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	-
Number of Cores/Bus Width	-
Speed	-
Co-Processors/DSP	-
RAM Controllers	-
Graphics Acceleration	-
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	-
Operating Temperature	-
Security Features	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=t4160nxn7pqb

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 (phyiscal and virtual), hardware accelerators, and network interfaces.

The Frame Manager (FMAN), a primary element of the DPAA, parses headers from incoming packets and classifies and selects data buffers with optional policing and congestion management. The FMAN passes its work to the Queue Manager (QMAN), which assigns it to cores or accelerators with a multilevel scheduling hierarchy. The T4240 processor's implementation of the DPAA offers accelerations for cryptography, enhanced regular expression pattern matching, and compression/decompression.

2.4 System peripherals and networking

For networking, there are dual FMANs with an aggregate of up to 16 any-speed MAC controllers that connect to PHYs, switches, and backplanes over RGMII, SGMII, QSGMII, HiGig2, XAUI, XFI, and 10Gbase-KR. The FMAN also supports new quality of service features through egress traffic shaping and priority flow control for data center bridging in converged data center networking applications. High-speed system expansion is supported through four PCI Express controllers that support varieties of lane lengths for PCIe specification 3.0, including endpoint SR-IOV with 128 virtual functions. Other peripherals include:

- SRIO
- Interlaken-LA
- SATA
- SD/MMC
- I^2C
- UART
- SPI
- NOR/NAND controller
- GPIO
- 1866 MT/s DDR3/L controller

3 Application examples

This chip is well-suited for applications that are highly compute-intensive, I/O-intensive, or both.

3.1 1U security appliance

This figure shows a 1U security appliance built around a single SoC. The QorIQ DPAA accelerates basic packet classification, filtering, and packet queuing, while the crypto accelerator (SEC 5.0), regex accelerator (PME 2.1), and compression/decompression accelerator (DCE 1.0) perform high throughput content processing. The high single threaded and aggregate DMIPS of the core CPUs provide the processing horsepower for complex classification and flow state tracking required for proxying applications as well as heuristic traffic analysis and policy enforcement.

The SoC's massive integration significantly reduces system BOM cost. SATA hard drives connect directly to the SoC's integrated controllers, and an Ethernet switch is only required if more than 16 1 GE ports or 4 10 GE ports are required. The SoC supports PCIe and Serial RapidIO for expansion.

