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 e500mc
Number of Cores/Bus Width	4 Core, 32-Bit
Speed	1.2GHz
Co-Processors/DSP	-
RAM Controllers	DDR3, DDR3L
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10/100/1000Mbps (5), 10Gbps (1)
SATA	SATA 3Gbps (2)
USB	USB 2.0 + PHY (2)
Voltage - I/O	1.5V, 1.8V, 2.5V, 3.3V
Operating Temperature	-40°C ~ 105°C (TA)
Security Features	-
Package / Case	1295-BBGA, FCBGA
Supplier Device Package	1295-FCPBGA (37.5x37.5)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=p3041nxn7mmc

1 P3041 Application Use Cases

1.1 Integrated Access Router (IAD)

Dual SATA ports provide high-speed, low-cost storage options for statistics or large databases. Compared to SGMII, 2.5-Gb/s Ethernet enables the next step in performance connectivity to switches.
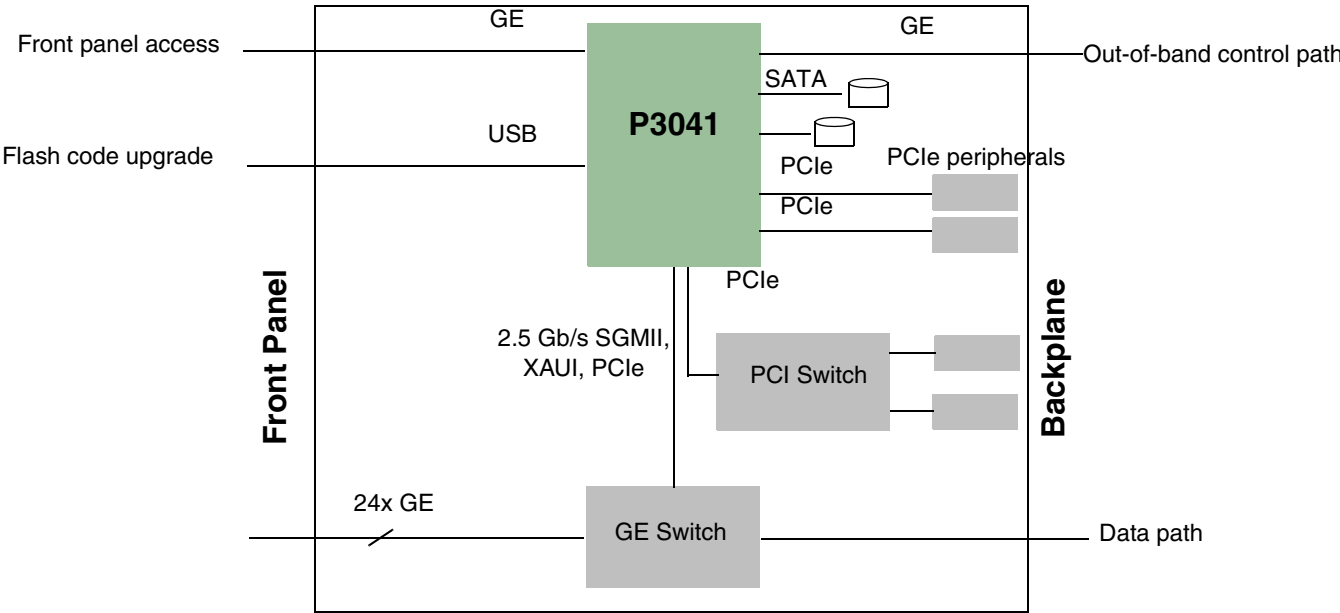

