

Welcome to [E-XFL.COM](#)

Understanding [Embedded - Microprocessors](#)

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of [Embedded - Microprocessors](#)

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details

Product Status	Active
Core Processor	PowerPC e6500
Number of Cores/Bus Width	12 Core, 64-Bit
Speed	1.8GHz
Co-Processors/DSP	-
RAM Controllers	DDR3, DDR3L
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	1Gbps (16), 10Gbps (4)
SATA	SATA 3Gbps (2)
USB	USB 2.0 + PHY (2)
Voltage - I/O	-
Operating Temperature	0°C ~ 105°C (TA)
Security Features	-
Package / Case	1932-BBGA, FCBGA
Supplier Device Package	1932-FCPBGA (45x45)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/t4240nsn7qtb

2 Summary of benefits

The T4 family of processors are ideal for combined control and data plane processing. A wide variety of applications can benefit from the processing, I/O integration, and power management capabilities. Similar to other QorIQ devices, the T4 family of processors' high level of integration offers significant space, weight, and power benefits compared to multiple discrete devices. Examples include:

- Service provider networking: RNC, metro networking, gateway, core/edge router, EPC, CRAN, ATCA, and AMC solutions.
- Enterprise equipment: router, switch services, and UTM appliances.
- Data centers: NFV, SDN, ADC, WOC, UTM, proxy, server appliance, and PCI Express (PCIe) offload.
- Storage controllers: FCoE bridging, iSCSI controller, and SAN controller.
- Aerospace, defense, and government: radar imaging, ruggedized network appliance, and cockpit display.
- Industrial computing: single-board computers and test equipment.

2.1 e6500 CPU core

The T4 family of processors are based on the Power Architecture® e6500 core. The e6500 core uses a seven-stage pipeline for low latency response while also boosting single-threaded performance. The e6500 core also offers high aggregate instructions per clock at lower power with an innovative "fused core" approach to threading. The e6500 core's fully resourced dual threads provide 1.7 times the performance of a single thread.

The e6500 cores are clustered in banks of four cores sharing a 2 MB L2 cache, allowing efficient sharing of code and data within a multicore cluster. Each e6500 core implements the Freescale AltiVec technology SIMD engine, dramatically boosting performance of heavy math algorithms with DSP-like performance.

The e6500 core features include:

- Up to 1.8 GHz dual threaded operation
- 7 DMIPS/MHz per core
- Advanced power saving modes, including state retention power gating

2.2 Virtualization

The T4 family of processors includes support for hardware-assisted virtualization. The e6500 core offers an extra core privilege level (hypervisor) and hardware offload of logical-to-real address translation. In addition, the T4 family of processors includes platform-level enhancements supporting I/O virtualization with DMA memory protection through IOMMUs and configurable "storage profiles" that provide isolation of I/O buffers between guest environments. Virtualization software for the T4 family includes kernel virtualization machine (KVM), Linux containers, and Freescale hypervisor and commercial virtualization software from vendors such as Enea®, Greenhills Software®, Mentor Graphics®, and Wind River.

2.3 Data Path Acceleration Architecture (DPAA)

The T4 family of processors enhance the QorIQ DPAA, an innovative multicore infrastructure for scheduling work to cores (physical and virtual), hardware accelerators, and network interfaces.

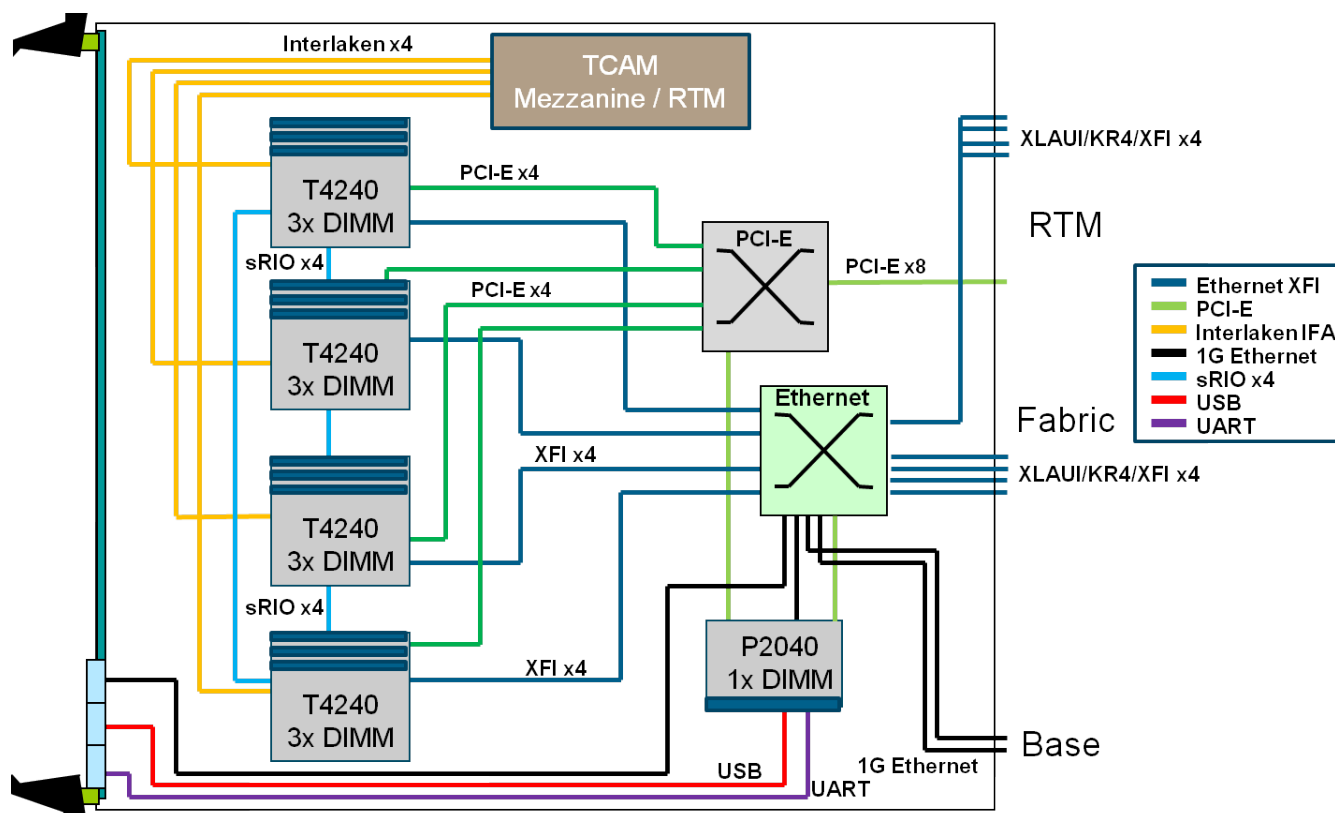


Figure 2. Network services ATCA blade

3.3 Radio node controller

Some of the more demanding packet-processing applications are found in the realm of wireless infrastructure. These systems have to interwork between wireless link layer protocols and IP networking protocols. Wireless protocol complexity is high, and includes scheduling, retransmission, and encryption with algorithms specific to cellular wireless access networks. Connecting to the IP network offers wireless infrastructure tremendous cost savings, but introduces all the security threats found in the IP world. The chip's network and peripheral interfaces provide it with the flexibility to connect to DSPs, and to wireless link layer framing ASICs/FPGAs (not shown). While the Data Path Acceleration Architecture offers encryption acceleration for both wireless and IP networking protocols, in addition to packet filtering capability on the IP networking side, multiple virtual CPUs may be dedicated to data path processing in each direction.