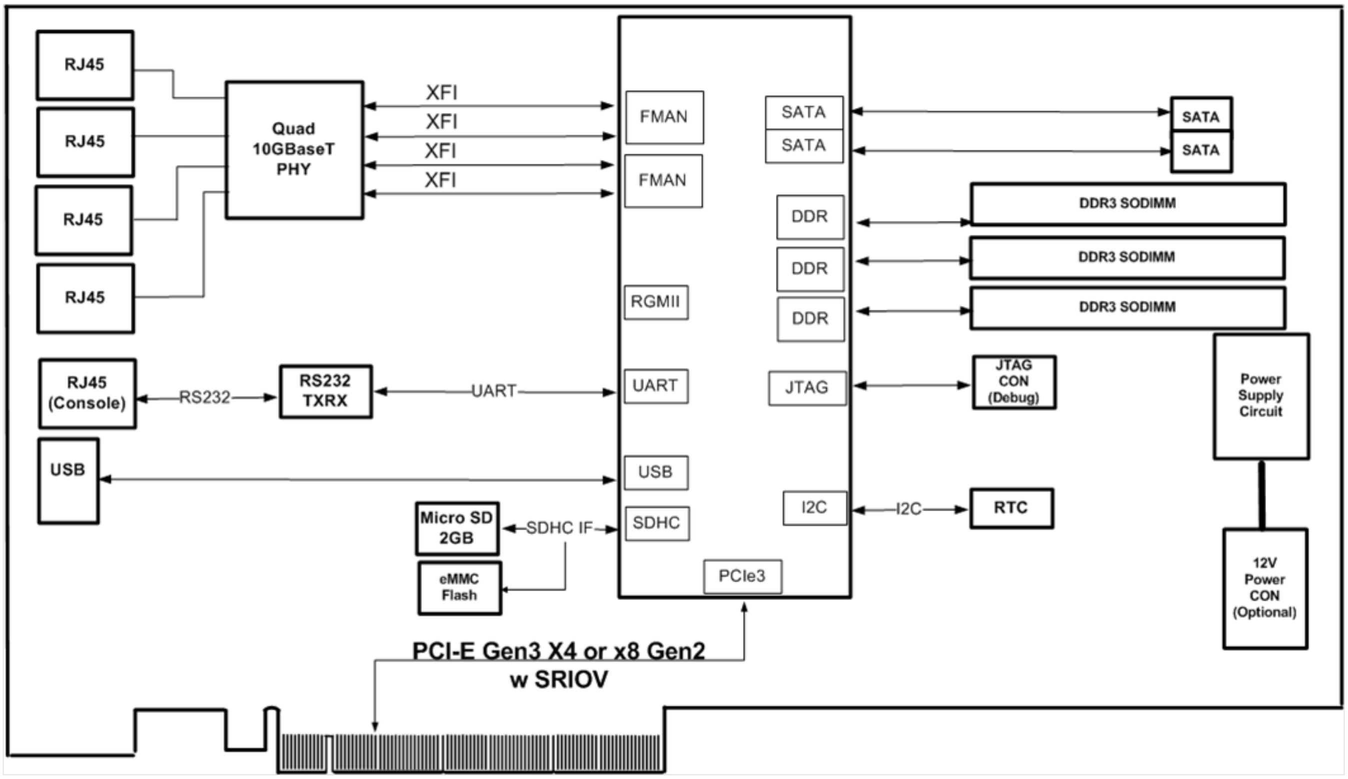


Figure 4. Intelligent network adapter

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.

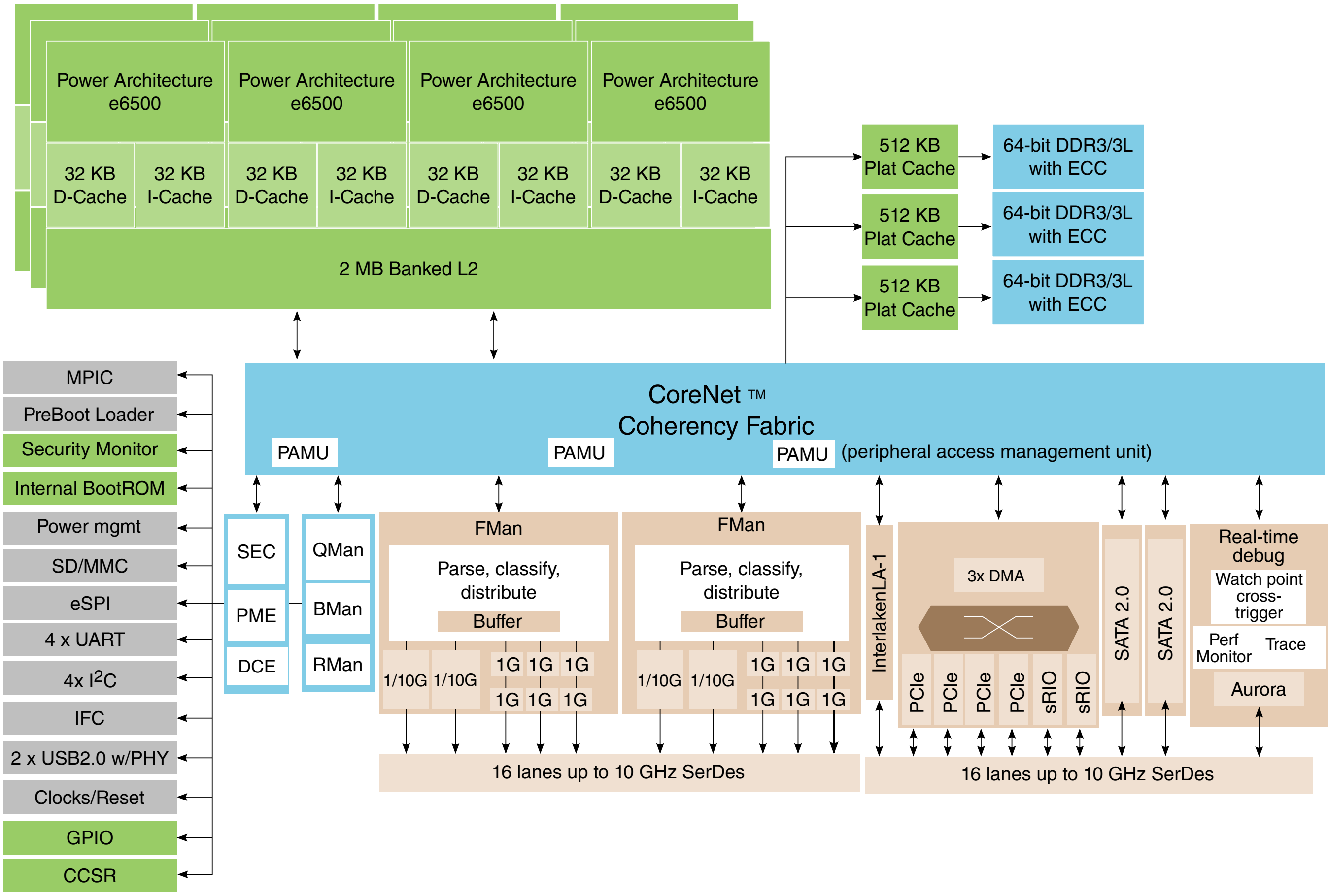


Figure 8. T4240 block diagram

5.2 Features summary

This chip includes the following functions and features:

- 12, dual-threaded e6500 cores for a total of 24/16/8 threads (T4240/T4160/T4080) built on Power Architecture® technology
    - Arranged as three clusters of four cores sharing a 2 MB L2 cache, 6 MB L2 cache total.
    - Up to 1.8 GHz with 64-bit ISA support (Power Architecture v2.06-compliant)
    - Three privilege levels of instruction: user, supervisor, and hypervisor
- Up to 1.5 MB CoreNet Platform Cache (CPC)
- Hierarchical interconnect fabric
    - CoreNet fabric supporting coherent and non-coherent transactions with prioritization and bandwidth allocation amongst CoreNet end-points
    - 1.46 Tbps coherent read bandwidth
- Up to three 64-bit DDR3/3L SDRAM memory controllers with ECC and interleaving support
    - Up to 1.867 GT/s data transfer rate
    - 64 GB per DDR controller
- Data Path Acceleration Architecture (DPAA) incorporating acceleration for the following functions:
    - Packet parsing, classification, and distribution (Frame Manager 1.1) up to 50 Gbps
    - Queue management for scheduling, packet sequencing, and congestion management (Queue Manager 1.1)
    - Queue Manager (QMan) fabric supporting packet-level queue management and quality of service scheduling
    - Hardware buffer management for buffer allocation and de-allocation (BMan 1.1)
    - Cryptography acceleration (SEC 5.0) at up to 40 Gbps

- 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 • 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.

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

        - Improved Programmable Interrupt Controller (PIC) automatically ACKs interrupts
        - Implements message send and receive functions for interprocessor communication, including receive filtering
    - External PID load and store facility
        - Provides system software with an efficient means to move data and perform cache operations between two disjoint address spaces
        - Eliminates the need to copy data from a source context into a kernel context, change to destination address space, then copy the data to the destination address space or alternatively to map the user space into the kernel address space

Details of the banked L2 are provided below.

- 2 MB cache with ECC protection (data, tag, & status)
    - Pipelined data array access with 2 cycle repeat rate
- 4 banks, supporting up to four concurrent accesses.
- 64-byte cache line size
- 16 way, set associative
    - Ways in each bank can be configured in one of several modes
    - Flexible way partitioning per vCPU
        - I-only, D-only, or unified
- Supports direct stashing of datapath architecture data into L2

The chip also contains up to 1.5 MB of shared L3 CoreNet Platform Cache (CPC), with the following features:

- Total 1.5 MB, implemented as three 512 KB arrays, one per DDR controller
    - ECC protection for Data, Tag and Status
    - 16-way set associative with configurable replacement algorithms
    - Allocation control for data read, data store, castout, decorated read, decorated store, instruction read and stash
    - Configurable SRAM partitioning

5.5 Inverted cache hierarchy

From the perspective of software running on an core vCPU, the SoC incorporates a 2.5-level cache hierarchy. These levels are as follows:

- Level 1: Individual core 32 KB Instruction and Data caches
- Level 2: Locally banked 2 MB cache (configurably shared by other vCPUs in the cluster)
- Level 2.5: Remote banked 2 MB caches (total 4 MB)

When vCPUs in different physical clusters are part of the same coherency domain, the CoreNet Coherency Fabric causes any cache miss in the vCPU's local L2 to be snooped by the remote L2s belonging to the other clusters. On a hit in a remote L2, the associated data is returned directly to the requesting vCPU, eliminating the need for a higher latency flush and retry protocol. This direct cache transfer is called cache intervention.

Previous generation QorIQ products also support cache intervention from their private backside L2 caches; however, the SoC's allocation policies make greater use of intervention. The sum of the SoC's L2 caches are 3x larger than the CPC. Ttherefore, the CPC is not intended to act as backing store for the L2s, as it typically is in the previous generation. This allows the CPCs to be dedicated to the non-CPU masters in the SoC, storing DPAA data structures and IO data that the CPUs and accelerators will most likely need.

Although the SoC supports allocation policies that would result in CPU instructions and in data being held in the CPC (CPC acting as vCPU L3), this is not the default. Because the CPC serves fewer masters, it serves those masters better, by reducing the DDR bandwidth consumed by the DPAA and improving the average latency.

- 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, I^2C 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

- 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 interoperablility 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.

