**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=t4160nsn7ttb |

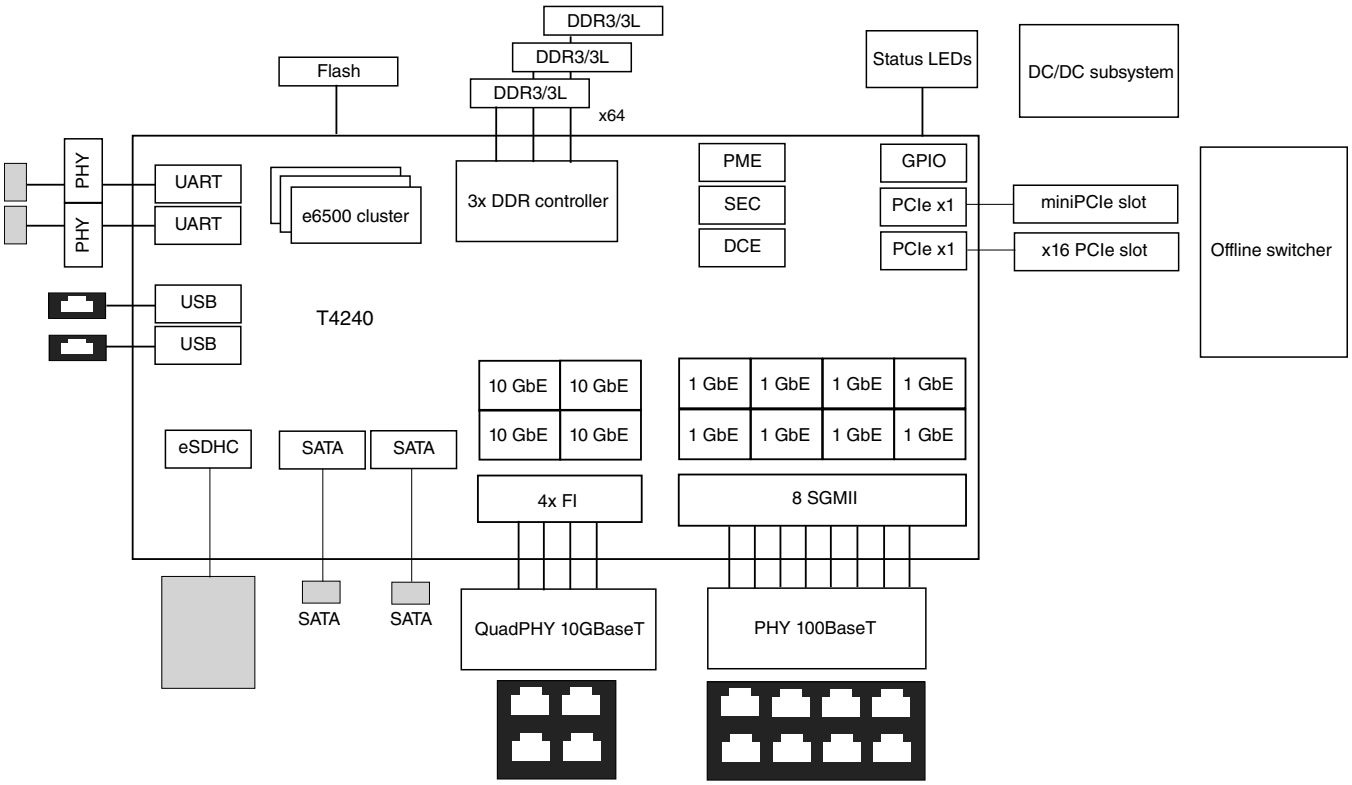


**Figure 1. SoC 1U security appliance**

## 3.2 Rack-mounted services blade

Networking and telecom systems are frequently modular in design, built from multiple standard dimension blades, which can be progressively added to a chassis to increase interface bandwidth or processing power. ATCA is a common standard form factor for chassis-based systems.

This figure shows a potential configuration for an ATCA blade with four chips and an Ethernet switch, which provides connectivity to the front panel and backplane, as well as between the chips. Potential systems enabled by chips in ATCA style modular architectures are described below.