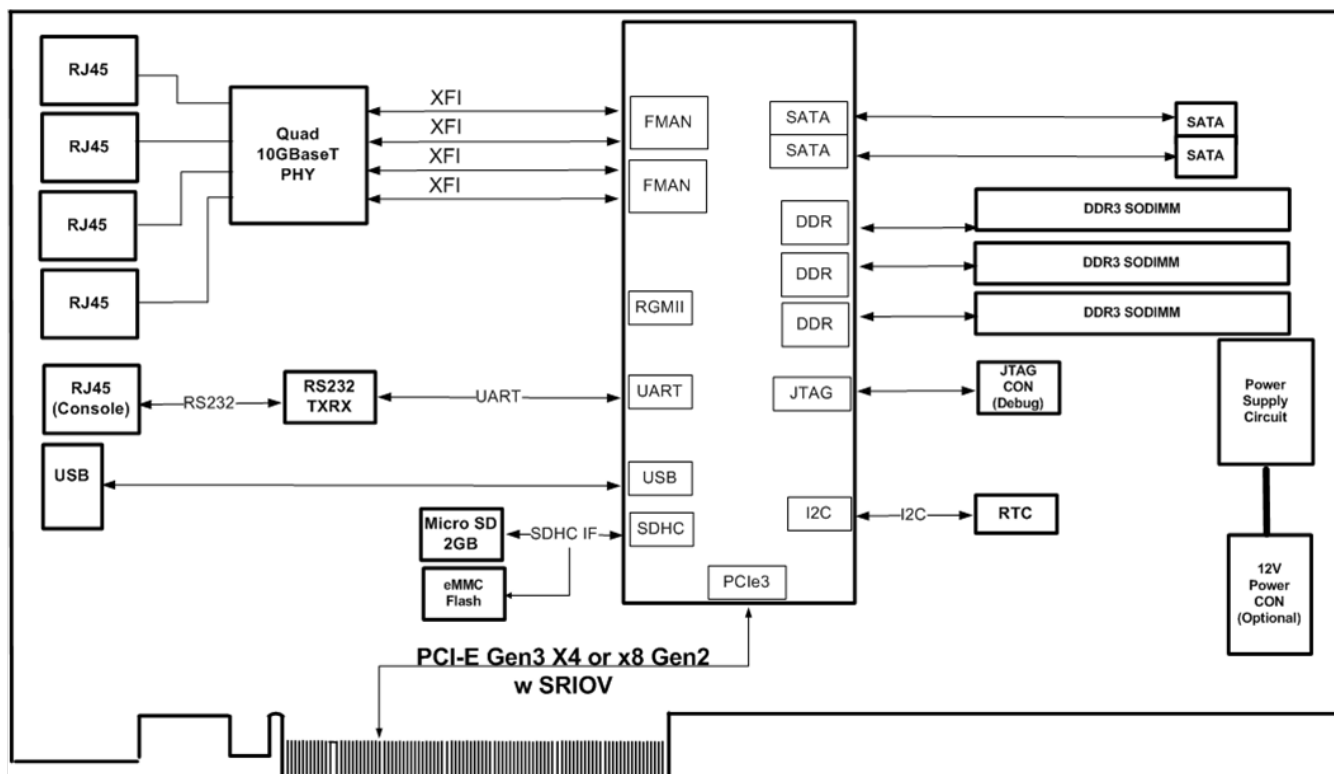


Figure 4. Intelligent network adapter

4 Multicore processing options

This flexible chip can be configured to meet many system application needs. The chip's CPUs (and hardware threads as virtual CPUs) can be combined as a fully-symmetric, multiprocessing, system-on-a-chip, or they can be operated with varying degrees of independence to perform asymmetric multiprocessing. High levels of processor independence, including the ability to independently boot and reset each core, is characteristic of the chip. The ability of the cores to run different operating systems, or run OS-less, provides the user with significant flexibility in partitioning between control, datapath, and applications processing. It also simplifies consolidation of functions previously spread across multiple discrete processors onto a single device.

While up to 24 Power Architecture threads (henceforth referred to as 'virtual CPUs', or 'vCPUs') offer a large amount of total, available computing performance, raw processing power is not enough to achieve multi-Gbps data rates in high-touch networking and telecom applications. To address this, this chip enhances the Freescale Data Path Acceleration Architecture (DPAA), further reducing data plane instructions per packet, and enabling more CPU cycles to work on value-added services as opposed to repetitive, low-level tasks. Combined with specialized accelerators for cryptography, pattern matching, and compression, the chip allows the user's software to perform complex packet processing at high data rates. There are many ways to map operating systems and I/O up to 24 chip vCPUs.

4.1 Asymmetric multiprocessing

As shown in this figure, the chip's vCPUs can be used in an asymmetric multi-processing model, with n copies of the same uni-processor OS, or n copies of OS 1, n copies of OS 2, and so on, up to 24 OS instances. The DPAA distributes work to the specific vCPUs based on basic classification or it puts work onto a common queue from which any vCPU can dequeue work.

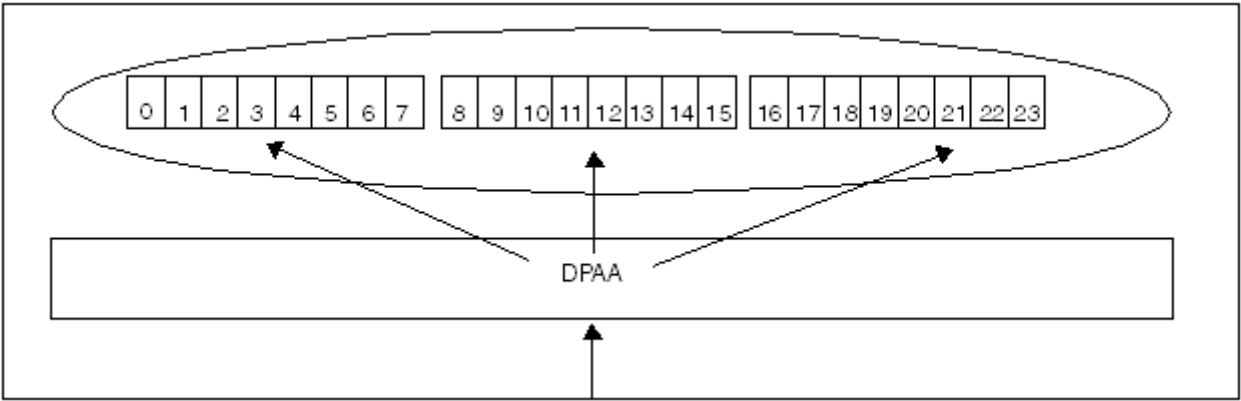


Figure 5. 24 vCPU AMP or SMP with affinity

4.2 Symmetric multiprocessing

Figure 5 also presents 24 vCPU SMP, where it is typical for data processing to involve some level of task affinity.

