Freescale's QUICC Engine™ Technology
Multi-protocol interworking fact sheet

Overview

The PowerQUICC® family of communication processors provides ideal solutions for wireless infrastructure, enterprise routing and other networking and gateway applications.

This document outlines Freescale's multi-protocol interworking functionality on the QUICC Engine™ module, which is integrated into the MPC8360E PowerQUICC II Pro and MPC8568E PowerQUICC III communication processors, plus MSC81xx StarCore® DSPs. The QUICC Engine module provides termination, interworking and switching functionality between a wide range of communication protocols that include Ethernet (IP), ATM HDLC, TDM, UTOPIA and Packet Over SONET (POS). Accompanying the QUICC Engine module is a feature-rich API, device drivers and reference application use-cases to facilitate quick and effective use of interworking functionality.

Interworking Advantages

In a communications processor, interworking refers to the ability to transfer payloads from one protocol and/or physical interface to another, on a per-packet basis, without CPU intervention. In contrast, termination involves CPU processing on a per-packet basis for transferring data from one interface to another. Figure 1 shows termination operation where both control and data plane traffic are terminated to the CPU.

In contrast, interworking operation is shown in Figure 2 where data plane traffic is directly forwarded by the QUICC Engine block.

Compared to termination, the use of interworking generally improves throughput and minimizes latency and jitter. Additionally, improved headroom is made available in the CPU for other application functionality. The most significant advantage is that the QUICC Engine block typically increases data plane throughput above what the CPU can achieve at equivalent CPU and QUICC Engine RISC frequencies.

Figure 1—PowerQUICC® Device Termination Operation

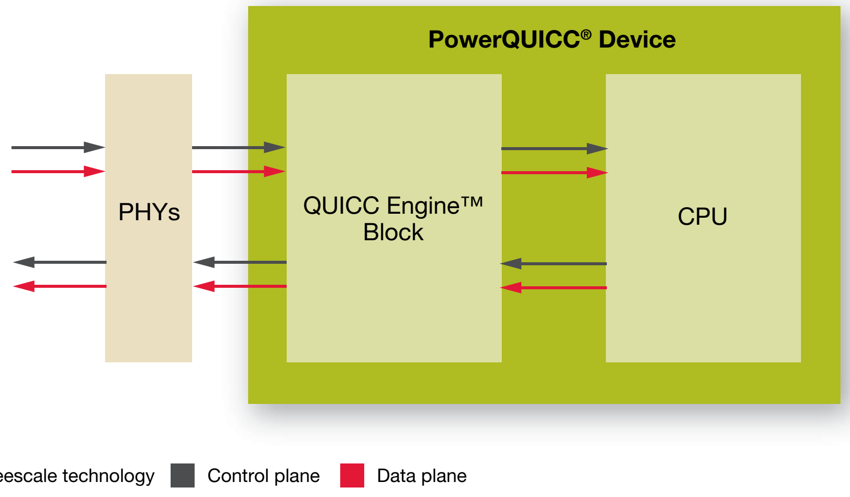
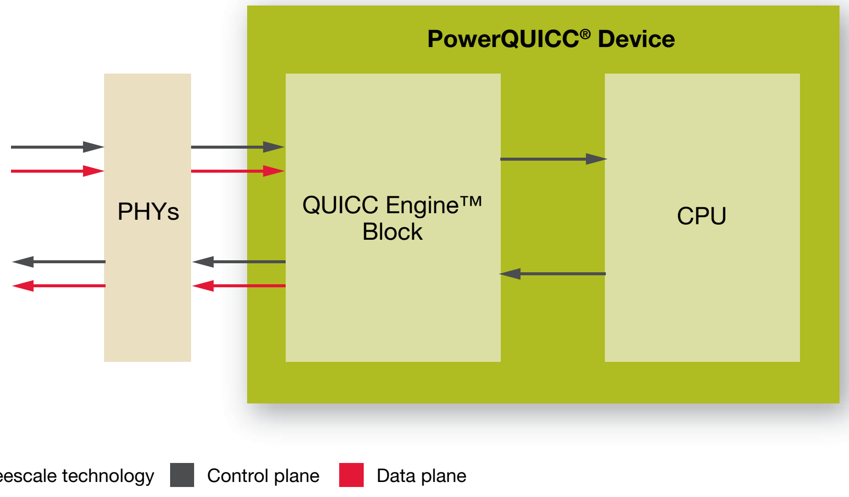


Figure 2—PowerQUICC® Device Interworking Operation



Interworking Features

As an example of the QUICC Engine technology interworking capabilities, consider an all-IP networking application that routes packets between Gigabit Ethernet interfaces as shown in Figure 3. One direction of the data flow is shown above as a sequence of logical processing stages performed in Freescale provided microcode. These stages include classification, policing and Weighted Random Early Discard (WRED) header manipulation such as for Network Address Port Translation (NAPT) and scheduling.

