Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Active
Core Processor	PowerPC e5500
Number of Cores/Bus Width	2 Core, 64-Bit
Speed	2GHz
Co-Processors/DSP	-
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	1V
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=p5020nxn1tnb

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.

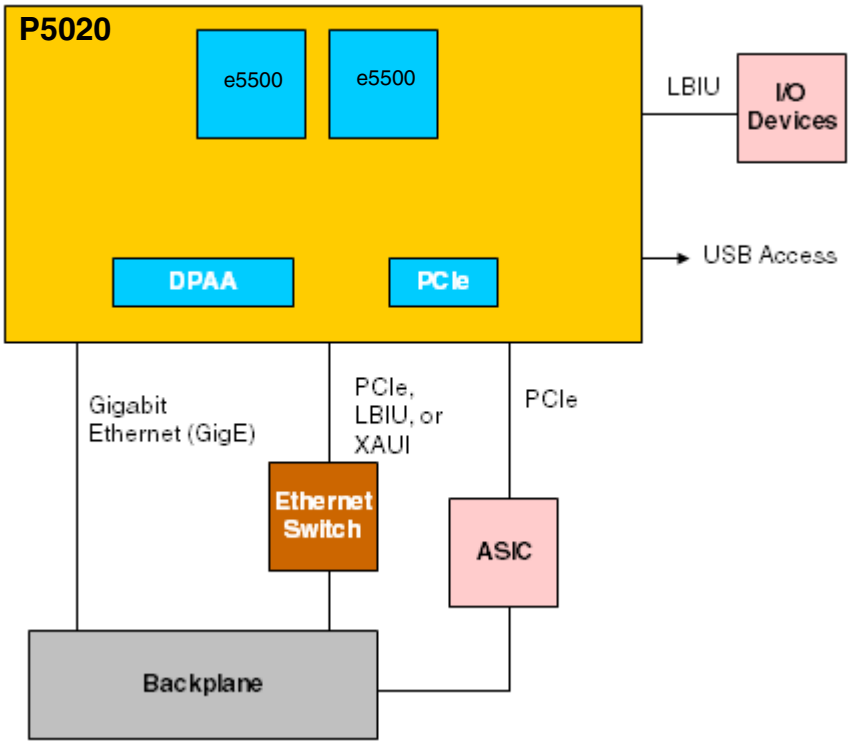


Figure 1. Control Plane Processor for a Router

1.2 DSP Farm Control Processor

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

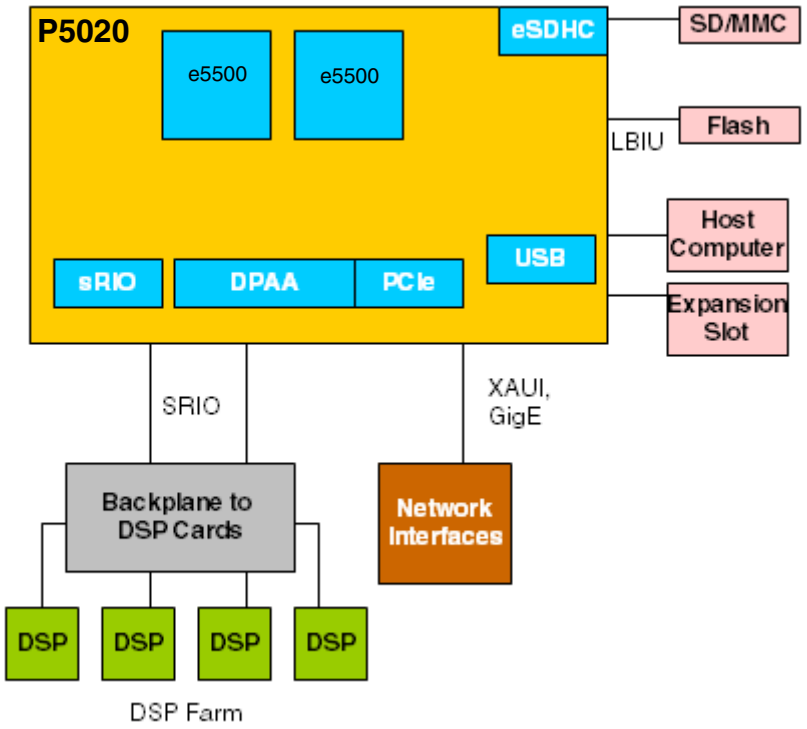


Figure 2. Control Plane Processor for a DSP Farm

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.

