Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Active
Core Processor	PowerPC e6500
Number of Cores/Bus Width	12 Core, 64-Bit
Speed	1.8GHz
Co-Processors/DSP	-
RAM Controllers	DDR3, DDR3L
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	1Gbps (16), 10Gbps (4)
SATA	SATA 3Gbps (2)
USB	USB 2.0 + PHY (2)
Voltage - I/O	-
Operating Temperature	0°C ~ 105°C (TA)
Security Features	-
Package / Case	1932-BBGA, FCBGA
Supplier Device Package	1932-FCPBGA (45x45)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=t4240nsn7ttb

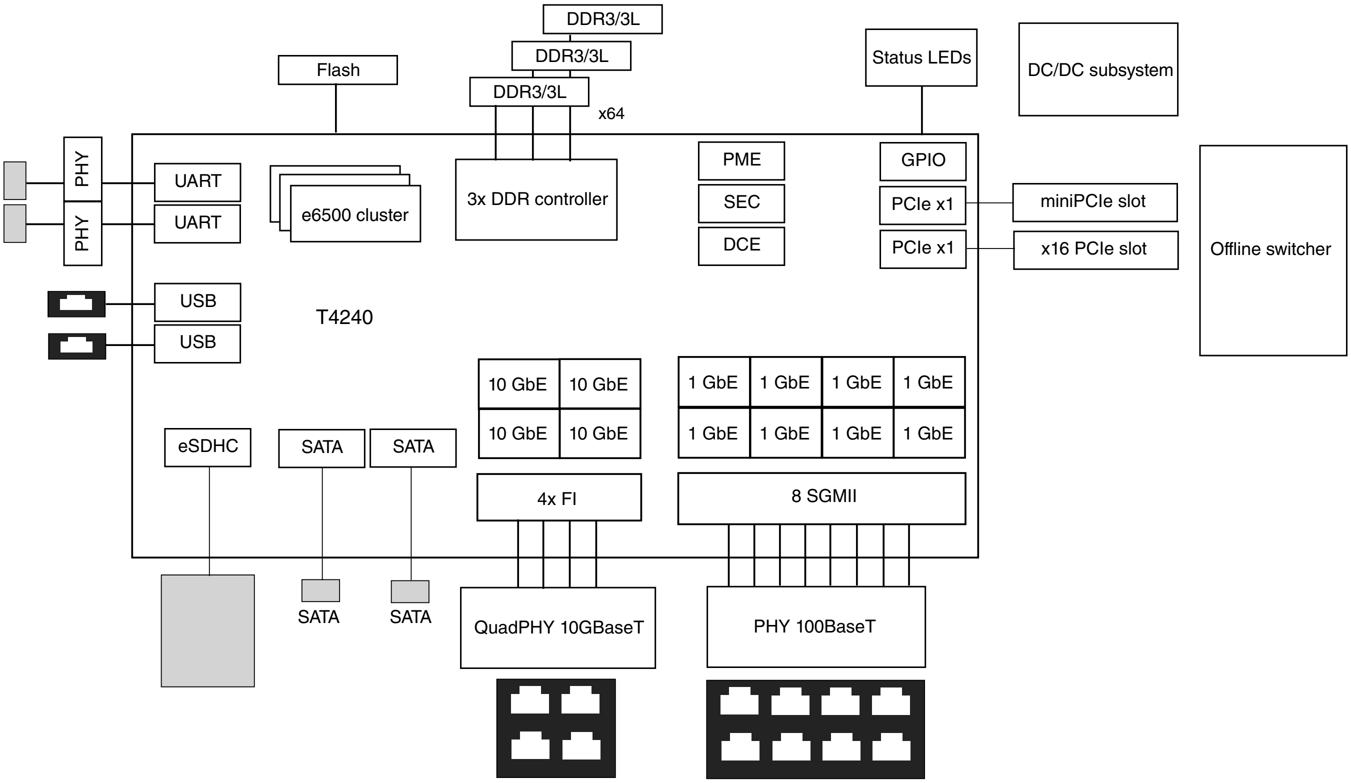

Figure 1. SoC 1U security appliance

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.