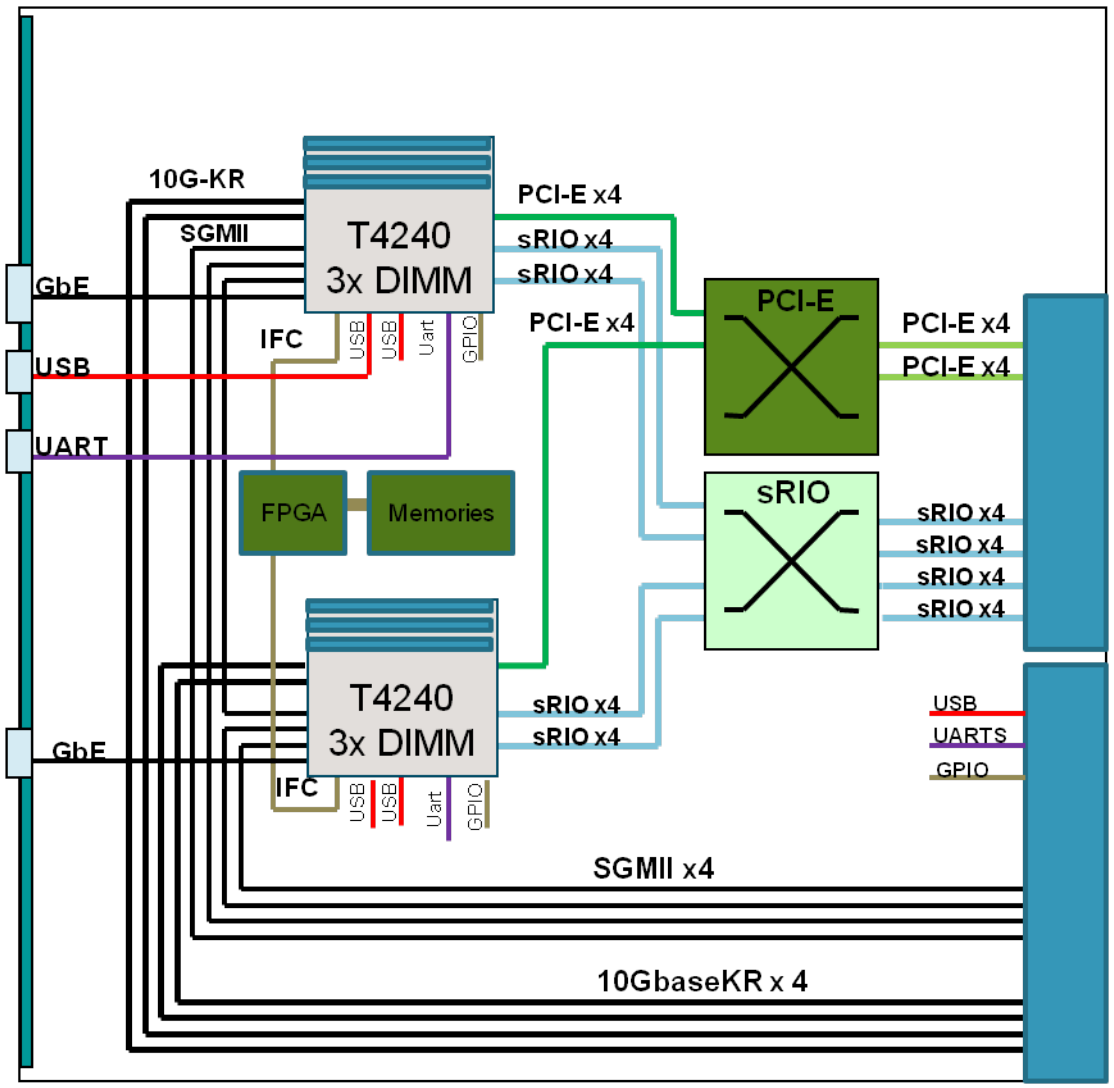


**Figure 3. Radio node controller**

## 3.4 Intelligent network adapter

The exact form factor of this card may vary, but the concepts are similar. A chip is placed on a small form factor card with an x8 PCIe connector and multiple 10 G Ethernet ports with HighGigE support for integrating with a Trident II device. This card is then used as inline accelerator that provides both line rate networking and intelligent programmable offload from a host processor subsystem in purpose built appliances and servers, such as Open vSwitch (OVS).

This figure shows an example of a T4240 built as a PCI Express form-factor supporting virtualization through SR-IOV with quad 10 G physical networking interfaces.

