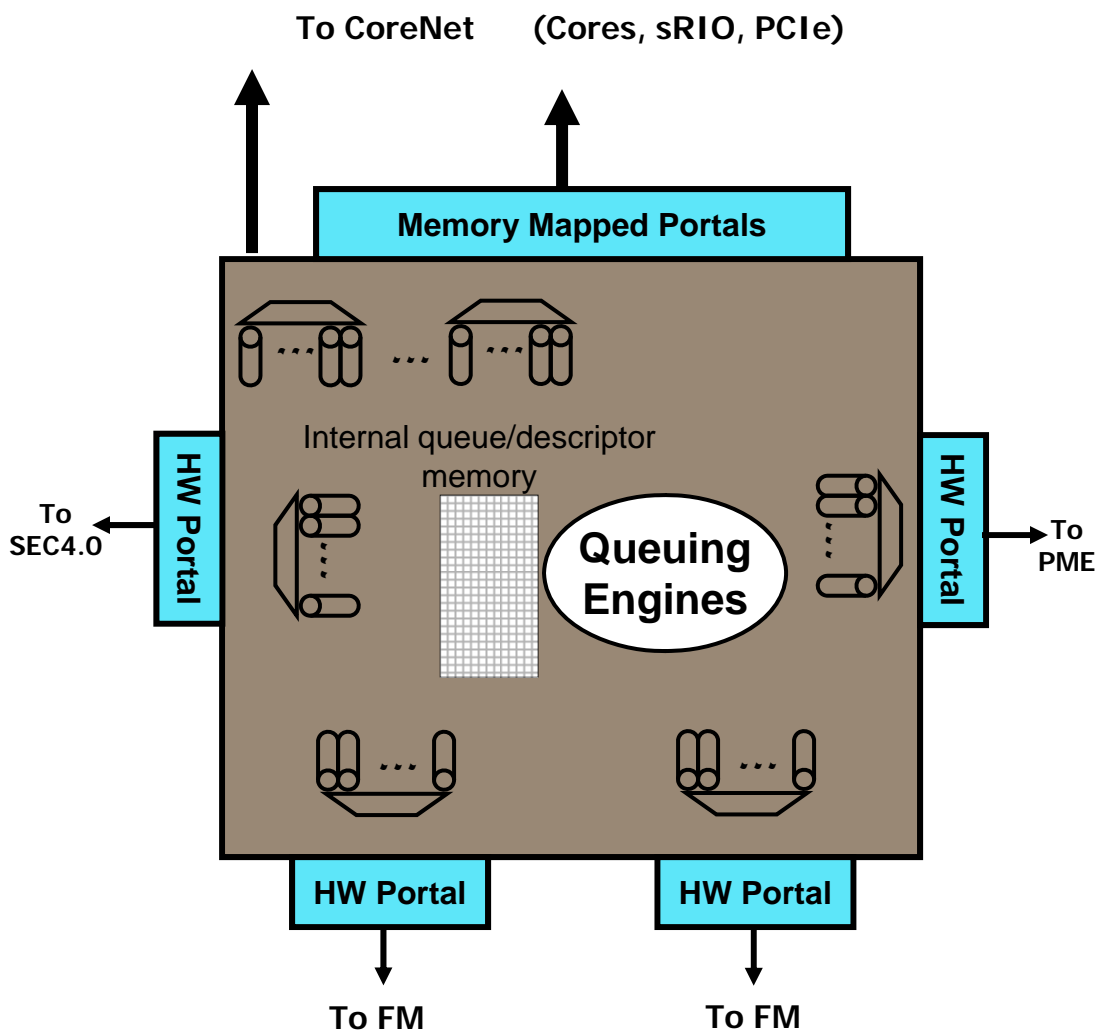




QoS supported through Channel Scheduler



Queue Manager (QMan)



- ▶ QMan provides a set of building blocks
 - Frame queues (FQ) which are enqueued onto
 - ...Work queues (WQ) which are organized into
 - ...Channels with prioritized class scheduling between WQs
 - ... which can be used to build connections between blocks (cores, network I/Fs, HW accelerators)
- ▶ Frame queues are “logical” queues
 - Actual data is stored in memory buffers, QMan queues “frame descriptors”
- ▶ QMan also supports active queue management:
 - Tail drop on FQs
 - WRED on groups of FQs
 - Tail drop on groups of FQs
- ▶ Channels can be shared between consumers which facilitates load spreading
- ▶ Channels may also be dedicated to a single consumer

Multi-buffer (Scatter/gather) Frame

Simple Frame

Frame Descriptor

D	PID
BPID	
Address	
000	Offset
Length	
Status/Cmd	

Buffer

Frame Descriptor

D	PID
BPID	
Address	
100	Offset
Length	
Status/Cmd	

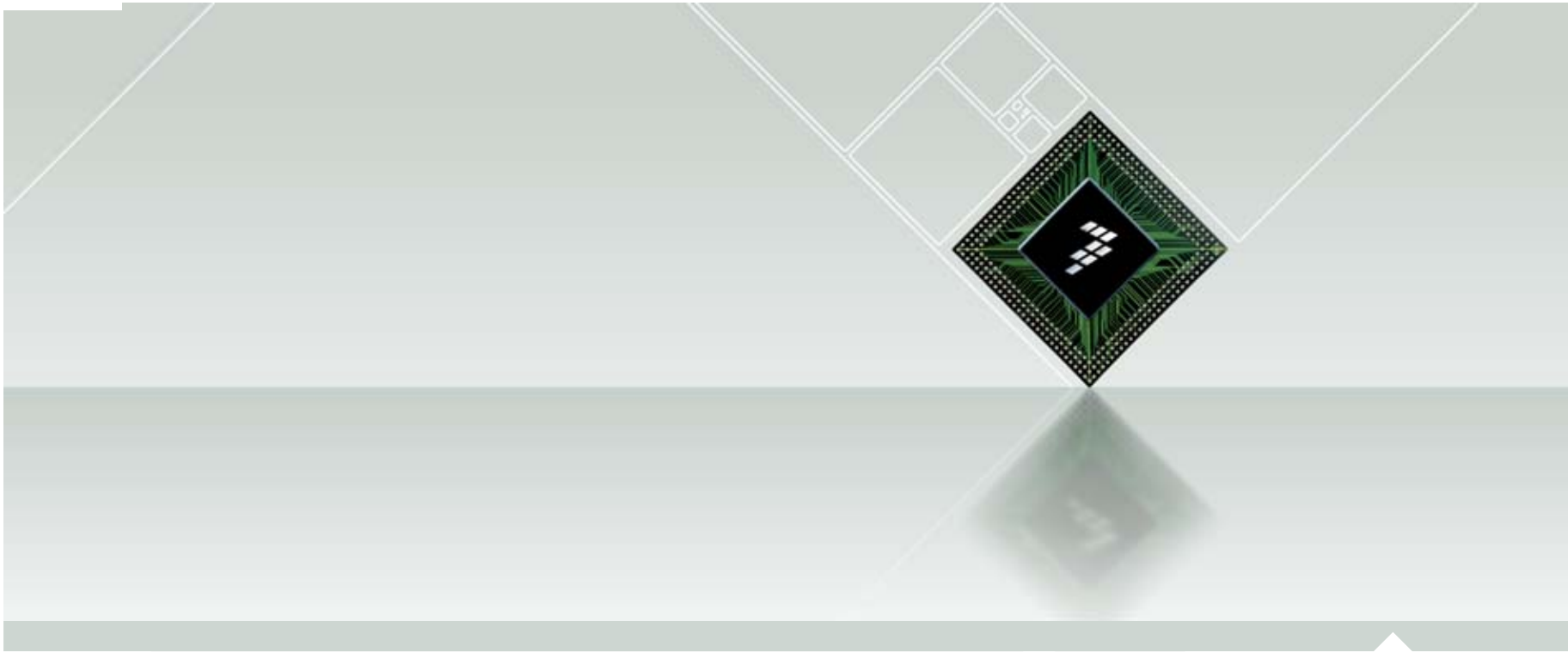
S/G List

Address	
00	Length
BPID	
Offset	
Address	
00	Length
BPID	
Offset	
...	
Address	
01	Length
BPID	
Offset	