4.3 Mixed symmetric and asymmetric multiprocessing

This figure shows one possibility for a mixed SMP and AMP processing. Two physical CPUs (vCPUs 0-3) are combined in an SMP cluster for control processing, with the Datapath using exact match classification to send only control packets to the SMP cluster. The remaining virtual cores could run 20 instances of datapath software.

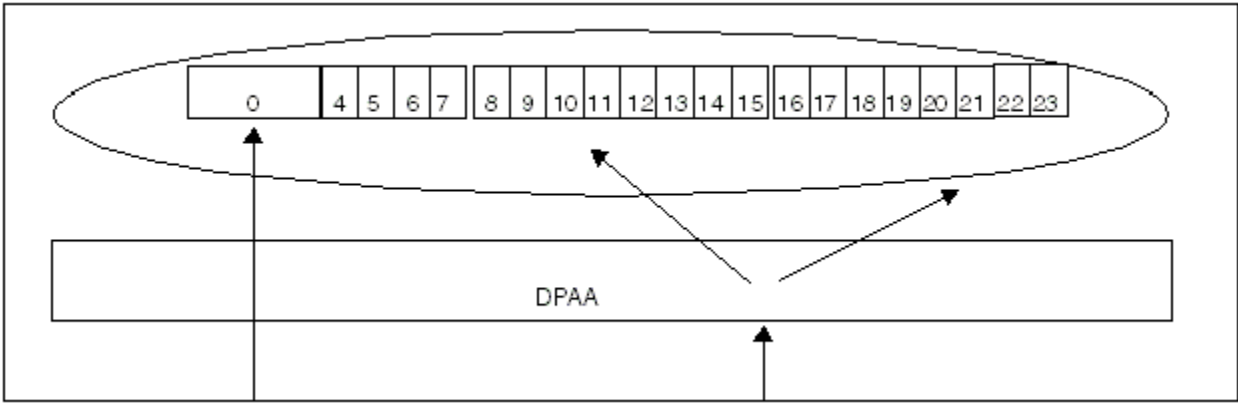


Figure 6. Mixed SMP and AMP option 1

This figure shows another possibility for mixed SMP and AMP processing. Two of the physical cores are run in single threaded mode; the remaining physical cores operate as four virtual CPUs. The Datapath directs traffic to specific software partitions based on physical Ethernet port, classification, or some combination.

- RegEx Pattern Matching Acceleration (PME 2.1) at up to 10 Gbps
- Decompression/Compression Acceleration (DCE 1.0) at up to 20 Gbps
- DPAA chip-to-chip interconnect via RapidIO Message Manager (RMAN 1.0)
- Up to 32 SerDes lanes at up to 10.3125 GHz
- Ethernet interfaces
 - Up to four 10 Gbps Ethernet XAUI or 10GBase-KR XFI MACs
 - Up to sixteen 1 Gbps Ethernet MACs
 - Up to two 1Gbps Ethernet RGMII MACs
 - Maximum configuration of 4 x 10 GE (XFI) + 10 x 1 GE (SGMII) + 2 x 1 GE (RGMII)
- High-speed peripheral interfaces
 - Up to four PCI Express 2.0 controllers, two supporting 3.0
 - Two Serial RapidIO 2.0 controllers/ports running at up to 5 GHz with Type 11 messaging and Type 9 data streaming support
 - Interlaken look-aside interface for serial TCAM connection at 6.25 and 10.3125 Gbps per-lane rates.
- Additional peripheral interfaces
 - Two serial ATA (SATA 2.0) controllers
 - Two high-speed USB 2.0 controllers with integrated PHY
 - Enhanced secure digital host controller (SD/MMC/eMMC)
 - Enhanced serial peripheral interface (eSPI)
 - Four I2C controllers
 - Four 2-pin or two 4-pin UARTs
 - Integrated Flash controller supporting NAND and NOR flash
- Three eight-channel DMA engines.
- Support for hardware virtualization and partitioning enforcement
- QorIQ Platform's Trust Architecture 2.0

5.3 Critical performance parameters

This table lists key performance indicators that define a set of values used to measure SoC operation.

Table 1. Critical performance parameters

Indicator	Values(s)
Top speed bin core frequency	1.8 GHz
Maximum memory data rate	1867 MHz (DDR3) ¹ , 1600 MHz for DDR3L <ul style="list-style-type: none"> • 1.5 V for DDR3 • 1.35 V for DDR3L
Integrated flash controller (IFC)	1.8 V
Operating junction temperature range	0-105 C
Package	1932-pin, flip-chip plastic ball grid array (FC-PBGA), 45 x 45mm

1. Conforms to JEDEC standard

5.4 Core and CPU clusters

This chip offers 12, high-performance, 64-bit Power Architecture, Book E-compliant cores. Each CPU core supports two hardware threads, which software views as a virtual CPU. The core CPUs are arranged in clusters of four with a shared 2 MB L2 cache.

Cmp features

This table shows the computing metrics the core supports.

Table 2. Power architecture metrics

Metric	Per core	Per cluster	Full device
DMIPS	10,800	43,200	129,600
Single-precision GFLOPs	18	72	Up to 216
Double-precision GFLOPs	3.6	14.4	Up to 42.4

The core subsystem includes the following features:

- Up to 1.8 GHz
- Dual-thread with simultaneous multi-threading (SMT)
 - Threading can be disabled on a per CPU basis
- 40-bit physical addressing
- L2 MMU
 - Supporting 4 KB pages
 - TLB0; 8-way set-associative, 1024-entries (4 KB pages)
 - TLB1; fully associative, 64-entry, supporting variable size pages and indirect page table entries
- Hardware page table walk
- 64-byte cache line size
- L1 caches, running at core frequency
 - 32 KB instruction, 8-way set-associative
 - 32 KB data, 8-way set-associative
 - Each with data and tag parity protection
- Hardware support for memory coherency
- Five integer units: 4 simple (2 per thread), 1 complex (integer multiply and divide)
- Two load-store units: one per thread
- Classic double-precision floating-point unit
 - Uses 32 64-bit floating-point registers (FPRs) for scalar single- and double-precision floating-point arithmetic
 - Designed to comply with IEEE Std. 754™-1985 FPU for both single and double-precision operations
- AltiVec unit
 - 128-bit Vector SIMD engine
 - 32 128-bit VR registers
 - Operates on a vector of
 - Four 32-bit integers
 - Four 32-bit single precision floating-point units
 - Eight 16-bit integers
 - Sixteen 8-bit integers
 - Powerful permute unit
 - Enhancements include: Move from GPRs to VR, sum of absolute differences operation, extended support for misaligned vectors, handling head and tails of vectors
- Supports Data Path Acceleration Architecture (DPAA) data and context "stashing" into L1 and L2 caches
- User, supervisor, and hypervisor instruction level privileges
- Addition of Elemental Barriers and "wait on reservation" instructions
- New power-saving modes including "drowsy core" with state retention and nap
 - State retention power-saving mode allows core to quickly wake up and respond to service requests
- Processor facilities
 - Hypervisor APU
 - "Decorated Storage" APU for improved statistics support
 - Provides additional atomic operations, including a "fire-and-forget" atomic update of up to two 64-bit quantities by a single access
 - Addition of Logical to Real Address translation mechanism (LRAT) to accelerate hypervisor performance
 - Expanded interrupt model