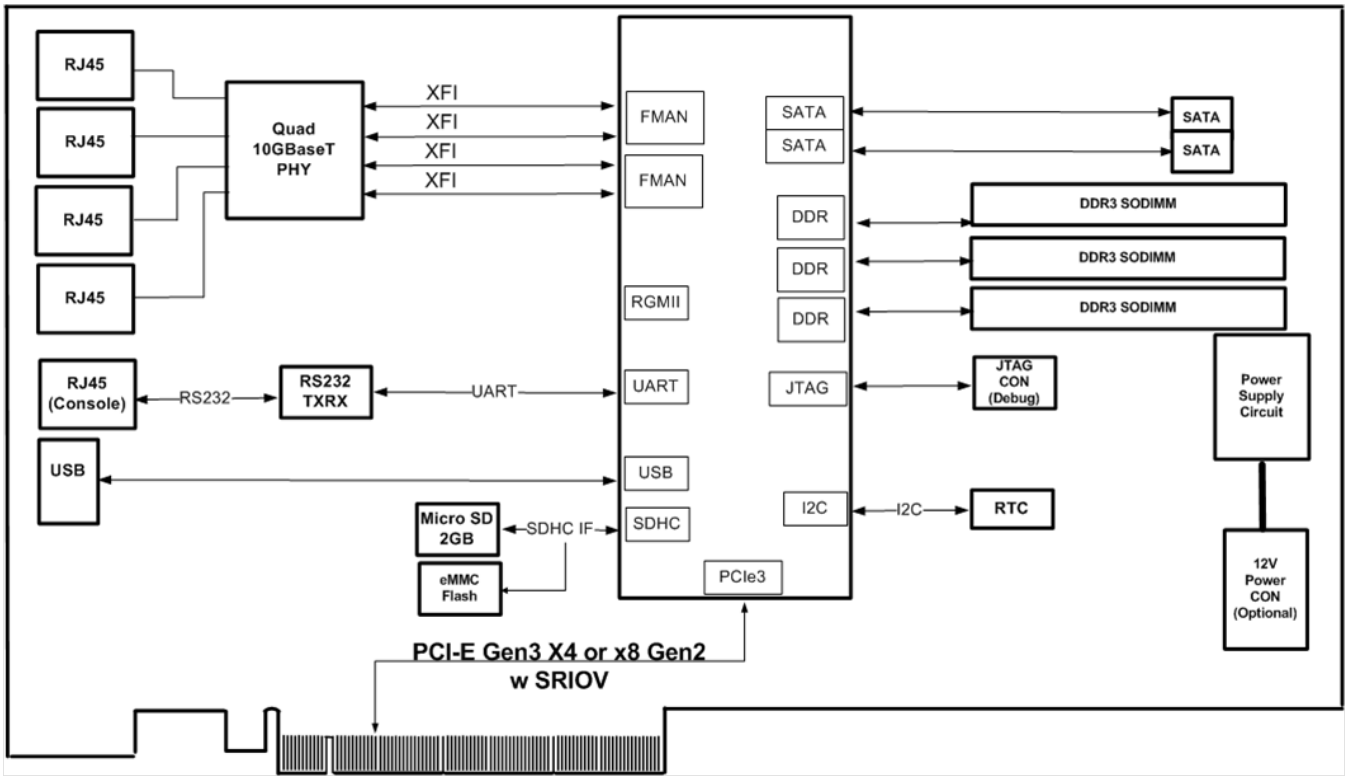


**Figure 4. Intelligent network adapter**

# 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.

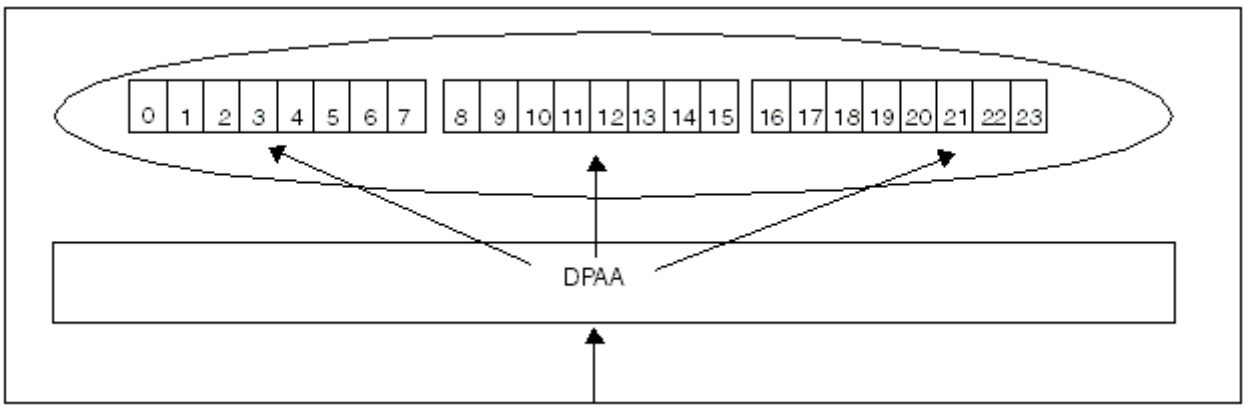


**Figure 5. 24 vCPU AMP or SMP with affinity**

## 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.

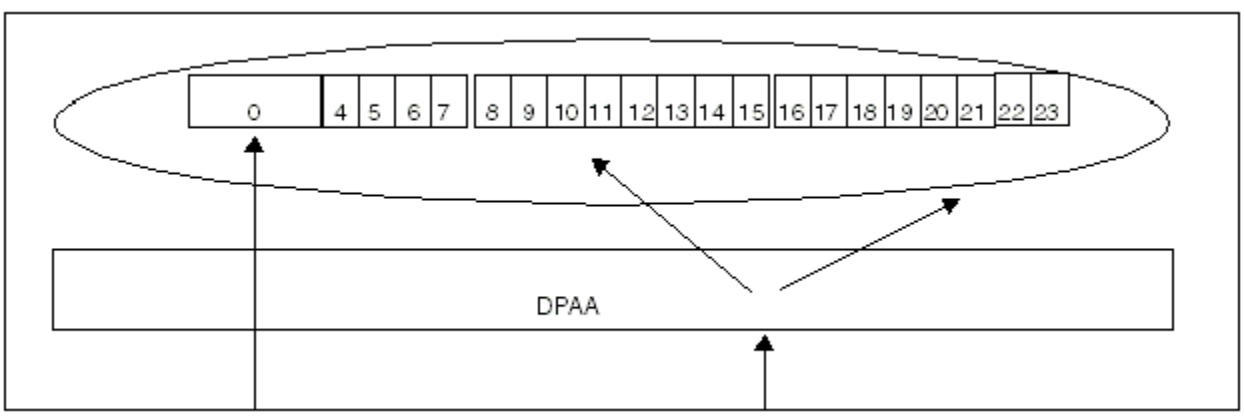


**Figure 6. Mixed SMP and AMP option 1**

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.