  - Supervisor
  - Hypervisor
- Independent boot and reset
- Secure boot capability

- Two 1-Mbyte shared CoreNet platform cache (CPC)
- Hierarchical interconnect fabric
  - CoreNet fabric supporting coherent and non-coherent transactions with prioritization and bandwidth allocation amongst CoreNet end-points
  - Queue manager fabric supporting packet-level queue management and quality of service scheduling
- Two 64-bit DDR3/3L SDRAM memory controllers with ECC and interleaving support
- Datapath acceleration architecture (DPAA) incorporating acceleration for the following functions:
  - Packet parsing, classification, and distribution
  - Queue management for scheduling, packet sequencing, and congestion management
  - Hardware buffer management for buffer allocation and de-allocation
  - Encryption/decryption (SEC 4.2)
  - RegEx pattern matching (PME 2.1)
  - RapidIO™ messaging manager (RMan)
  - RAID5/6 Engine
    - Support for XOR and Galois Field parity calculation
    - Support for data protection information (DPI)
- Ethernet interfaces
  - One 10 Gbps Ethernet (XAUI) controller
  - Five 1 Gbps or four 2.5 Gbps Ethernet controllers
- High speed peripheral interfaces
  - Four PCI Express 2.0 controllers/ports running at up to 5 GHz
  - Two serial RapidIO 2.0 controllers/ports (version 1.3 with features of 2.1) running at up to 5 GHz with Type 11 messaging and Type 9 data streaming support
- Additional peripheral interfaces
  - Dual SATA supporting 1.5 and 3.0 Gb/s operation
  - Two USB 2.0 controllers with integrated PHY
  - SD/MMC controller (eSDHC)
  - Enhanced SPI controller
  - Four I^2C controllers
  - Two Dual DUARTs
  - Enhanced local bus controller (eLBC)
- 18 SerDes lanes to 5 GHz
- Multicore Programmable Interrupt Controller (MPIC)

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+

  - 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

3.9 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:

- Packet Distribution and Queue/Congestion Management
- Accelerating Content Processing

3.9.1 Packet Distribution and Queue/Congestion Management

The following table lists some packet distribution and queue/congestion management offload functions.

Table 2. Offload Functions

Function Type	Definition
Data buffer management	Supports allocation and deallocation of buffers belonging to pools originally created by software with configurable depletion thresholds. Implemented in a module called the Buffer Manager (BMan).
Queue management	Supports queuing and quality-of-service scheduling of frames to CPUs, network interfaces and DPAA logic blocks, maintains packet ordering within flows. Implemented in a module called the Queue Manager (QMan). The QMan, besides providing flow-level queuing, is also responsible for congestion management functions such as RED/WRED, congestion notifications and tail discards.
Packet distribution	Supports in-line packet parsing and general classification to enable policing and QoS-based packet distribution to the CPUs for further processing of the packets. This function is implemented in the block called the Frame Manager (FMan).
Policing	Supports in-line rate-limiting by means of two-rate, three-color marking (RFC 2698). Up to 256 policing profiles are supported. This function is also implemented in the FMan.

3.9.2 Accelerating Content Processing

Properly implemented acceleration logic can provide significant performance advantages over most optimized software with acceleration factors on the order of 10–100x. Accelerators in this category typically touch most of the bytes of a packet (not just headers). To avoid consuming CPU cycles in order to move data to the accelerators, these engines include well-pipelined DMAs. The following table lists some specific content-processing accelerators on the P5020.

Table 3. Content-Processing Accelerators

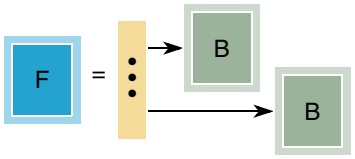
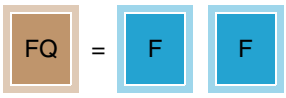
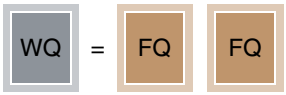
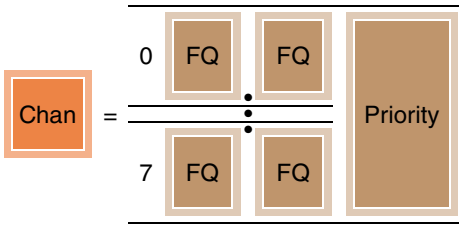
Interface	Definition
SEC 4.2	Crypto-acceleration for protocols such as IPsec, SSL, and 802.16
PME 2.1	Regex style pattern matching for unanchored searches, including cross-packet stateful patterns

Note: Prior versions of the SEC and PME are integrated into multiple members of the PowerQUICC and QorIQ family. Both of these engines have been enhanced to work within the DPAA, and also upgraded in both features and performance.

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)

When all SERDES are otherwise allocated, it is possible to enable two of dTSECs by means of RGMII or RMII physical interfaces.