Chip features

- Boot chip-select (CS0) available after system reset, with boot block size of 8 KB, for execute-in-place boot loading from NAND Flash
- Up to terabyte Flash devices supported

5.7.2.1.2 NOR Flash features

- Data bus width of 8/16/32
- Compatible with asynchronous NOR Flash
- Directly memory mapped
- Supports address data multiplexed (ADM) NOR device
- Flexible timing control allows interfacing with proprietary NOR devices
- Boot chip-select (CS0) available at system reset

5.7.2.1.3 General-purpose chip-select machine (GPCM)

The IFC's GPCM supports the following features:

- Normal GPCM
 - Support for x8/16/32-bit device
 - Compatible with general purpose addressable device, for example, SRAM and ROM
 - External clock is supported with programmable division ratio (2, 3, 4, and so on, up to 16)
- Generic ASIC Interface
 - Support for x8/16/32-bit device
 - Address and Data are shared on I/O bus
 - Following address and data sequences are supported on I/O bus:
 - 32-bit I/O: AD
 - 16-bit I/O: AADD
 - 8-bit I/O: AAAADDDD

5.7.2.2 Serial memory controllers

In addition to the parallel NAND and NOR flash supported by the IFC, the SoC supports serial flash using eSPI, I²C and SD/MMC/eMMC card and device interfaces. The SD/MMC/eMMC controller includes a DMA engine, allowing it to move data from serial flash to external or internal memory following straightforward initiation by software.

Detailed features of the eSDHC include the following:

- Conforms to the SD Host Controller Standard Specification version 2.0, including Test event register support
- Compatible with the MMC System Specification version 4.2
- Compatible with the SD Memory Card Specification version 2.0, and supports the high capacity SD memory card
- Designed to work with SD memory, SD combo, MMC, and their variants like mini and micro.
- Card bus clock frequency up to 52 MHz
- Supports 1-/4-bit SD, 1-/4-/8-bit MMC modes
- Supports single-block and multi-block read, and write data transfer
- Supports block sizes of 1-2048 bytes
- Supports the mechanical write protect detection. In the case where write protect is enabled, the host will not initiate any write data command to the card
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports Auto CMD12 for multi-block transfer
- Host can initiate command that do not use data lines, while data transfer is in progress
- Embodies a configurable 128x32-bit FIFO for read/write data
- Supports SDMA, ADMA1, and ADMA2 capabilities

5.9.2 Serial RapidIO

The Serial RapidIO interface is based on the *RapidIO Interconnect Specification, Revision 2.1*. RapidIO is a high-performance, point-to-point, low-pin-count, packet-switched system-level interconnect that can be used in a variety of applications as an open standard. The rich feature set includes high data bandwidth, low-latency capability, and support for high-performance I/O devices as well as message-passing and software-managed programming models. Receive and transmit ports operate independently, and with 2 x 4 Serial RapidIO controllers, the aggregate theoretical bandwidth is 32 Gbps.

The chip offers two Serial RapidIO controllers, muxed onto the SerDes blocks. The Serial RapidIO interface is based on the *RapidIO Interconnect Specification, Revision 2.1*. Receive and transmit ports operate independently and with 2 x 4 Serial RapidIO controllers; the aggregate theoretical bandwidth is 32 Gbps. The Serial RapidIO controllers can be used in conjunction with "Rapid IO Message Manager (RMAN), as described in [RapidIO Message Manager \(RMan\)](#)."

Key features of the Serial RapidIO interface unit include the following:

- Support for *RapidIO Interconnect Specification, Revision 2.1* (All transaction flows and priorities.)
- 2x, and 4x LP-serial link interfaces, with transmission rates of 2.5, 3.125, or 5.0 Gbaud (data rates of 1.0, 2.0, 2.5, or 4.0 Gbps) per lane
- Auto-detection of 1x, 2x, or 4x mode operation during port initialization
- 34-bit addressing and up to 256-byte data payload
- Support for SWRITE, NWRITE, NWRITE_R and Atomic transactions
- Receiver-controlled flow control
- RapidIO error injection
- Internal LP-serial and application interface-level loopback modes

The Serial RapidIO controller also supports the following capabilities, many of which are leveraged by the RMan to efficient chip-to-chip communication through the DPAA:

- Support for RapidIO Interconnect Specification 2.1, "Part 2: Message Passing Logical Specification"
- Supports RapidIO Interconnect Specification 2.1, "Part 10: Data Streaming Logical Specification"
- Supports RapidIO Interconnect Specification 2.1, "Annex 2: Session Management Protocol"
 - Supports basic stream management flow control (XON/XOFF) using extended header message format
- Up to 16 concurrent inbound reassembly operations
 - One additional reassembly context is reservable to a specific transaction type
- Support for outbound Type 11 messaging
- Support for outbound Type 5 NWRITE and Type 6 SWRITE transactions
- Support for inbound Type 11 messaging
- Support for inbound Type 9 data streaming transactions
- Support for outbound Type 9 data streaming transactions
 - Up to 64 KB total payload
- Support for inbound Type 10 doorbell transactions
 - Transaction steering through doorbell header classification
- Support for outbound Type 10 doorbell transactions
 - Ordering can be maintained with respect to other types of traffic.
- Support for inbound and outbound port-write transactions
 - Data payloads of 4 to 64 bytes

5.9.3 SATA

Each of the SoC's two SATA controllers is compliant with the *Serial ATA 2.6 Specification*. Each of the SATA controllers has the following features:

- Supports speeds: 1.5 Gbps (first-generation SATA), and 3Gbps (second-generation SATA)
- Supports advanced technology attachment packet interface (ATAPI) devices
- Contains high-speed descriptor-based DMA controller
- Supports native command queuing (NCQ) commands

- Supports port multiplier operation
- Supports hot plug including asynchronous signal recovery

5.9.4 Interlaken Look-Aside Controller (LAC) and interface

Interlaken Look-Aside is a high speed serial channelized chip-to-chip interface. To facilitate interoperability between a GPU or NPU and a look-aside co-processor, the Interlaken Look-Aside protocol is defined for short transaction with small data & command transfers. Although based on the Interlaken protocol, Interlaken Look-Aside is not directly compatible with the Interlaken streaming specification, and can be considered a different operational mode. The SoC's Interlaken LAC is Look-Aside only.