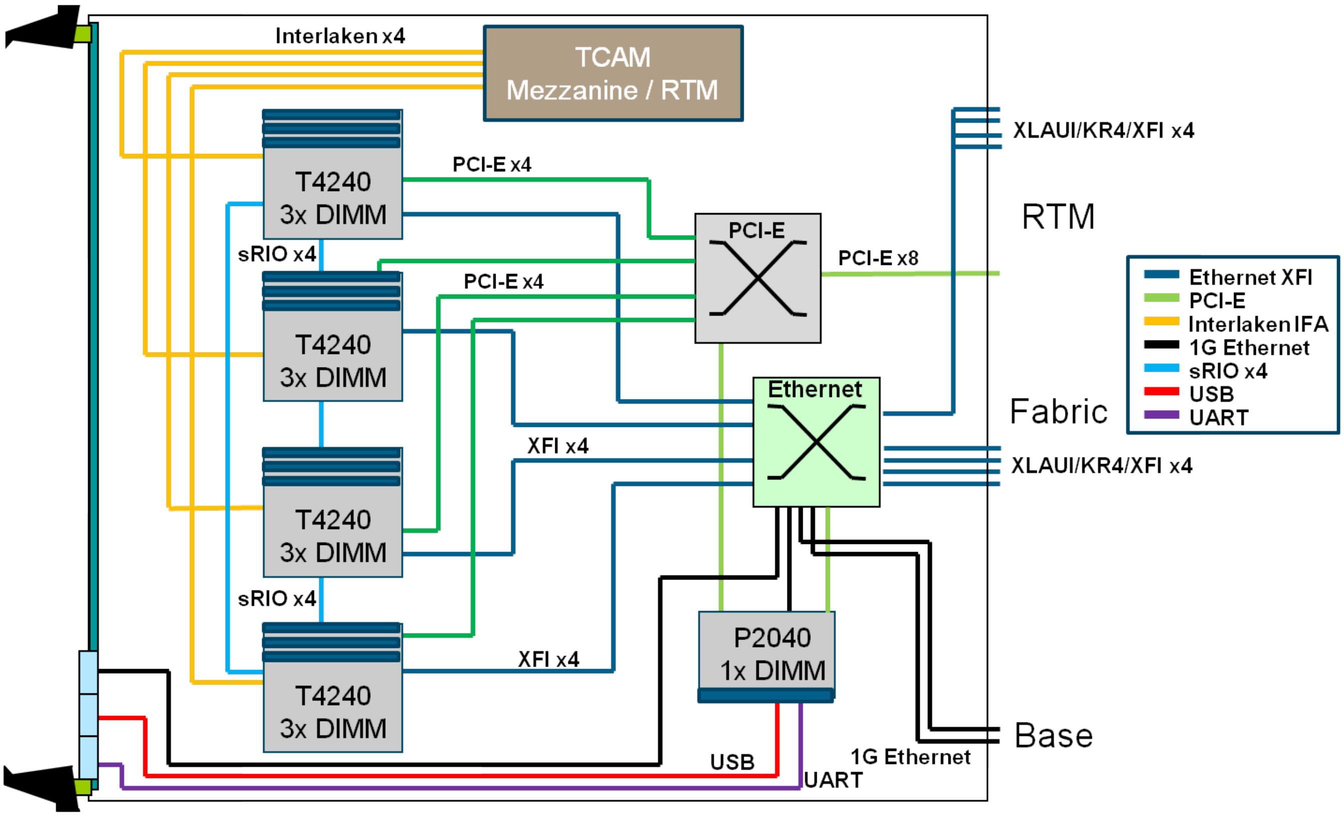

Figure 2. Network services ATCA blade

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.