Data

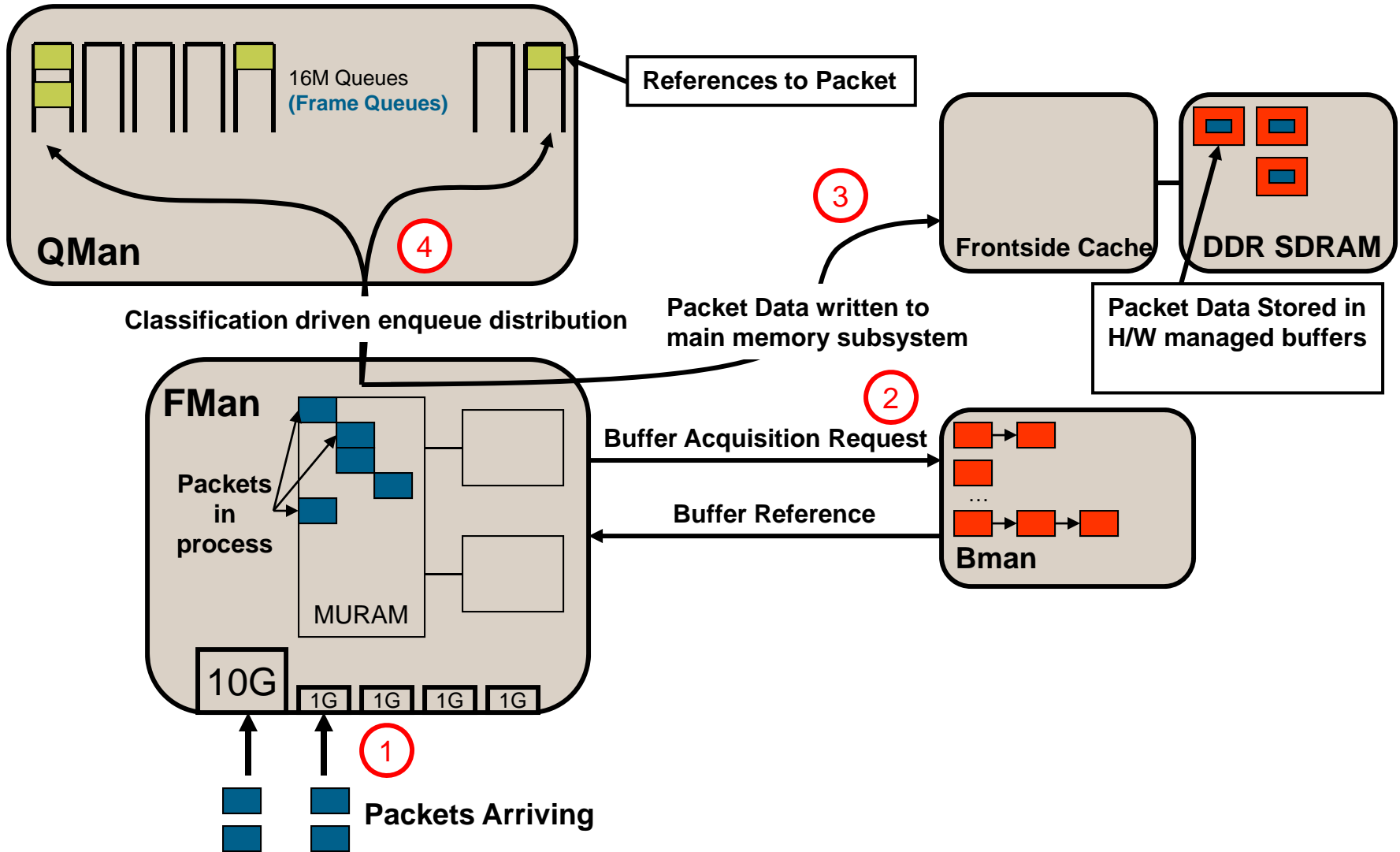
Data

Data

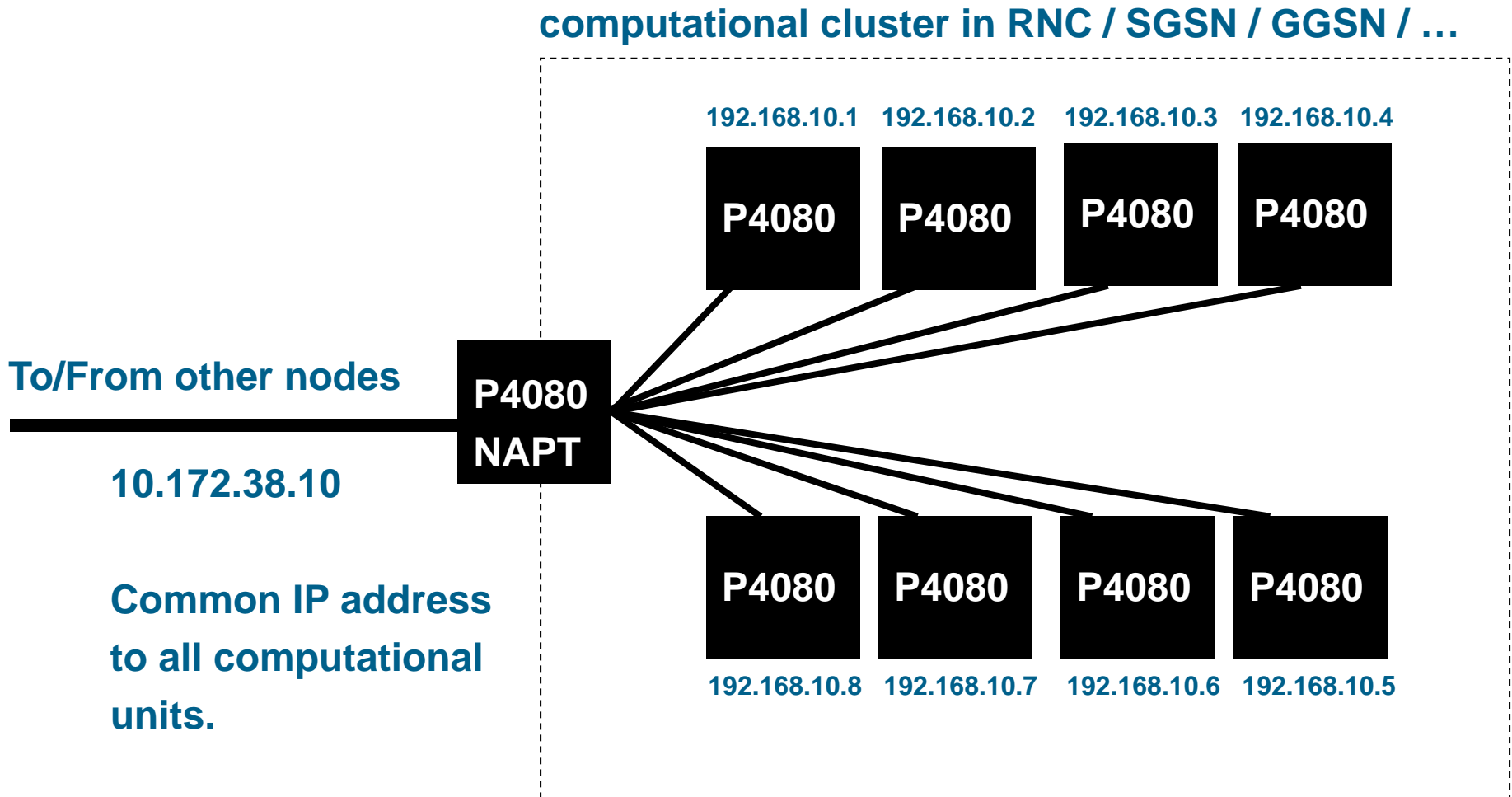


Packet I/O

FMan/QMan Ingress Packet Processing

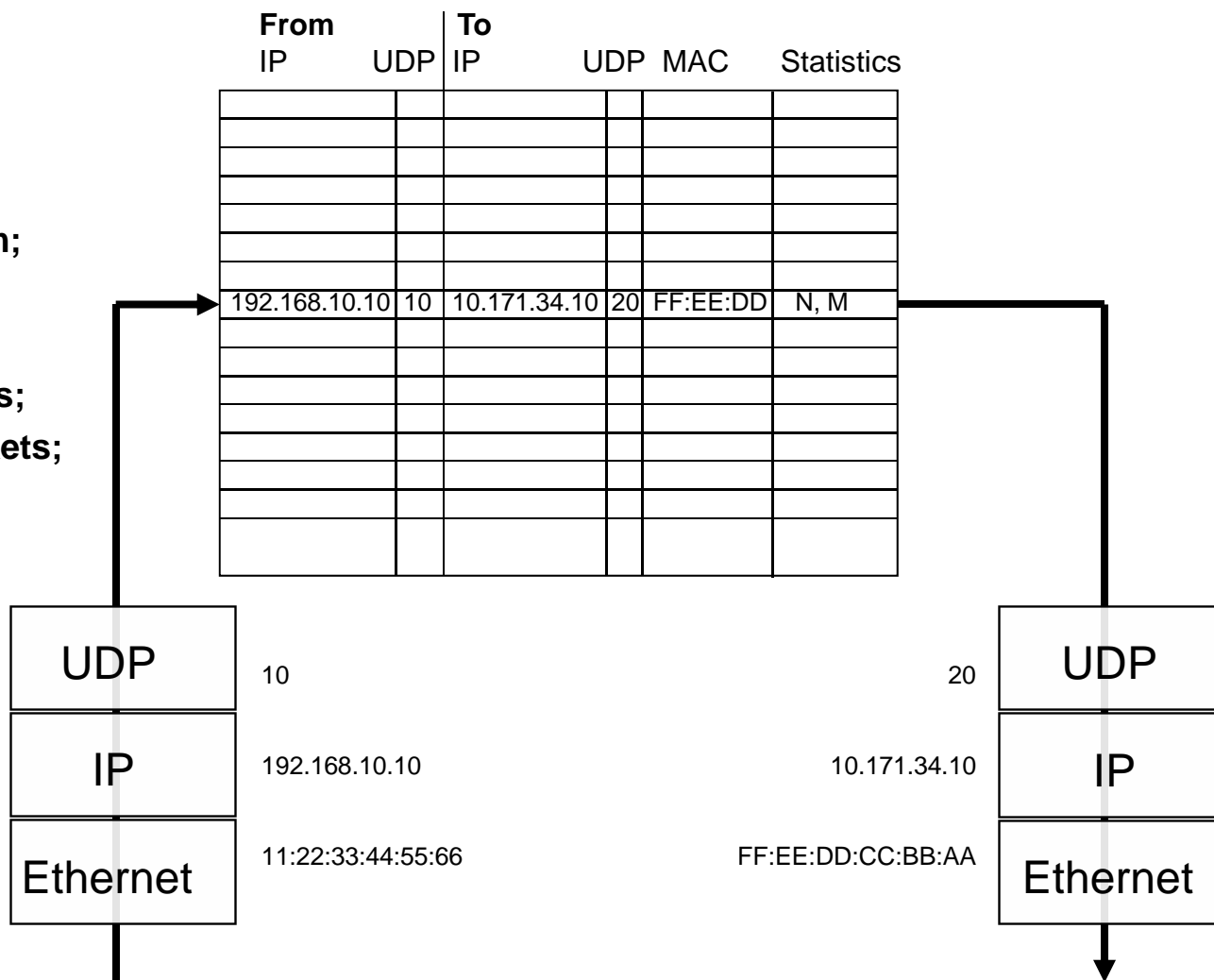


Example Location of NAPT Functionality



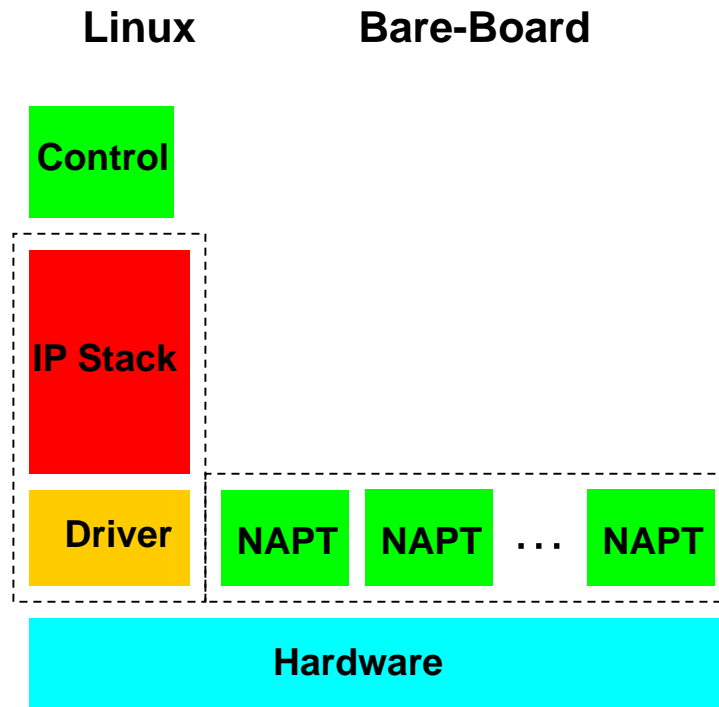
```
typedef struct SHLL_node
```

```
{
  SHLL_node *Next;
  uint32_t IpFrom;
  uint32_t IpTo;
  uint16_t PortFrom;
  uint16_t PortTo;
  union mac_address_t EthTo;
  uint64_t StatBytes;
  uint64_t StatPackets;
}
```



System Configuration: Linux® OS and Bare-metal

- ▶ Demo implementation based on separated slow- and fastpath
- ▶ Linux running on 1 core with control software on top
- ▶ 7 cores bare-metal application processing packets
- ▶ Hypervisor protects the partitions from each other



