



Welcome to [E-XFL.COM](https://www.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 e5500
Number of Cores/Bus Width	2 Core, 64-Bit
Speed	2.0GHz
Co-Processors/DSP	Security; SEC 4.2
RAM Controllers	DDR3, DDR3L
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	1Gbps (5), 10Gbps (1)
SATA	SATA 3Gbps (2)
USB	USB 2.0 + PHY (2)
Voltage - I/O	-
Operating Temperature	-40°C ~ 105°C (TA)
Security Features	Boot Security, Cryptography, Secure Fusebox, Secure JTAG, Secure Memory, Tamper Detection
Package / Case	1295-BBGA, FCBGA
Supplier Device Package	1295-FCPBGA (37.5x37.5)
Purchase URL	<a href="https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=p5020nxe1tnb">https://www.e-xfl.com/pro/item?MUrl=&amp;PartUrl=p5020nxe1tnb</a>

# 1 P5020 Application Use Cases

## 1.1 Router Control Processor

The following figure shows the P5020 in a linecard control plane application, where the linecard is part of a high-end network router.

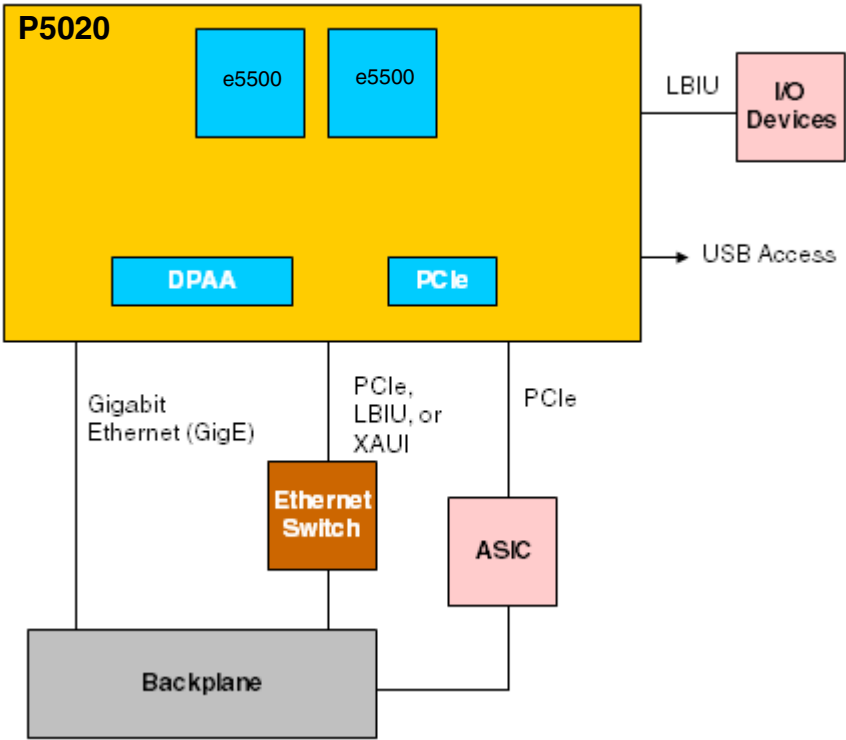


Figure 1. Control Plane Processor for a Router

# 1.2 DSP Farm Control Processor

The following figure shows a DSP farm enabled by the P5020 utilizing serial RapidIO.

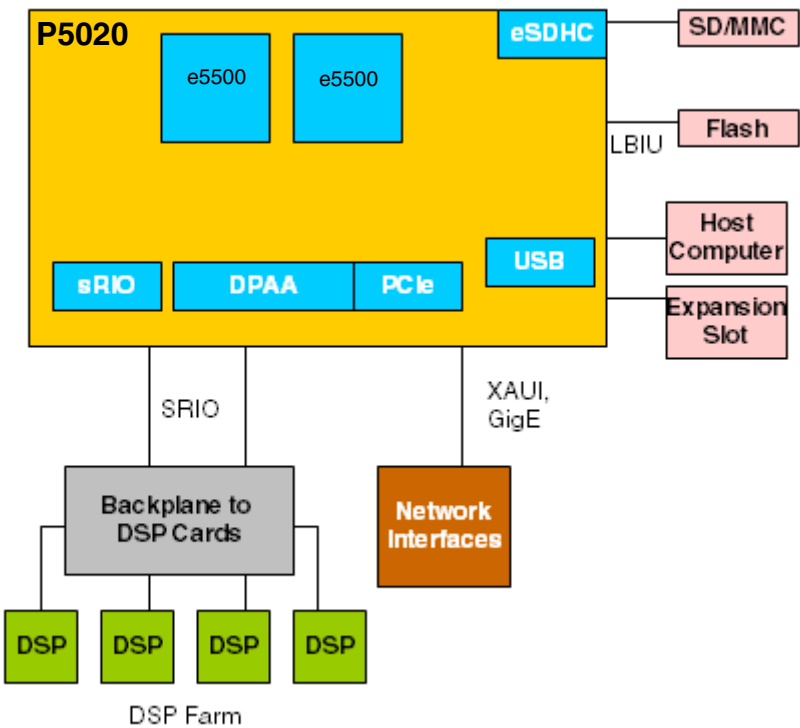
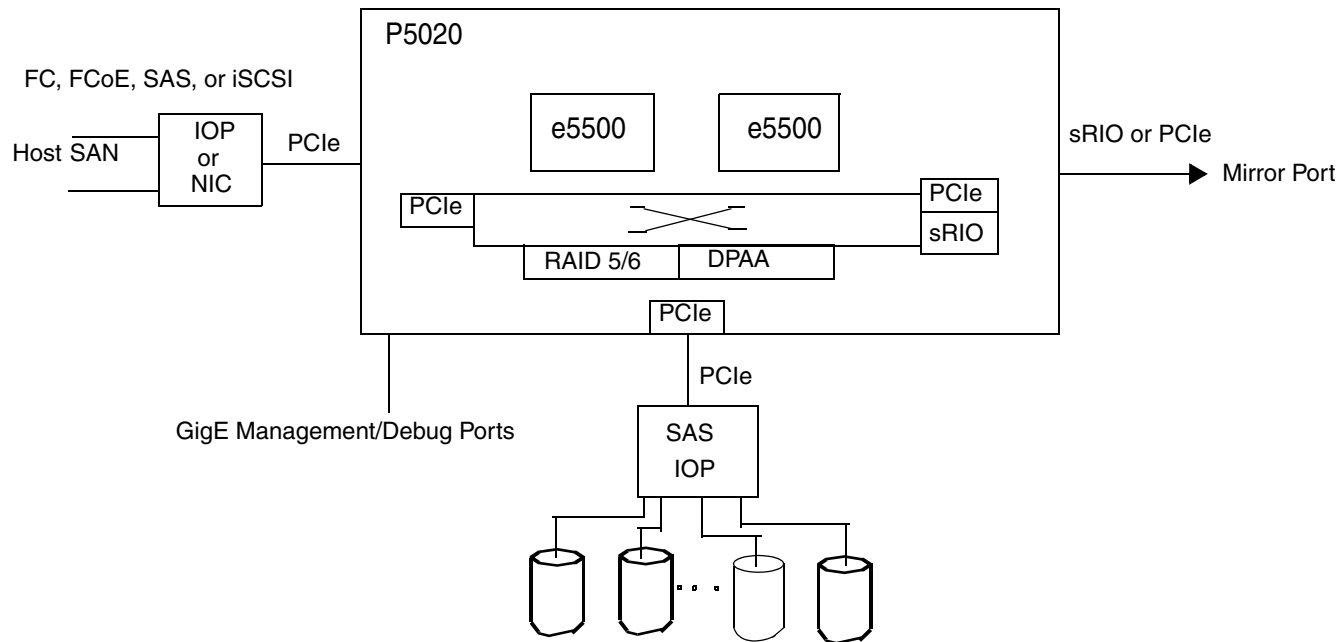