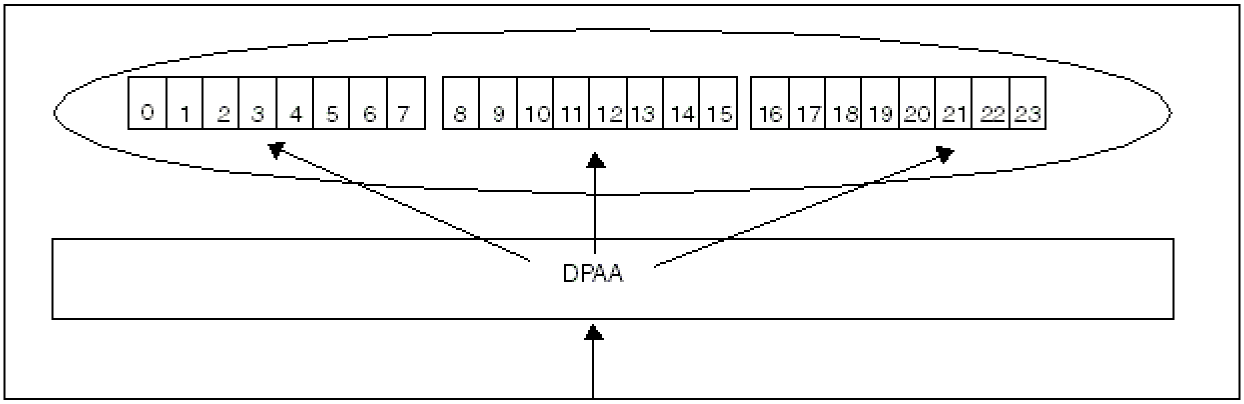

Figure 5. 24 vCPU AMP or SMP with affinity

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.

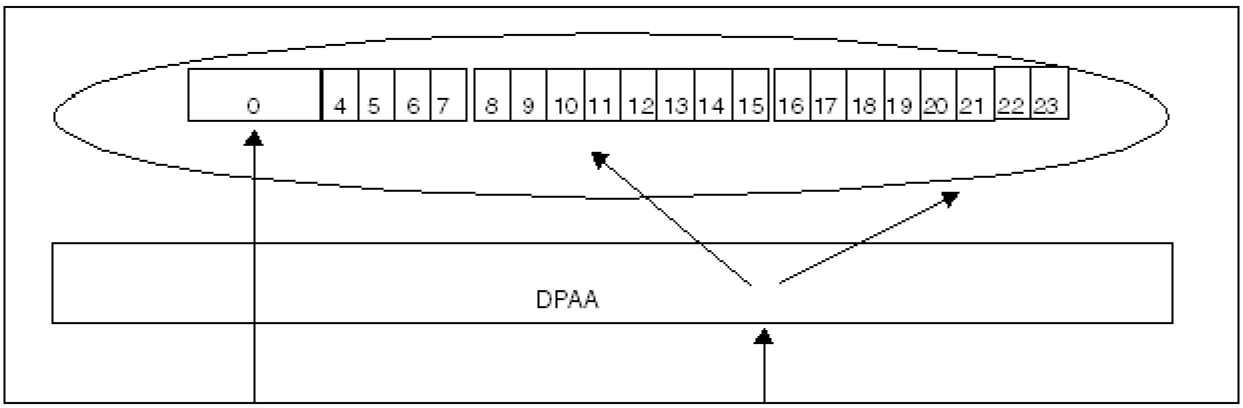

Figure 6. Mixed SMP and AMP option 1

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.

