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/product-detail/nxp-semiconductors/t4240nse7ttb

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.

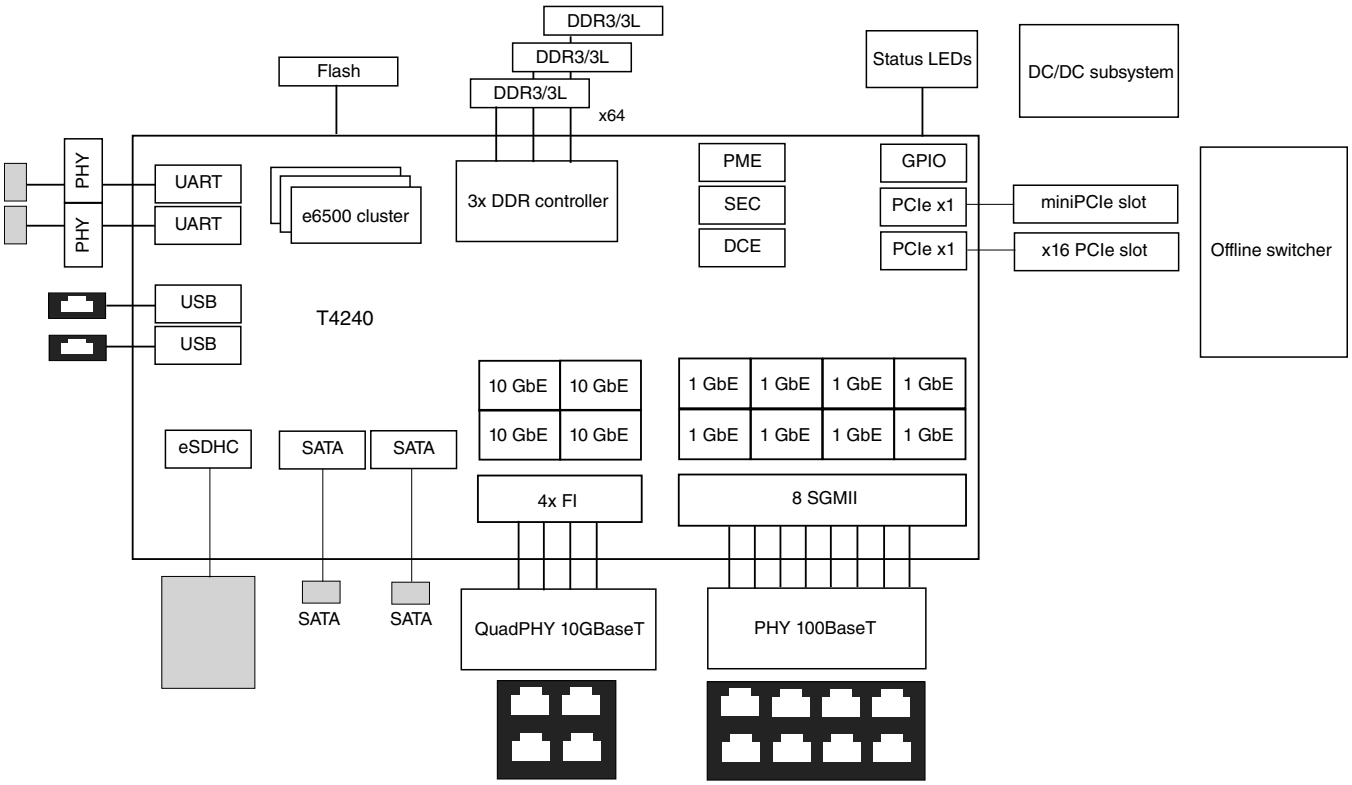


Figure 1. SoC 1U security appliance

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.