Figure 2. Control Plane Processor for a DSP Farm

# 1.3 SAN RAID 6 Controller

The following figure shows a RAID-enabled Disk Array Controller in an redundant active-active system for block-oriented storage systems. The P5020 Data Path Acceleration Architecture (DPAA) accelerates RAID 5/6 calculations and low-overhead data movement while optionally supporting data-at rest encryption and Data Integrity Field support.



**Figure 3. SAN RAID Controller**

## 2 P5020 Dual-Core Processing Options

The device cores can run either on an OS or run OS-less using a simple scheduler.

### 2.1 Running on an OS

There are different multi-processing options with the device cores running on an OS:

- Symmetric multi-processing
- Cooperative asymmetric multi-processing
  - Two copies of the same OS that are non-SMP enabled
  - Two separate operating systems

### 2.2 Running OS-Less Using a Simple Scheduler

It is also possible for one or more cores to run OS-less, using a simple scheduler. This is a likely scenario when cores are performing datapath operations with bounded real-time requirements. This use case is greatly enhanced by the provisioning of a 512-Kbyte private back-side L2 cache for each e5500 core. These caches can operate as a traditional unified cache, or be set to operate as instruction only, data only, or even locked and used as memory-mapped SRAM.

CPU cores operating asymmetrically can be run at asynchronous clock rates. Each processor can source its input clock from one of the multiple PLLs inside the P5020. This allows each core to operate at the minimum frequency required to perform its assigned function, saving power. The cores are also capable of running at half and quarter ratios of their input PLL frequency and can switch between PLLs and ratios

nearly instantaneously. This allows lightly utilized CPUs to be slowed (under software control) for power savings, rather than performing more complex task migration operations.

### 3 Features

#### 3.1 Block Diagram

The following figure shows the major functional units within the P5020.