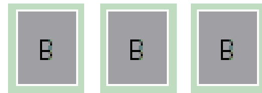
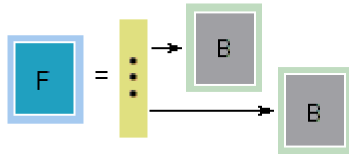
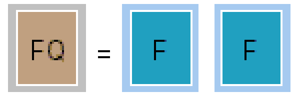
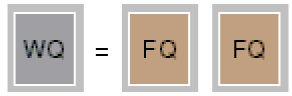
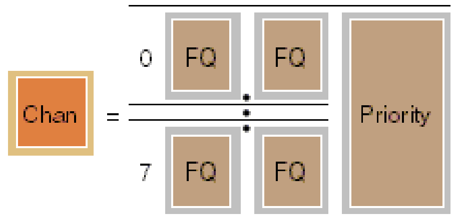
Examples of the types of packet-processing services that this architecture is optimized to support are as follows:

- 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

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.

This figure is a logical view of the DPAA.

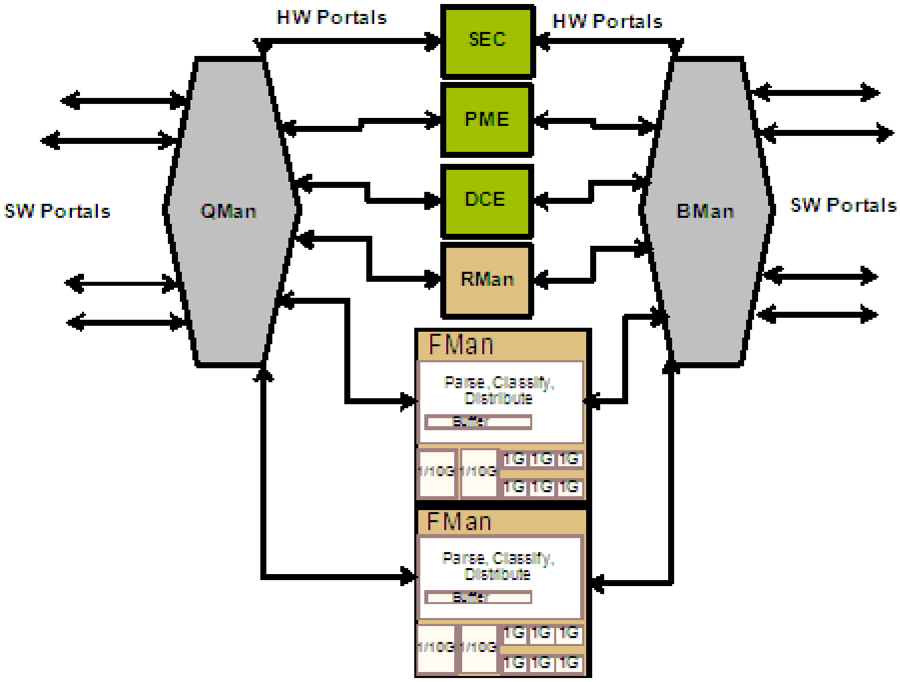


Figure 10. Logical representation of DPAA

5.10.5.1 Frame Manager and network interfaces

The chip incorporates two enhanced Frame Managers. The Frame Manager improves on the bandwidth and functionality offered in the P4080.

Each Frame Manager, or FMan, combines Ethernet MACs with packet parsing and classification logic to provide intelligent distribution and queuing decisions for incoming traffic. Each FMan supports PCD at 37.2 Mpps, supporting line rate 2x10G + 2x2.5G at minimum frame size.

These Ethernet combinations are supported:

- 10 Gbps Ethernet MACs are supported with XAUI (four lanes at 3.125 GHz) or XFI (one lane at 10.3125 GHz SerDes).
- 1 Gbps Ethernet MACs are supported with SGMII (one lane at 1.25 GHz with 3.125 GHz option for 2.5 Gbps Ethernet).
  - SGMIIs can be run at 3.125 GHz so long as the total Ethernet bandwidth does not exceed 25 Gbps on the associated FMan.
  - If not already assigned to SGMII, two MACs can be used with RGMII.
- Four x1Gbps Ethernet MACs can be supported using a single lane at 5 GHz (QSGMII).
- HiGig is supported using four lanes at 3.125 GHz or 3.75 GHz (HiGig2).