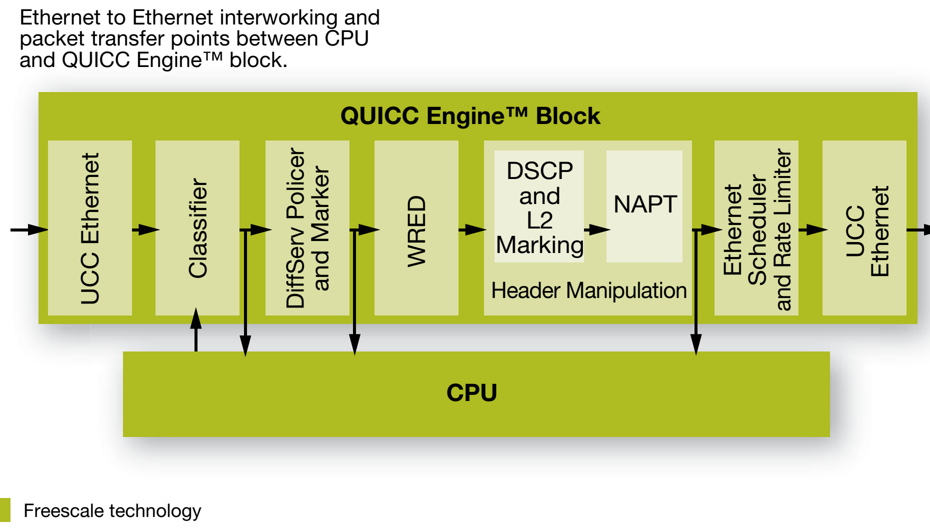
Also shown are possible packet transfer points between the CPU and the QUICC Engine block. This permits the CPU to take advantage of the QUICC Engine technology in some scenarios such as for classification or policing. For instance, policing of packets destined for the CPU can prevent the CPU from being overwhelmed under Denial-of-Service attack scenarios.

Classification

The QUICC Engine microcode provides convenient, powerful and flexible header parsing capabilities with protocol-specific optimizations for rapid parsing. The parser generates a lookup key based on user specified fields from Layers 2 to 4. Protocol agnostic parsing, where the user specifies multiple header offsets and numbers of bytes to extract from each offset to construct a lookup key, is also supported. Up to 128 bytes from the beginning of a packet can be parsed to produce an 8, 16 or 24 byte lookup key.

The generated lookup key is then used to perform a lookup in a lookup table to determine the next stage of processing. Various lookup table types are supported, some of which are tabulated in Table 1. Note that hierarchical or chained lookups are also supported.

Figure 3—Data Flow for QUICC Engine™ Technology


Ethernet to Ethernet interworking and packet transfer points between CPU and QUICC Engine™ block.
QUICC Engine™ Block
UCC Ethernet
Classifier
DiffServ Policer and Marker
WRED
DSCP and L2 Marking
NAPT
Header Manipulation
Ethernet Scheduler and Rate Limiter
UCC Ethernet
CPU
Freescale technology

Header Manipulation

In IPv4 forwarding applications, header manipulation is commonly used to perform Network Address Port Translation (NAPT). Such NAPT header manipulation, together with the associated IP and TCP/UDP checksum updates, are supported by the microcode.

In addition to such replacement of fields, insertion of fields such as VLAN tags or arbitrary user-specified fields and removal of fields is also supported. Insertion and removal of bytes at user-specified offsets and byte counts can be performed. Header modification can be configured on a per-flow basis or for a user-determined group of flows.

Many protocol-specific header and trailer modifications specific to interworking between ATM, Ethernet and PPP protocols are supported.

Table 1—Table Lookup Types

Table Type	# Entries	Key Size	Usage
CAM Emulation Table	Up to 32 entries	8, 16 or 24 bytes	For prioritized linear search, usually 16 entries or less, e.g. IP protocol type filtering Internal memory only
Index Table	2N entries, N=8, 9, 10, …, 24 (256 to 16 million entries)	8- to 24-bits Keys up to 32-bits can be compressed	Index tables are commonly used for ATM VPI/VCI or VLAN tags
Hash Table	2N sets, N=1, 2, 3, …, 16 (2 to 64K sets) Configurable as 4-way or 8-way sets	8, 16 or 24 bytes	Hash tables are used for exact match Internal or external memory

Quality of Service (QoS)

Extensive QoS features are supported for ATM, Ethernet and PPP. As an example, consider the QoS features that are supported with Ethernet. In the ingress direction, classification can take into account VLAN priority fields and/or IPv4 Type of Service fields or DiffServ Code Points. Per-flow policing with dual leaky buckets is supported. WRED, used for congestion avoidance in TCP traffic, may also be applied on a per-flow basis. Figure 4 shows a profile applied to packets "marked" with different colors by the policer. Multiple configurable WRED discard profiles are supported, and each flow can be assigned to a profile.

In the egress direction, each Ethernet interface supports up to eight queues. Combined strict priority and Weighted Fair Queuing (WFQ) can be supported by the scheduler as shown in Figure 5. Rate-limiting may also be applied using the traffic shaper.