▶ Frame Manager Ingress

1. Buffer allocation in DDR
2. Parser / classification / policing
3. IP checksum
4. UDP checksum
5. Enqueue to core

Accelerator

▶ Software

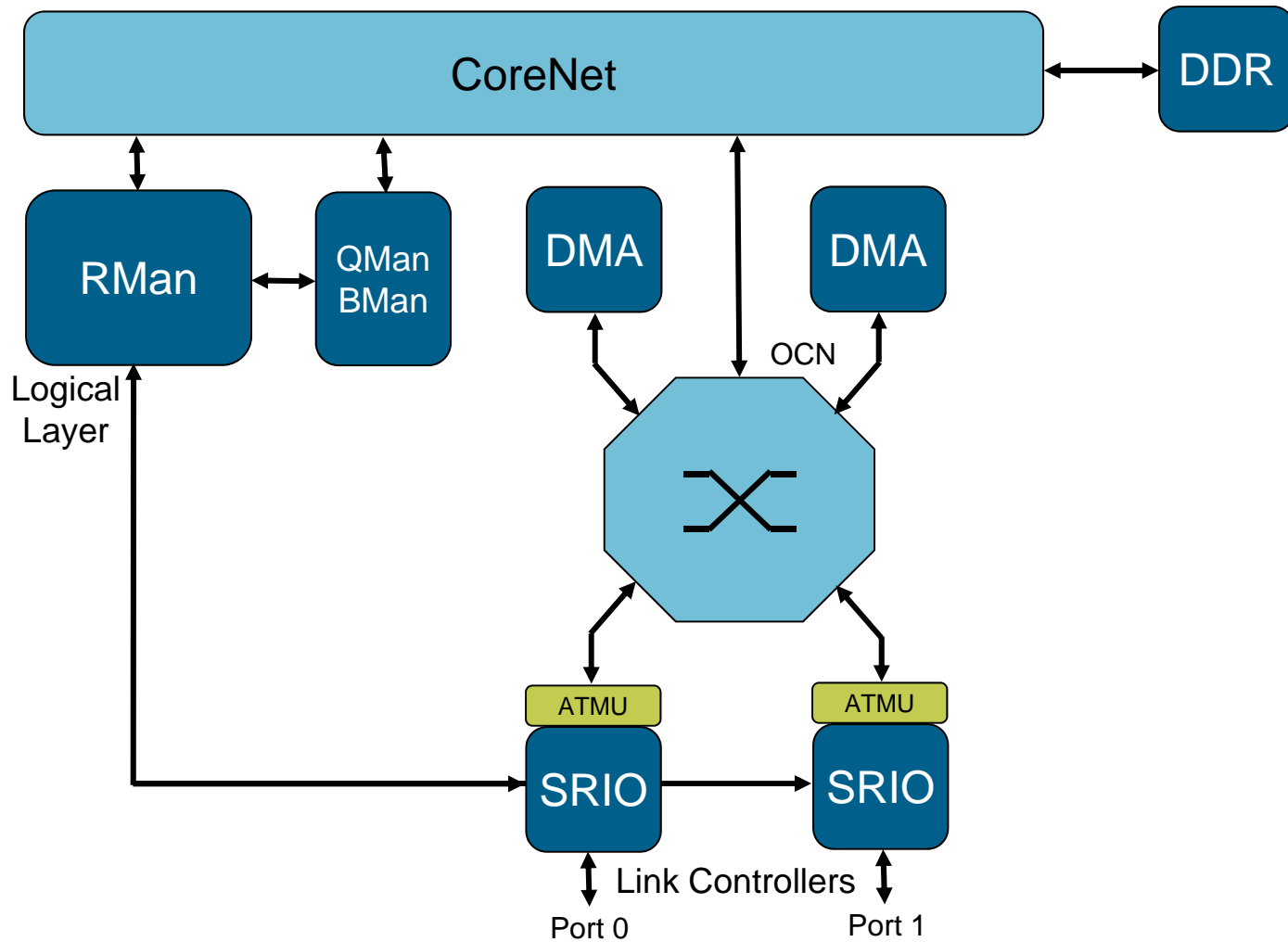
6. Dequeue from Frame Manager
7. Translation lookup
8. Enqueue to Frame Manager

Core

▶ Frame Manager Egress

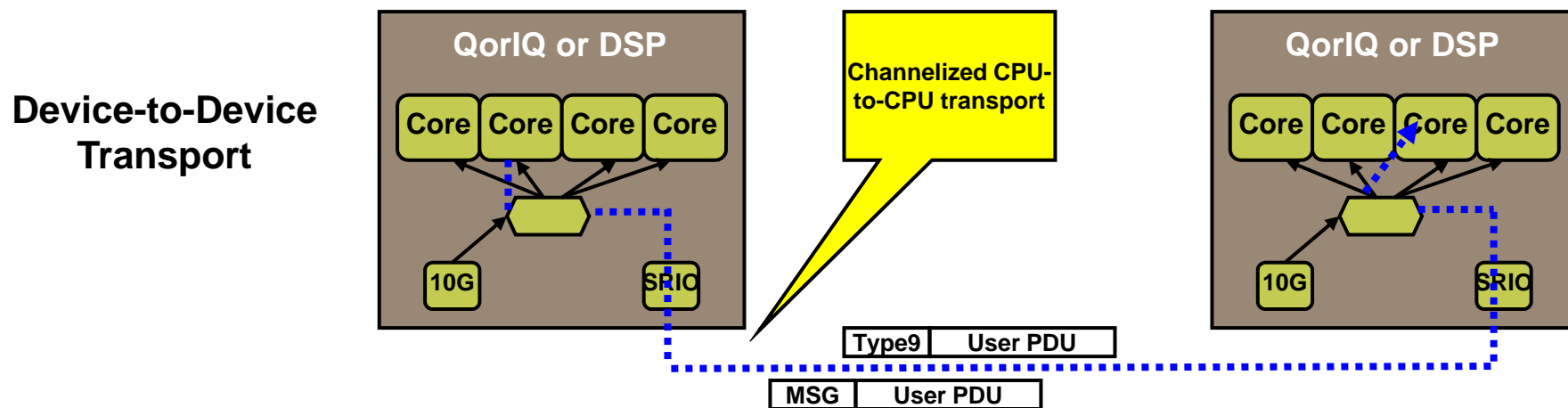
9. Dequeue from core
10. Add UDP checksum
11. Add IP checksum
12. Buffer deallocation from DDR

Accelerator



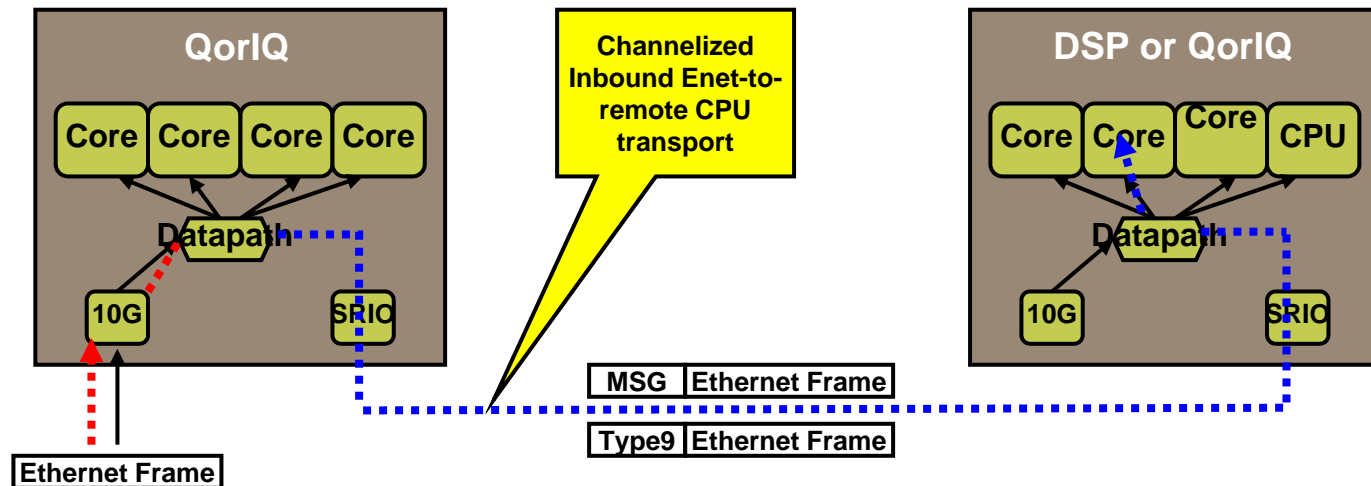
RMan for QorIQ: Greater Performance and Functionality

- ▶ Many queues allow multiple inbound/outbound queues per core
 - Hardware queue management via QorIQ Datapath 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

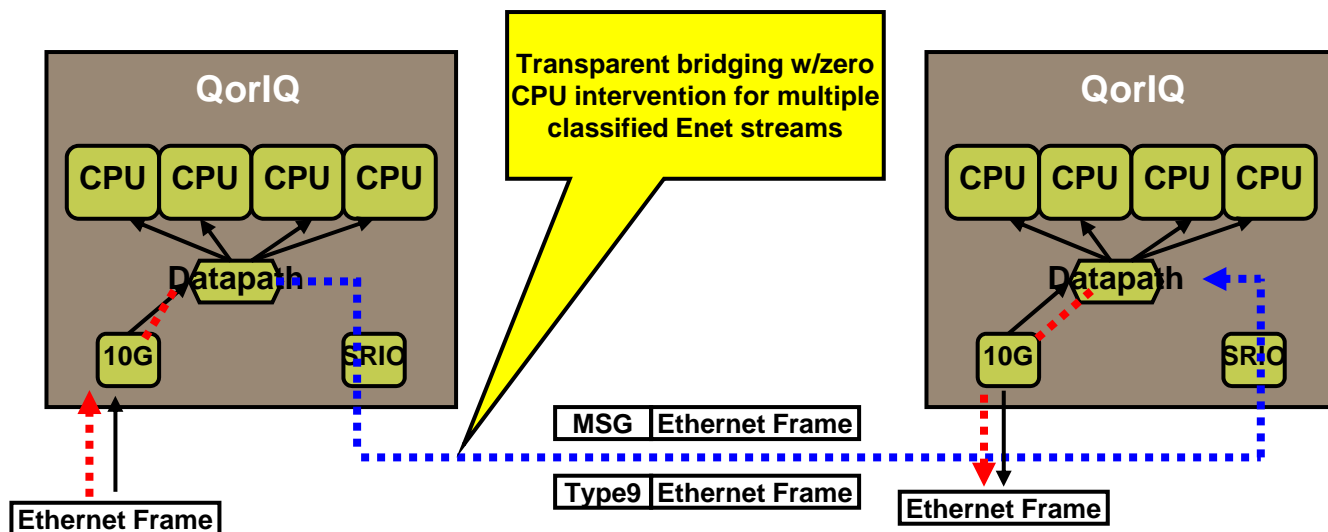


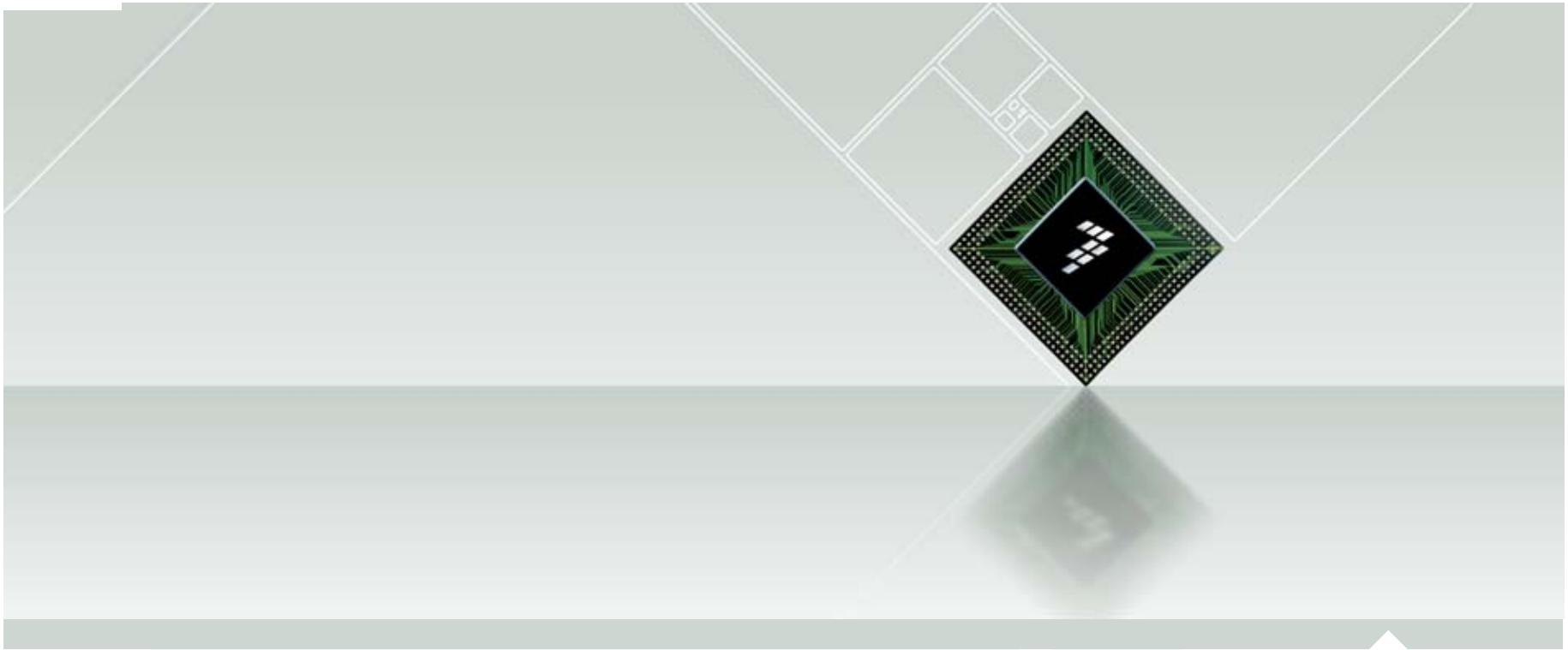
RMan Enables New Zero-CPU Overhead Use Models