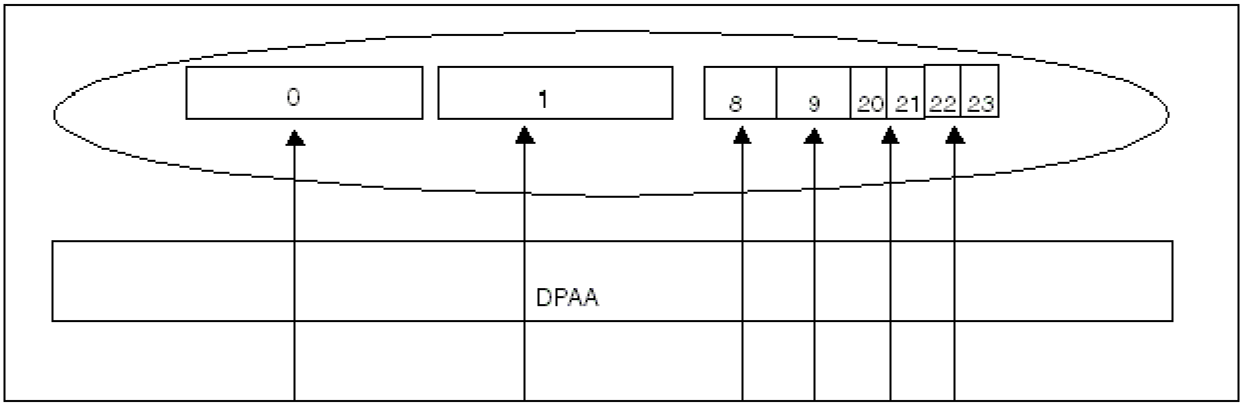

Figure 7. Mixed SMP and AMP option 2

5 Chip features

This section describes the key features and functionalities of the T4240 chip. See the T4160 and T4080 appendices for those device's specific block diagrams.

5.1 Block diagram

This figure shows the major functional units within the chip.