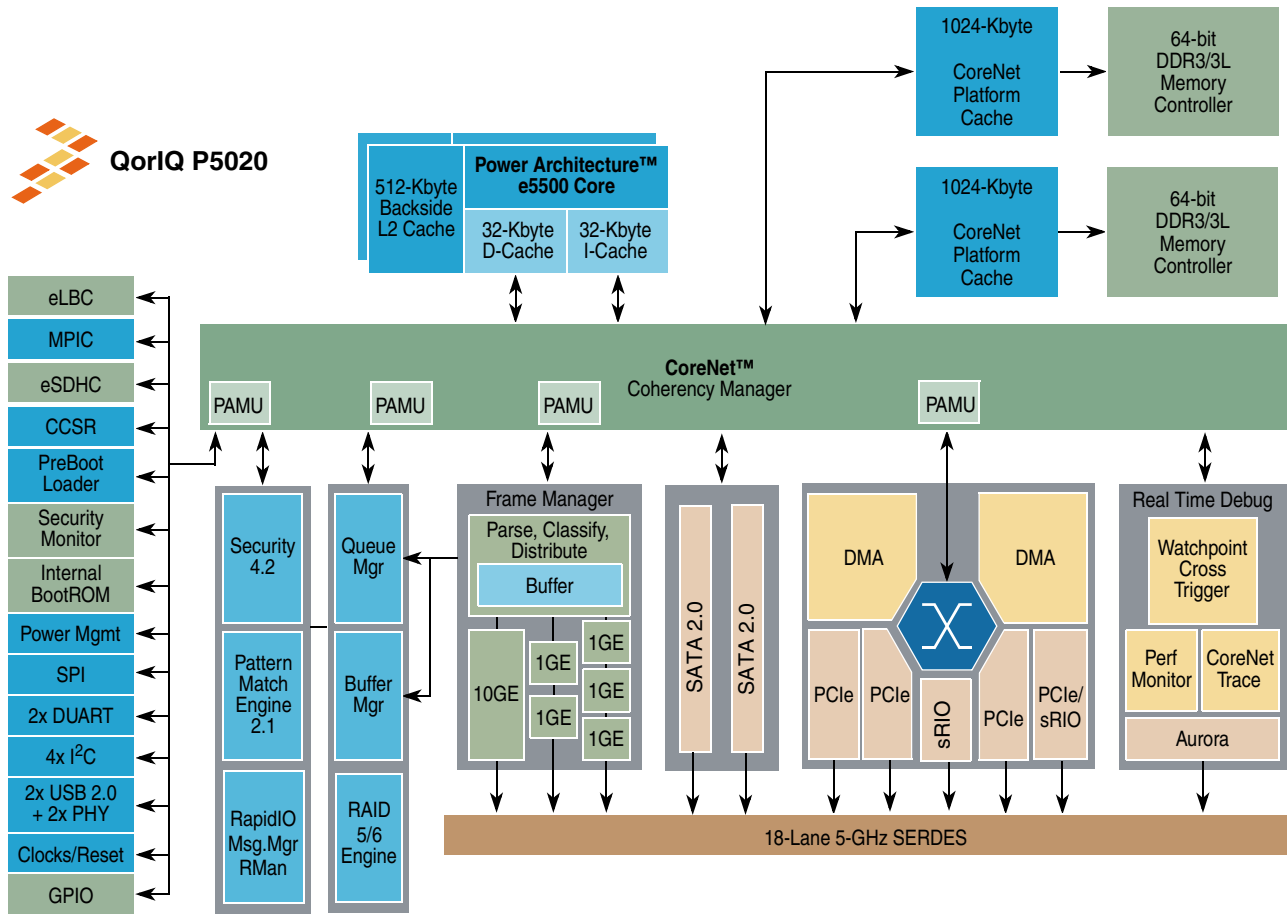


Figure 4. P5020 Preliminary Block Diagram

#### 3.2 P5020 Features Summary

The P5020 SoC includes the following functions and features:

- Two e5500 cores built on Power Architecture technology, each with a private 512-Kbyte private backside cache
  - Up to 2 GHz
  - Three levels of instructions:
    - User

- Two 4-channel DMA engines

### 3.3 P5020 Benefits

The P5020's e5500 cores can be combined as a fully-symmetric, multi-processing, system-on-a-chip, or they can be operated with varying degrees of independence to perform asymmetric multi-processing. Full processor independence, including the ability to independently boot and reset each e5500 core, is a defining characteristic of the device. 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.

### 3.4 Data Path Acceleration Architecture (DPAA) Benefits

While the two Power Architecture cores offer a major leap in available processor performance in many throughput-intensive, packet-processing networking applications, raw processing power is not enough to achieve multi-Gbps data rates. To address this, the P5020 uses Freescale's Data Path Acceleration Architecture (DPAA) (see [Section 3.9, "Data Path Acceleration Architecture \(DPAA\)"](#)), which significantly reduces data plane instructions per packet, enabling more CPU cycles to work on value-added services rather than repetitive low-level tasks. Combined with specialized accelerators for cryptography and pattern matching, the P5020 allows the user's software to perform complex packet processing at high data rates.

### 3.5 Critical Performance Parameters

The following table lists key performance indicators that define a set of values used to measure P5020 operation.

**Table 1. P5020 Critical Performance Parameters**

Indicator	Values(s)
Top speed bin core frequency	2.0 GHz
Maximum memory data rate	1.3 GHz (DDR3/3L) <sup>1</sup> <ul style="list-style-type: none"> <li>• 1.5-V for DDR3</li> <li>• 1.35-V for DDR3L</li> </ul>
Local bus	<ul style="list-style-type: none"> <li>• 3.3 V</li> <li>• 2.5 V</li> <li>• 1.8 V</li> </ul>
Operating junction temperature range	0–105 C
Package	1295-pin FC-PBGA (flip-chip plastic ball grid array)

**Notes:**

<sup>1</sup> Conforms to JEDEC standard

## 3.6 e5500 Core and Cache Memory Complex

Each e5500 is a superscalar dual issue processor, supporting out-of-order execution and in-order completion, which allows the Power Architecture e5500 to perform more instructions per clock than other RISC and CISC architectures.

### 3.6.1 e5500 Core Features

- Up to 2.0 GHz core clock speed
- 36 bit physical addressing
- 64 TLB SuperPages
- 512-entry, 4-Kbyte pages front end
- 3 Integer Units: 2 simple, 1 complex (integer multiply and divide)
- 64-byte cache line size
- L1 caches, running at same frequency of CPU
  - 32-Kbyte Instruction, 8-way
  - 32-Kbyte Data, 8-way
  - Both with data and tag parity protection
- Supports data path acceleration architecture (DPAA) data and context “stashing” into the L1 data cache and the backside L2 cache
- User, supervisor, and hypervisor instruction level privileges
- New processor facilities
  - Hypervisor APU
  - Classic double precision floating point unit
    - Uses 32 64-bit floating-point registers (FPRs) for scalar single- and double-precision floating-point arithmetic
    - Replaces the embedded floating-point facility (SPE) implemented on the e500v1 and e500v2
    - Designed to comply with IEEE Std. 754™1985 FPU for both single- and double-precision operations
  - “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
  - 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

The nonvolatile memory interfaces accessible by the PBL are as follows:

- The eLBC may be accessed by software running on the CPUs following boot; it is not dedicated to the PBL. It also can be used for both volatile (SRAM) and nonvolatile memory as well as a control and low-performance data port for external memory-mapped P5020s. See [Section 3.6.7, “Enhanced Local Bus Controller.”](#)
- The serial memory controllers may be accessed by software running on the CPUs following boot; they are not dedicated to the PBL. See [Section 3.6.7.1, “Serial Memory Controllers.”](#)

### 3.6.7 Enhanced Local Bus Controller

The enhanced local bus controller (eLBC) port connects to a variety of external memories, DSPs, and ASICs.

