E·XFL



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	Obsolete
Core Processor	PowerPC e6500
Number of Cores/Bus Width	24 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	Boot Security, Cryptography, Secure Fusebox, Secure Debug, Tamper Detection, Volatile key Storage
Package / Case	1932-BBGA, FCBGA
Supplier Device Package	1932-FCPBGA (45x45)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/t4240nsn7pqb

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong





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.



Multicore processing options



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.



wuncore processing options



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.



Ump features



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.



- 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.





5.7.1.1 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 I²C boot sequencer but on behalf of a large number of interfaces.

It supports IFC, I²C, 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 (I²C, 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



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



Table 6.	Parser header types (continued)	
		_

Header type	Definition
	For example, a frame that always contains a proprietary header before the Ethernet header would be non-self-describing. Both self-describing and non-self-describing headers are supported by means of parsing rules in the FMan.
Proprietary	Can be defined as being self-describing or non-self-describing

The underlying notion is that different frames may require different treatment, and only through detailed parsing of the frame can proper treatment be determined.

Parse results can (optionally) be passed to software.

5.10.5.1.2 FMan distribution and policing

After parsing is complete, there are two options for treatment, as shown in this table.

Treatment	Function	Benefits
Hash	 Hashes select fields in the frame as part of a spreading mechanism. The result is a specific frame queue identifier. To support added control, this FQID can be indexed by values found in the frame, such as TOS or p-bits, or any other desired field(s). 	Useful when spreading traffic while obeying QoS constraints is required
Classification look-up	 Looks up certain fields in the frame to determine subsequent action to take, including policing. The FMan contains internal memory that holds small tables for this purpose. The user configures the sets of lookups to perform, and the parse results dictate which one of those sets to use. Lookups can be chained together such that a successful look-up can provide key information for a subsequent look-up. After all the look-ups are complete, the final classification result provides either a hash key to use for spreading, or a FQ ID directly. 	 Useful when hash distribution is insufficient and a more detailed examination of the frame is required Can determine whether policing is required and the policing context to use

Table 7. Post-parsing treatment options

Key benefits of the FMan policing function are as follows:

- Because the FMan has up to 256 policing profiles, any frame queue or group of frame queues can be policed to either drop or mark packets if the flow exceeds a preconfigured rate.
- Policing and classification can be used in conjunction to mitigate Distributed Denial of Service Attack (DDOS).
- The policing is based on the two-rate-three-color marking algorithm (RFC2698). The sustained and peak rates, as well as the burst sizes, are user-configurable. Therefore, the policing function can rate-limit traffic to conform to the rate that the flow is mapped to at flow set-up time. By prioritizing and policing traffic prior to software processing, CPU cycles can focus on important and urgent traffic ahead of other traffic.

Each FMan also supports PCD on traffic arriving from within the chip. This is referred to as off-line parsing, and it is useful for reclassification following decapsulation of encrypted or compressed packets.

FMan PCD supports virtualization and strong partitioning by delaying buffer pool selection until after classification. In addition to determining the FQ ID for the classified packet, the FMan also determines the 'storage profile.' Configuration of storage profiles (up to 32 per physical port) allows the FMan to store received packets using buffer pools owned by a single software partition, and enqueue the associated Frame Descriptor to a frame queue serviced by only that software partition.



Crip features

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.



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

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

5.10.6 DPAA capabilities

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

5.10.6.1 Ingress policing and congestion management

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

FMan policing capabilities include the following:

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

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

5.10.6.2 Customer-edge egress-traffic management (CEETM)

Customer-edge egress-traffic management (CEETM) is a DPAA enhancement first appearing in the T4240. T4240 continues to support the work queue and frame queue scheduling functionality available in the P4080 and other first generation QorIQ chips, but introduces alternative 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)





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

5.10.6.2.2 CEETM configuration

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

NOTE

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



Crip features





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



5.12 Advanced power management

Power dissipation is always a major design consideration in embedded applications; system designers need to balance the desire for maximum compute and IO density against single-chip and board-level thermal limits.

Advances in chip and board level cooling have allowed many OEMs to exceed the traditional 30 W limit for a single chip, and Freescale's flagship T4240 multicore chip, has consequently retargeted its maximum power dissipation. A top-speed bin T4240 dissipates approximately 2x the power dissipation of the P4080; however, the T4240 increases computing performance by ~4x, yielding a 2x improvement in DMIPs per watt.

Junction temperature is a critical factor in comparing embedded processor specifications. Freescale specs max power at 105C junction, standard for commercial, embedded operating conditions. Not all multicore chips adhere to a 105C junction for specifying worst case power. In the interest of normalizing power comparisons, the chip's typical and worst case power (all CPUs at 1.8 GHz) are shown at alternate junction temperatures.

To achieve the previously-stated 2x increase in performance per watt, the chip implements a number of software transparent and performance transparent power management features. Non-transparent power management features are also available, allowing for significant reductions in power consumption when the chip is under lighter loads; however, non-transparent power savings are not assumed in chip power specifications.

5.12.1 Transparent power management

This chip's commitment to low power begins with the decision to fabricate the chip in 28 nm bulk CMOS. This process technology offers low leakage, reducing both static and dynamic power. While 28 nm offers inherent power savings, transistor leakage varies from lot to lot and device to device. Leakier parts are capable of faster transistor switching, but they also consume more power. By running devices from the leakier end of the process spectrum at less than nominal voltage and devices from the slower end of the process spectrum at higher nominal voltage, T4240-based systems can achieve the required operating frequency within the specified max power. During manufacturing, Freescale will determine the voltage required to achieve the target frequency bin and program this Voltage ID into each device, so that initialization software can program the system's voltage regulator to the appropriate value.

Dynamic power is further reduced through fine-grained clock control. Many components and subcomponents in the chip automatically sleep (turn off their clocks) when they are not actively processing data. Such blocks can return to full operating frequency on the clock cycle after work is dispatched to them. A portion of these dynamic power savings are built into the chip max power specification on the basis of impossibility of all processing elements and interfaces in the chip switching concurrently. The percent switching factors are considered quite conservative, and measured typical power consumption on QorIQ chips is well below the maximum in the data sheet.

As noted in Frame Manager and network interfaces, the chip supports Energy-Efficient Ethernet. During periods of extended inactivity on the transmit side, the chip transparently sends a low power idle (LPI) signal to the external PHY, effectively telling it to sleep.

Additional power savings can be achieved by users statically disabling unused components. Developers can turn off the clocks to individual logic blocks (including CPUs) within the chip that the system is not using. Based on a finite number of SerDes, it is expected that any given application will have some inactive Ethernet MACs, PCI Express, or serial RapidIO controllers. Re-enabling clocks to a logic block generally requires an chip reset, which makes this type of power management infrequent (effectively static) and transparent to runtime software.



5.12.2 Non-transparent power management

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

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

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

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

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

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

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

5.12.3 Cascade power management

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

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

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



Conclusion

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.



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

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

Appendix B T4080

B.1 Introduction

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

This figure shows the major functional units within the chip.





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.



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.