Figure 1. P3041 Integrated Access Router Interface

1.2 Base Station Network Interface Card (NIC)

Dual Serial RapidIO ports (up to 5 GHz) can be used for redundancy or multiple connections, both to the backplane or to the DSP farm. With improved Type 11 messaging and new support for Type 9 data streaming, the Serial RapidIO interconnect can now be used not only as a control plane interface, but can also achieve its intended potential as a highly-efficient, data path.
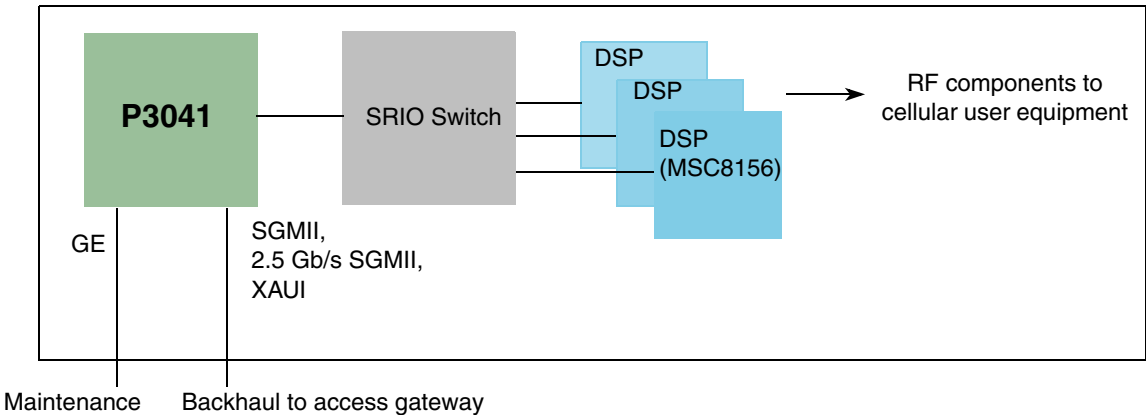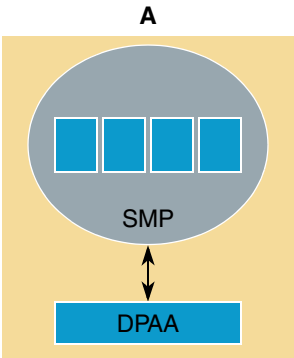

Figure 2. P3041 LTE Wireless Base Station Interface

**P3041 QorIQ Communications Processor Product Brief, Rev. 0**

A

All CPUs are running a single operating system, with any specialization of CPU function occurring through OS techniques such as Task Affinity. The I/Os and acceleration hardware are under the control of the SMP OS. Typically all CPUs operate at the same frequency.

B
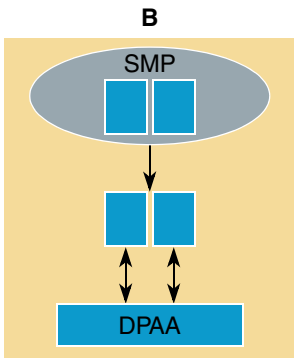
Some number of the cores are operated as an SMP cluster, most likely running high complexity control plane operations. The control plane configures and manages the remaining processors, which are running individual copies of an RTOS or scheduler to perform dataplane operations. In this use case, the SMP CPUs typically operate at the same frequency, the remaining CPUs can run at a different frequency from the SMP CPUs, and even from each other.

C
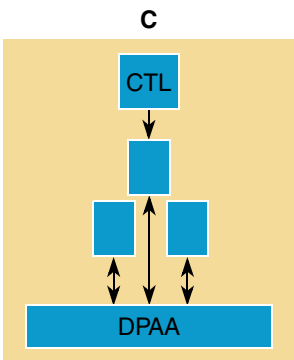
A single CPU is used as the control processor, configuring and managing the other three processors, which are running individual copies of an RTOS or scheduler, as in B. CPU operating frequencies are an independent parameter.

D
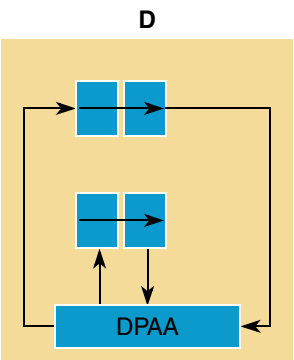
All CPUs are used for datapath operations, here shown as two sets of pipelined functions, each interacting independently with the I/Os and accelerators. Operating frequencies for each CPU in the pipeline can be set independently, and the provision of a 128-Kbyte back-side L2 provides significant flexibility in partitioning and rebalancing the pipeline as processing requirements change.