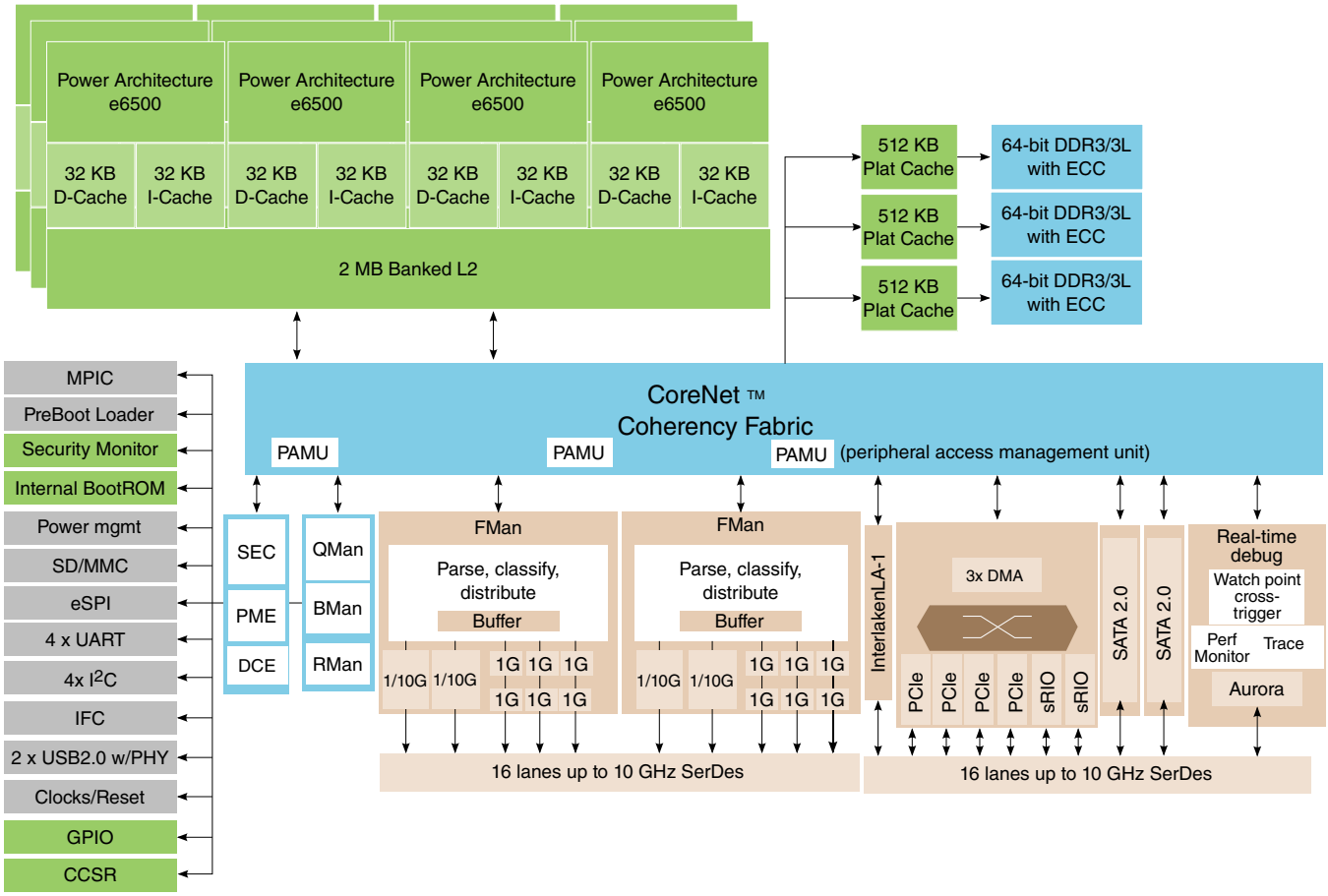


**Figure 8. T4240 block diagram**

## 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<br>• 1.5 V for DDR3<br>• 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.

## 5.6 CoreNet fabric and address map

The CoreNet fabric provides the following:

- A highly concurrent, fully cache coherent, multi-ported fabric
- Point-to-point connectivity with flexible protocol architecture allows for pipelined interconnection between CPUs, platform caches, memory controllers, and I/O and accelerators at up to 733 MHz
- The CoreNet fabric has been designed to overcome bottlenecks associated with shared bus architectures, particularly address issue and data bandwidth limitations. The chip's multiple, parallel address paths allow for high address bandwidth, which is a key performance indicator for large coherent multicore processors.
- Eliminates address retries, triggered by CPUs being unable to snoop within the narrow snooping window of a shared bus. This results in the chip having lower average memory latency.

This chip's 40-bit, physical address map consists of local space and external address space. For the local address map, 32 local access windows (LAWs) define mapping within the local 40-bit (1 TB) address space. Inbound and outbound translation windows can map the chip into a larger system address space such as the RapidIO or PCIe 64-bit address environment. This functionality is included in the address translation and mapping units (ATMUs).

## 5.7 Memory complex

The SoC's memory complex consists of up to three DDR controllers for main memory, and the memory controllers associated with the Integrated Flash Controller (IFC).