Key features of the eLBC include the following:

- Multiplexed 32-bit address and 32-bit data bus operating at up to 93 MHz
- Eight chip selects for eight external slaves
- Up to eight-beat burst transfers
- 8-, 16-, or 32-bit port sizes controlled by an internal memory controller
- Three protocol engines on a per-chip-select basis
- Parity support
- Default boot ROM chip select with configurable bus width (8-, 16-, or 32-bit)
- Support for parallel NAND and NOR flash

Three separate state machines share the same external pins and can be programmed separately to access different types of devices. Some examples are as follows:

- The general-purpose chip-select machine (GPCM) controls accesses to asynchronous devices using a simple handshake protocol.
- The user-programmable machine (UPM) can be programmed to interface to synchronous devices or custom ASIC interfaces.
- The NAND flash control machine (FCM) further extends interface options.
- Each chip select can be configured so that the associated chip interface is controlled by the GPCM, UPM, or FCM controller.

All controllers can be enabled simultaneously. The eLBC internally arbitrates among the controllers, allowing each to read or write a limited amount of data before allowing another controller to use the bus.

#### 3.6.7.1 Serial Memory Controllers

In addition to the parallel NAND and NOR flash supported by means of the eLBC, the P5020 supports serial flash using SPI and SD/MMC/eMMC card. 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.



## 3.7 Universal Serial Bus (USB) 2.0

The two USB 2.0 controllers with integrated PHY provide point-to-point connectivity complying with the USB specification, Rev. 2.0. Each USB controller can be configured to operate as a stand-alone host, and USB #2 can be configured as a stand-alone device, or with both host and device functions operating simultaneously.

Key features of the USB 2.0 controller include the following:

- Compatible with USB specification, Rev. 2.0
- Supports full-speed (12 Mbps), and low-speed (1.5 Mbps) operations
- Supports the required signaling for the USB transceiver macrocell interface (UTMI). The PHY interfacing to the UTMI is an internal PHY.
- Both controllers support operation as a stand-alone USB host controller
  - Support USB root hub with one downstream-facing port
  - Enhanced host controller interface (EHCI)-compatible
- One controller supports operation as a stand-alone USB device
  - Supports one upstream-facing port
  - Supports six programmable USB endpoints

The host and device functions are both configured to support all four USB transfer types:

- Bulk
- Control
- Interrupt
- Isochronous

## 3.8 High-Speed Peripheral Interface Complex

All high-speed peripheral interfaces connect via 18 lanes of 5-GHz SerDes to a common crossbar switch referred to as OCeAN. Two high-speed I/O interface standards are supported: PCI Express (PCIe), and Serial RapidIO (sRIO). The P5020 integrates the following:

- Four PCIe controllers
- Two Serial RapidIO controllers
- RapidIO message manager (RMan).

### 3.8.1 PCI Express Controllers

Each of the four PCIe interfaces is compliant with the *PCI Express Base Specification Revision 2.0*. Key features of the PCIe interface include the following:

- Power-on reset configuration options allow root complex or endpoint functionality.
- The physical layer operates at 2.5 or 5 Gbaud data rate per lane.
- Receive and transmit ports operate independently, with an aggregate theoretical bandwidth of 32 Gbps.

- x8, x4, x2, and x1 link widths supported
- Both 32- and 64-bit addressing and 256-byte maximum payload size
- Full 64-bit decode with 36-bit wide windows
- Inbound INTx transactions
- Message Signaled Interrupt (MSI) transactions

### 3.8.2 Serial RapidIO

The Serial RapidIO interface is based on the *RapidIO Interconnect Specification, Revision 1.3*, with features from 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.

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

- Support for *RapidIO Interconnect Specification, Revision 1.3* (all transaction flows and priorities)
- 1x, 2x, and 4x LP-serial link interfaces, with transmission rates of 2.5, 3.125, or 5.0 Gbaud (data rates of 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

#### 3.8.2.1 RapidIO Message Manager (RMan)

The key features of the RapidIO message manager (RMan) include the following:

- Manages two inbox/outbox mailboxes (queues) for data and one doorbell message structure
- Can multi-cast a single-segment 256-byte message to up to 32 different destination DevIDs
- Has four outbound segmentation units supporting RapidIO Type 5–6 and Type 8–11

### 3.8.3 Serial ATA (SATA) 2.0 Controllers

The key features of each of the two SATA include the following:

- Designed to comply with Serial ATA 2.6 Specification
- Supports host SATA I per spec Rev 1.0a
  - OOB
  - Port multipliers
  - ATAPI 6+

## 3.9.3 DPAA Terms and Definitions

The following table lists common DPAA terms and their definitions.

**Table 4. DPAA Terms and Definitions**

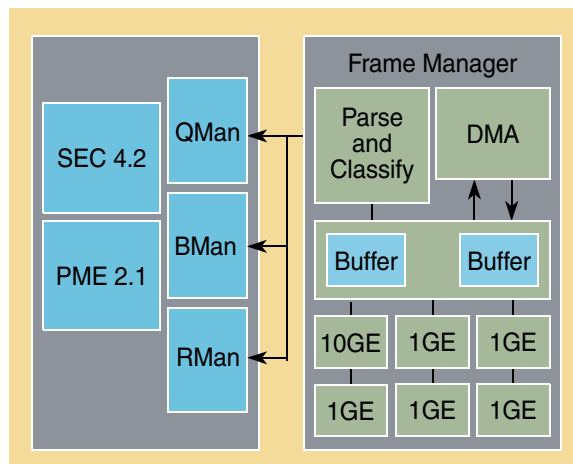
Term	Definition	Graphic Representation
Buffer	Region of contiguous memory, allocated by software, managed by the DPAA BMan	
Buffer pool	Set of buffers with common characteristics (mainly size, alignment, access control)	
Frame	Single buffer or list of buffers that hold data, for example, packet payload, 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, or SEC.	—
Pool channel	A channel statically assigned to a group of end points, from which any of the end points may dequeue frames.	—