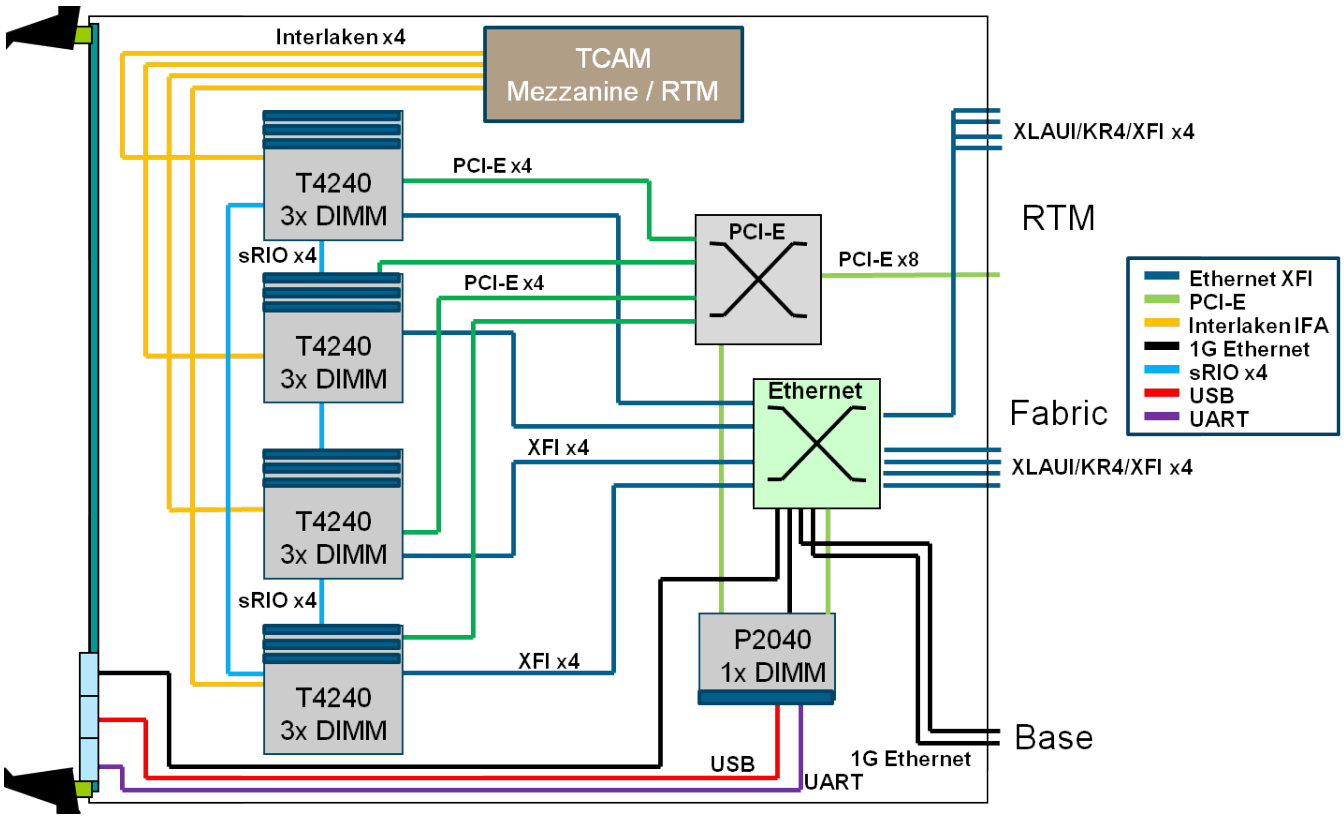


Figure 2. Network services ATCA blade

3.3 Radio node controller

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

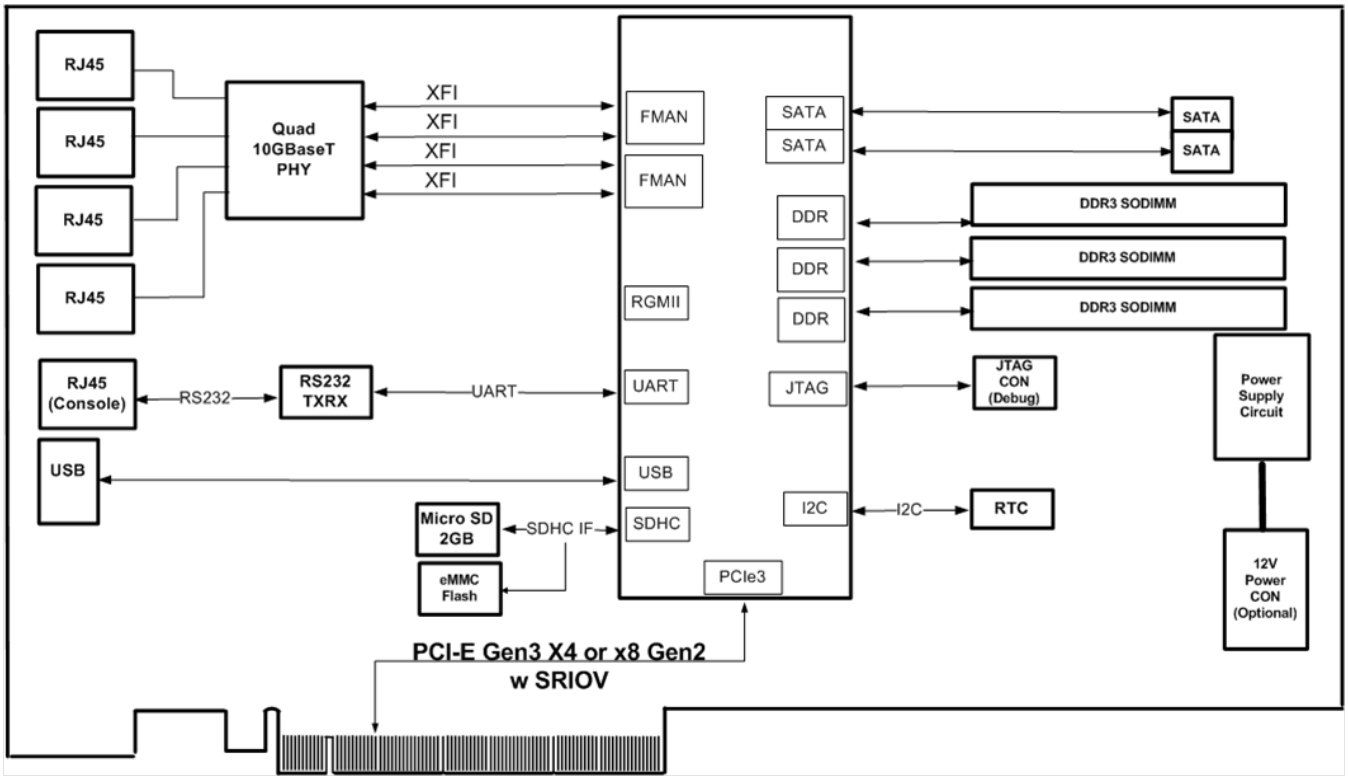


Figure 4. Intelligent network adapter

4 Multicore processing options

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

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

4.1 Asymmetric multiprocessing

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

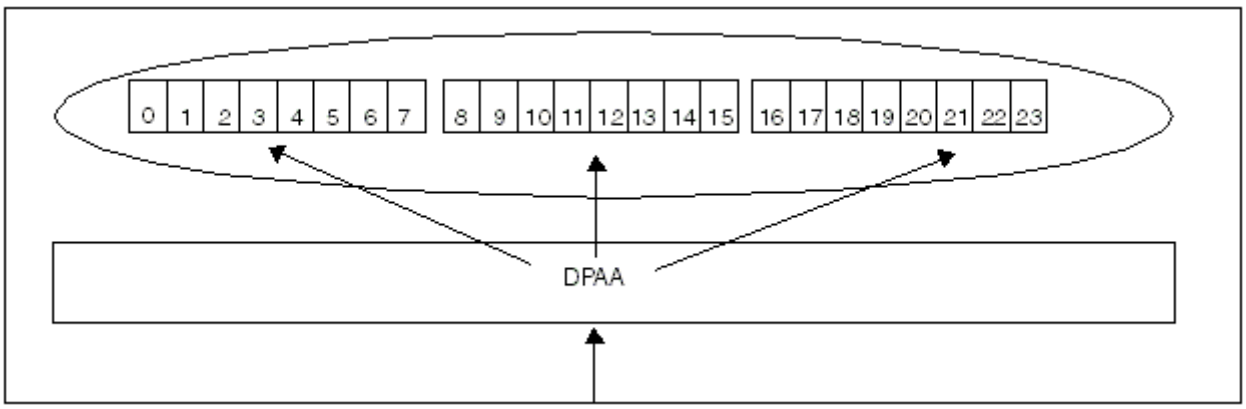


Figure 5. 24 vCPU AMP or SMP with affinity

4.2 Symmetric multiprocessing

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

4.3 Mixed symmetric and asymmetric multiprocessing

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

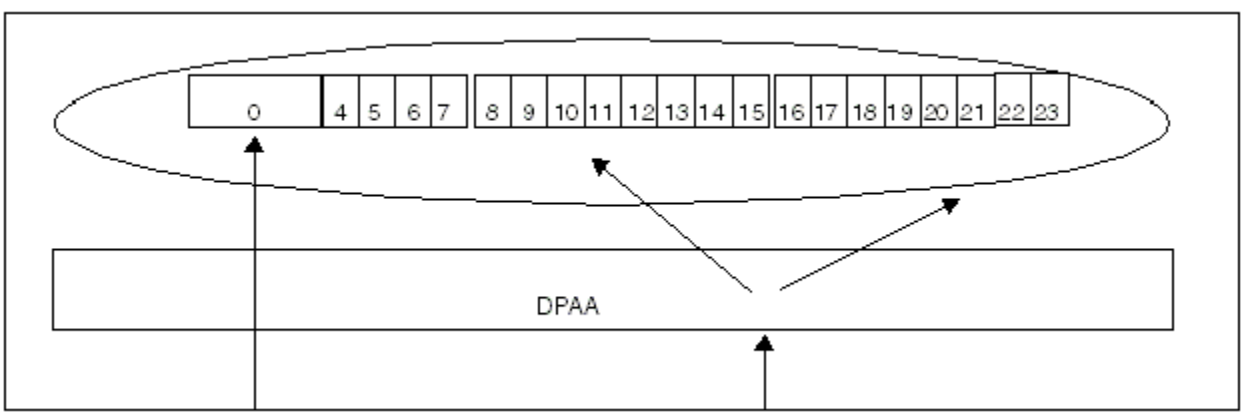


Figure 6. Mixed SMP and AMP option 1

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

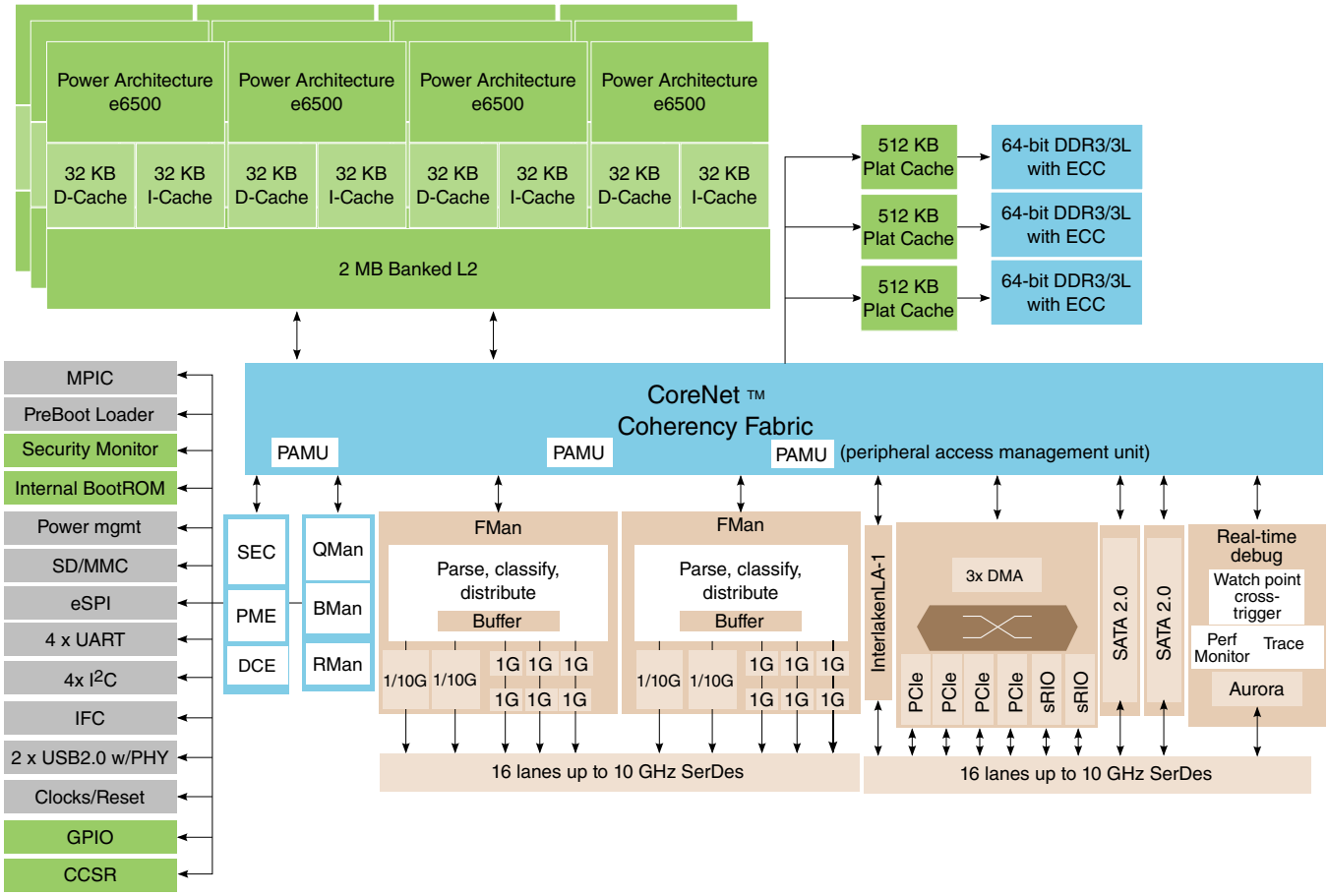


Figure 8. T4240 block diagram