The Interlaken LAC features:

- Supports Interlaken Look-Aside Protocol definition, Rev. 1.1
- Supports up to 32 software portals, with stashing option
- Supports inband per-channel flow control options, with a simple xon/xoff semantics
- Supports a range of SerDes frequencies (6.25 GHz to 10.3125 GHz) and widths (x4, x8)
- 64B/67B data encoding and scrambling
- Programmable BURSTMAX (256 to 512-byte) and BURSTSHORT (8 to 16 bytes)
- Error detection: illegal burst sizes, bad 64/67 word type, CRC-24 error, receiver data overflow
- Built in statistics and error counters
- Dynamic power-down of each software portal

Although not part of the DPAA, the LAC leverages DPAA concepts, including software portals and stashing. Each vCPU has a private software portal into the LAC, through which it issues commands and receives its results. Software commands to the LAC commands are translated into the Interlaken control words and data words, which are transmitted across the SerDes lanes to the co-processor, generally expected to be a TCAM.

TCAM responses received by the LAC (control words and data words) are then written to memory mapped space defined for the software portal of the vCPU that initiated the request. These writes can be configured to stash data directly into the vCPU's cache to reduce latency.

Each vCPU can generally have four outstanding transactions with the LAC; however, if not all vCPUs are configured to use the LAC, those that are configured can have more outstanding transactions. Order is maintained for all transactions issued by a single portal.

5.10 Data Path Acceleration Architecture (DPAA)

This chip includes an enhanced implementation of the QorIQ Datapath Acceleration Architecture (DPAA). This architecture provides the infrastructure to support simplified sharing of networking interfaces and accelerators by multiple CPUs. These resources are abstracted as enqueue/dequeue operations by CPU 'portals' into the datapath. Beyond enabling multicore sharing of resources, the DPAA significantly reduces software overheads associated with high-touch packet-processing operations.

Examples of the types of packet-processing services that this architecture is optimized to support are as follows:

- Traditional routing and bridging
- Firewall
- Security protocol encapsulation and encryption

The functions off-loaded by the DPAA fall into two broad categories:

- Packet distribution and queue-congestion management
- Accelerating content processing

5.10.1 Packet distribution and queue/congestion management

This table lists some packet distribution and queue/congestion management offload functions.

Table 3. Offload functions

Function type	Definition
Data buffer management	Supports allocation and deallocation of buffers belonging to pools originally created by software with configurable depletion thresholds. Implemented in a module called the Buffer Manager (BMan).
Queue management	Supports queuing and quality-of-service scheduling of frames to CPUs, network interfaces and DPAA logic blocks, maintains packet ordering within flows. Implemented in a module called the Queue Manager (QMan). The QMan, besides providing flow-level queuing, is also responsible for congestion management functions such as RED/WRED, congestion notifications and tail discards.
Packet distribution	Supports in-line packet parsing and general classification to enable policing and QoS-based packet distribution to the CPUs for further processing of the packets. This function is implemented in the block called the Frame Manager (FMan).
Policing	Supports in-line rate-limiting by means of two-rate, three-color marking (RFC 2698). Up to 256 policing profiles are supported. This function is also implemented in the FMan.
Egress Scheduling	Supports hierarchical scheduling and shaping, with committed and excess rates. This function is supported in the QMan, although the FMan performs the actual transmissions.

5.10.2 Accelerating content processing

Properly implemented acceleration logic can provide significant performance advantages over most optimized software with acceleration factors on the order of 10-100x. Accelerators in this category typically touch most of the bytes of a packet (not just headers). To avoid consuming CPU cycles in order to move data to the accelerators, these engines include well-pipelined DMAs. This table lists some specific content-processing accelerators on the chip.

Table 4. Content-processing accelerators

Interface	Definition
SEC	Crypto-acceleration for protocols such as IPsec, SSL, and 3GPP RLC
PME	Regex style pattern matching for unanchored searches, including cross-packet stateful patterns
DCE	Compression/Decompression acceleration for ZLib and deflate

5.10.3 Enhancements of T4240 compared to first generation DPAA

A short summary of T4240 enhancements over the first generation DPAA (as implemented in the P4080) is provided below:

- Frame Manager
 - 2x performance increase (up to 25 Gbps per FMan)
 - Storage profiles.
 - HiGig (3.125 GHz) and HiGig2 (3.125 GHz and 3.75 GHz)
 - Energy Efficient Ethernet
- SEC 5.0
 - 2x performance increase for symmetric encryption and protocol processing

- Up to 20 Gbps for IPsec @ Imix
 - 10x performance increase for public key algorithms
 - Support for 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3 (ZUC)
- DCE 1.0, new accelerator for compression/decompression
- RMan (Serial RapidIO Manager)
- DPAA overall capabilities
 - Data Center Bridging
 - Egress Traffic Shaping

5.10.4 DPAA terms and definitions

The QorIQ Platform's Data Path Acceleration Architecture (henceforth DPAA) assumes the existence of network flows, where a flow is defined as a series of network datagrams, which have the same processing and ordering requirements. The DPAA prescribes data structures to be initialized for each flow. These data structures define how the datagrams associated with that flow move through the DPAA. Software is provided a consistent interface (the software portal) for interacting with hardware accelerators and network interfaces.

All DPAA entities produce data onto frame queues (a process called enqueueing) and consume data from frame queues (dequeuing). Software enqueues and dequeues through a software portal (each vCPU has two software portals), and the FMan, RMan, and DPAA accelerators enqueue/dequeue through hardware portals. This figure illustrates this key DPAA concept.

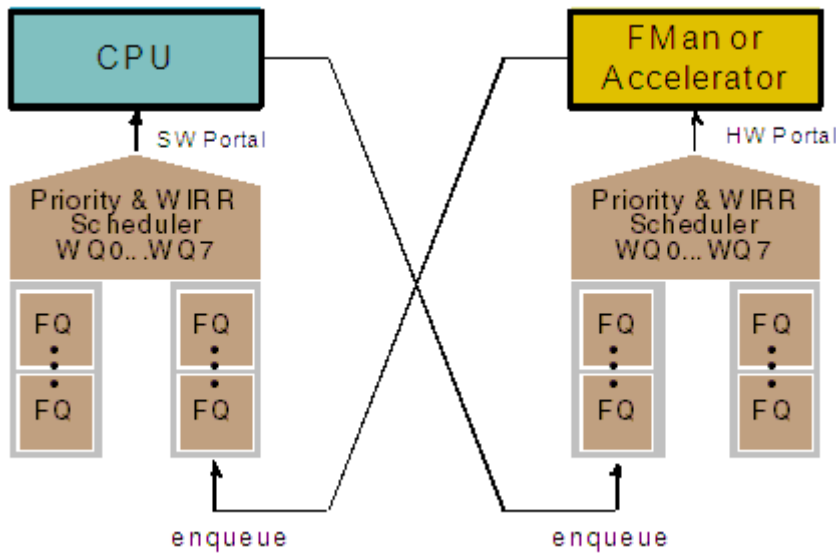


Figure 9. DPAA enqueueing and dequeuing

This table lists common DPAA terms and their definitions.