The Frame Manager's Ethernet functionality also supports the following:

- 1588v2 hardware timestamping mechanism in conjunction with IEEE Std. 802.3bf (Ethernet support for time synchronization protocol)
- Energy Efficient Ethernet (IEEE Std. 802.3az)
- IEEE Std. 802.3bd (MAC control frame support for priority based flow control)
- IEEE Std. 802.1Qbb (Priority-based flow control) for up to eight queues/priorities
- IEEE Std. 802.1Qaz (Enhanced transmission selection) for three or more traffic classes

5.10.5.1.1 Receiver functionality: parsing, classification, and distribution

Each Frame Manager matches its 25 Gbps Ethernet connectivity with 25 Gbps (37.2 Mpps) of Parsing, Classification, and Distribution (PCD) performance. PCD is the process by which the Frame Manager identifies the frame queue on which received packets should be enqueued. The consumer of the data on the frame queues is determined by Queue Manager configuration; however, these activities are closely linked and managed by the FMan Driver and FMan Configuration Tool, as in previous QorIQ SoCs.

This figure provides a logical view of the FMan's processing flow, illustrating the PCD features.

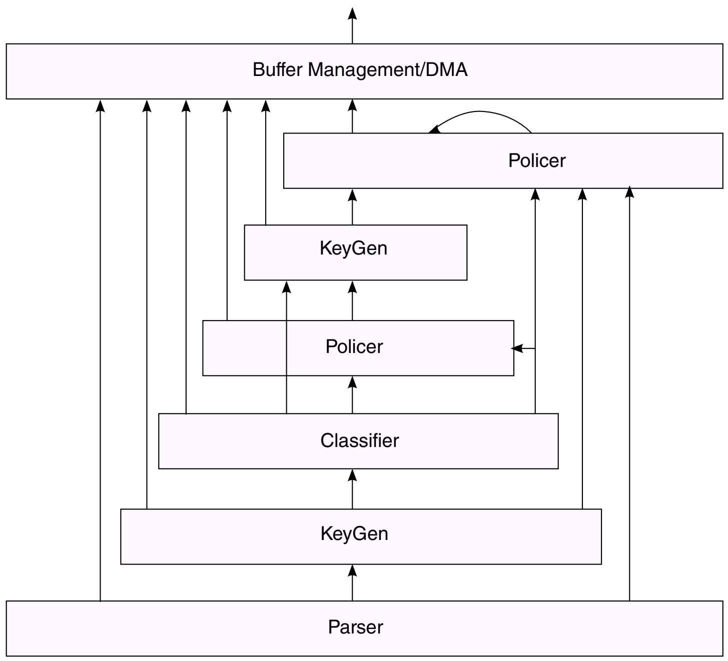


Figure 11. Logical view of FMan processing

Each frame received by the FMan is buffered internally while the Parser, KeyGen, and Classification functions operate.

The parse function can parse many standard protocols, including options and tunnels, and it supports a generic configurable capability to allow proprietary or future protocols to be parsed. Hard parsing of the standard protocol headers can be augmented with user-defined soft parsing rules to handle proprietary header fields. Hard and soft parsing occurs at wire speed.

This table defines several types of parser headers.

Table 6. Parser header types

Header type	Definition
Self-describing	Announced by proprietary values of Ethertype, protocol identifier, next header, and other standard fields. They are self-describing in that the frame contains information that describes the presence of the proprietary header.
Non-self-describing	Does not contain any information that indicates the presence of the header.

Table continues on the next page...

- 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)

    - 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 configure 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 classe(s); 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 dequeue 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.