### 5.7.1 DDR memory controllers

The chip offers up to three 64-bit DDR controllers supporting ECC protected memories. These DDR controllers operate at up to 1.867 GT/s for DDR3, and, in more power sensitive applications, up to 1.6 GHz for DDR3L. Some key DDR controller features are as follows:

- Interleaving options
  - None, three fully independent controllers
  - Two interleaved, one independent
  - Three interleaved
  - Interleaving can be configured on 1 KB, 4 KB, and 8 KB granules
- Support x4, x8, and x16 memory widths
  - Programmable support for single, dual, and quad ranked devices and modules
  - Support for both unbuffered and registered DIMMs
  - 4 chip-selects per controller
  - 64 GB per controller, 192 GB per chip
- The SoC can be configured to retain the currently active SDRAM page for pipelined burst accesses. Page mode support of up to 64 simultaneously open pages can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, page mode can save up to ten memory clock cycles for subsequent burst accesses that hit in an active page.
- Using ECC, the SoC detects and corrects all single-bit errors and detects all double-bit errors and all errors within a nibble.
- Upon detection of a loss of power signal from external logic, the DDR controllers can put compliant DDR SDRAM DIMMs into self-refresh mode, allowing systems to implement battery-backed main memory protection.
- In addition, the DDR controllers offer an initialization bypass feature for use by system designers to prevent re-initialization of main memory during system power-on after an abnormal shutdown.
- Support active zeroization of system memory upon detection of a user-defined security violation.

- 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

### 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 enqueuing) 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.

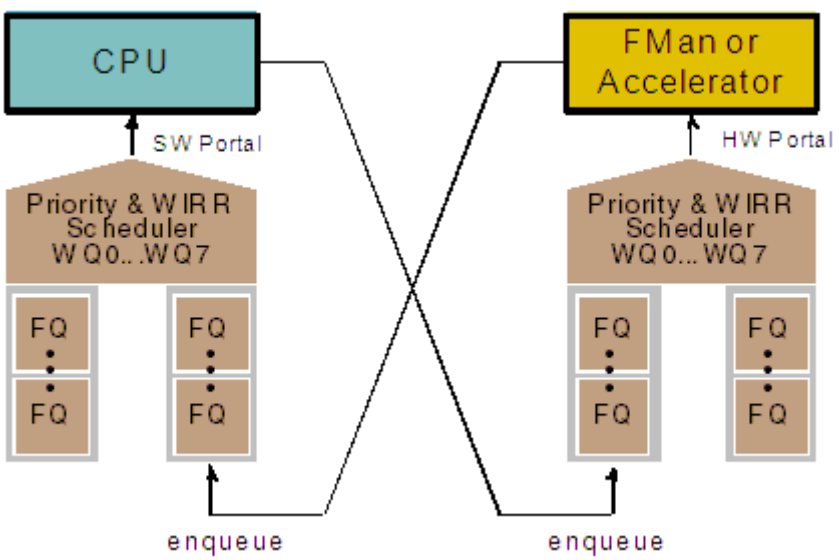


**Figure 9. DPAA enqueuing and dequeuing**

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 | B |

*Table continues on the next page...*

This figure is a logical view of the DPAA.