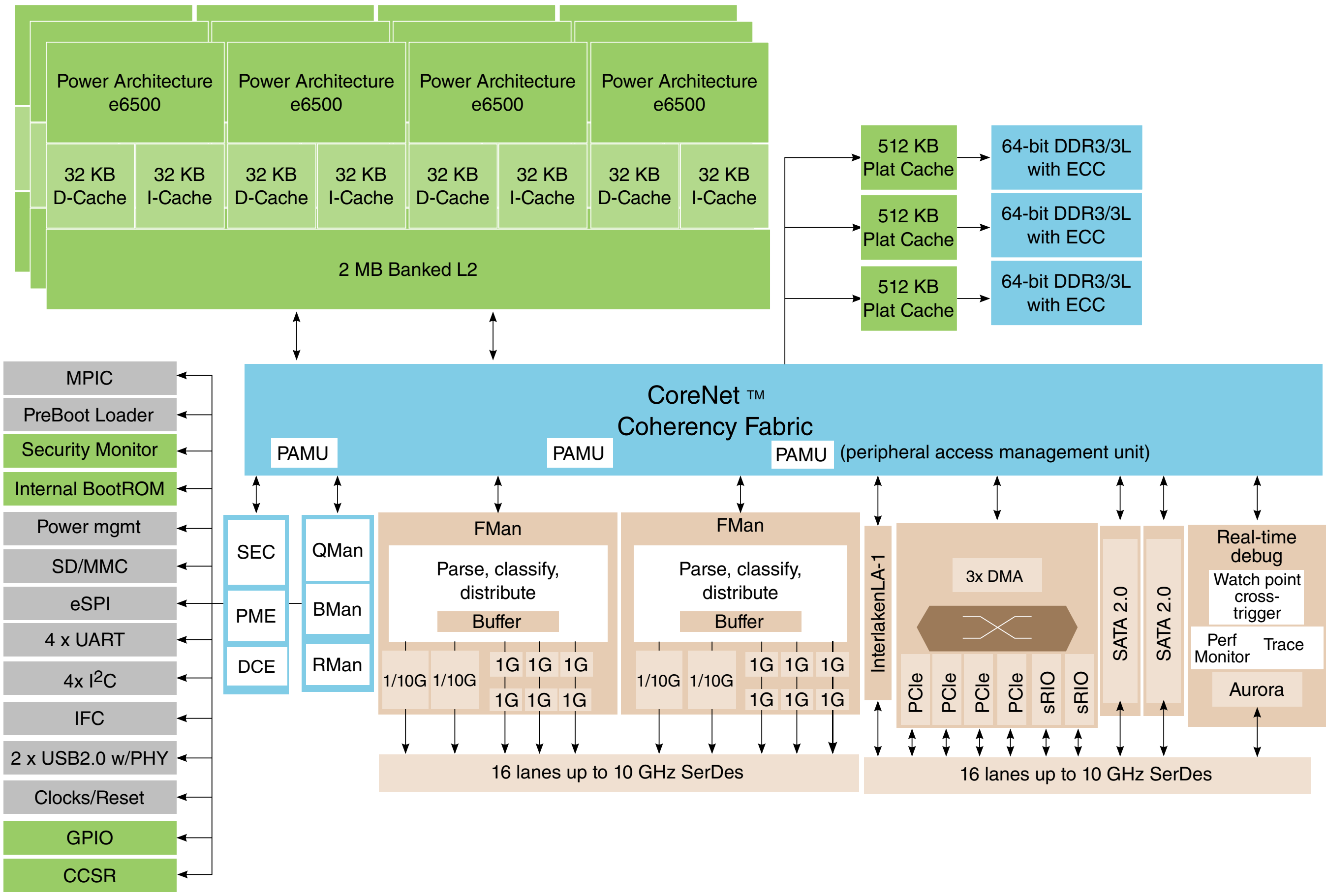

Figure 8. T4240 block diagram

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

5.9.2 Serial RapidIO

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

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

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

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

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

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

5.9.3 SATA

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

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

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

5.9.4 Interlaken Look-Aside Controller (LAC) and interface

Interlaken Look-Aside is a high speed serial channelized chip-to-chip interface. To facilitate 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

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.

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

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

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

- Public Key Hardware Accelerators (PKHA)
    - RSA and Diffie-Hellman (to 4096b)
    - Elliptic curve cryptography (1023b)
- Data Encryption Standard Accelerators (DESA)
    - DES, 3DES (2-key, 3-key)
    - ECB, CBC, OFB, and CFB modes
- Advanced Encryption Standard Accelerators (AESA)
    - Key lengths of 128-bit, 192-bit, and 256-bit
    - ECB, CBC, CTR, CCM, GCM, CMAC, OFB, CFB, xcbc-mac, and XTS
- ARC Four Hardware Accelerators (AFHA)
    - Compatible with RC4 algorithm
- Message Digest Hardware Accelerators (MDHA)
    - SHA-1, SHA-256, 384, 512-bit digests
    - MD5 128-bit digest
    - HMAC with all algorithms
- Kasumi/F8 Hardware Accelerators (KFHA)
    - F8, F9 as required for 3GPP
    - A5/3 for GSM and EDGE, GEA-3 for GPRS
- Snow 3G Hardware Accelerators (SNOWf8 and SNOWf9)
    - Implements Snow 3.0, F8 and F9 modes
- ZUC Hardware Accelerators (ZUCE and ZUCA)
    - Implements 128-EEA3 & 128-EIA3
- CRC Unit
    - Standard and user-defined polynomials
- Random Number Generator
    - Incorporates TRNG entropy generator for seeding and deterministic engine (SHA-256)
    - Supports random IV generation

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

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

    - 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)