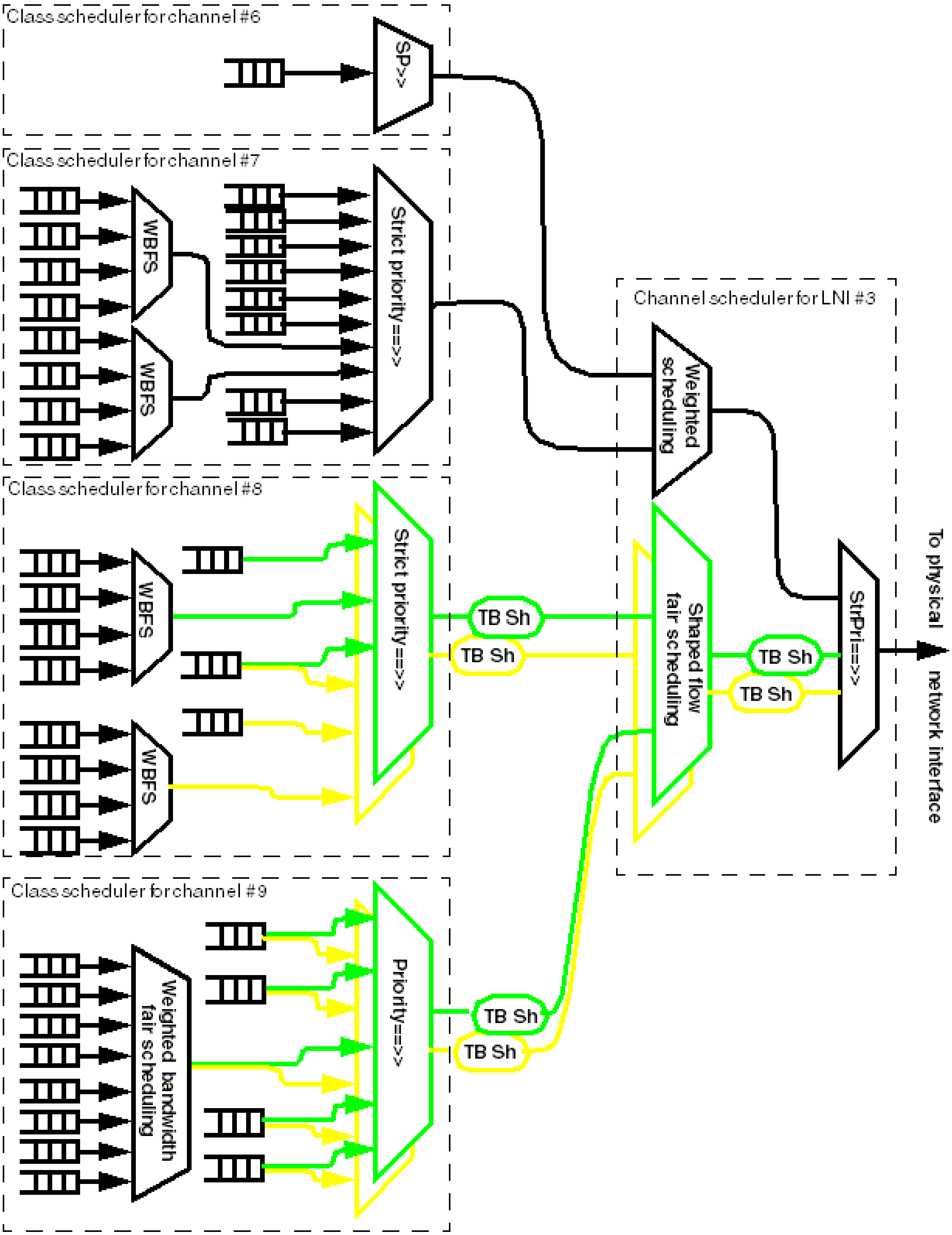


Figure 13. CEETM scheduler: illustrative configuration scenario

Figure 13 illustrates the following scenario:

- Channels #6, #7, #8 and #9 have been configured to be scheduled by the channel scheduler for LNI#3 (for example, all the packets from these channels are directed to the physical network interface configurably coupled to LNI#3).
- Channels #6 and #7 have been configured to be "unshaped." Packets from these channels will not be subjected to shaping at the channel level and will feed the top priority level within the LNI, which is also not subjected to shaping. Their class schedulers will not distinguish between CR and ER opportunities.
- Channels #8 and #9 have been configured to be "shaped." Their class schedulers will distinguish between CR and ER opportunities. The CR/ER packets to be sent from each channel shall be subjected to a pair of CR/ER token bucket shapers specific to that channel. The aggregate of CR/ER packets from these channels are subject to a pair of CR/ER token bucket shapers specific to LNI#3.
- Channel #6 has only one class in use. That class queue behaves as if it were a channel queue and as a peer to Channel #7. Unused classes do not have to be configured as such; they are simply not used.
- Channel #7 has all 16 classes in use.
    - The group classes have been configured as two groups (A and B) of four classes.
    - The priority of the groups A and B have both been set to be immediately below independent class 5. In a case of similar configuration group A has higher priority than group B.
- Channel #8 has three independent classes and two groups of four grouped classes in use.
    - The priorities of the class groups A and B have been set to be immediately below independent class 0 and class 2 respectively.
    - Independent class 0 and class group A have been configured to request and fulfill only CR packet opportunities.
    - Independent class 1 has been configured to request and fulfill both CR and ER packet opportunities.
    - Independent class 2 and class group B have been configured to request and fulfill only ER packet opportunities.
- Channels #9 has four independent classes and one group of eight grouped classes in use.
    - The group classes have been configured as one group (A) of eight classes.
    - All independent classes and the class group (A) have been configured to request and fulfill both CR and ER packet opportunities.