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.

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.

5.7.1.1 DDR bandwidth optimizations

Multicore SoCs are able to increase CPU and network interface bandwidths faster than commodity DRAM technologies are improving. As a result, it becomes increasingly important to maximize utilization of main memory interfaces to avoid a memory bottleneck. The T4 family's DDR controllers are Freescale-developed IP, optimized for the QorIQ SoC architecture, with the goal of improving DDR bandwidth utilization by fifty percent when compared to first generation QorIQ SoCs.

Most of the WRITE bandwidth improvement and approximately half of the READ bandwidth improvement is met through target queue enhancements; in specific, changes to the scheduling algorithm, improvements in the bank hashing scheme, support for more transaction re-ordering, and additional proprietary techniques.

The remainder of the READ bandwidth improvement is due to the addition of an intelligent data prefetcher in the memory subsystem.

5.7.1.2 Prefetch Manager (PMan)

NOTE

All transactions to DDR pass through the CPC; this means the CPC can miss (and trigger prefetching) even on data that is not intended for allocation into the CPC.

The PMAN monitors CPC misses for opportunities to prefetch, using a "confidence"-based algorithm to determine its degree of aggressiveness. It can be configured to monitor multiple memory regions (each of different size) for prefetch opportunities. Multiple CPC misses on accesses to a tracked region for consecutive cache blocks increases confidence to start prefetching, and a CPC miss of a tracked region with same stride will instantly cause prefetching.

The PMan uses feedback to increase or decrease its aggressiveness. When the data it prefetches is being used, it prefetches further ahead. If the request stride length changes or previously prefetched data isn't consumed, prefetching slows or stops (at least for that region/requesting device/transaction type).

5.7.2 PreBoot Loader and nonvolatile memory interfaces

The PreBoot Loader (PBL) operates similarly to an I^2C boot sequencer but on behalf of a large number of interfaces.

It supports IFC, I^2C, eSPI, eSDHC.

The PBL's functions include the following:

- Simplifies boot operations, replacing pin strapping resistors with configuration data loaded from nonvolatile memory
- Uses the configuration data to initialize other system logic and to copy data from low speed memory interfaces (I^2C, IFC, eSPI, and SD/MMC) into fully initialized DDR or the 2 MB front-side cache

5.7.2.1 Integrated Flash Controller

The SoC incorporates an Integrated Flash Controller similar to the one used in some previous generation QorIQ SoCs. The IFC supports both NAND and NOR flash, as well as a general purpose memory mapped interface for connecting low speed ASICs and FPGAs.