Figure 3. CPU Usage Use Cases

Figure 4 shows an additional use case, which involves the use of one of the CPUs as an I/O processor. The DPAA can greatly simplify and accelerate processing for packets entering the system by means of the

3 P3041 Features

3.1 Block Diagram

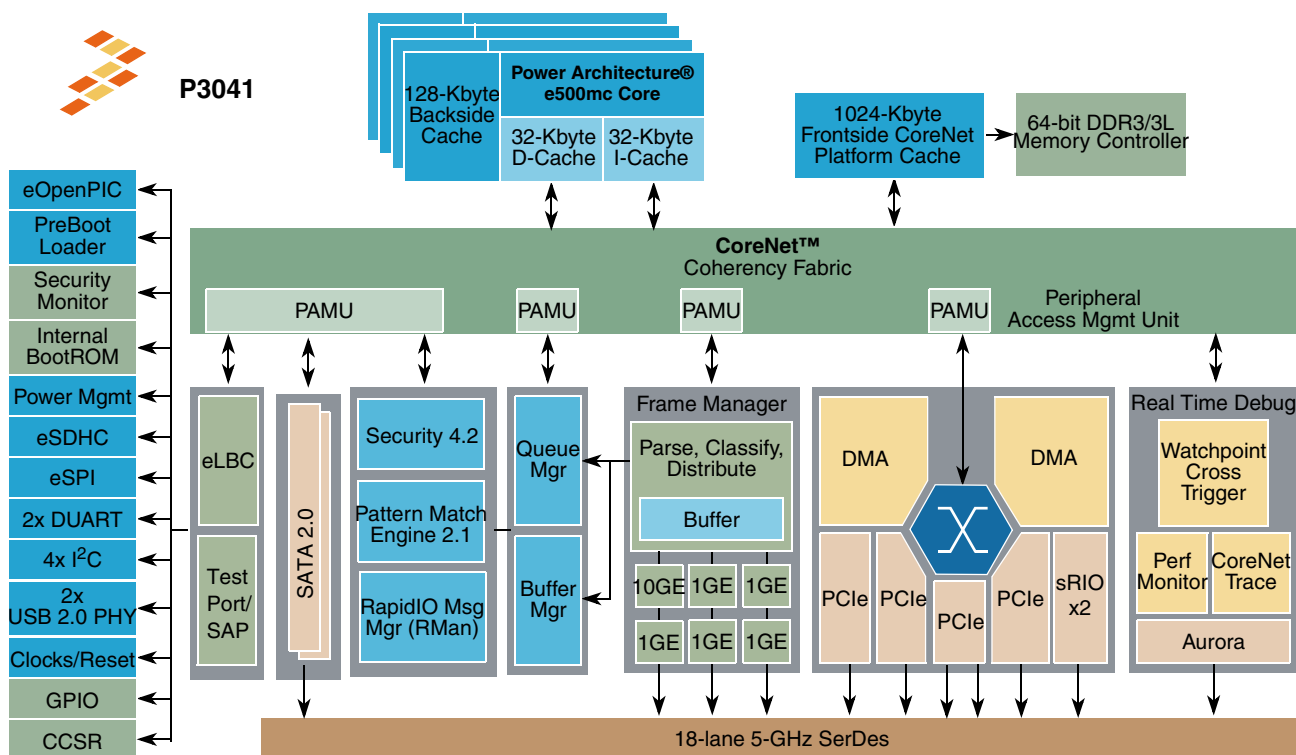Figure 5 shows the major functional units within the P3041.

Figure 5. P3041 Preliminary Block Diagram

3.2 P3041 Features Summary

The P3041 SoC includes the following functions and features:

- Four e500mc cores built on Power Architecture technology, each with a private 128-Kbyte backside cache
  — Three levels of instructions:
    – User
    – Supervisor
    – Hypervisor
  — Independent boot and reset
  — Secure boot capability
- 1-Mbyte shared CoreNet platform cache (CPC)
- Hierarchical interconnect fabric

defining characteristic of the P3041. 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 four 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 P3041 uses Freescale's Data Path Acceleration Architecture (DPAA) (see Section 3.11, "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 P3041 allows the user's software to perform complex packet processing at high data rates.

3.5   P3041 Critical Performance Parameters

Table 1 lists key performance indicators that define a set of values used to measure P3041 operation.