## 3.9.4 Major DPAA Components

The Data Path Acceleration Architecture (DPAA) includes the following major components:

- [Section 3.9.4.1, “Frame Manager \(FMan\)”](#)
- [Section 3.9.4.2, “Queue Manager \(QMan\)”](#)
- [Section 3.9.4.3, “Buffer Manager \(BMan\)”](#)
- [Section 3.9.4.6, “RapidIO Message Manager \(RMan\)”](#)

- [Section 3.9.4.4, “Security Engine \(SEC 4.2\)”](#)
- [Section 3.9.4.5, “Pattern Matching Engine \(PME 2.1\)”](#)



**Figure 5. QorIQ Data Path Acceleration Architecture (DPAA)**

### 3.9.4.1 Frame Manager (FMan)

The Frame Manager (FMan) combines the Ethernet network interfaces with packet distribution logic to provide intelligent distribution and queuing decisions for incoming traffic. This integration allows the FMan to perform configurable parsing and classification of the incoming frame with the purpose of selecting the appropriate input frame queue for expedited processing by a CPU or pool of CPUs.

#### 3.9.4.1.1 FMan Network Interfaces

The FMan integrates five data path, tri-speed Ethernet controllers (dTSECs) and one 10-Gbit Ethernet controller.

Note that the more basic parsing and filing capability found in prior PowerQUICC eTSECs is removed from the MACs themselves, and aggregated in the more flexible and robust parsing and classification logic described in [Section 3.9.4.1.2, “FMan Parse Function.”](#)

The Ethernet controllers support the following:

- Programmable CRC generation and checking
- RMON statistics
- Jumbo frames of up to 9.6 Kbytes

They are designed to comply with IEEE Std 802.3®, IEEE 802.3u, IEEE 802.3x, IEEE 802.3z, IEEE 802.3ac, IEEE 802.3ab, and additionally the 1Gbps MACs support IEEE-1588 v2 (clock synchronization over Ethernet).

The dTSECs are capable of full- and half-duplex Ethernet support (1000 Mbps supports only full duplex); the 10-Gbit MAC is a single-speed full duplex. It supports IEEE 802.3 full-duplex flow control (automatic PAUSE frame generation or software-programmed PAUSE frame generation and recognition).

- 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 for mitigating Distributed Denial of Service Attack (DDOS).
- The policing is based on two-rate-three-color marking algorithm (RFC2698). The sustained and peak rates as well as the burst sizes are user-configurable. Hence, the policing function can rate-limit traffic to conform to the rate the flow is mapped to at flow set-up time. By prioritizing and policing traffic prior to software processing, CPU cycles can be focused on the important and urgent traffic ahead of other traffic.

### **3.9.4.2 Queue Manager (QMan)**

The Queue Manager (QMan) is the main component in the DPAA that allows 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 CPU cores. The QMan features are as follows:

- 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
- 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
- Lossless flow control for ingress network interfaces
- Congestion avoidance (RED/WRED) and congestion management with tail discard and up to 256 congestion groups with each group composed of a user-configured number of frame queues.

### **3.9.4.3 Buffer Manager (BMan)**

The buffer manager (BMan) manages pools of buffers on behalf of software for both hardware (accelerators and network interfaces) and software use. The BMan features are as follows:

- Common interface for software and hardware
- Guarantees atomic access to shared buffer pools
- Supports 32 buffer pools. Software and hardware buffer consumers can request both 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 that optimizes cache usage and allocation

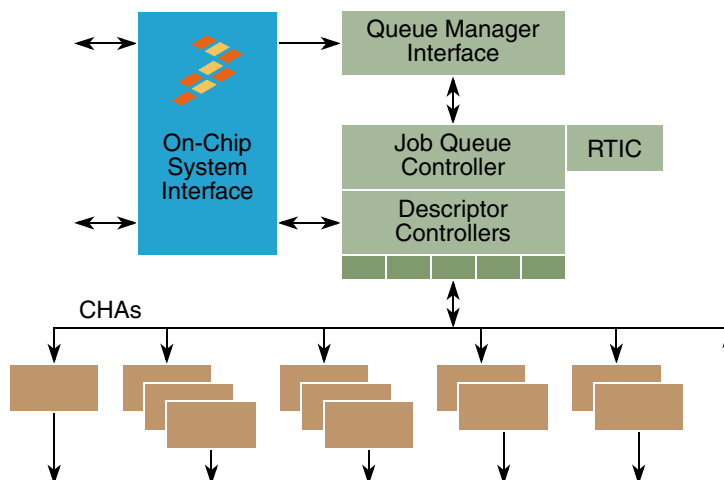


Figure 6. SEC 4.2 Block Diagram

### 3.9.4.5 Pattern Matching Engine (PME 2.1)

The PME is a self-contained hardware module capable of autonomously scanning data from streams for patterns that match a specification in a database dedicated to it. The PME 2.1 is an updated version of the PME used in previous members of the PowerQUICC family. Specific updates include the following:

- QMan interface supporting the DPAA Queue Interface Driver
- 2x increase in the number of patterns supported (16 Kbytes to 32 Kbytes)
- Increase in number of stateful rules supported (8 Kbytes to 16 Kbytes)
- Raw scanning performance is ~ 5 Gbps.

Patterns that can be recognized, or “matched,” by the PME are of two general forms:

- Byte patterns are simple matches such as “abcd123” existing in both the data being scanned and in the pattern specification database.
- Event patterns are a sequence of multiple byte patterns. In the PME, event patterns are defined by stateful rules.