Table 5. DPAA terms and definitions

Term	Definition	Graphic representation
Buffer	Region of contiguous memory, allocated by software, managed by the DPAA BMan	

Table continues on the next page...

Table 5. DPAA terms and definitions (continued)

Term	Definition	Graphic representation
Buffer pool	Set of buffers with common characteristics (mainly size, alignment, access control)	
Frame	Single buffer or list of buffers that hold data, for example, packet payload, header, and other control information	
Frame queue (FQ)	FIFO of frames	
Work queue (WQ)	FIFO of FQs	
Channel	Set of eight WQs with hardware provided prioritized access	
Dedicated channel	Channel statically assigned to a particular end point, from which that end point can dequeue frames. End point may be a CPU, FMan, PME,DCE,RMan or SEC.	-
Pool channel	A channel statically assigned to a group of end points, from which any of the end points may dequeue frames.	

5.10.5 Major DPAA components

The SoC's Datapath Acceleration Architecture, shown in the figure below, includes the following major components:

- Frame Manager (FMan)
- Queue Manager (QMan)
- Buffer Manager (BMan)
- RapidIO Message Manager (RMan 1.0)
- Security Engine (SEC 5.0)
- Pattern Matching Engine (PME 2.1)
- Decompression and Compression Engine (DCE 1.0)

The QMan and BMan are infrastructure components, which are used by both software and hardware for queuing and memory allocation/deallocation. The Frame Managers and RMan are interfaces between the external world and the DPAA. These components receive datagrams via Ethernet or Serial RapidIO and queue them to other DPAA entities, as well as dequeue datagrams from other DPAA entities for transmission. The SEC, PME, and DCE are content accelerators that dequeue processing requests (typically from software) and enqueue results to the configured next consumer. Each component is described in more detail in the following sections.

The SEC 5.0 can perform full protocol processing for the following security protocols:

- IPsec
- SSL/TLS
- 3GPP RLC encryption/decryption
- LTE PDCP
- SRTP
- IEEE 802.1AE MACSec
- IEEE 802.16e WiMax MAC layer

The SEC 5.0 supports the following algorithms, modes, and key lengths as raw modes, or in combination with the security protocol processing described above.

- Public Key Hardware Accelerators (PKHA)
 - RSA and Diffie-Hellman (to 4096b)
 - Elliptic curve cryptography (1023b)
- Data Encryption Standard Accelerators (DESA)
 - DES, 3DES (2-key, 3-key)
 - ECB, CBC, OFB, and CFB modes
- Advanced Encryption Standard Accelerators (AESA)
 - Key lengths of 128-bit, 192-bit, and 256-bit
 - ECB, CBC, CTR, CCM, GCM, CMAC, OFB, CFB, xcbc-mac, and XTS
- ARC Four Hardware Accelerators (AFHA)
 - Compatible with RC4 algorithm
- Message Digest Hardware Accelerators (MDHA)
 - SHA-1, SHA-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 (SNOWf8 and SNOWf9)
 - Implements Snow 3.0, F8 and F9 modes
- ZUC Hardware Accelerators (ZUCE and ZUCA)
 - Implements 128-EEA3 & 128-EIA3
- CRC Unit
 - Standard and user-defined polynomials
- Random Number Generator
 - Incorporates TRNG entropy generator for seeding and deterministic engine (SHA-256)
 - Supports random IV generation

The SEC 5.0 is designed to support bulk encryption at up to 40 Gbps, large packet/record IPsec/SSL at up to 30 Gbps, and 20 Gbps for IPsec ESP at Imix packet sizes. 3G and LTE algorithms are supported at 10 Gbps or more.

The SEC dequeues data from its QMan hardware portal and, based on FQ configuration, also dequeues associated instructions and operands in the Shared Descriptor. The SEC processes the data then enqueues it to the configured output FQ. The SEC uses the Status/CMD word in the output Frame Descriptor to inform the next consumer of any errors encountered during processing (for example, received packet outside the anti-replay window.)

- Ability to match patterns across data "work units" or packet boundaries
 - Can be used to correlate patterns, qualify matches (for example, contextual match), or to track protocol state change
- Easily support "greedy" wildcards
 - For example, ABC.*DEF == two patterns tied together by a stateful rule
- Delays the need for software post-processing. Software is alerted after all byte patterns are detected in the proper sequence, rather than any time a byte pattern is detected.
- Implements a significant subset of the regex pattern definition syntax as well as many constructs which cannot be expressed in standard PCRE
- PME 2.1 supports up to 32K stateful rules, linking multiple byte patterns

The PME 2.1 dequeues data from its QMan hardware portal and, based on FQ configuration, scans the data against one of 256 pattern sets, 16 subsets per pattern set.

When the PME finds a byte pattern match, or a final pattern in a stateful rule, it generates a report.

5.10.5.6 Decompression and Compression Engine (DCE 1.0)

The Decompression and Compression Engine (DCE 1.0) is an accelerator compatible with Datapath Architecture providing lossless data decompression and compression for the QorIQ family of SoCs. The DCE supports the raw DEFLATE algorithm (RFC1951), GZIP format (RFC1952) and ZLIB format (RFC1950). The DCE also supports Base 64 encoding and decoding (RFC4648).

The DEFLATE algorithm is a basic building block for data compression in most modern communication systems. It is used by HTTP to compress web pages, by SSL to compress records, by gzip to compress files and email attachments, and by many other applications.

Deflate involves searching for repeated patterns previously seen in a Frame, computing the length and the distance of the pattern with respect to the current location in the Frame, and encoding the resulting information into a bitstream.

The decompression algorithm involves decoding the bitstream and replaying past data. The Decompression and Compression Engine is architected to minimize the system memory bandwidth required to do decompression and compression of Frames while providing multi-gigabits per second of performance.