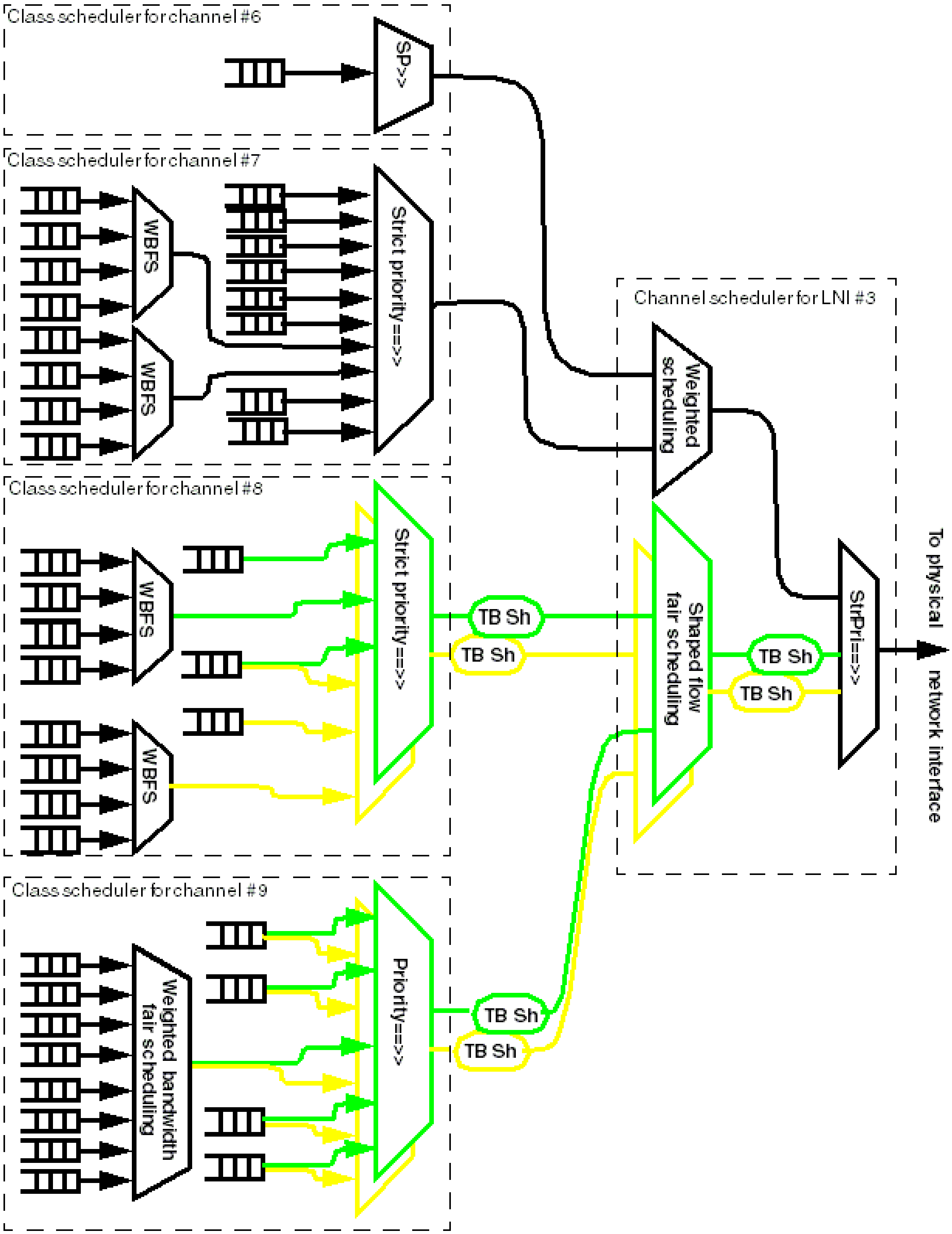

Figure 13. CEETM scheduler: illustrative configuration scenario

Figure 13 illustrates the following scenario:

- 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

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).

5.13 Debug support

The reduced number of external buses enabled by the move to multicore chips greatly simplifies board level lay-out and eliminates many concerns over signal integrity. Even though the board designer may embrace multicore CPUs, software engineers have real concerns over the potential to lose debug visibility. Despite the problems external buses can cause for the hardware engineer, they provide software developers with the ultimate confirmation that the proper instructions and data are passing between processing elements.

Processing on a multicore chip with shared caches and peripherals also leads to greater concurrency and an increased potential for unintended interactions between device components. To ensure that software developers have the same or better visibility into the device as they would with multiple discrete communications processors, Freescale developed an Advanced Multicore Debug Architecture.

The debugging and performance monitoring capability enabled by the device hardware coexists within a debug ecosystem that offers a rich variety of tools at different levels of the hardware/software stack. Software development and debug tools from Freescale (CodeWarrior), as well as third-party vendors, provide a rich set of options for configuring, controlling, and analyzing debug and performance related events.

6 Conclusion

Featuring 24 virtual cores, and based on the dual-threaded e6500 Power Architecture core, the T4240 processor, along with its 16 (T4160) and 8 (T4080) virtual-core variants, offers frequencies up to 1.8 GHz, large caches, hardware acceleration, and advanced system peripherals. All three devices target applications that benefit from consolidation of control and data plane processing in a single chip. In addition, each e6500 core implements the Freescale AltiVec technology SIMD engine, dramatically boosting the performance of math-intensive algorithms without using additional DSP components on the board. A wide variety of applications can benefit from the processing, I/O integration, and power management offered for the T4 series processors. Similar to other QorIQ devices, the T4 family processors' high level of integration offers significant space, weight, and power benefits compared to multiple discrete devices. Freescale also offers fully featured development support, which includes the QorIQ T4240 QDS Development System, QorIQ T4240 Reference Design Board, Linux SDK for QorIQ Processors, as well as popular operating systems and development tools from a variety of vendors. See the Freescale website for the latest information on tools and SW availability.

For more information about the QorIQ T4 family, contact your Freescale sales representative.