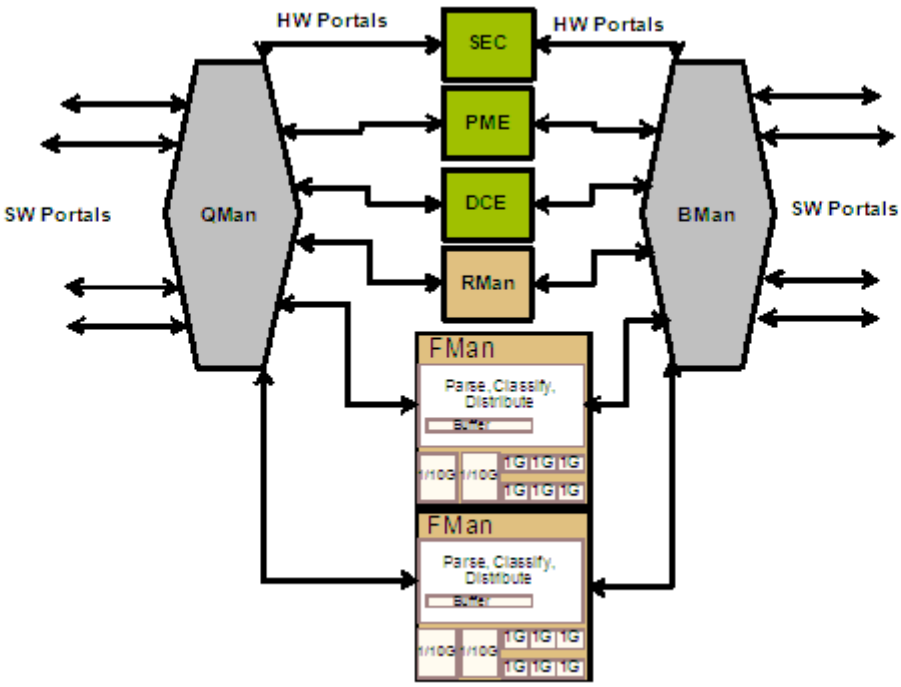


**Figure 10. Logical representation of DPAA**

### 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.

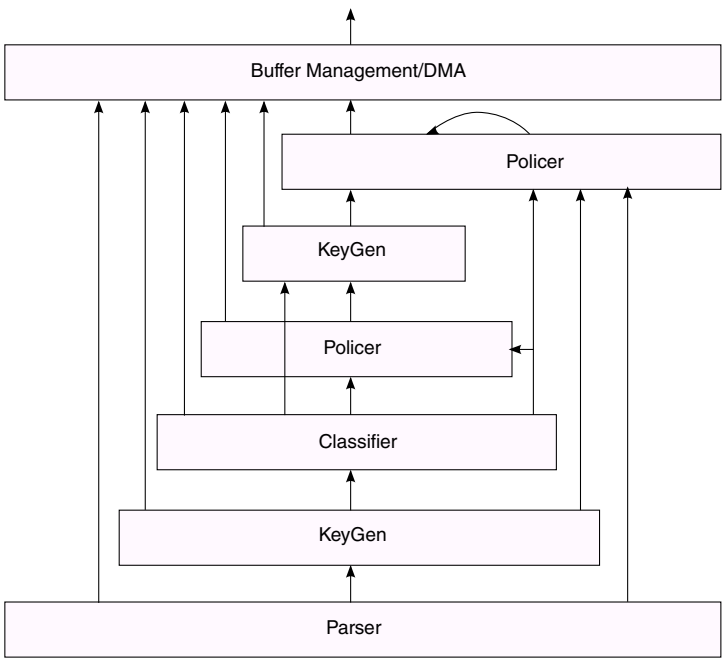


**Figure 11. Logical view of FMan processing**

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...*

This capability includes copying from one buffer pool to another if the traffic is received via the FMan's off-line parsing port. Packets can be copied to multiple buffer pools and enqueued to multiple frame queues to support broadcast and multicast requirements.

### 5.10.5.2 Queue Manager

The Queue Manager (QMan) is the primary infrastructure component in the DPAA, allowing for simplified sharing of network interfaces and hardware accelerators by multiple CPU cores. It also provides a simple and consistent message and data passing mechanism for dividing processing tasks amongst multiple vCPUs.

The Queue Manager offers the following features:

- Common interface between software and all hardware
  - Controls the prioritized queuing of data between multiple processor cores, network interfaces, and hardware accelerators.
  - Supports both dedicated and pool channels, allowing both push and pull models of multicore load spreading.
- Atomic access to common queues without software locking overhead
- Mechanisms to guarantee order preservation with atomicity and order restoration following parallel processing on multiple CPUs
- Egress queuing for Ethernet interfaces
  - Hierarchical (2-level) scheduling and dual-rate shaping
  - Dual-rate shaping to meet service-level agreements (SLAs) parameters (1 Kbps...10 Gbps range, 1 Kbps granularity across the entire range)
  - Configurable combinations of strict priority and fair scheduling (weighted queuing) between the queues
  - Algorithms for shaping and fair scheduling are based on bytes
- Queuing to cores and accelerators
  - Two level queuing hierarchy with one or more Channels per Endpoint, eight work queues per Channel, and numerous frame queues per work queue
  - Priority and work conserving fair scheduling between the work queues and the frame queues
- Loss-less flow control for ingress network interfaces
- Congestion avoidance (RED/WRED) and congestion management with tail discard

### 5.10.5.3 Buffer Manager

The Buffer Manager (BMan) manages pools of buffers on behalf of software for both hardware (accelerators and network interfaces) and software use.

The Buffer Manager offers the following features:

- Common interface for software and hardware
- Guarantees atomic access to shared buffer pools
- Supports 64 buffer pools
  - Software, hardware buffer consumers can request different size buffers and buffers in different memory partitions
- Supports depletion thresholds with congestion notifications
- On-chip per pool buffer stockpile to minimize access to memory for buffer pool management
- LIFO (last in first out) buffer allocation policy
  - Optimizes cache usage and allocation
  - A released buffer is immediately used for receiving new data

### 5.10.5.4 SEC 5.0

The SEC 5.0 is Freescale's fifth generation crypto-acceleration engine. The SEC 5.0 is backward-compatible with the SEC 4.x, as implemented in the first generation of high-end QorIQ products, which includes the P4080. As in the SEC 4.x, the SEC 5.0 offers high performance symmetric and asymmetric encryption, keyed and unkeyed hashing algorithms, NIST-compliant random number generation, and security protocol header and trailer processing.

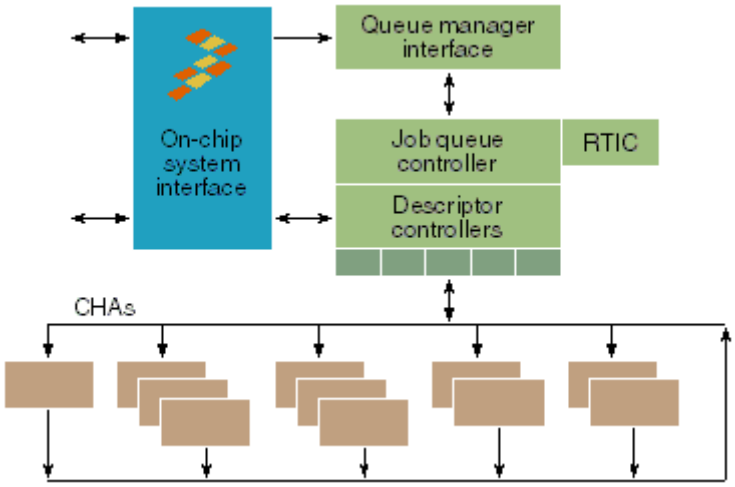


**Figure 12. SEC 5.0 block diagram**

The SEC 5.0 is also part of the QorIQ Platform's Trust Architecture, which gives the SoC the ability to perform secure boot, runtime code integrity protection, and session key protection. The Trust Architecture is described in Resource partitioning and QorIQ Trust Architecture.