Detailed features include the following:

- 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
- Compression
 - ZLIB, GZIP and DEFLATE header insertion
 - ZLIB and GZIP CRC computation and insertion
 - Zlib sync flush and partial flush for chunked compression (for example, for HTTP1.1)
 - Four modes of compression
 - No compression (just add DEFLATE header)
 - Encode only using static/dynamic Huffman codes
 - Compress and encode using static Huffman codes
 - Compress and encode using dynamic Huffman codes
 - Uses a 4KB sliding history window
 - Supports Base 64 encoding (RFC4648) after compression
 - Provides at least 2.5:1 compression ratio on the Calgary Corpus
- Decompression supports:
 - ZLIB, GZIP and DEFLATE header removal
 - ZLIB and GZIP CRC validation
 - 32KB history
 - Zlib flush for chunked decompression (for HTTP1.1 for example)

cmp features

- All standard modes of decompression
- No compression
- Static Huffman codes
- Dynamic Huffman codes
- Provides option to return original compressed Frame along with the uncompressed Frame or release the buffers to BMan
- Does not support use of ZLIB preset dictionaries (FDICT flag = 1 is treated as an error).
- Base 64 decoding (RFC4648) prior to decompression

The DCE 1.0 is designed to support up to 8.8 Gbps for either compression or decompression, or 17.5 Gbps aggregate at ~4 KB data sizes.

5.10.6 DPAA capabilities

Some DPAA features and capabilities have been described in the sections covering individual DPAA components. This section describes some capabilities enabled by DPAA components working together.

5.10.6.1 Ingress policing and congestion management

In addition to selecting FQ ID and storage profile, classification can determine whether policing is required for a received packet, along with the specific policing context to be used.

FMan policing capabilities include the following:

- RFC2698: two-rate, three-color marking algorithm
- RFC4115: Differentiated service two-rate, three-color marker with efficient handling of in-profile traffic
- Up to 256 internal profiles

The sustained and peak rates, and burst size for each policing profile are user-configurable.

5.10.6.2 Customer-edge egress-traffic management (CEETM)

Customer-edge egress-traffic management (CEETM) is a DPAA enhancement first appearing in the T4240. T4240 continues to support the work queue and frame queue scheduling functionality available in the P4080 and other first generation QorIQ chips, but introduces alternative functionality, CEETM, that can be mode selected on a network interface basis to support the shaping and scheduling requirements of carrier Ethernet connected systems.

5.10.6.2.1 CEETM features

Each instance of CEETM (one per FMan) provides the following features:

- Supports hierarchical multi-level scheduling and shaping, which:
 - is performed in an atomic manner; all context at all levels is examined and updated synchronously.
 - employs no intermediate buffering between class queues and the direct connect portal to the FMan.
- Supports dual-rate shaping (paired committed rate (CR) shaper and excess rate (ER) shaper) at all shaping points.
 - Shapers are token bucket based with configurable rate and burst limit.
 - Paired CR/ER shapers may be configured as independent or coupled on a per pair basis; coupled means that credits to the CR shaper in excess of its token bucket limit is credited to the ER bucket
- Supports eight logical network interfaces (LNI)
 - Each LNI:
 - aggregates frames from one or more channels.
 - priority schedules unshaped frames (aggregated from unshaped channels), CR frames, and ER frames (aggregated from shaped channels)

- applies a dual-rate shaper to the aggregate of CR/ER frames from shaped channels
- can be configured (or reconfigured for lossless interface failover) to deliver frames to any network interface.
- Supports 32 channels available for allocation across the eight LNIs
- Each channel:
 - can be configured to deliver frames to any LNI.
 - can be configured to be unshaped or shaped; when shaped, a dual rate shaper applies to the aggregate of CR/ER frames from the channel.
 - has eight independent classes and eight grouped classes; grouped classes can be configured as one class group of eight or as two class groups of four.
 - supports weighted bandwidth fairness within grouped class groups with weights configured on a channel and class basis.
 - strict priority scheduling of the eight independent classes and the aggregate(s) of the grouped class(es); the priority of each of the two class groups can be independently configured to be immediately below any of the independent classes.
 - is configurable such that each of the eight independent classes and two class groups can supply CR frames, ER frames or both when channel is configured to be shaped.
 - is configured independently.
- Each class:
 - has a dedicated class queue (CQ) with equivalent congestion management functionality available to FQs.
 - can have a dedicated or shared Congestion Management Record supports sufficient number of CMRs for all CQs to have a dedicated CMR, if desired.
 - can be flow-controlled by traffic-class flow control messages via portal; achieves backward compatibility with by allowing each of these 16 classes to be configured (per LNI) to respect one or none of the 8 on/off control bits within existing message format (as was defined for 8-class non-CEETM channels).
 - is identified via a "logical frame queue identifier" to maintain semantic compatibility with enqueue commands to frame queues (non-CEETM queues).
 - supports the identification of intra-class flows (logically equivalent to FQs but not queued separately) in order to apply static context (Context_A and Context_B) to frames as they are dequeued from CQs; this provides functionality equivalent to that available when a frame is dequeued from a frame queue (non-CEETM queues).

5.10.6.2.2 CEETM configuration

The CEETM configuration, shown in [Figure 13](#), is very asymmetrical and is intended to demonstrate the degrees of configurability rather than an envisioned use case.

NOTE

The color green denotes logic units and signal paths that relate to the request and fulfillment of committed rate (CR) packet transmission opportunities. The color yellow denotes the same for excess rate (ER). The color black denotes logic units and signal paths that are used for unshaped opportunities or that operate consistently whether used for CR or ER opportunities.