Scalable Multicore System



Ethernet Bridging





PME: Traffic Monitoring and Management

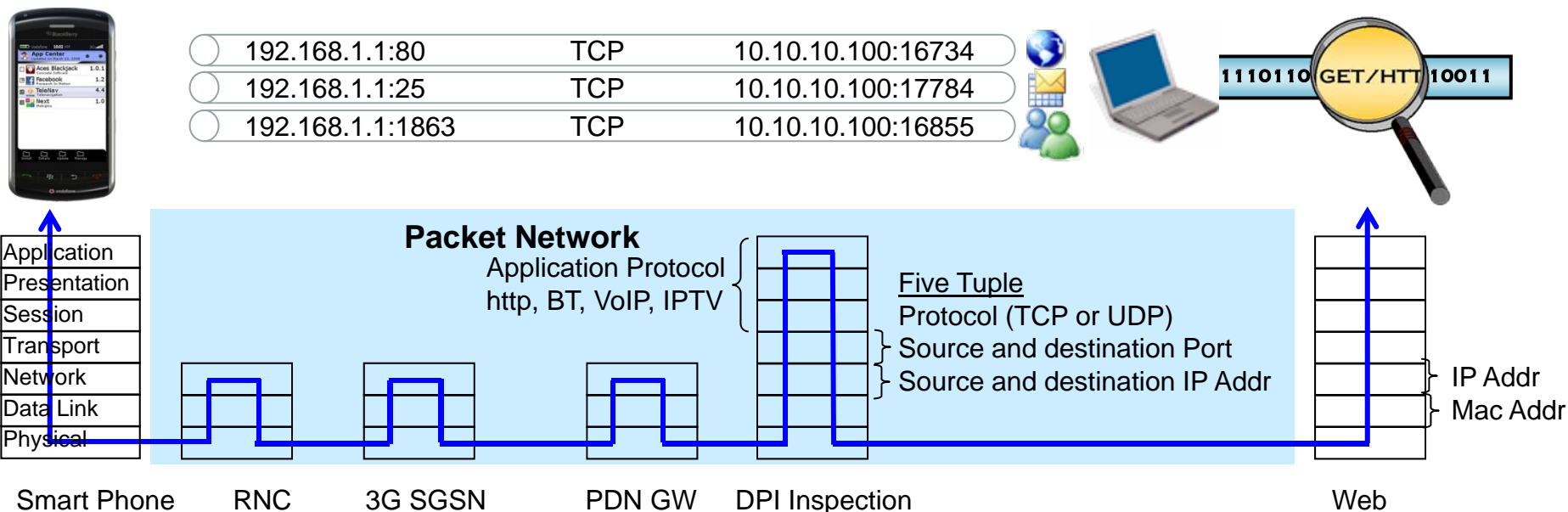
Deploy Deep Packet Inspection and Policy Control

► DPI enables companies to:

- Understand the network traffic and pattern
- Gather business intelligence
- Identify trends and adapt to those trends

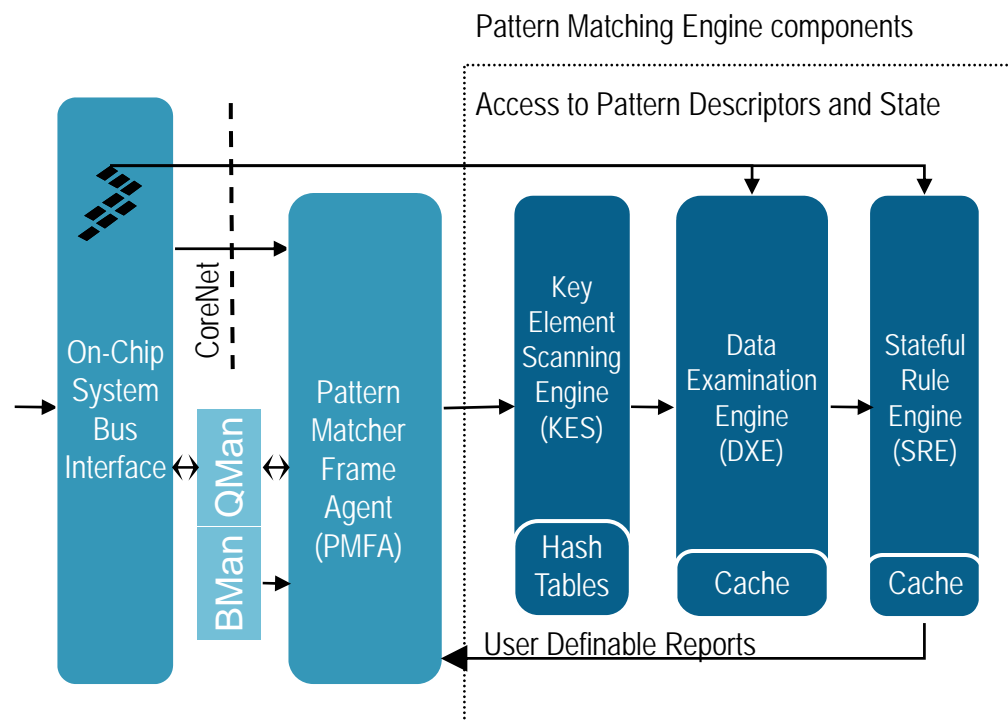
► Policy Control enables:

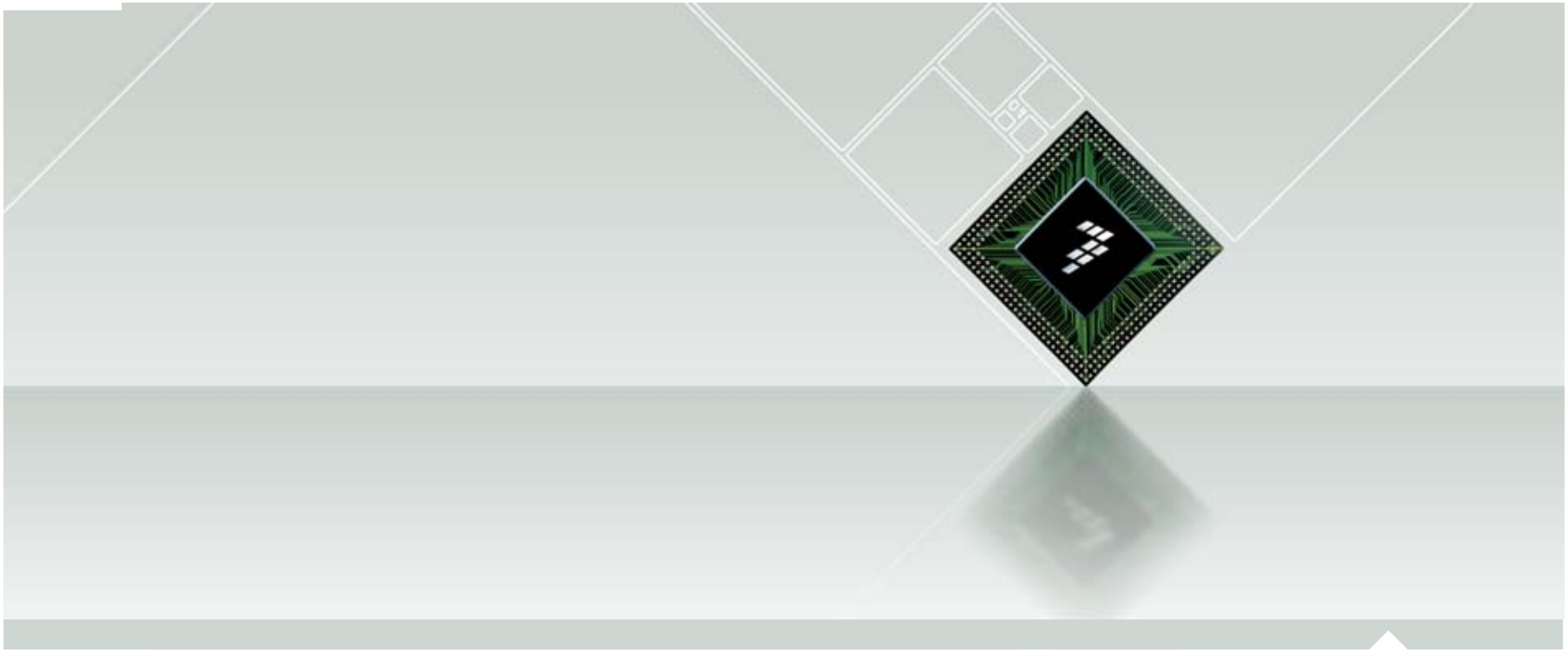
- A smarter pipe requires fine-tuned network controls
- Control and manage growing usage
- Fair usage to all network users



Policy Control: DPI with Pattern Matcher

- ▶ **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
- ▶ **Stateful rules operate on a per session basis**
 - User-defined logic reacts to pattern matches detected by the DXE
 - Can be used to further qualify the pattern match
 - Protocol state tracking (e.g. track the “normal” transitions of SMTP)