### 5.10.5.5 Pattern Matching Engine (PME 2.1)

The PME 2.1 is Freescale's second generation of extended NFA style pattern matching engine. Unchanged from the first generation QorIQ products, it supports ~10 Gbps data scanning.

Key benefits of a NFA pattern matching engine:

- No pattern "explosion" to support "wildcarding" or case-insensitivity
  - Comparative compilations have shown 300,000 DFA pattern equivalents can be achieved with ~8000 extended NFA patterns
- Pattern density much higher than DFA engines.
  - Patterns can be stored in on-chip tables and main DDR memory
  - Most work performed solely with on-chip tables (external memory access required only to confirm a match)
  - No need for specialty memories; for example, QDR SRAM, RLDRAM, and so on.
- Fast compilation of pattern database, with fast incremental additions
  - Pattern database can be updated without halting processing
  - Only affected pattern records are downloaded
  - DFA style engines can require minutes to hours to recompile and compress database

Freescale's basic NFA capabilities for byte pattern scanning are as follows:

- The PME's regex compiler accepts search patterns using syntax similar to that in software-based regex engines, such as Perl-Compatible Regular Expression (PCRE).
  - Supports Perl meta-characters including wildcards, repeats, ranges, anchors, and so on.
  - Byte patterns are simple matches, such as gabcd123h, existing in both the data being scanned and in the pattern specification database.
- Up to 32 KB patterns of length 1-128 bytes

Freescale's extensions to NFA style pattern matching are principally related to event pattern scanning. Event patterns are sequences of byte patterns linked by 'stateful rules.' Freescale uses event pattern scanning and stateful rule processing synonymously. Stateful rules are hardware instructions by which users define reactions to pattern match events, such as state changes, assignments, bitwise operations, addition, subtraction, and comparisons.

Some key characteristics and benefits of the Stateful Rule extensions include:

- 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 functionary, 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)

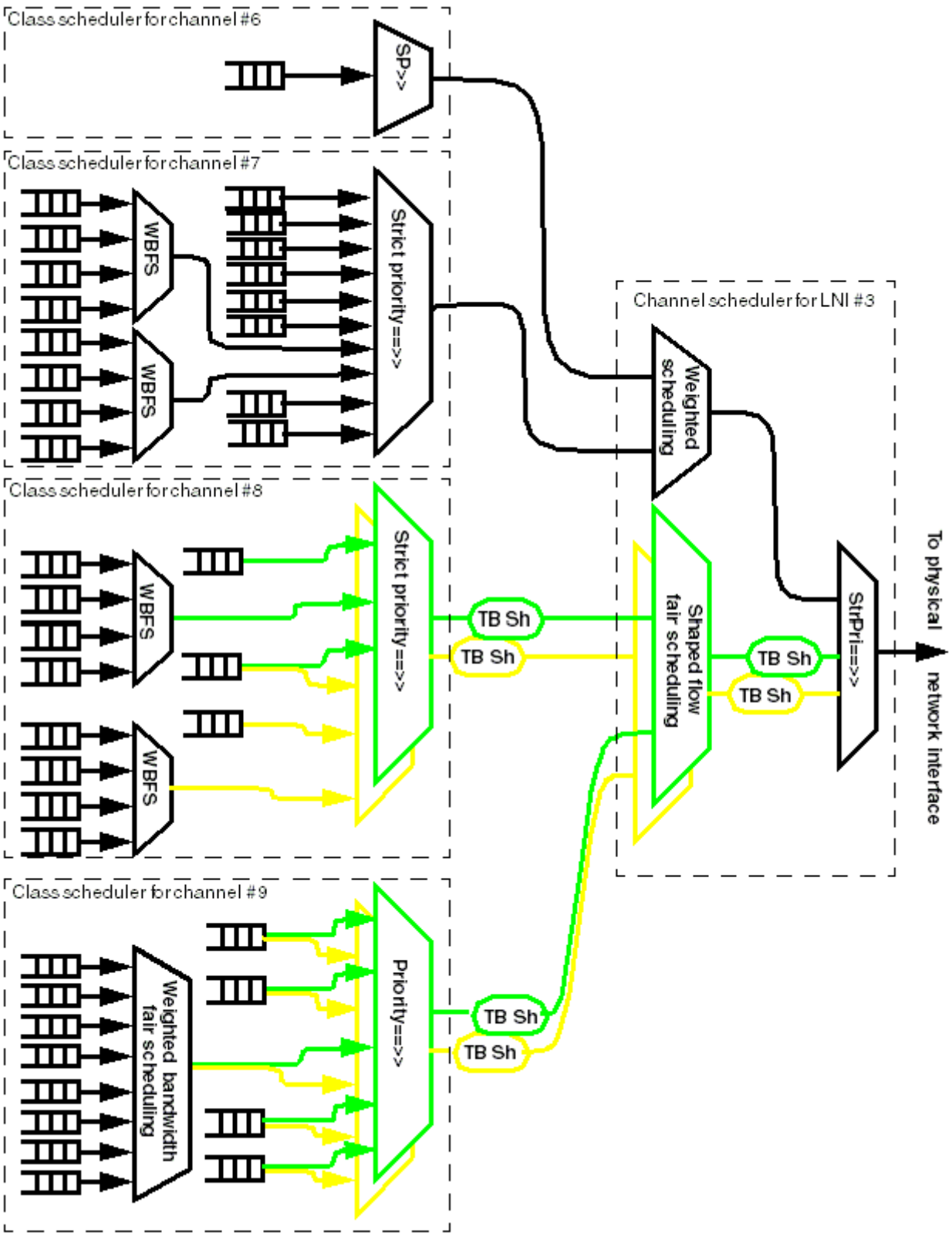


**Figure 13. CEETM scheduler: illustrative configuration scenario**

Figure 13 illustrates the following scenario:

## 5.11 Resource partitioning and QorIQ Trust Architecture

Consolidation of discrete CPUs into a single, multicore chip introduces many opportunities for unintended resource contentions to arise, particularly when multiple, independent software entities reside on a single chip. A system may exhibit erratic behavior if multiple software partitions cannot effectively partition resources. Device consolidation, combined with a trend toward embedded systems becoming more open (or more likely to run third-party or open-source software on at least one of the cores), creates opportunities for malicious code to enter a system.

This chip offers a new level of hardware partitioning support, allowing system developers to ensure software running on any CPU only accesses the resources (memory, peripherals, and so on) that it is explicitly authorized to access. This section provides an overview of the features implemented in the chip that help ensure that only trusted software executes on the CPUs, and that the trusted software remains in control of the system with intended isolation.

### 5.11.1 Core MMU, UX/SX bits, and embedded hypervisor

The chip's first line of defense against unintended interactions amongst the multiple CPUs/OSes is each core vCPU's MMU. A vCPU's MMU is configured to determine which addresses in the global address map the CPU is able to read or write. If a particular resource (memory region, peripheral device, and so on) is dedicated to a single vCPU, that vCPU's MMU is configured to allow access to those addresses (on 4 KB granularity); other vCPU MMUs are not configured for access to those addresses, which makes them private. When two vCPUs need to share resources, their MMUs are both configured so that they have access to the shared address range.

This level of hardware support for partitioning is common today; however, it is not sufficient for many core systems running diverse software. When the functions of multiple discrete CPUs are consolidated onto a single multicore chip, achieving strong partitioning should not require the developer to map functions onto vCPUs that are the exclusive owners of specific platform resources. The alternative, a fully open system with no private resources, is also unacceptable. For this reason, the core's MMU also includes three levels of access permissions: user, supervisor (OS), and hypervisor. An embedded hypervisor (for example, KVM, XEN, QorIQ ecosystem partner hypervisor) runs unobtrusively beneath the various OSes running on the vCPUs, consuming CPU cycles only when an access attempt is made to an embedded hypervisor-managed shared resource.

The embedded hypervisor determines whether the access should be allowed and, if so, proxies the access on behalf of the original requestor. If malicious or poorly tested software on any vCPU attempts to overwrite important device configuration registers (including vCPU's MMU), the embedded hypervisor blocks the write. High and low-speed peripheral interfaces (PCI Express, UART), when not dedicated to a single vCPU/partition, are other examples of embedded hypervisor managed resources. The degree of security policy enforcement by the embedded hypervisor is implementation-dependent.

In addition to defining regions of memory as being controlled by the user, supervisor, or hypervisor, the core MMU can also configure memory regions as being non-executable. Preventing CPUs from executing instructions from regions of memory used as data buffers is a powerful defense against buffer overflows and other runtime attacks. In previous generations of Power Architecture, this feature was controlled by the NX (no execute) attribute. In new Power Architecture cores such as the e6500 core, there are separate bits controlling execution for user (UX) and supervisor (SX).

### 5.11.2 Peripheral access management unit (PAMU)

MMU-based access control works for software running on CPUs; however, these are not the only bus masters in the SoC. Internal components with bus mastering capability (FMan, RMan, PCI Express controller, PME, SEC, and so on) also need to be prevented from reading and writing to certain memory regions. These components do not spontaneously generate access attempts; however, if programmed to do so by buggy or malicious software, any of them could read or write sensitive data registers and crash the system. For this reason, the SoC also includes a distributed function referred to as the peripheral access management unit (PAMU).

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.

## Table C-1. Revision history

| Rev. number | Date | Substantive change(s) |
|---|---|---|
| 1 | 10/2014 | • Added support for T4080 throughout document.<br>• Updated Introduction.<br>• In Summary of benefits, updated the first sentence to include "...SDN switches or controllers, network function virtualization..." and added the following subsections:<br>  • e6500 CPU core<br>  • Virtualization<br>  • Data Path Acceleration Architecture (DPAA)<br>  • System peripherals and networking<br>• In Intelligent network adapter, added examples.<br>• Updated Block diagram.<br>• 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.<br>• In Critical performance parameters, removed the typical power consumption table.<br>• 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".<br>• Updated the read bandwidth in CoreNet fabric and address map.<br>• Added HiGig 2 in Enhancements of T4240 compared to first generation DPAA.<br>• Updated bullet two in CoreNet fabric and address map and updated the last bullet in High-speed peripheral interface complex (HSSI).<br>• Updated Non-transparent power management.<br>• Rewrote Conclusion to add more information and a list of Freescale resources.<br>• 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