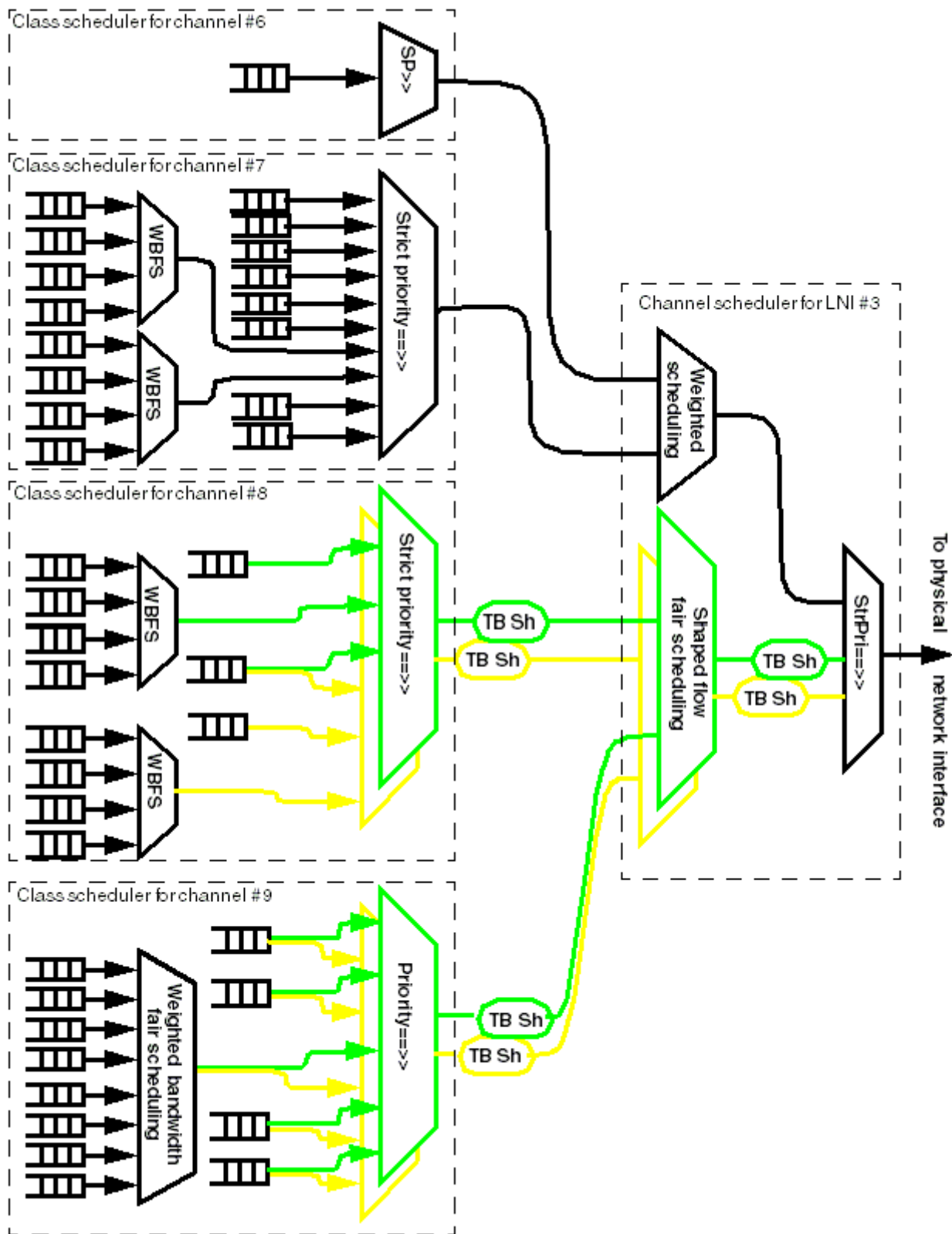


Figure 13. CEETM scheduler: illustrative configuration scenario

Figure 13 illustrates the following scenario:

5.12.2 Non-transparent power management

Many load-based power savings are use-case specific static configurations (thereby software transparent), and were described in the previous section. This section focuses on SoC power management mechanisms, which software can dynamically leverage to reduce power when the system is lightly loaded. The most important of these mechanisms involves the cores.

A full description of core low-power states with proper names is provided in the SoC reference manual. At a high level, the most important of these states can be viewed as "PH10" and "PH20," described as follows. Note that these are relative terms, which do not perfectly correlate to previous uses of these terms in Power Architecture and other ISAs:

- In PH10 state CPU stops instruction fetches but still performs L1 snoops. The CPU retains all state, and instruction fetching can be restarted instantly.
- In PH20 state CPU stops instruction fetches and L1 snooping, and turns off all clocks. Supply voltage is reduced, using a technique Freescale calls State Retention Power Gating (SRPG). In the "napping" state, a CPU uses ~75% less power than a fully operational CPU, but can still return to full operation quickly (~100 platform clocks).

The core offers two ways to enter these (and other) low power states: registers and instructions.

As the name implies, register-based power management means that software writes to registers to select the CPU and its low power state. Any CPU with write access to power management registers can put itself, or another CPU, into a low power state; however, a CPU put into a low power state by way of register write cannot wake itself up.

Instruction-based power management means that software executes special WAIT instruction to enter a low power state. CPUs exit the low power state in response to external triggers, interrupts, doorbells, stashes into L1-D cache, or clear reservation on snoop. Each vCPU can independently execute WAIT instructions; however, the physical CPU enters PH20 state after the second vCPU executes its wait. The instruction-based "enters PH20 state" state is particularly well-suited for use in conjunction with Freescale's patented Cascade Power Management, which is described in the next section.

While significant power savings can be achieved through individual CPU low power states, the SoC also supports a register-based cluster level low power state. After software puts all CPUs in a cluster in a PH10 state, it can additionally flush the L2 cache and have the entire cluster enter PH20 state. Because the L2 arrays have relatively low static power dissipation, this state provides incremental additional savings over having four napping CPUs with the L2 on.

5.12.3 Cascade power management

Cascade power management refers to the concept of allowing SoC load, as defined by the depth of queues managed by the Queue Manager, to determine how many vCPUs need to be awake to handle the load. Recall from [Queue Manager](#) that the QMan supports both dedicated and pool channels. Pool channels are channels of frame queues consumed by parallel workers (vCPUs), where any worker can process any packet dequeued from the channel.

Cascade Power Management exploits the QMan's awareness of vCPU membership in a pool channel and overall pool channel queue depth. The QMan uses this information to tell vCPUs in a pool channel (starting with the highest numbered vCPU) that they can execute instructions to "take a nap." When pool channel queue depth exceeds configurable thresholds, the QMan wakes up the lowest numbered vCPU.

The SoC's dynamic power management capabilities, whether using the Cascade scheme or a master control CPU and load to power matching software, enable up to a 75% reduction to each core in power consumption versus data sheet max power.

Table A-1. Differences between T4240 and T4160 (continued)

Feature	T4240	T4160
Max number of Anyspeed MACs configured for 10 GE operation	2 per Frame Manager	1 per Frame Manager
SerDes and pinout		
Total number of SerDes lanes	4 x 8	2 x 4 and 2 x 8
High-speed IO		
PCIe	4	3 (PCIe 3 is disabled)

Appendix B T4080

B.1 Introduction

The T4080 is a low power version of the T4160. The T4080 has four dual threaded Power Architecture e6500 cores with the same two memory complexes (CoreNet platform cache and DDR3 memory controller) with the same high-performance datapath acceleration, networking, and peripheral bus interfaces.

This figure shows the major functional units within the chip.

Table C-1. Revision history

Rev. number	Date	Substantive change(s)
1	10/2014	<ul style="list-style-type: none"> Added support for T4080 throughout document. Updated Introduction. In Summary of benefits, updated the first sentence to include "...SDN switches or controllers, network function virtualization..." and added the following subsections: <ul style="list-style-type: none"> e6500 CPU core Virtualization Data Path Acceleration Architecture (DPAA) System peripherals and networking In Intelligent network adapter, added examples. Updated Block diagram. In Features summary, added T4160 and T4080 thread specifications, added 10GBase-KR to the Ethernet interfaces, updated the coherent read bandwidth, and removed the note. In Critical performance parameters, removed the typical power consumption table. In Core and CPU clusters, updated the 16 way, set associative sub-bullets and changed the double-precision, full device value from "42.2" to "up to 42.4". Updated the read bandwidth in CoreNet fabric and address map. Added HiGig 2 in Enhancements of T4240 compared to first generation DPAA. Updated bullet two in CoreNet fabric and address map and updated the last bullet in High-speed peripheral interface complex (HSSI). Updated Non-transparent power management. Rewrote Conclusion to add more information and a list of Freescale resources. In the Appendix A T4160 Introduction, removed the T4240-specific information.
0	06/2013	Initial public release.