3.9.4.1.2 FMan Parse Function

The primary function of the packet parse logic is to identify the incoming frame for the purpose of determining the desired treatment to apply. This parse function can parse many standard protocols, including options and tunnels, and supports a generic configurable capability to allow proprietary or future protocols to be parsed.

There are several types of parser headers, shown in the following table.

Table 5. Parser Header Types

Header Type	Definition
Self-describing	Announced by proprietary values of Ethertype, protocol identifier, next header, and other standard fields. They are self-describing in that the frame contains information that describes the presence of the proprietary header.
Non-self-describing	Does not contain any information that indicates the presence of the header. 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.

3.9.4.1.3 FMan Distribution and Policing

After parsing is complete, there are two options for treatment (see Table 6).

Table 6. Post-Parsing Treatment Options

Treatment	Function	Benefits
Hash	• Hashes selected 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

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 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

3.9.4.4 Security Engine (SEC 4.2)

The SEC 4.2 is QorIQ's fourth generation crypto-acceleration engine. In addition to off-loading cryptographic algorithms, the SEC 4.2 offers header and trailer processing for several established security protocols. The SEC 4.2 includes several Descriptor Controllers (DECOs), which are updated versions of the previous SEC crypto-channels. DECOs are responsible for header and trailer processing, and managing context and data flow into the CHAs assigned to it for the length of an operation.

The DECOs can perform header and trailer processing, as well as single pass encryption/integrity checking for the following security protocols:

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

In prior versions of the SEC, the individual algorithm accelerators were referred to as Execution Units (EUs). In the SEC 4.2, these are referred to as Crypto Hardware Accelerators (CHAs) to distinguish them from prior implementations. Specific CHAs available to the DECOs are listed below.

- Advanced encryption standard unit (AESA)
- ARC four execution unit (AFHA)
- Cyclic redundancy check accelerator (CRCA)
- Data encryption standard execution unit (DESA)
- Kasumi execution unit (KFHA)
- SNOW 3 G hardware accelerator (STHA)
- Message digest execution unit (MDHA)
- Public key execution unit (PKHA)
- Random number generator (RNGB)

Depending on the security protocol and specific algorithms, the SEC 4.2's aggregate symmetric encryption/integrity performance is 5 Gbps, while asymmetric encryption (RSA public key) performance is ~5,000 1024b RSA operations per second.

The SEC 4.2 is also part of the QorIQ Trust Architecture, which gives the P5020 the ability to perform secure boot, runtime code integrity protection, and session key protection. The Trust Architecture is described in Section 3.10, "Avoiding Resource Contentions Using the QorIQ Trust Architecture."

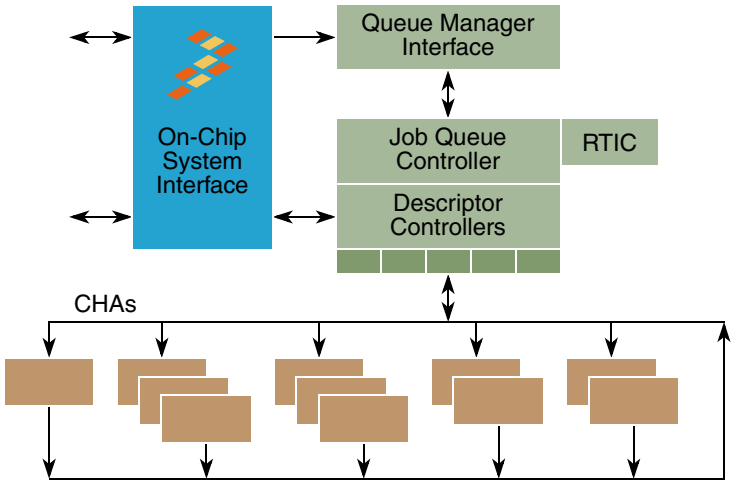


Figure 6. SEC 4.2 Block Diagram

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.

- Extended mailbox
- Letter

Should the packet remain unclassified, the traffic is retried with an error in the case of Type 10 and 11 traffic and dropped in the case of Type 9 traffic. Dropped traffic is logged and upon a threshold can assert an error interrupt.

Classification allows Type 9, 10 and 11 traffic to be distributed across 64 possible Frame queues. A single dedicated inbound Type 8 Port-write Frame queue is provided.

For all outbound traffic types (Type 8, 9, 10 and 11), the Datapath Acceleration Architecture allows a very large number of outbound Frame queues effectively limited by system, software and performance constraints.

3.9.4.7 RAID5/6 Engine

The P5020 includes a RAID5/6 Engine for storage applications, which significantly extends the capability and performance of earlier PowerQUICC RAID (XOR) functionality. The RAID5/6 Engine supports a variety of storage-related functions such as Move, Generate XOR, RAID 6 Parity, Fill and Check. The following table summarizes the functions supported by the engine.