SEC: Ciphering and Packet Formatting

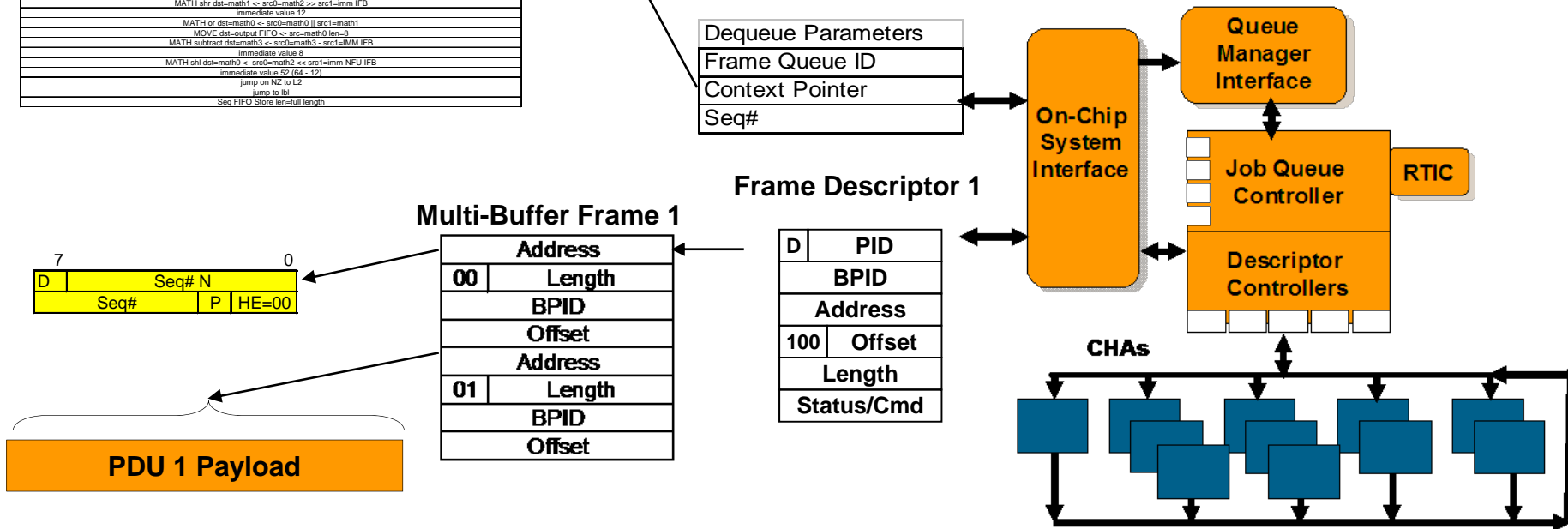
P4080 RLC AM Downlink Processing

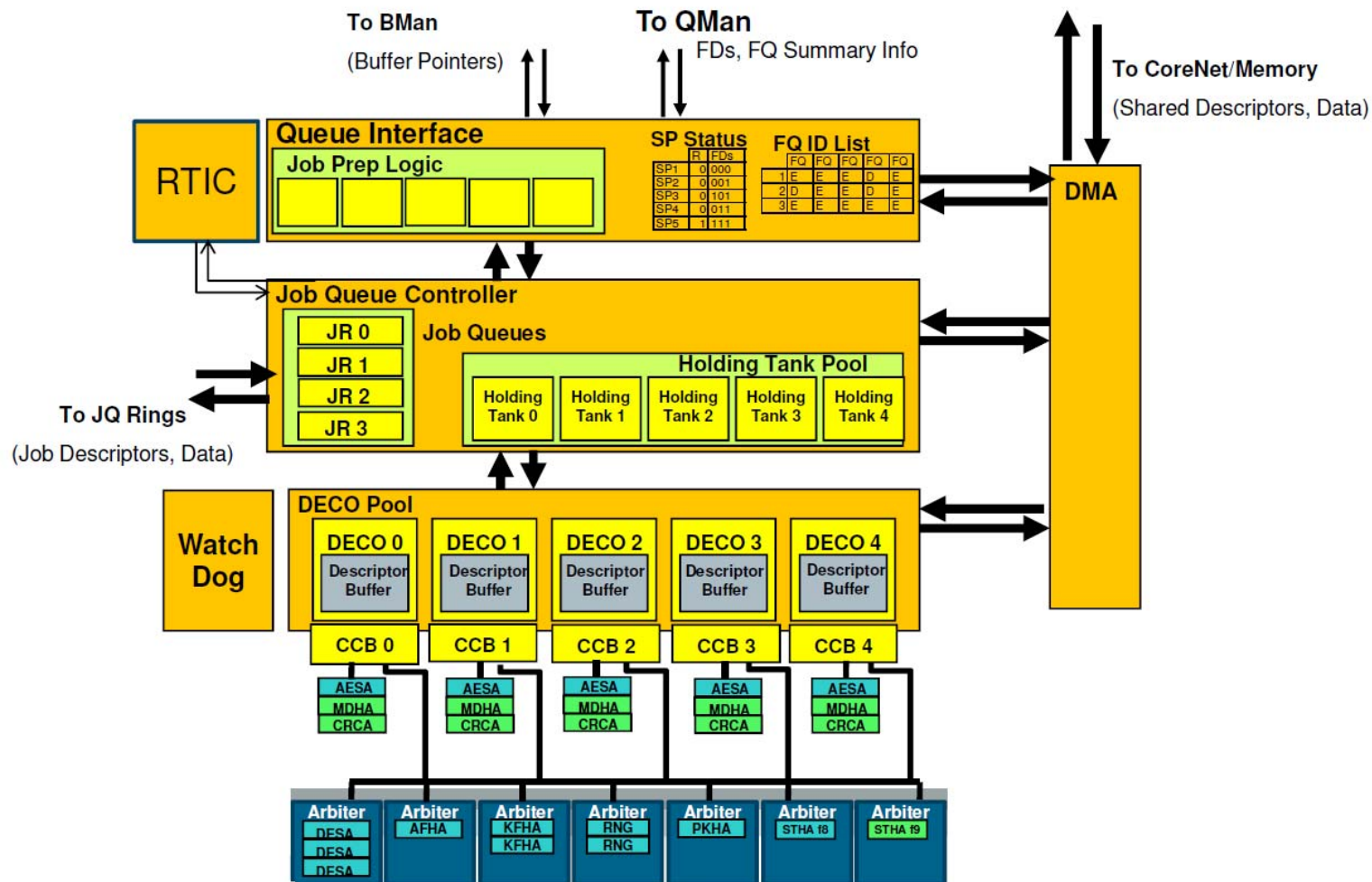
Downlink Shared Descriptor

31	shared descriptor header	0
LOAD imm. dst=math0 <- (the 16 byte ppdb)		
20 bit "ESN" reserved for CA & CE		
beamer	dir	
SEQ LOAD dst=math1 len=8		
SEQ FIFO LOAD len=encrypted payload length		
KEY inline len=16 class=1 dst=key		
16-byte inline key		
MATH shr dst=math1 <- src0=math1 src1=imm IFB		
immediate value 4		
MOVE dst=math2 src=math1		
MATH shr dst=math2 <- src0=math2 << src1=imm IFB (right align SN)		
immediate value 07		
MATH and dst=math2 <- src0=math2 & src1=IMM IFB		
immediate value FFF		
MATH add dst=math0 <- src0=math0 + src1=math2		
MOVE dst=ctx src=math0 offset=0		
OP Kas=8		
wait for done		
MOVE src=output FIFO dst=math3 len=encrypted payload length		
MOVE dst=math2 <- src=output FIFO len=8		
MATH shr dst=math1 <- src0=math2 >> src1=imm IFB		
immediate value 12		
MATH or dst=math0 <- src0=math0 src1=math1		
MOVE dst=output FIFO <- src=math0 len=8		
MATH subtract dst=math3 <- src0=math3 - src1=IMM IFB		
immediate value 9		
MATH shl dst=math0 <- src0=math2 << src1=imm NFU IFB		
immediate value 52 (64 - 12)		
jump on NZ to L2		
jump to B7		
Seq FIFO Store len=full length		

The SEC dequeues from its highest priority FQ, an action which gets it the Shared Descriptor, the FD, SG List, and initial data.

The Shared Descriptor is only fetched once per FQ dequeue, and a configurable max number of FDs can be processed from that FQ.

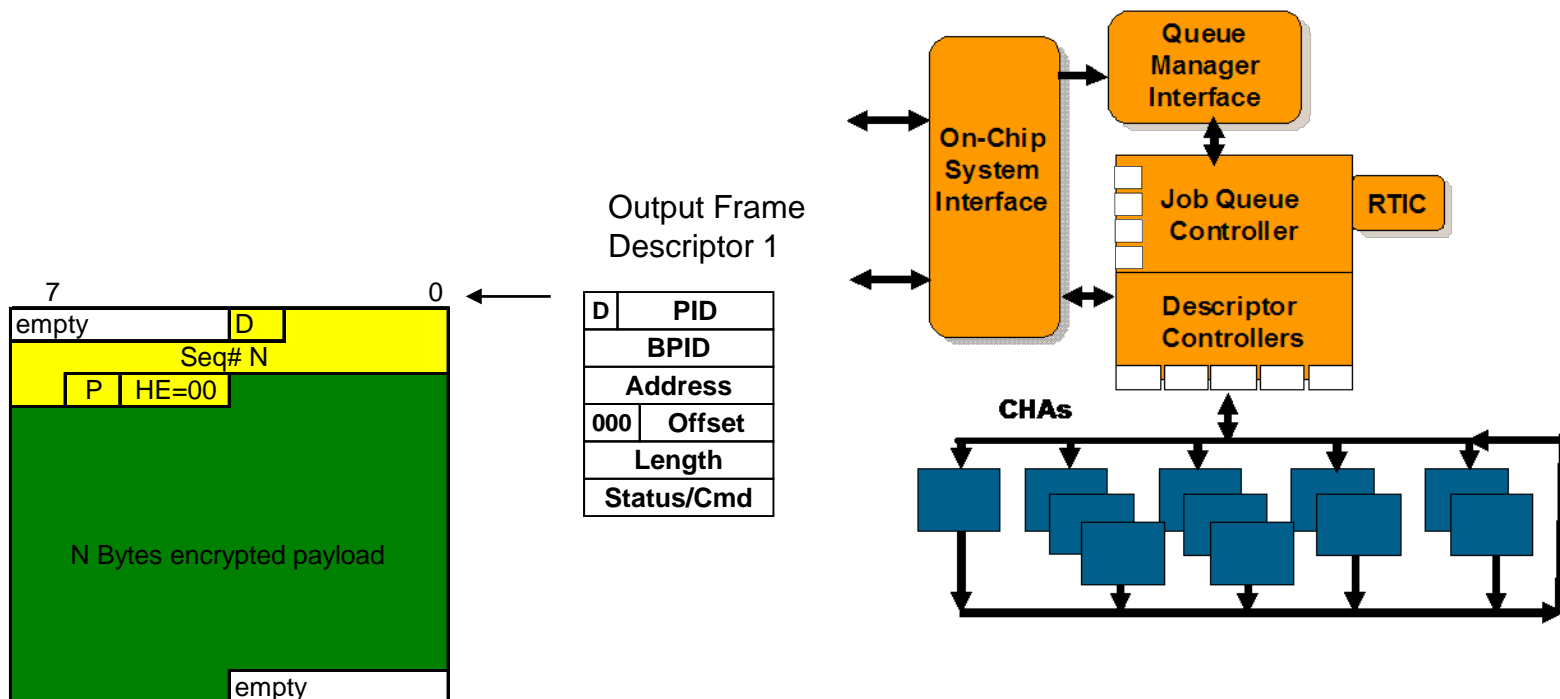




- ▶ Load Commands:
 - [SEQ] KEY – Load cipher keys
 - [SEQ] LOAD – Load register
 - [SEQ] FIFO LOAD – Loads In/Out FIFO
- ▶ Store Commands:
 - [SEQ] STORE – Store register
 - [SEQ] FIFO STORE – Store In/Out FIFO
- ▶ MOVE – Move data between SEC registers.
- ▶ MATH – Perform arithmetic operation (sum, and, or, xor, bitshift, ...)
- ▶ OPERATION – Execute cipher operation (Kasumi, Snow, AES, ...)
- ▶ JUMP – Branch in the descriptor
- ▶ SEQ IN/OUT PTR – Sets pointer for in/out sequence data.
- ▶ JD/SD HEADER – First word in the descriptor.
- ▶ SIGNATURE – Last word in trusted descriptor.