5.7.2.1.1 NAND Flash features

- x8/x16 NAND Flash interface
- Optional ECC generation/checking
- Flexible timing control to allow interfacing with proprietary NAND devices
- SLC and MLC Flash devices support with configurable page sizes of up to 4 KB
- Support advance NAND commands like cache, copy-back, and multiplane programming

- 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

    - 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.

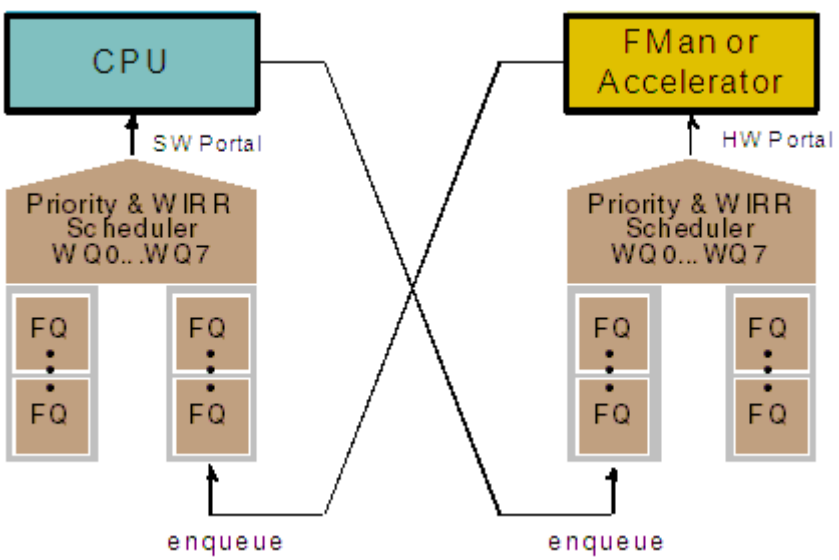


Figure 9. DPAA enqueuing and dequeuing

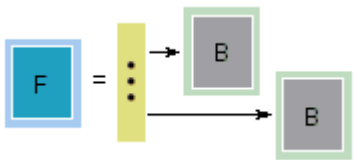
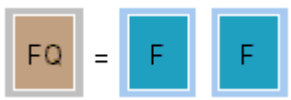
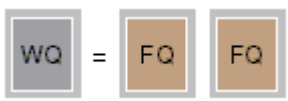
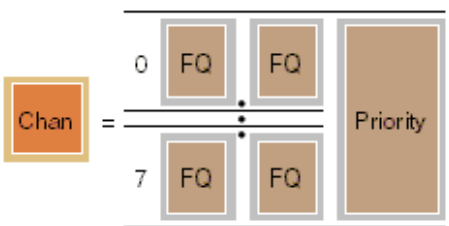
This table lists common DPAA terms and their definitions.

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...

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.

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.

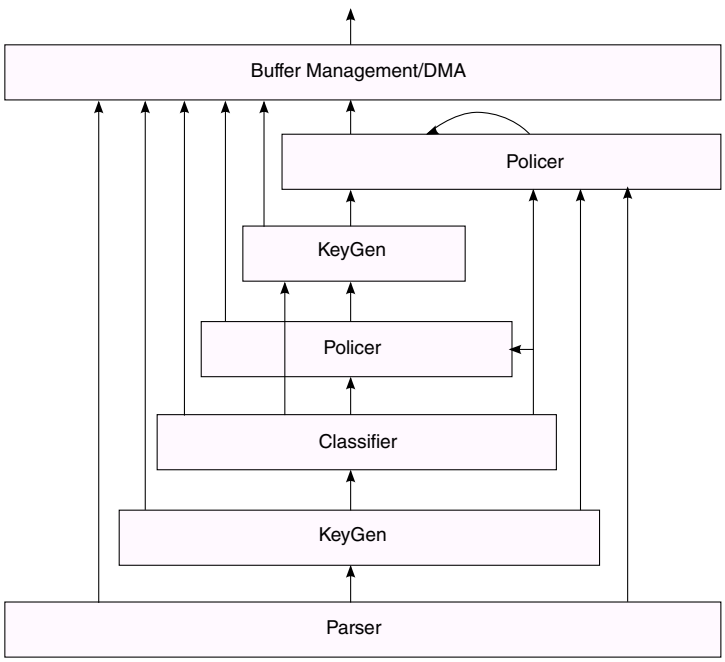


Figure 11. Logical view of FMan processing

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...