Table 1. P3041 Critical Performance Parameters

Indicator	Values(s)
Top speed bin e500mc core frequency	1.5 GHz
Maximum memory data rates	1.3 GHz (DDR3/3L)[1] • 1.5-V for DDR3 • 1.35-V for DDR3L
Local bus	• 3.3 V • 2.5 V • 1.8 V
Operating junction temperature range	0–105 C with the option for –40 to 105 C
Package	• 1295-pin FC-PBGA (flip-chip plastic ball grid array) • Pin-compatible with P4040, P4080, P5010, and P5020

Notes:
[1] Conforms to JEDEC standard

– 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

3.6.2   e500mc 128-Kbyte Private Backside Cache

Each e500mc core features a 128-Kbyte private backside cache running at the same frequency of the CPU, which supports the following:

- Write-back, pseudo LRU replacement algorithm

- Tag parity and ECC data protection

- 8-way, with arbitrary partitioning between instruction and data. For example, 3-ways instruction, 5-ways data, and so on

- Supports direct stashing of Data Path Acceleration Architecture (DPAA) data into cache

3.6.3   CoreNet Platform Cache (CPC)

The P3041 contains 1-Mbyte of shared CoreNet platform cache (CPC). The key features of the CPC include the following:

- Configurable as write-back or write-through

- Pseudo LRU replacement algorithm

- ECC protection

- 64-byte coherency granule

- 1 cache line read 64 bytes per cycle at 750 MHz, 0.4 terabits/sec read bandwidth

- 32-way cache array configurable to any of several modes on a per-way basis.
  — Unified cache, I-only, D-only
  — I/O stash (configurable portion of each packet copied to CPC on write to main memory)
    – stashing of all transactions and sizes supported
    – explicit (CoreNet signalled) and implicit (address range based) stash allocation
  — Addressable SRAM (32-Kbyte granularity)

3.7   CoreNet Fabric and Address Map

The CoreNet fabric is Freescale's next generation Front-side Interconnect Standard for multicore products, and provides the following:

- A highly concurrent, fully cache coherent, multi-ported fabric

- Point-to-point connectivity with flexible protocol architecture allows for pipelined interconnection between CPUs, platform caches, memory controllers, and I/O and accelerators at up to 750 MHz

- The CoreNet fabric has been designed to overcome bottlenecks associated with shared bus architectures, particularly address issue and data bandwidth limitations. The P3041's multiple,

parallel address paths allow for high address bandwidth, which is a key performance indicator for large coherent multicore processors

- Eliminates address retries, triggered by CPUs being unable to snoop within the narrow snooping window of a shared bus. This results in the P3041 having lower average memory latency

The flexible P3041's 36-bit, physical address map consists of local space and external address space. For the local address map, 32 local access windows (LAWs) define mapping within the local 36-bit (64-Gbyte) address space. Inbound and outbound translation windows can map the P3041 into a larger system address space such as the RapidIO or PCIe 64-bit address environment. This functionality is included in the address translation and mapping units (ATMUs).

3.8 Memory Complex

The P3041 memory complex consists of one DDR controller for main memory, and the memory controllers associated with the Enhanced Local Bus Controller (eLBC).

3.8.1 DDR Memory Controller

The P3041 DDR memory controllers have the following functionalities:

- Supports DDR3/3L SDRAM. The P3041 also supports chip-select interleaving within a controller.
- The P3041 can be configured to retain the currently active SDRAM page for pipelined burst accesses. Page mode support of up to 32 simultaneously open pages can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, page mode can save up to 10 memory clock cycles for subsequent burst accesses that hit in an active page.
- Using ECC, the P3041 detects and corrects all single-bit errors and detects all double-bit errors and all errors within a nibble.
- Upon detection of a loss of power signal from external logic, the DDR controllers can put compliant DDR SDRAM DIMMs into self-refresh mode, allowing systems to implement battery-backed main memory protection.
- Supports initialization bypass feature for use by system designers to prevent re-initialization of main memory during system power-on after an abnormal shutdown
- Supports active zeroization of system memory upon detection of a user-defined security violation

3.8.2 PreBoot Loader (PBL) and Nonvolatile Memory Interfaces

The PreBoot Loader (PBL) is a new logic module that operates similarly to an I^{2}C boot sequencer but on behalf of a larger number of interfaces.

The PBL's functions include the following:

- Simplifies boot operations, replacing pin strapping resistors with configuration data loaded from nonvolatile memory
- Uses the configuration data to initialize other system logic and to copy data from low speed memory interfaces (I^{2}C, eLBC, SPI, and SD/MMC) into fully initialized DDR or the 1-Mbyte front-side cache.

3.9   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:

- Complies with USB specification, Rev. 2.0
- Supports high-speed (480 Mbps), 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

3.10   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 P3041 integrates the following:

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

3.10.1   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.