Appendix A T4160

A.1 Introduction

The T4160 is a lower power version of the T4240. The T4160 combines eight dual threaded Power Architecture e6500 cores and two memory complexes (CoreNet platform cache and DDR3 memory controller) with the same high-performance datapath acceleration, networking, and peripheral bus interfaces.

This figure shows the major functional units within the chip.

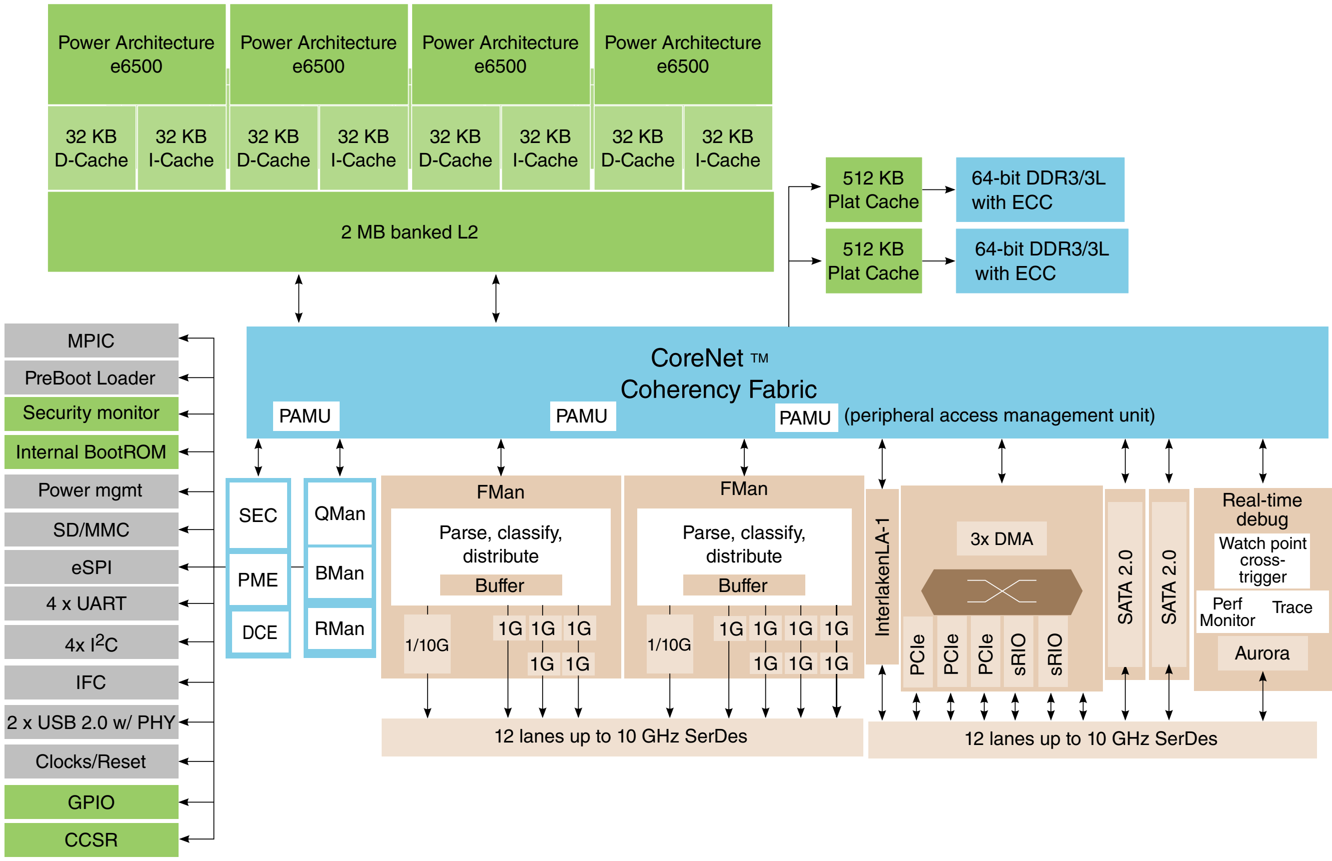

Figure B-1. T4080 block diagram

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.

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