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).

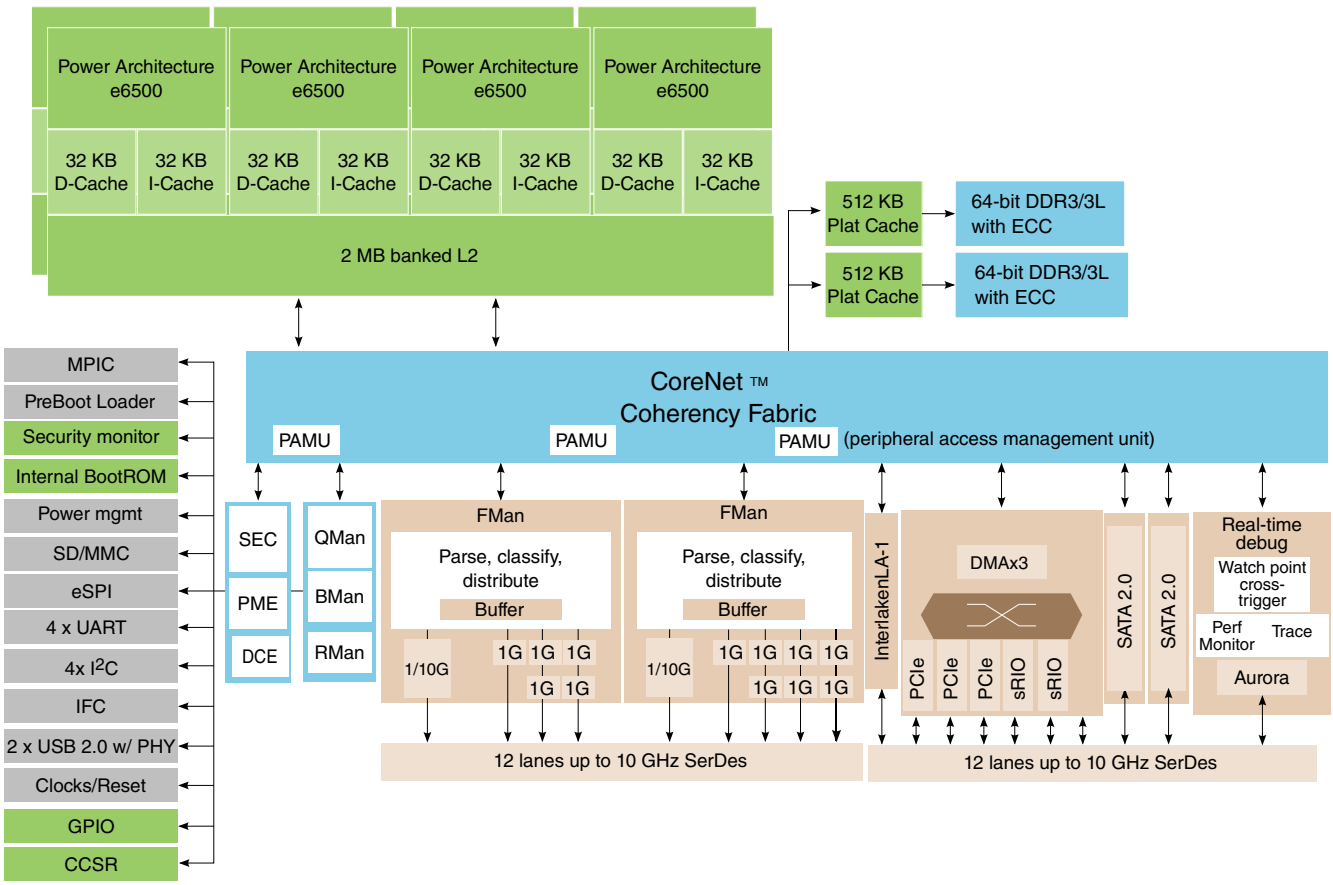


Figure A-1. T4160 block diagram

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...