    — ATAPI 6+

    — Spread spectrum clocking on receive

- Support for SATA II extensions

    — Asynchronous notification

    — Hot plug including asynchronous signal recovery

    — Link power management

    — Native command queuing

    — Staggered spin-up and port multiplier support

- Support for SATA I and II data rates (1.5 and 3.0 Gbaud)

- Standard ATA master-only emulation

- Includes ATA shadow registers

- Implements SATA superset registers (SError, SControl, SStatus)

- Interrupt driven

- Power management support

- Error handling and diagnostic features

    — Far end/near end loopback

    — Failed CRC error reporting

    — Increased ALIGN insertion rates

    — Scrambling and CONT override

3.11   Data Path Acceleration Architecture (DPAA)

The DPAA provides the infrastructure to support simplified sharing of networking interfaces and accelerators by multiple CPU cores. These resources are abstracted into enqueue/dequeue operations by means of a common DPAA Queue Manager (QMan) driver. Beyond enabling multicore resource sharing, the DPAA significantly reduces software overheads associated with high-touch packet-forwarding operations. Examples of the types of packet-processing services this architecture is optimized to support are as follows:
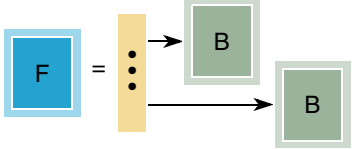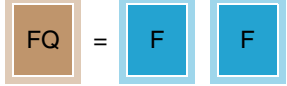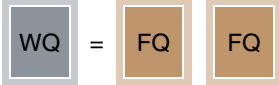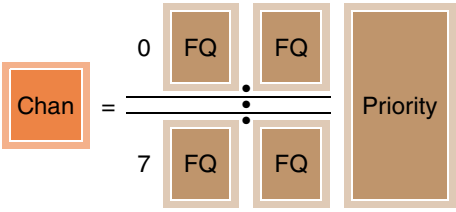
- Traditional routing and bridging

- Firewall

- VPN termination for both IPsec and SSL VPNs

- Intrusion detection/prevention (IDS/IPS)

- Network anti-virus (AV)

The DPAA generally leaves software in control of protocol processing, while reducing CPU overheads through off-load functions, which fall into two, broad categories:

- Section 3.11.1, "Packet Distribution and Queue/Congestion Management"

- Section 3.11.2, "Accelerating Content Processing"

3.11.3 DPAA Terms and 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.	

3.11.4 Major DPAA Components

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

- Section 3.11.4.1, "Frame Manager (FMan)
- Section 3.11.4.2, "Queue Manager (QMan)
- Section 3.11.4.3, "Buffer Manager (BMan)
- Section 3.10.2.2, "RapidIO Message Manager (RMan)
- Section 3.11.4.4, "Security Engine (SEC 4.2)

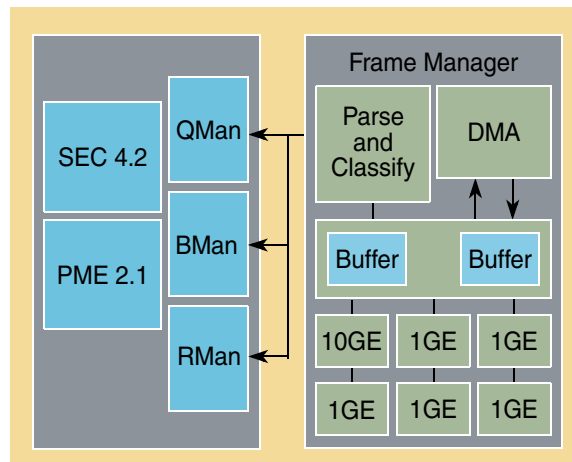- Section 3.11.4.5, "Pattern Matching Engine (PME 2.1)"

Figure 6. QorIQ Data Path Acceleration Architecture (DPAA)

3.11.4.1   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 at line rate (7.5 Mpps). 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.

3.11.4.1.1   FMan Network Interfaces

The P3041 FMan integrates five datapath, 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 3.11.4.1.2, "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.

3.11.4.2   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
- Loss-less 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.