#### 3.9.4.5.1 PME Regular Expressions (Regex)

The PME specifies patterns of bytes as regular expressions (regex). The P5020 (by means of an online or offline process) converts Regex patterns into the PME’s pattern specification database. Generally, there is a one-to-one mapping between a regex and a PME byte pattern. The PME’s use of regex pattern matching offers built-in case-insensitivity and wildcard support with no pattern explosion, while the PME’s NFA-style architecture offers fast pattern database compilation and fast incremental updates. Up to 32,000 regex patterns are supported, each up to 128 bytes long. The 32,000 regex patterns can be combined by means of stateful rules to detect a far larger set of event patterns. Comparative compilations against DFA style regex engines have shown that 300,000 DFA pattern equivalents can be achieved with ~8000 PME regexes with stateful rules.

### 3.9.4.5.2 PME Match Detection

Within the PME, match detection proceeds in stages. The key element scanner performs initial byte pattern matching, with handoff to the data examination engine for elimination of false positives through more complex comparisons. As the name implies, the stateful rule engine receives confirmed basic matches from the earlier stages, and monitors a stream for addition for subsequent matches that define an event pattern.

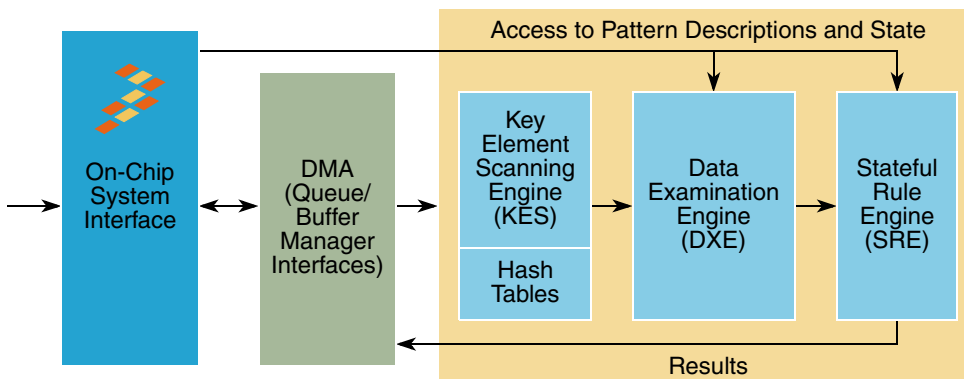


Figure 7. PME 2.1 Block Diagram

### 3.9.4.6 RapidIO Message Manager (RMan)

The RapidIO message manager (RMan) produces and consumes Type 8 Port-write, Type 9 Data Streaming, Type 10 Doorbells and Type 11 Messaging traffic and is capable of producing Type 5 NWRITE and Type 6 SWRITE transactions.

For inbound traffic, the RMan supports up to 17 open reassembly contexts as a arbitrary mix of Type 9, and Type 11 traffic.

As ingress packets arrives at the RMan, they are compared against up to 64 classification rules to determine the target queue. These rules support Type 8, 9, 10 and 11 transaction types. They may be wildcarded and are configured as masks over selected header fields. The following fields are maskable as part of each classification rule:

Transaction types:

- RapidIO port
- Source ID
- Destination ID
- Flow level

Type 9 messaging-specific fields:

- Class-of-service (CoS)
- StreamID

Type 11 messaging-specific fields:

- Mailbox

4-Kbyte granularity); other CPU MMUs are not configured for access to the other CPU's private memory range. When two CPUs 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 SoC, achieving strong partitioning should not require the developer to map functions onto cores 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 MMU also includes embedded Hypervisor extensions.

Each core MMU supports three levels of instructions:

- User
- Supervisor (OS)
- Hypervisor: An embedded Hypervisor micro-kernel (provided by Freescale as source code) runs unobtrusively beneath the various OSes running on the CPUs, 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 core attempts to overwrite important P5020 configuration registers (including CPU MMUs), the embedded Hypervisor blocks the write. Other examples of embedded Hypervisor managed resources are high- and low-speed peripheral interfaces (PCIe, UART) if those resources are not dedicated to a single CPU/partition.

### 3.10.3 Peripheral Access Management Unit (PAMU)

The P5020 includes a distributed function collectively referred to as the peripheral access management unit (PAMU), which provides address translation and access control for all bus masters in the system (PME, SEC, FMan, and so on). The PAMU access control can be one of the following:

- Absolute—The FMan, PME, SEC, and other bus masters can never access memory range XYZ.
- Conditional—Based on the Partition ID of the CPU that programmed the bus master

Being MMU-based, the embedded Hypervisor is only able to stop unauthorized software access attempts. Internal components with bus mastering capability also need to be prevented from reading and writing to specific memory regions. These devices do not spontaneously generate access attempts, but, if programmed to do so by buggy or malicious software, any of them could overwrite sensitive configuration registers and crash the system.

### 3.10.4 Secure Boot and Sensitive Data Protection

The core MMUs and PAMU allow the device to enforce a consistent set of memory access permissions on a per-partition basis. When combined with embedded Hypervisor for safe sharing of resources, the P5020 becomes highly resilient when poorly tested or malicious code is run. For system developers building high reliability/high security platforms, rigorous testing of code of known origin is the norm.



be disabled by means of the DEVDIS register. Re-enabling clocks to a logic block requires an SoC reset, which makes this type of power management operation infrequent (effectively static).