Benefits of the CEETM include the following:

- Provides "virtual" ports for multiple applications or users with different QoS/CoS requirements which are sharing an egress interface
- Supports DSCP capable scheduling for the following virtual link with configurable combinations of strict priority and weighted scheduling
    - Weighted scheduling closely approximating WFQ
- Supports traffic shaping
    - dual rate shaping of the virtual links
- Supports aggregating traffic from multiple virtual links and shaping this aggregate
- Hierarchical scheduling and shaping
- Class-based scheduling and dual rate shaping
- Supports a subset of the IEEE Data Center Bridging (DCB) standards

5.10.6.3 Data Center Bridging (DCB)

Data Center Bridging (DCB) refers to a series of inter-related IEEE specifications collectively designed to enhance Ethernet LAN traffic prioritization and congestion management. Although the primary objective is the data center environment (consisting of servers and storage arrays), some aspects of DCB are applicable to more general uses of Ethernet, within and between network nodes.

The SoC DPAA is compliant with the following DCB specifications :

- IEEE Std. 802.1Qbb: Priority-based flow control (PFC)
    - PAUSE frame per Ethernet priority code point (8)
    - Prevents single traffic class from throttling entire port
- IEEE Std. 802.1Qaz: Enhanced transmission selection (ETS)
    - Up to three Traffic Class Groups (TCG), where a TCG is composed of one or more priority code points
    - Bandwidth allocation and transmit scheduling (1% granularity) by traffic class group
    - If one of the TCGs does not consume its allocated bandwidth, unused bandwidth is available to other TCGs

PAMUs provide address translation and access control for all non-CPU initiators in the system. PAMU access control is based on the logical I/O device number (LIODN) advertised by a bus master for a given transaction. LIODNs can be static (for example, PCI Express controller #1 always uses LIODN 123) or they can be dynamic, based on the ID of the CPU that programmed the initiator (for example, the SEC uses LIODN 456 because it was given a descriptor by vCPU #2). In the dynamic example, the SoC architecture provides positive identification of the vCPU programming the SEC, preventing LIODN spoofing.

5.11.3 IO partitioning

The simplest IO configuration in chips running multiple independent software partitions is to dedicate specific IO controllers (PCI Express, SATA, Serial RapidIO controllers) to specific vCPUs. The core MMUs and PAMUs can enforce these access permissions to insure that only the software partition owning the IO is able to use it. The obvious problem with this approach is that there are likely to be more software partitions wanting IO access than there are IO controllers to dedicate to each.

Safe IO sharing can be accomplished through the use of a hypervisor; however, there is a performance penalty associated with virtual IO, as the hypervisor must consume CPU cycles to schedule the IO requests and get the results back to the right software partition.

The DPAA (described in Data Path Acceleration Architecture (DPAA)") was designed to allow multiple partitions to efficiently share accelerators and IOs, with its major capabilities centered around sharing Ethernet ports. These capabilities were enhanced in the chip with the addition of FMan storage profiles. The chip's FMans perform classification prior to buffer pool selection, allowing Ethernet frames arriving on a single port to be written to the dedicated memory of a single software partition. This capability is fully described in Receiver functionality: parsing, classification, and distribution."

The addition of the RMan extends the chip's IO virtualization by allowing many types of traffic arriving on Serial RapidIO to enter the DPAA and take advantage of its inherent virtualization and partitioning capabilities.

The PCI Express protocol lacks the PDU semantics found in Serial RapidIO, making it difficult to interwork between PCI Express controllers and the DPAA; however, PCI Express has made progress in other areas of partition. The Single Root IO Virtualization specification, which the chip supports as an endpoint, allows external hosts to view the chip as multiple two physical functions (PFs), where each PF supports up to 64 virtual functions (VFs). Having multiple VFs on a PCI Express port effectively channelizes it, so that each transaction through the port is identified as belonging to a specific PF/VF combination (with associated and potentially dedicated memory regions). Message signalled interrupts (MSIs) allow the external Host to generate interrupts associated with a specific VF.

5.11.4 Secure boot and sensitive data protection

The core MMUs and PAMU allow the SoC to enforce a consistent set of memory access permissions on a per-partition basis. When combined with an embedded hypervisor for safe sharing of resources, the SoC becomes highly resilient to poorly tested or malicious code. For system developers building high reliability/high security platforms, rigorous testing of code of known origin is the norm.

For this reason, the SoC offers a secure boot option, in which the system developer digitally signs the code to be executed by the CPUs, and the SoC insures that only an unaltered version of that code runs on the platform. The SoC offers both boot time and run time code authenticity checking, with configurable consequences when the authenticity check fails. The SoC also supports protected internal and external storage of developer-provisioned sensitive instructions and data. For example, a system developer may provision each system with a number of RSA private keys to be used in mutual authentication and key exchange. These values would initially be stored as encrypted blobs in external non-volatile memory; but, following secure boot, these values can be decrypted into on-chip protected memory (portion of platform cache dedicated as SRAM). Session keys, which may number in the thousands to tens of thousands, are not good candidates for on-chip storage, so the SoC offers session key encryption. Session keys are stored in main memory, and are decrypted (transparently to software and without impacting SEC throughput) as they are brought into the SEC 5.0 for decryption of session traffic.