3.11.4.3   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
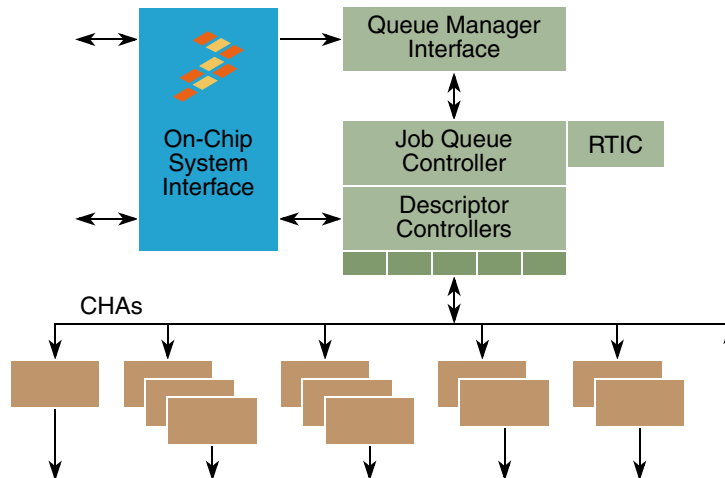
Figure 7. SEC 4.2 Block Diagram

3.11.4.5   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.

3.11.4.5.1   PME Regular Expressions (Regex)

The PME specifies patterns of bytes as regular expressions (regex). The P3041 (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.

3.11.4.5.2   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.
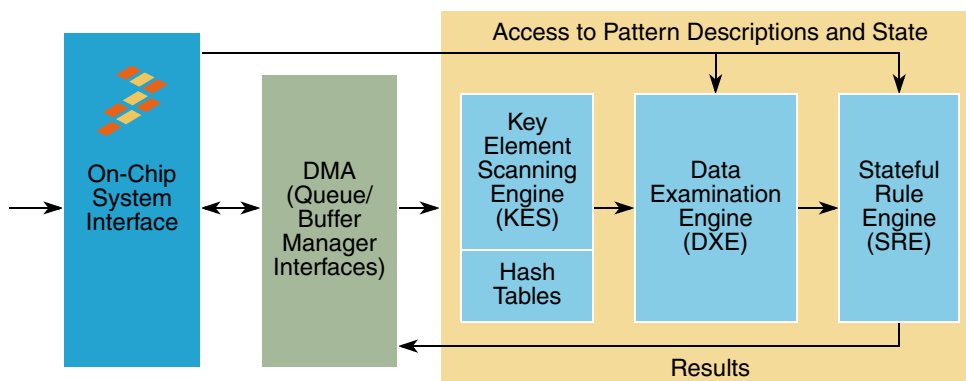

Figure 8. PME 2.1 Block Diagram

3.12   Avoiding Resource Contentions Using the QorIQ Trust Architecture

Consolidation of discrete CPUs into a single, multicore SoC and potential repartitioning of legacy software on those cores introduces many opportunities for unintended resource contentions to arise, but the QorIQ Trust Architecture can reduce the risk of these issues.

3.12.1   QorIQ Trust Architecture Benefits

A system may exhibit erratic behavior if the multiple CPUs do not effectively partition and share system resources. While it can be challenging to prevent unintended resource contention, stopping malicious software is much more difficult. 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.

The P3041 offers a new level of hardware partitioning support, allowing system developers to ensure software running on any CPU only accesses the resources (memory, peripherals, etc.) that it is explicitly authorized to access. This may not seem like a challenge in an SMP environment, because the OS performs resource allocation for the applications running on it. However, it is a very difficult problem to overcome in AMP environments where there may be multiple instances of the same OS, or even different OSes running on the various CPU cores. Even OS protections in an SMP system may be insufficient in the presence of malicious software.

3.12.2   e500mc MMU and Embedded Hypervisor

The P3041's first line of defense against unintended interactions amongst the multiple CPUs/OSes is each e500mc core's MMU, which are configured to determine which addresses in the global address map the CPU is able to read or write. If a particular resource (such as a portion of memory or a peripheral device) is dedicated to a single CPU, that CPU's MMU is configured to allow access to those addresses (on 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, both of their MMUs are configured to 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 shouldn't 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 e500mc MMU also includes embedded Hypervisor extensions.

Each e500mc 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 device 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.

3.12.3   Peripheral Access Management Unit (PAMU)

The P3041 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.

3.12.4 Secure Boot and Sensitive Data Protection

The e500mc MMUs and PAMU allow the P3041 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 P3041 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.