### 3.11.3 Avoiding Full System Failure Due to Thermal Overload

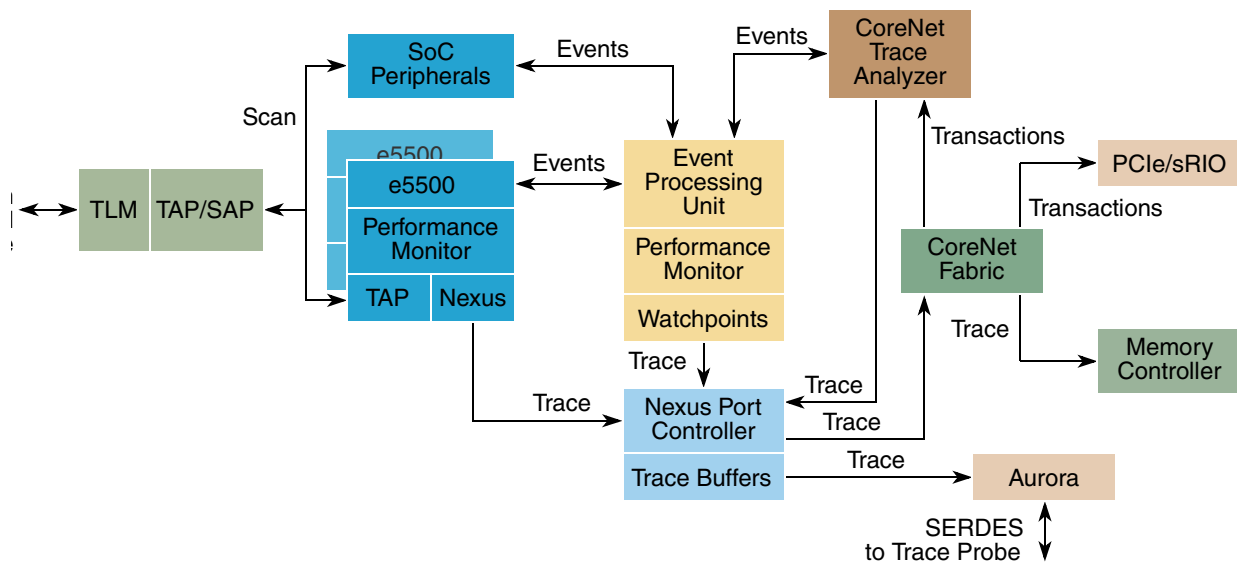
Changing PLL frequency dividers (/2, /4) can be used to achieve large and rapid reductions in dynamic power consumptions, and with the help of external temperature detection circuitry, can serve as a thermal overload protection scheme. If the junction temperature or system ambient temperature of the device achieves some critical level, external temperature detection circuitry can drive a high-priority interrupt into the P5020, causing it to reduce selected CPU frequencies by half or more. This allows the system to continue to function in a degraded mode, rather than failing entirely. This technique is much simpler than turning off selected CPUs, which can involve complex task migration in an AMP system. When system temperatures have been restored to safe ranges, all CPUs can be returned to normal frequency within a few clock cycles.

When less drastic frequency changes are desired, software can switch the CPU to a slower speed PLL, such as 1 GHz versus 1.5 GHz. Many cores could be switched to a slower PLL during periods of light traffic, with the ability to immediately return those cores to the full rate PLL should traffic suddenly increase. The more traditional Power Architecture single-core power management modes (such as Core Doze, Core Nap, and Core Sleep) are also available in the core.

## 3.12 Debug Support

The reduced number of external buses enabled by the move to multicore SoCs greatly simplifies board level lay-out and eliminates many concerns over signal integrity. While 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 HW engineer, they provide software developers with the ultimate confirmation that the proper instructions and data are passing between processing elements.

Processing on a multicore SOC with shared caches and peripherals also leads to greater concurrency and an increased potential for unintended CPU interactions. To ensure that software developers have the same or better visibility into the P5020 as they would with multiple discrete Freescale communications processors, Freescale developed the debug architecture shown in the following figure.



**Figure 8. Debug Architecture**

Debug features include the following:

- Debug and performance monitoring registers in both the core and platform
  - Accessible by target resident debug software and non-resident debug tools
  - Capable of generating debug interrupts and trace event messages
- Run control with enhancements
  - Classic
  - Cross-core and SoC watchpoint triggering
- High speed trace port (Aurora-based)
  - Supports Nexus class 2 instruction trace including timestamps
    - Process id trace, watchpoint trace
  - Supports “light” subset of Nexus class 3 data trace
    - Enabled by cores, by event triggers, by Instruction Address Compare/Data Address Compare events
  - Data Acquisition Trace
    - Compatible with Nexus class 3
    - Instrumented code can generate data trace messages for values of interest
    - Performed by writing values to control registers within each core
  - Watchpoint Trace
    - Can generate cross-core correlated breakpoints
    - Breakpoint on any core can halt execution of selected additional cores with minimal skid
- CoreNet transaction analyzer
  - Provides visibility to transactions across CoreNet (CoreNet fabric is otherwise transparent to software)

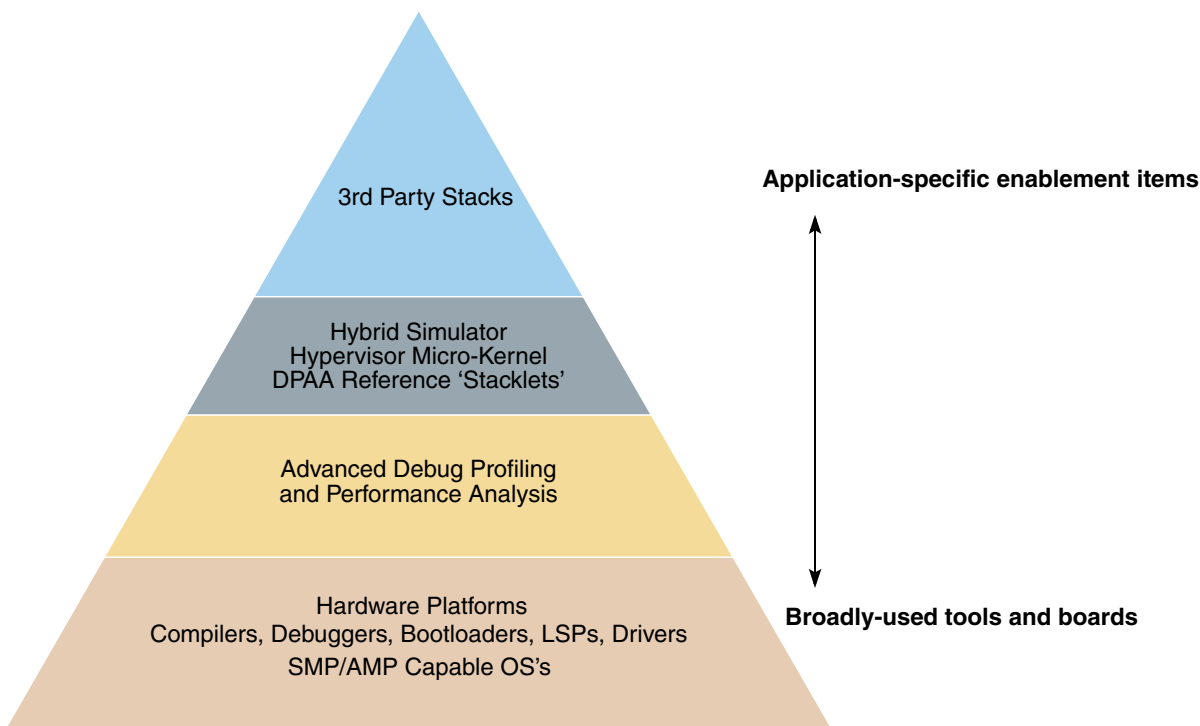
- Generates trace messages to Nexus Port Controller
- Supports filtering of accesses of interest
  - Data Address Compare (4)
  - Data Value Compare (2)
  - Transaction Attribute Compare (2)