Table 7. RAID5/6 Engine Supported Functions

Function	No. of Sources	No. of Destinations	Command Options	
			Scatter/Gather	DIF
No Op	—	—	—	—
Single Source Move	1	1	Y	N
Multicast Move	1	2	Y	N
Add DIF	1	1 or 2	Y	Y
Remove DIF	1	1 or 2	Y	Y
Update DIF	1	1 or 2	Y	Y
Generate Q Parity	2–16	1	Y	Y
Generate Q and Q Parity	2–16	2	Y	Y
Fill Pattern	—	1	Y	Y
Check Pattern	1	—	Y	Y
Fill LFSR	—	1	Y	N
Check LFSR	1	—	Y	N
Compare	2	—	Y	Y
Gather DIF	1	1	Y	Y

The RAID5/6 Engine supports commands with between 1 and 16 sources for relevant functions. A simple DMA move operation is supported along with a two-destination multicast move that duplicates the source data. Both of these simple operations are the foundation for commands that support Data Protection Information (DIF) insertion, updating and checking. A single RAID5/6 parity generate function is

3.10.4.1 Secure Boot Option

The system developer digitally signs the code to be executed by the CPU coming out of reset, and the device ensures that only an unaltered version of that code runs on the platform. The P5020 offers both boot time and run time code authenticity checking and configurable consequences when the authenticity check fails.

3.10.4.2 Sensitive Data Protection Option

The P5020 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 device 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.

3.11 Advanced Power Management

The P5020's advanced power management capabilities are based around fine-grained static clock control and software-controlled dynamic frequency management.

3.11.1 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 P5020 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 P5020 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 device. 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.

3.11.2 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).

3.11.3 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 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 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.

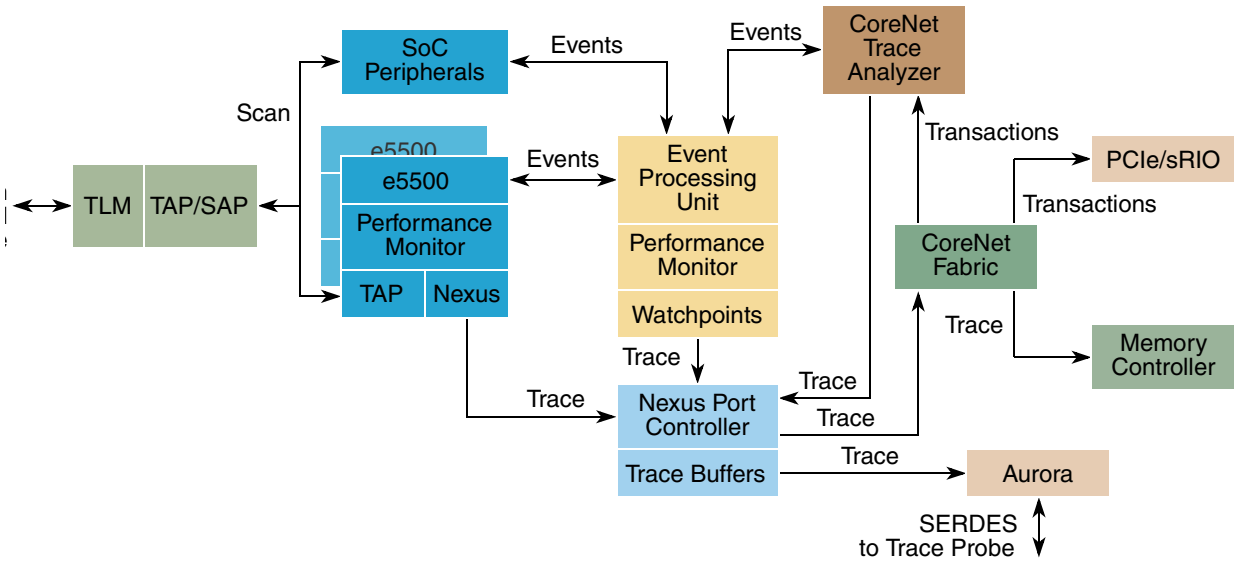


Figure 8. Debug Architecture

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)

- 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, Section 3.7, "Universal Serial Bus (USB) 2.0."
0	12/2011	Initial public release.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: http://www.reg.net/v2/webservices/Freescale/Docs/TermsandConditions.htm.

Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, C-Ware, Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony, and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, ColdFire+, CoreNet, Flexis, MagniV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SafeAssure, SMARTMOS, TurboLink, Vybrid, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2011-2013 Freescale Semiconductor, Inc.

Document Number: P5020PB
Rev. 1
02/2013

freescale™