3.12.4.1 Secure Boot Option

The system developer digitally signs the code to be executed by the CPU coming out of reset, and the P3041 ensures that only an unaltered version of that code runs on the platform. The P3041 offers both boot time and run time code authenticity checking and configurable consequences when the authenticity check fails.

3.12.4.2 Sensitive Data Protection Option

The P3041 supports protected internal and external storage of developer-provisioned sensitive instructions and data.

For example, a system developer may provision each system with a number of RSA private keys to be used in mutual authentication and key exchange. These values would initially be stored in external non-volatile memory, but following secure boot, these values can be decrypted into on-chip protected memory (portion of platform cache dedicated as SRAM). Session keys, which may number in the thousands to tens of thousands, are not good candidates for on-chip storage, so the P3041 offers session key encryption. Session keys are stored in main memory, and are decrypted (transparently to software and without impacting SEC throughput) as they are brought into the SEC 4.2 for decryption of session traffic.

3.13 Advanced Power Management

The P3041's advanced power management capabilities are based around fine-grained static clock control and software-controlled dynamic frequency management.

3.13.1 Saving Power by Managing Internal Clocks

Dynamic voltage and frequency scaling (DVFS) are useful techniques for reducing typical/average power and maximizing battery life in laptop environments, but embedded applications must be designed for rapid response to bursts of traffic and max power under worst-case environmental conditions. While the P3041 does not implement DVFS in the PC sense, it does actively manage internal clocks to avoid wasting energy. Clock signals are disabled to idle components, reducing dynamic power. These blocks can return to full operating frequency on the clock cycle after work is dispatched to them.

The P3041 also supports (under software control) dynamic changes to CPU operating frequencies and voltages. Each CPU sources its input clock from one of two independent PLLs inside the P3041. Each CPU can also source its input clock from an integer frequency divider from two of the three independent PLLs. CPUs can switch their source PLL, and their frequency divider glitchlessly and nearly instantaneously. This allows each core to operate at the minimum frequency required to perform its assigned function, saving power.

3.13.2   Turning Off Unneeded Clocks

Fine-grained static control allows developers to turn off the clocks to individual logic blocks within the SoC that the system has no need for. Based on a finite number of SerDes, it is expected that any given application will have some Ethernet MACs, PCIe, or Serial RapidIO controllers inactive. These blocks can 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).

3.13.3   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 P3041 achieves some critical level, external temperature detection circuitry can drive a high-priority interrupt into the P3041, 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 G Hz 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 e500mc.

3.14   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 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 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 P3041 as they would with multiple discrete devices, the P3041 implements the debug architecture shown in Figure 9.
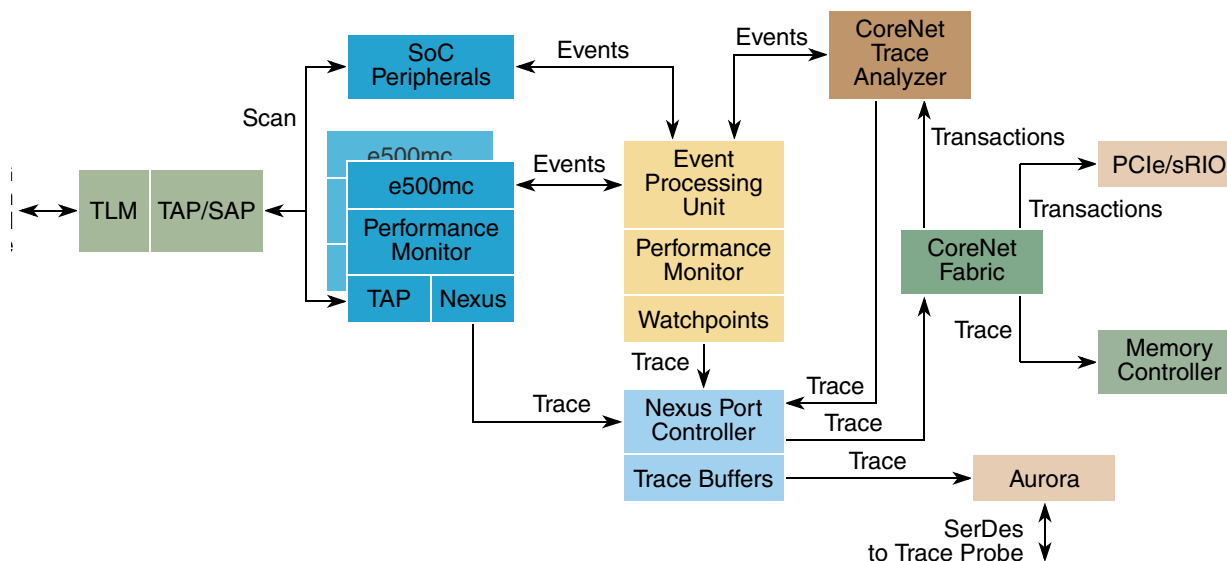
Figure 9. P3041 Debug Architecture Block Diagram