## 4 Developer Environment

Software developers creating solutions with the Power Architecture technology have long benefited from a vibrant support ecosystem, including high quality tools, OSes, and network protocol stacks. Freescale is working with our ecosystem partners to ensure that this remains the case for multicore, Power Architecture-based products, including the P5020.

The various levels of the developer environment are shown in [Figure 9](#), with the more broadly used tools and boards at the base of the pyramid, and increasingly application-specific enablement items at the top. Each level is described further, as follows:

- [Section 4.1, “Base of the Pyramid: Broadly-Used Tools and Boards](#)
- [Section 4.2, “First Level of the Pyramid: Debug and Performance Analysis](#)
- [Section 4.3, “Second Level of the Pyramid: Simulation, Hypervisor, and DPAA Reference “Stacklets”](#)
- [Section 4.4, “Top Level of the Pyramid: Application-Specific Enablement](#)



**Figure 9. Levels of Developer Environment**

## 4.1 Base of the Pyramid: Broadly-Used Tools and Boards

### 4.1.1 Hardware Platforms

This category includes both development systems and the reference designs. Development systems are available from both Freescale and our partners, with some partner systems being offered with form factors and BOMs to support use as reference designs. Freescale development systems are supported by the open source GNU tool set including compilers, linkers, and debuggers.

### 4.1.2 Compilers, Debuggers, Bootloaders, LSPs, Drivers

In active partnership with the open source community and Linux distribution and support suppliers, these tools will be updated to fully and efficiently support the device.

### 4.1.3 SMP/AMP Capable OS's

Open source tools will be part of an overall P5020 development board Linux support package, which will include AMP and SMP versions of the Linux OS, and P5020 drivers for the accelerators and networking and peripheral interfaces featured in the P5020. AMP Linux support will include the ability to boot multiple instances of Linux on different cores. Power Architecture ecosystem partners are committed to providing board support packages for the P5020.

## 4.2 First Level of the Pyramid: Debug and Performance Analysis

### 4.2.1 Advanced Debug

Advanced debug supports real-time trace analysis. It allows the developer to perform initial system bring-up and development, and is required to deal with the special challenges of software debugging and performance analysis in multicore systems.

### 4.2.2 Profiling and Performance Analysis

Freescale will bring tools support for profiling and performance analysis (such as enhanced statistics gathering) to the market both by means of our CodeWarrior line of tools and in partnership with industry standard tools suppliers.

## 4.3 Second Level of the Pyramid: Simulation, Hypervisor, and DPAA Reference “Stacklets”

### 4.3.1 Hybrid Simulator

In conjunction with Virtutech, Freescale will provide a hybrid simulator that combines both functional and performance measurement models of the P5020. The hybrid simulator allows the user to switch between “fast functional mode” and “detailed performance mode” with capabilities that include the following:

- Global visibility
- Determinism
- Bug reproducibility
- Reverse execution
- Special abilities to detect race conditions
- Ability to detect race conditions

### 4.3.2 Hypervisor Micro-Kernel

The P5020's e5500 cores offer a new embedded Hypervisor capability to address the need for a single operating system performing coordination and access control functions, managing shared resources in an efficient manner. The embedded Hypervisor provides the software layer needed to manage the operating systems and supervisor-level applications as they access shared resources. Recognizing that each developer's system design may call for a different partitioning of resources, and involve different combinations of OSEs and RTOSes, Freescale and our ecosystem partners will provide reference implementations of the embedded Hypervisor's peripheral virtualization and access control which the developer can modify to match unique system requirements.

### 4.3.3 DPAA Reference "Stacklets"

It is expected that some CPUs will be dedicated as datapath processors, working closely with the DPAA. Freescale will provide reference protocol "stacklets," optimizing performance critical regions of protocol processing and their interaction with the DPAA hardware.

## 4.4 Top Level of the Pyramid: Application-Specific Enablement

This category includes 3rd-party stacks optimized for DPAA, RegEx, AV TCP, IPv4/6, IPsec/SSL.

Many of the expected applications for the P5020 involve network protocol processing. Partitioning between control CPUs and datapath CPUs, and developing the protocol processing firmware which runs on the datapath CPUs is an area for significant value added services for Freescale partners at the top level of the enablement pyramid. OEMs wishing to engage with these partners can realize significant "time-to-performance" advantages.

## 5 Document Revision History

The following table provides a revision history for this product brief.

**Table 8. Revision History**

Revision	Date	Substantive Change(s)
1	02/2013	Modified USB Specification, <a href="#">Section 3.7</a> , "Universal Serial Bus (USB) 2.0."
0	12/2011	Initial public release.