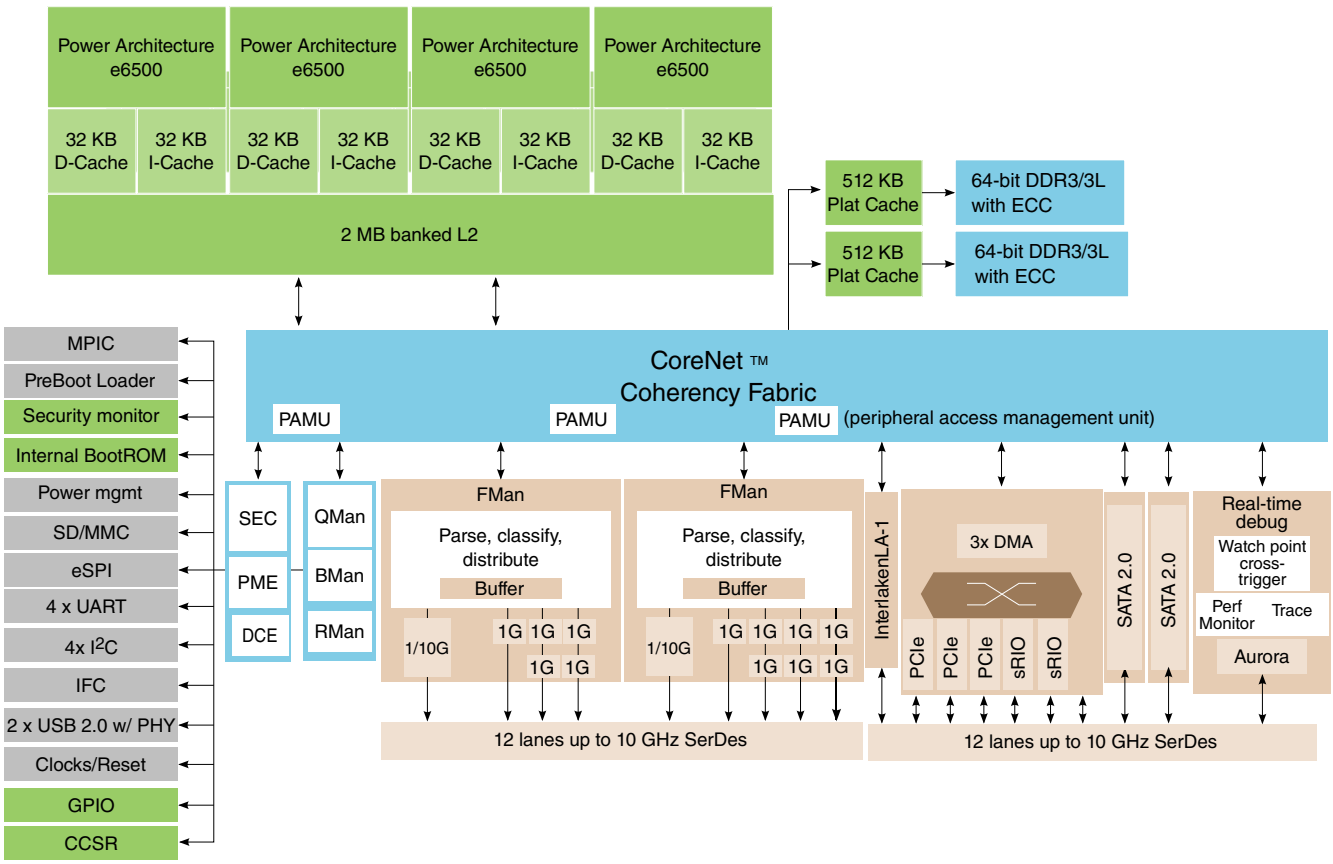


Figure B-1. T4080 block diagram

B.2 Overview of differences between T4160 and T4080

Table B-1. Differences between T4160 and T4080

Feature	T4160	T4080
Cores		
Number of physical cores	8	4
Number of threads	16	8
Number of clusters	2	1

Appendix C Revision history

C.1 Revision history

This table provides a revision history for this document.