Debug features include the following:

- Debug and performance monitoring registers in both the e500mc 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 e500mc 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 (NPC)
— 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 P3041.

The various levels of the developer environment are shown in Figure 10, 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 in the following subsections:

- 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"
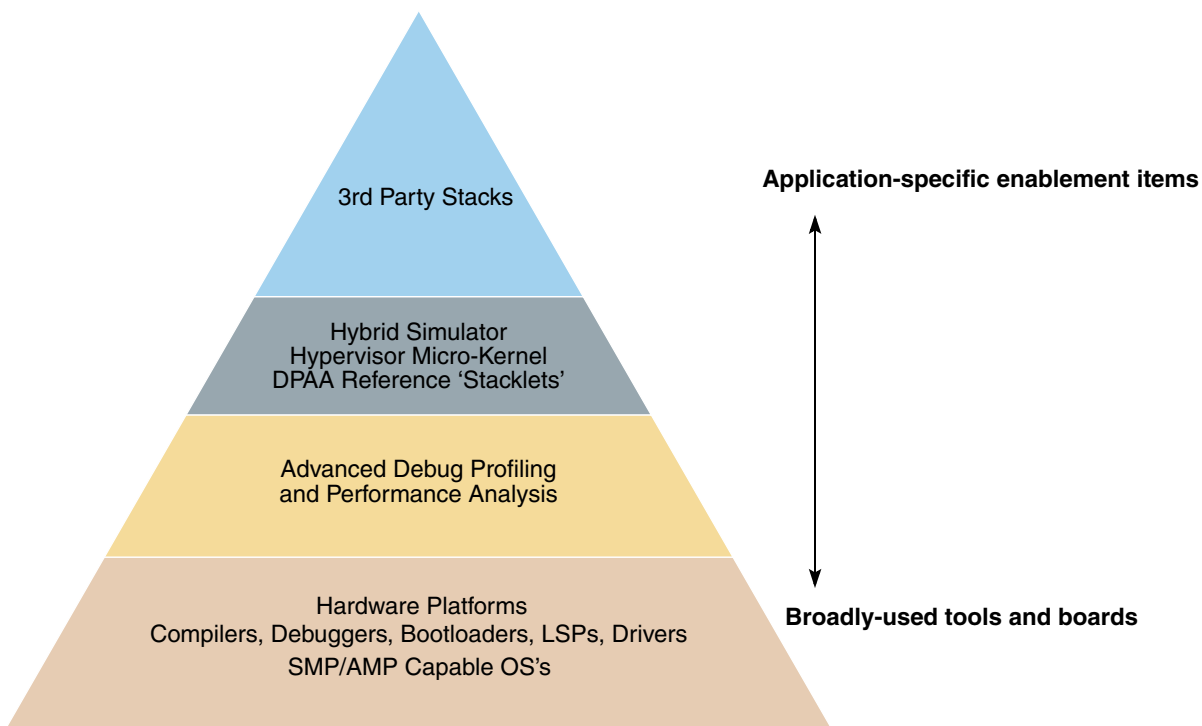
Figure 10. Levels of Developer Environment

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 P3041.

4.1.3 SMP/AMP Capable OS's

Open source tools will be part of an overall P3041 development board Linux support package, which will include AMP and SMP versions of the Linux OS, and device drivers for the accelerators and networking and peripheral interfaces featured in the P3041. 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 P3041.

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 P3041. The hybrid simulator allows the user to switch between "fast functional mode" and "detailed performance mode" with capabilities that include the following:

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +303-675-2140
Fax: +303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: P3041PB
Rev. 0
11/2011