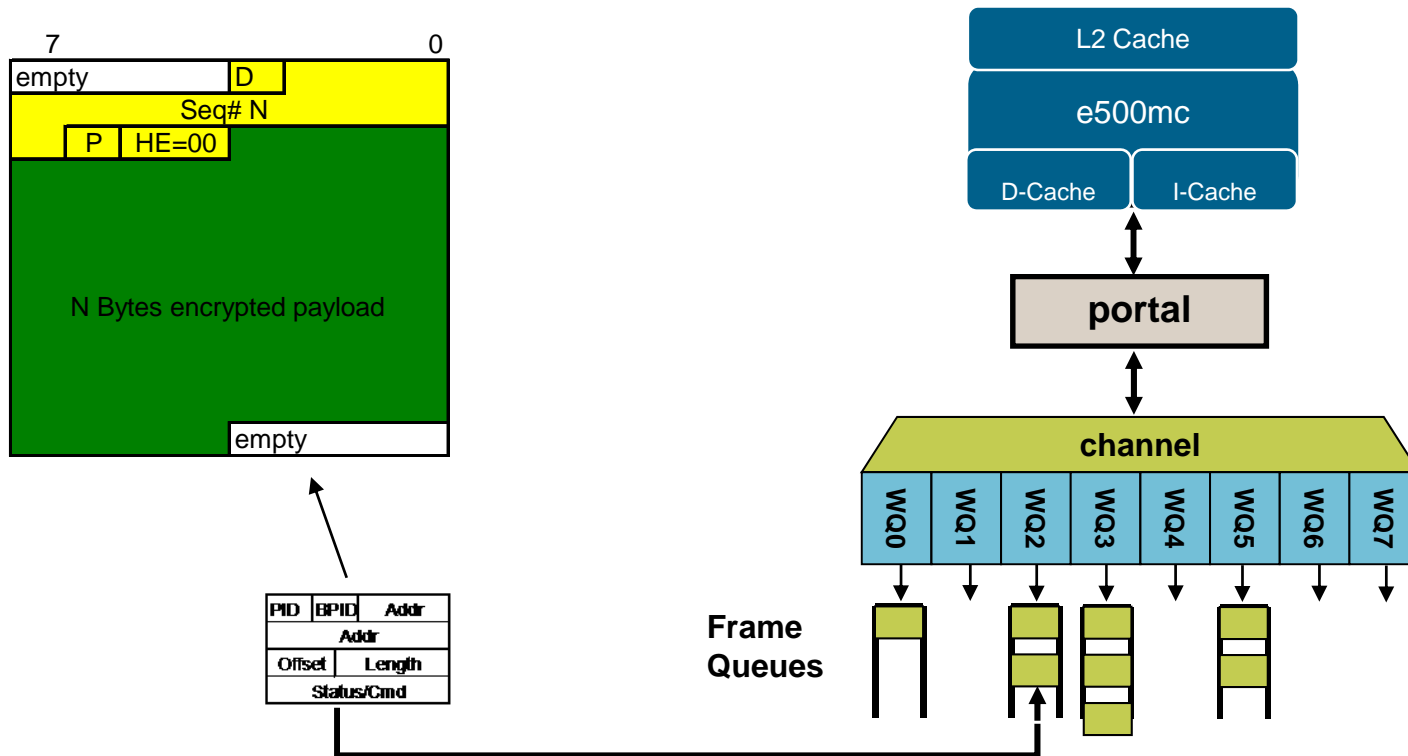
P4080 RLC AM Downlink Processing

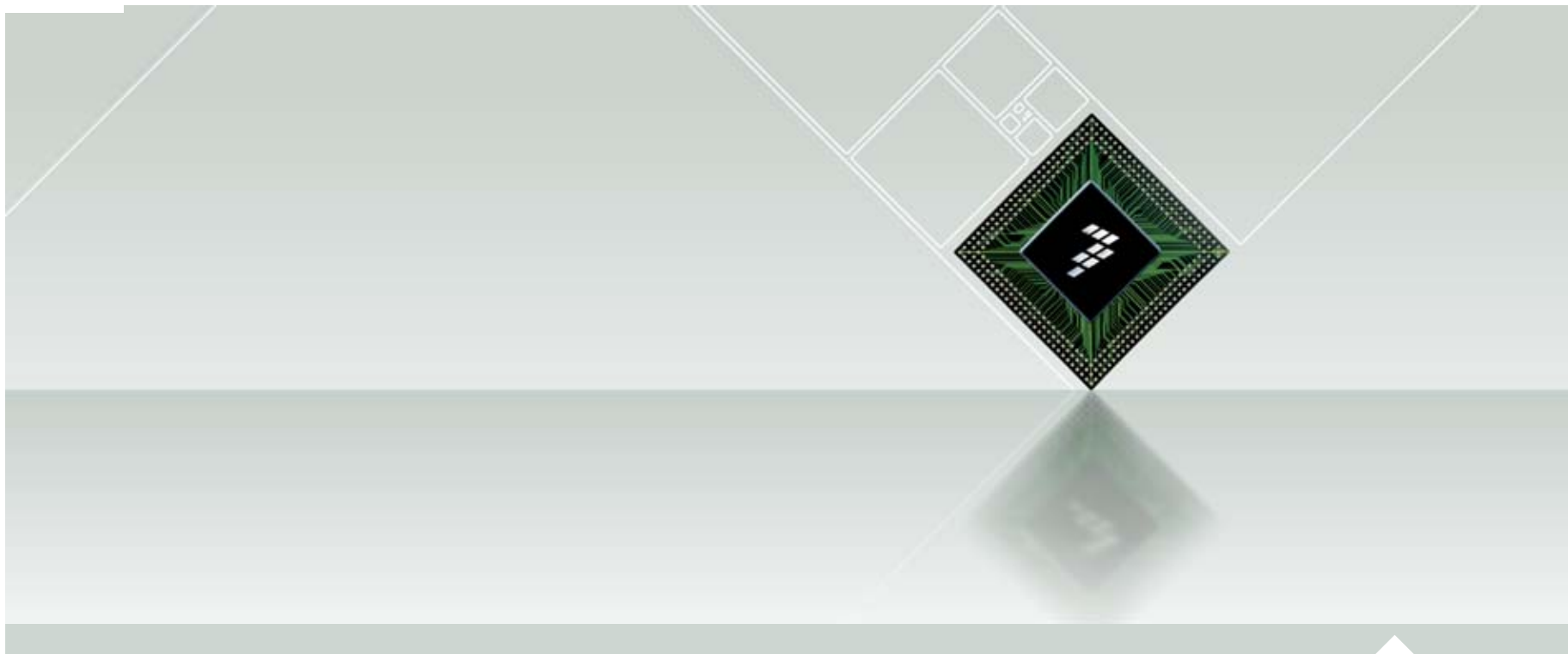
An output Frame Descriptor is sent out from the SEC that in turn points to the generated output data. Input data is either left in memory but can also be freed and the buffer returned to BMan.



P4080 RLC AM Downlink Processing

The frame descriptors is enqueued into a work queue / channel back to a core for further processing.





Use Case: Voice Packet Processing

- ▶ Voice Packets are ciphered/formatted in RNC (3G) and (e)NB (LTE and 3G flat architecture).
- ▶ Voice packets are large to the number but small to the size (~20 bytes).
- ▶ Typically each active flow has a packet per ~20ms, i.e. it is not possible to group packets of same flow or latency will increase and phone conversation will not work (lag severely).
- ▶ Hardware Acceleration is a big benefit but overhead/byte of sending to accelerator has been a big problem since packets are so small.

Frame Descriptor

D	PID
BPID	
Address	
100	Offset
Length	
Status/Cmd	
Context 0	

Data S/G List

00	Length
BPID	
Offset	
Address	
10	Length
BPID	
Offset	
Address	

Input S/G List

00	Length
BPID	
Offset	
Address	
00	Length
BPID	
Offset	
Address	
...	
Address	
01	Length
BPID	
Offset	

Job Definition
(# of packages, size etc)

Packet 0

...

Data N

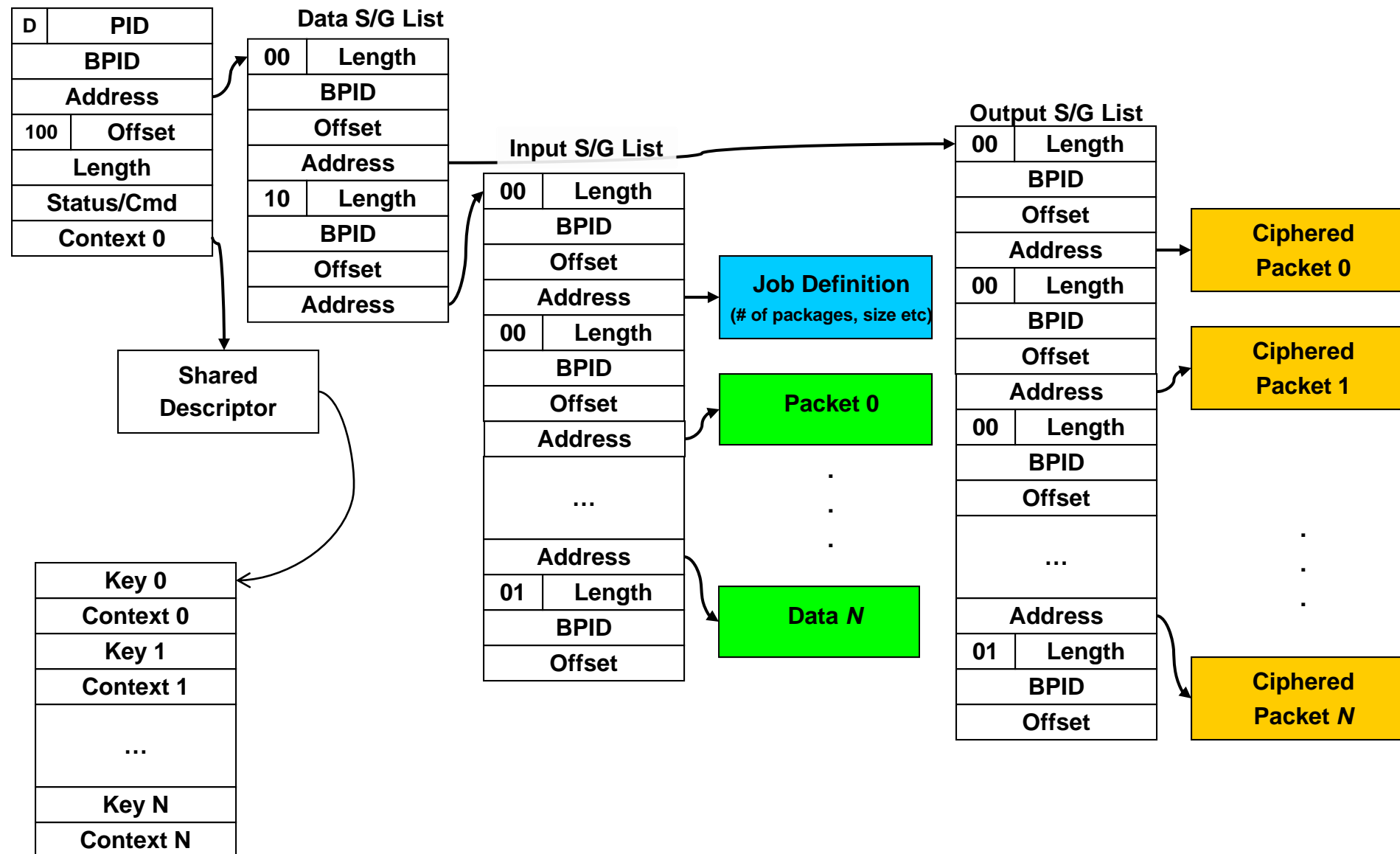
Output

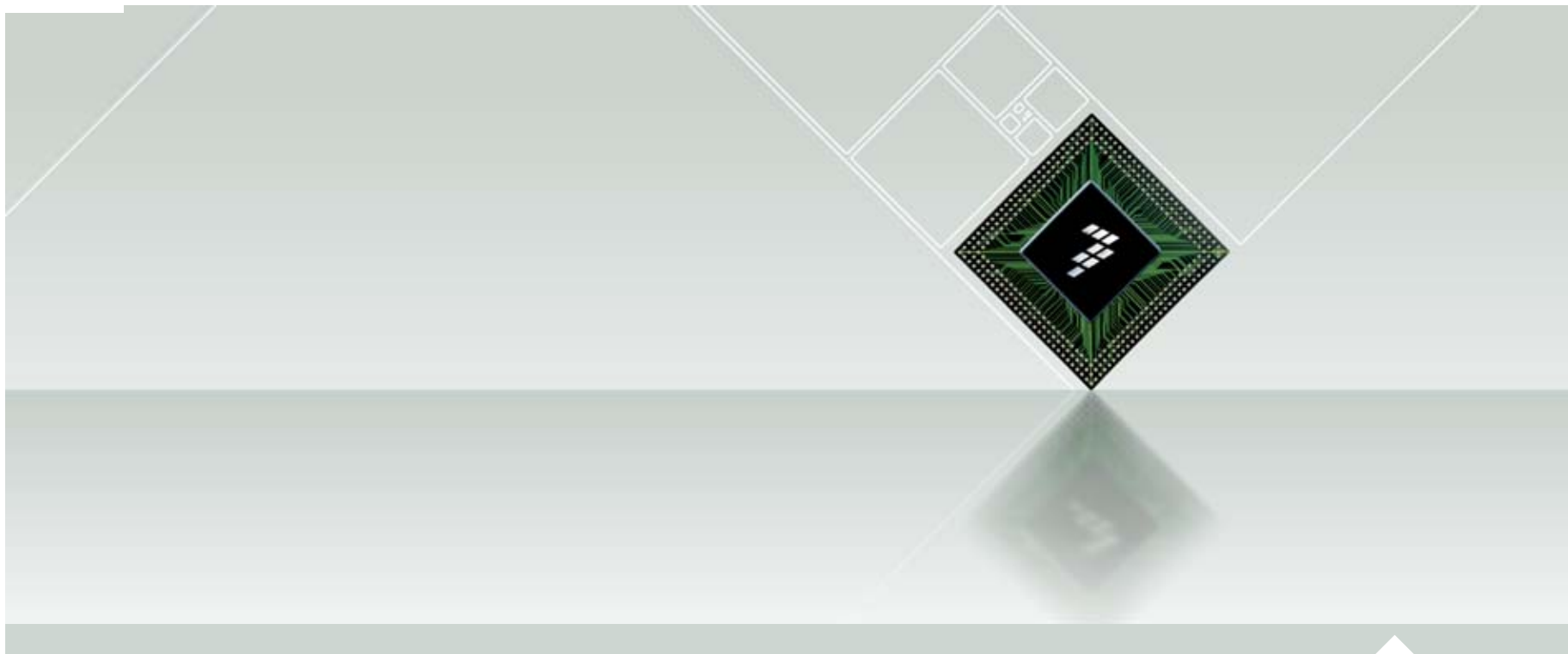
Ciphered Packet 0
Ciphered Packet 1
...
Ciphered Packet N