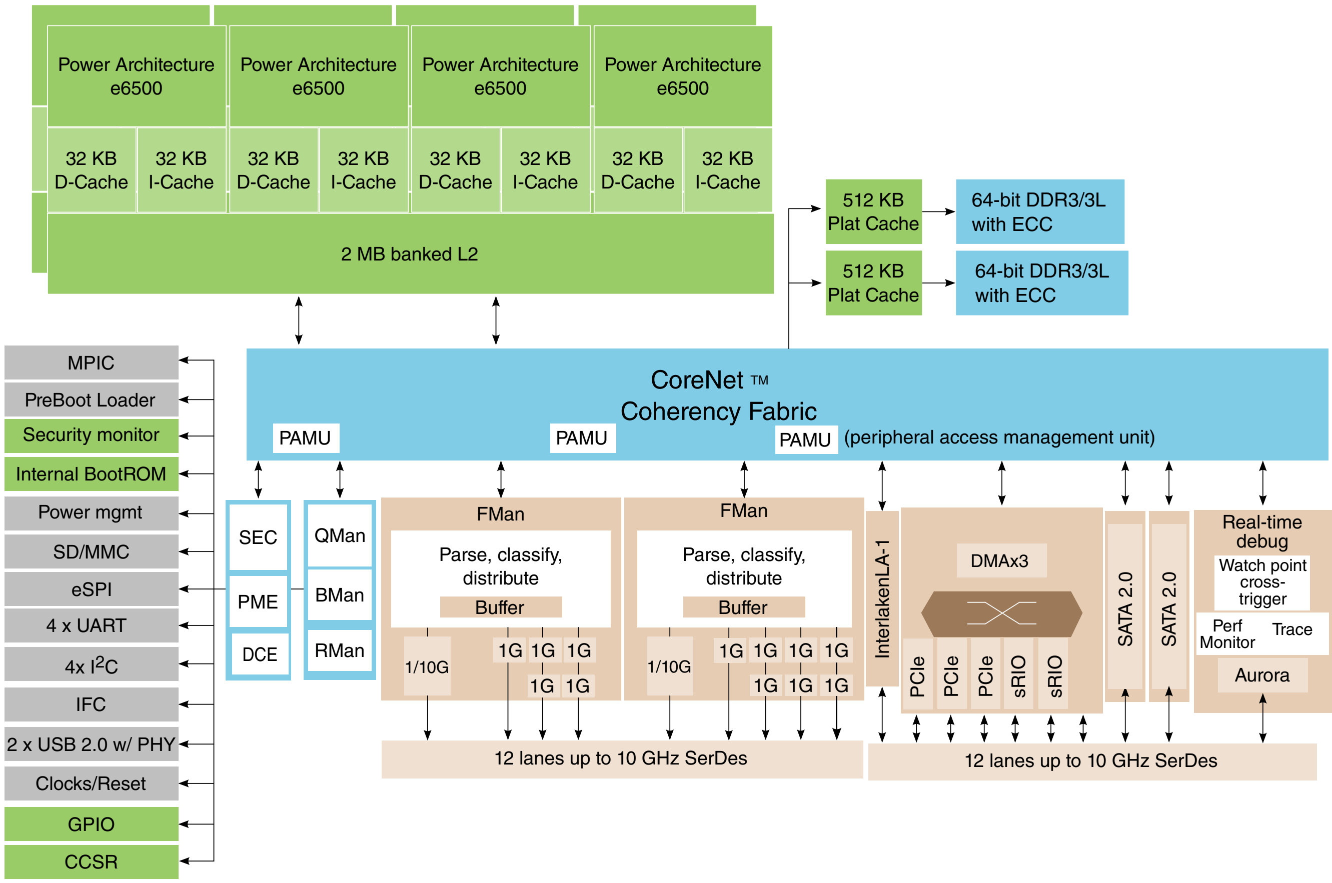


Figure A-1. T4160 block diagram

A.2 Overview of differences between T4240 and T4160

Table A-1. Differences between T4240 and T4160

Feature	T4240	T4160
Cores		
Number of physical cores	12	8
Number of threads	24	16
Number of clusters	3	2
Memory subsystem		
Total CPC memory	3 x 512 KB	2 x 512 KB
Number of DDR controllers	3	2
Peripherals		
Number of Frame Managers	2	2
Total number of Anyspeed MACs	8 per Frame Manager	6 (FMan1) and 8 (FMan2)

Table continues on the next page...

Table C-1. Revision history

Rev. number	Date	Substantive change(s)

1	10/2014	• 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:
		  • 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.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number T4240PB
Revision 1, 10/2014