Shared Descriptor

Key 0
Context 0
Key 1
Context 1
...
Key N
Context N

S/G output – buffers at random location





Use Case: HSPA Mobile Broadband

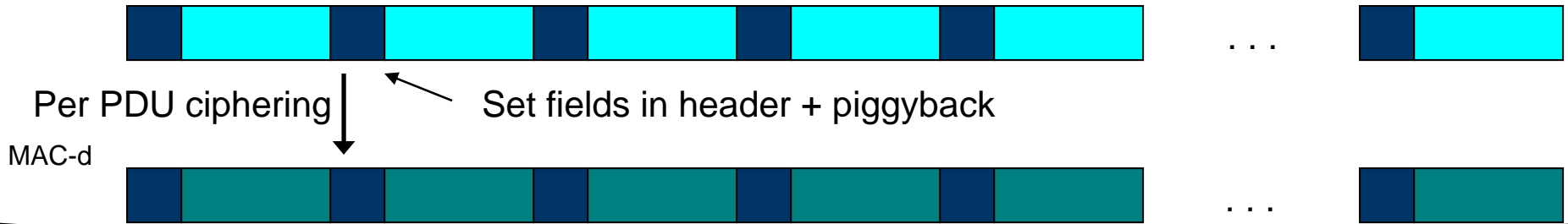
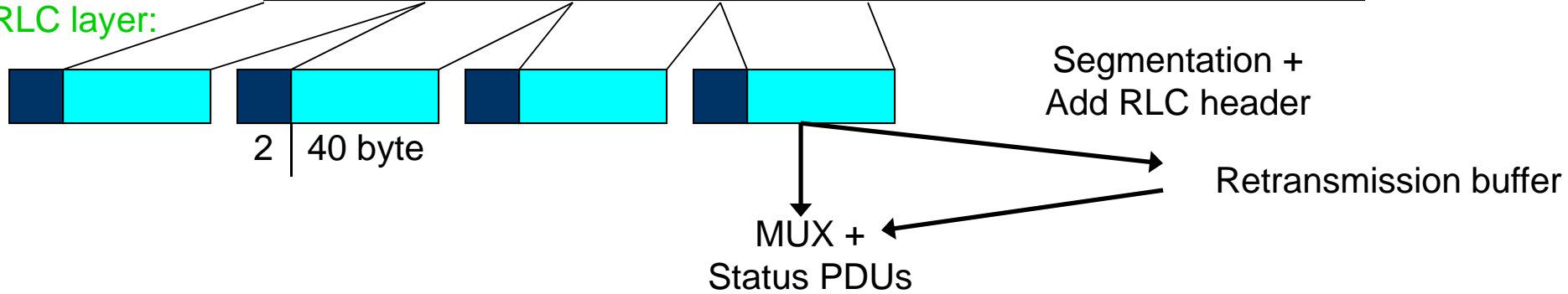
- ▶ Huge total bandwidth!
- ▶ Very high individual user peak rate.
- ▶ Complex protocol processing.

Overall Flow HS

MAC/GTP-U layer:

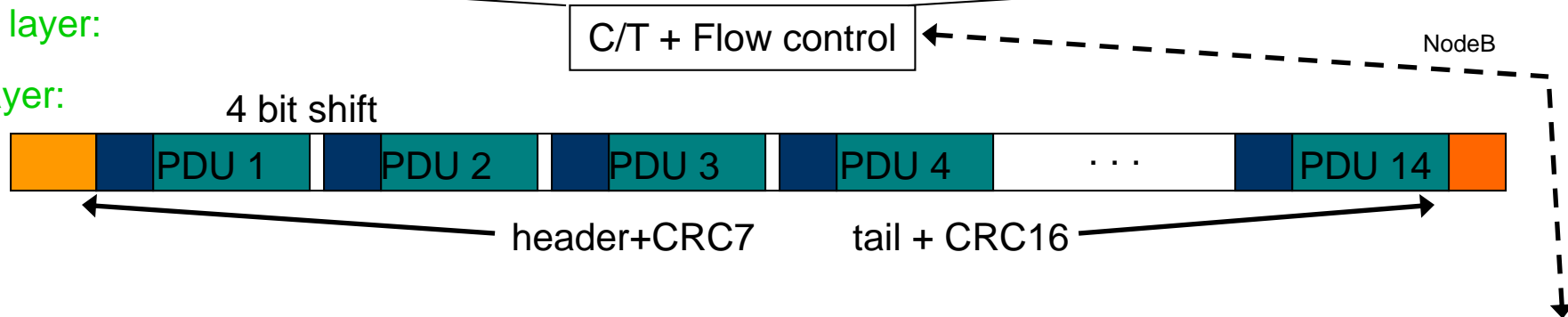
User data: IP packet (max 65535 bytes)

RLC layer:



MAC layer:

FP layer:



Frame Type 1 with MAC and RLC PDU – R7

Gray = Depends on configuration and version

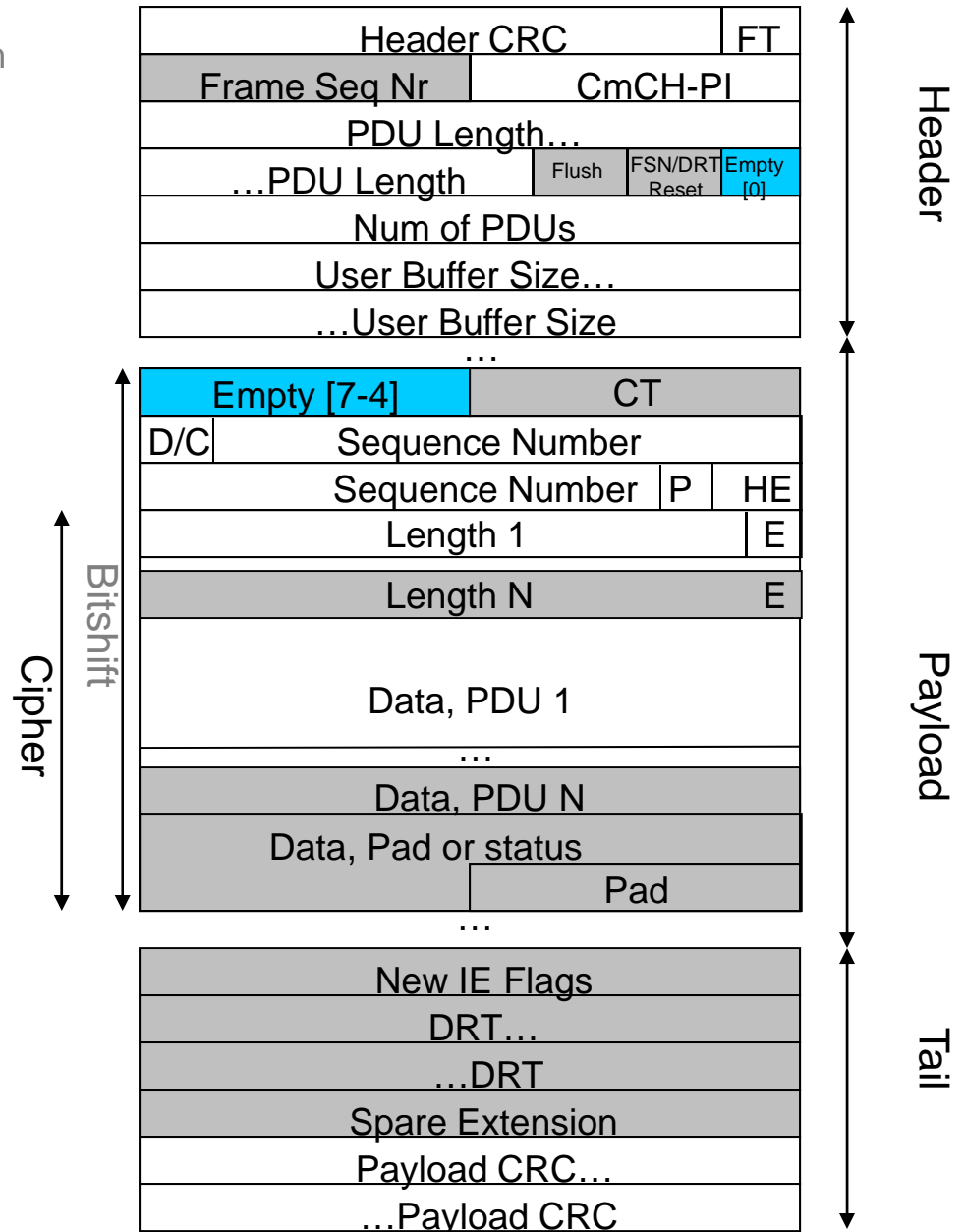
Blue = Empty

Example of special cases:

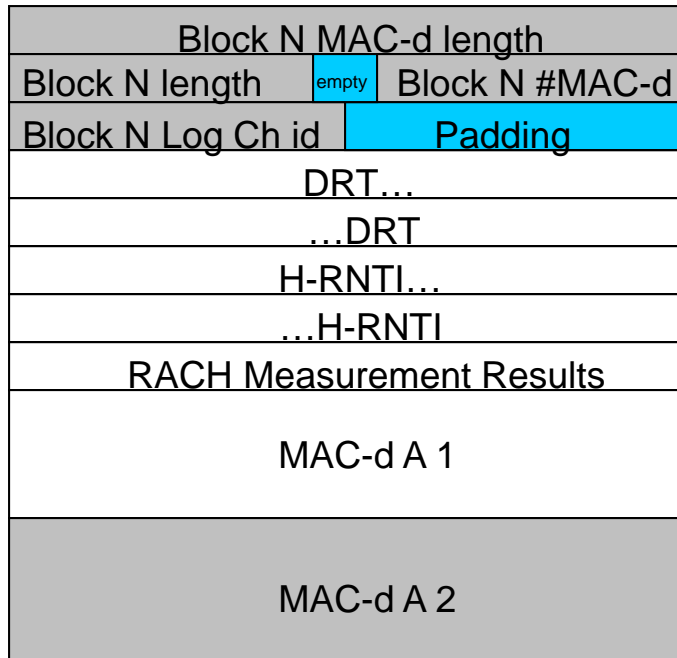
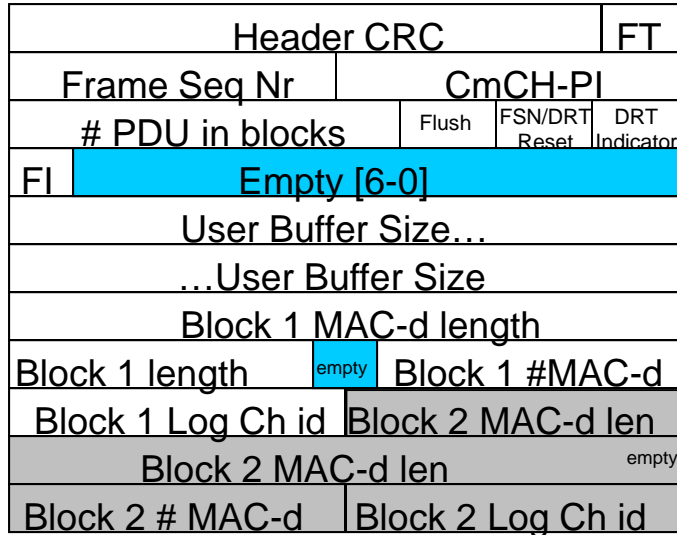
E bit can have “alternative interpretation”

HE valid values differs over revisions

Length field can be 7 or 15 bits



Frame Type 2 with MAC and RLC PDU – R7

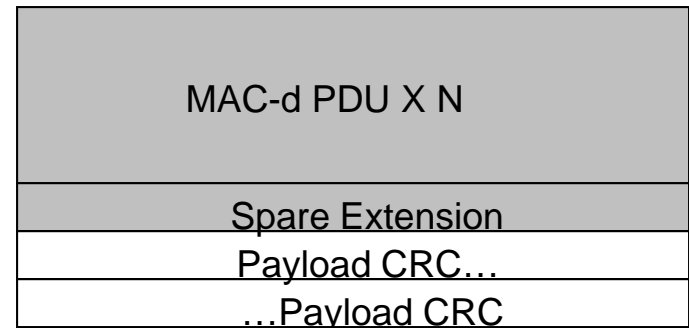
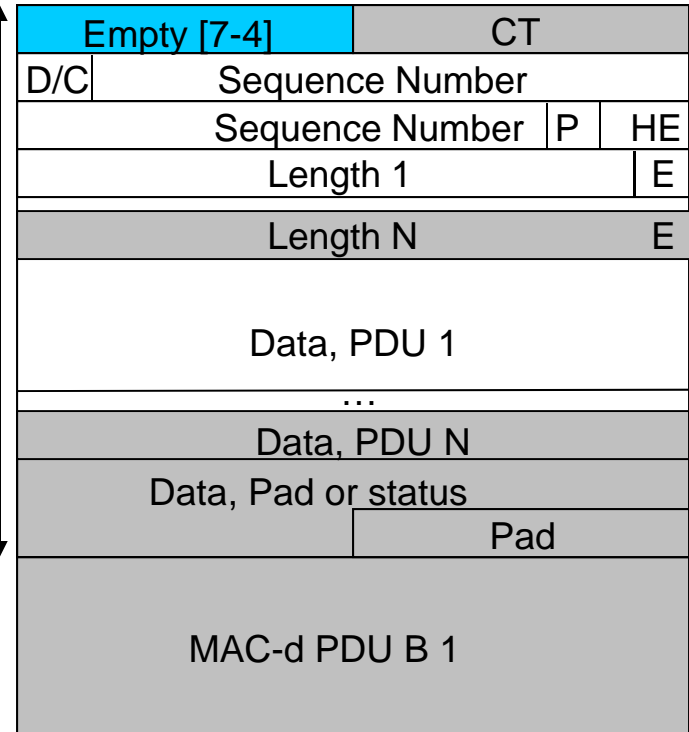


Header

Payload

Cipher

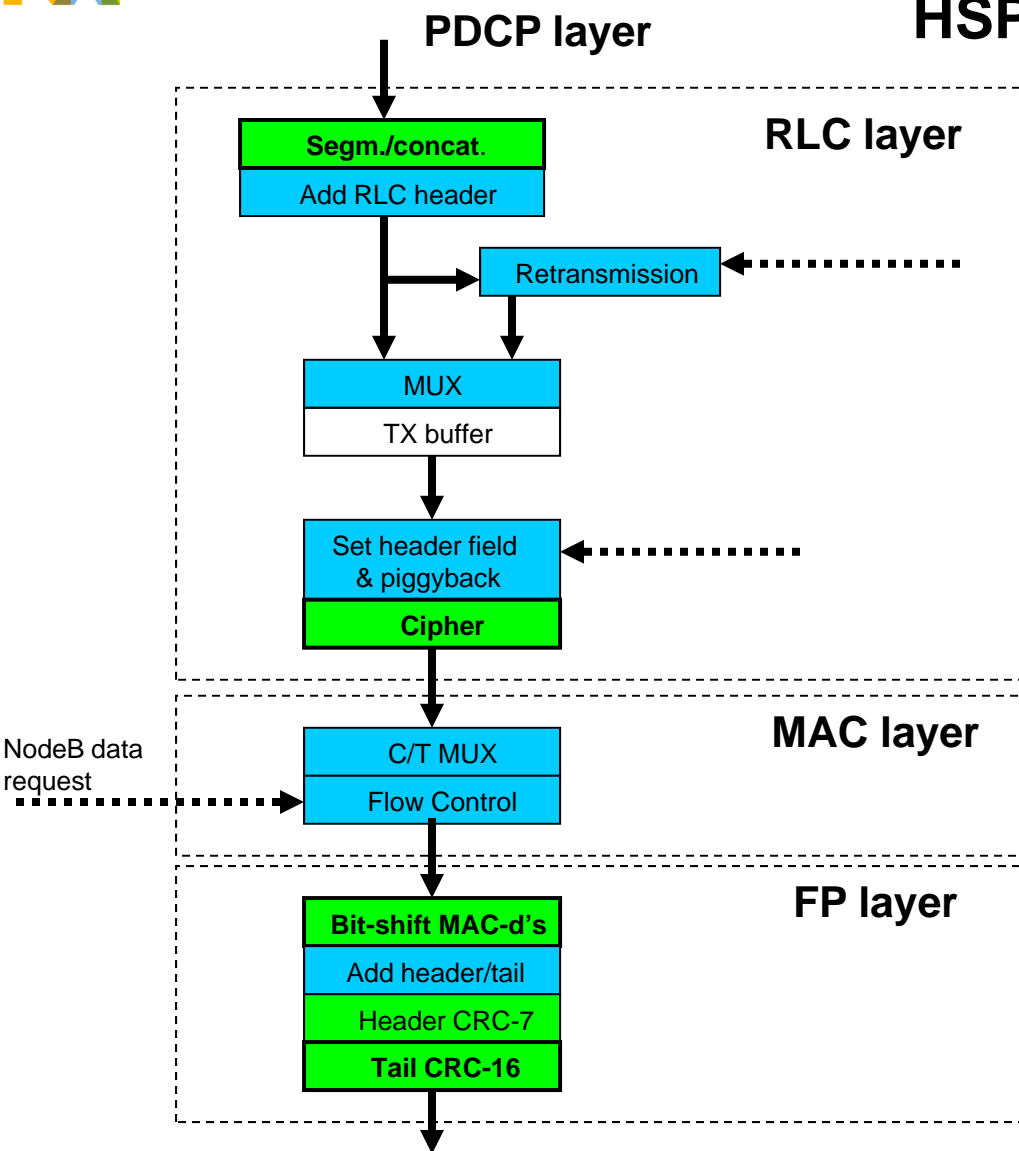
Bitshift



MAC-d PDU A N

Payload

Tail



Bold frame indicates intensive work.

Blue = Core

Green = Accelerator

Features for mobile broadband:

- ▶ Cipher blocks of packets
- ▶ Auto-update sequence number
- ▶ Bit-shifting
- ▶ CRC-7 / CRC-11 / CRC-16
- ▶ Dual CRC (header/payload)

RLC / MAC / FP flow on P4080: Example A

- ▶ User data assembled in memory, from GTP-U.
- ▶ When MAC flow control indicates that we can send data:

Core:

1. Checks retransmission buffer.
2. Checks status transmission buffer.
3. Creates RLC headers with status.
4. Checks if C/T MUX is used.
5. Adds pointers to segments and RLC headers.
6. Concatenation/piggyback/pad. of last user-data packet
7. Saves pointers for retransmission buffer
8. Creates FP header/tail.

SEC: segments, adds header, bitshift and cipher. Adds FP header/tail and computes CRC-7 and CRC-16.

D	PID
BPID	
Address	
100	Offset
Length	
Status/Cmd	

S/G List	
00	Length
BPID	
Offset	
Address	
10	Length
BPID	
Offset	
Address	

Input S/G List

00	Length
BPID	
Offset	
Address	
00	Length
BPID	
Offset	
Address	
00	Length
BPID	
Offset	
Address	
00	Length
BPID	
Offset	
Address	
00	Length
BPID	
Offset	
Address	
01	Length
BPID	
Offset	

Job Definition

Header 0

Payload 0

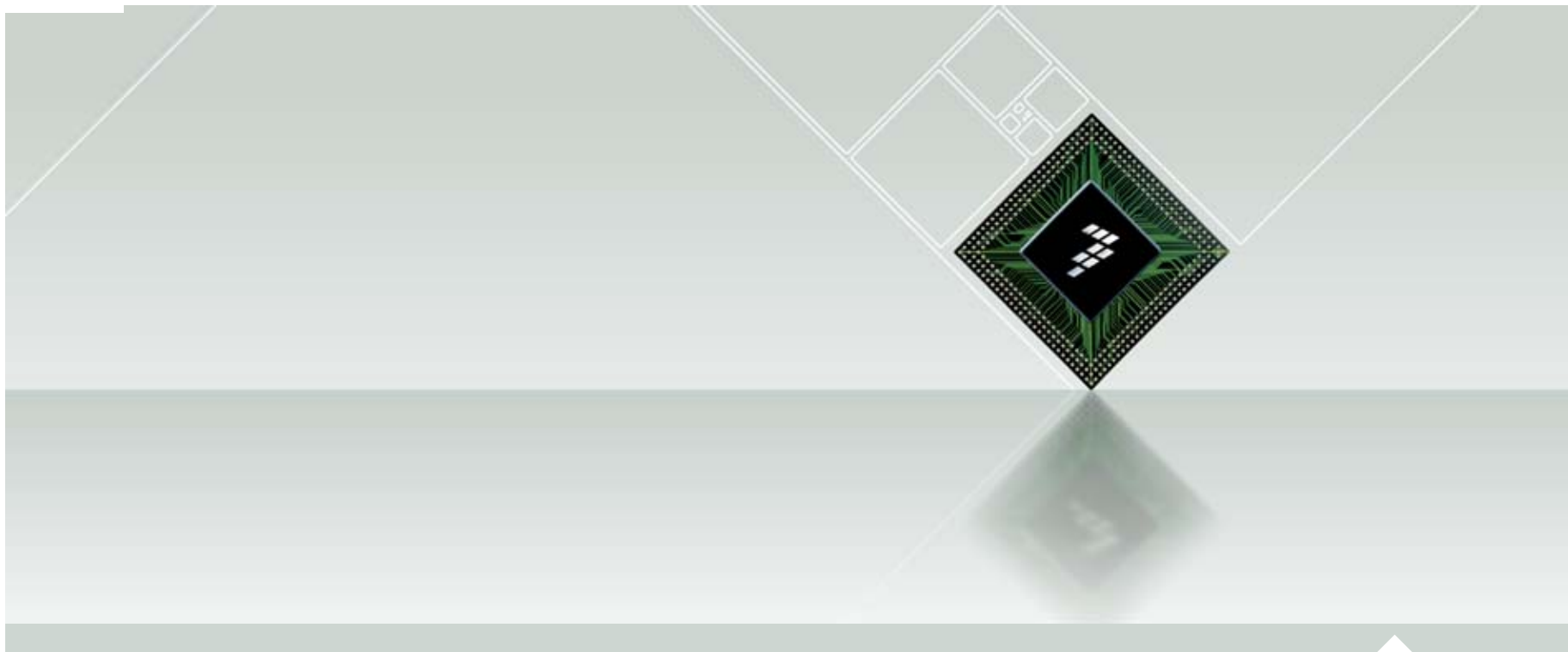
Header 1

...

Payload N

Output

7	0
Header CRC	
Fr Seq #	
MAC-d PDU Length	
# of PDUs	
User Buffer Size	
empty	
Seq# N	
P HE=00	
82B encrypted payload	
empty	
empty	
Seq# N	
P HE=00	
82B encrypted payload	
empty	
empty	
Seq# N	
P HE=00	
82B encrypted payload	
empty	
New IE flags	
DRT	
Spare Extension	
CRC16 (LSB)	
CRC16 (MSB)	



DEMO