Figure 4—Weighted Random Early Discard (WRED) Packet Profile

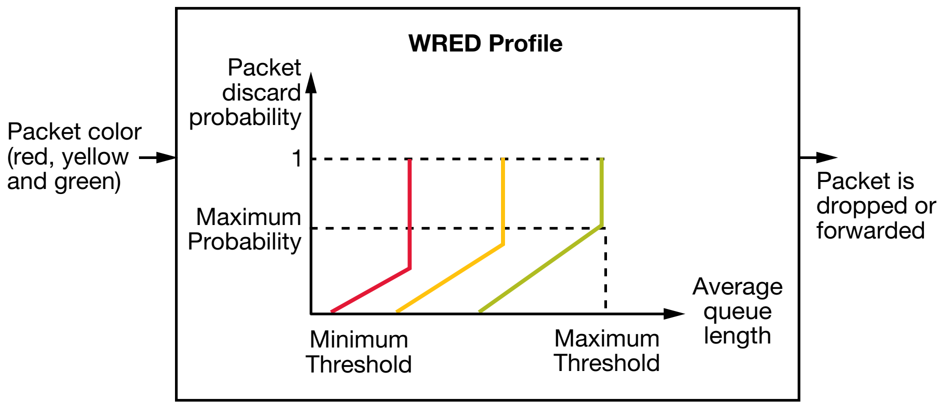
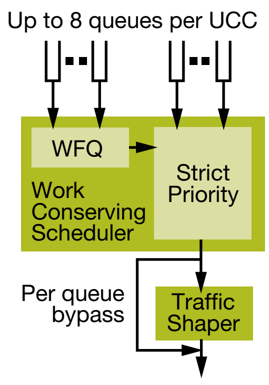


Figure 5—Weighted Fair Queuing (WFQ) and Scheduling

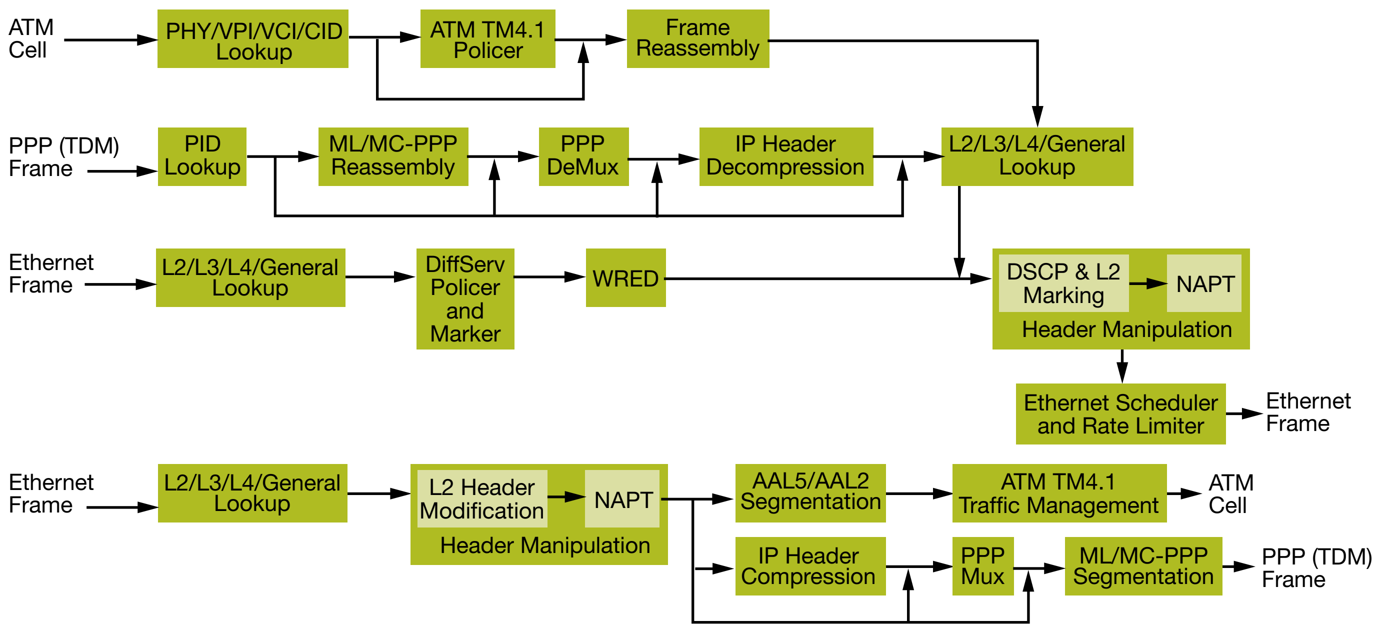


Multi-Protocol Interworking

Figure 6 outlines the main interworking data flow paths for ATM to Ethernet interworking, PPP to Ethernet and Ethernet to Ethernet interworking packages.

When interworking between Ethernet and PPP or ATM, most of the interworking features described earlier are available in the Ethernet ingress and egress directions. In the case of ATM, QoS capabilities specific to ATM traffic management standards are supported. With PPP, additional parsing and header manipulation capabilities such as IP header compression are supported.

Figure 6—Data Flow Paths for Interworking



Device Drivers and API

The ease with which the rich protocol features can be used is paramount. This is facilitated by powerful and flexible drivers and API. The current list of microcode packages, device drivers, application use cases and associated documentation are provided by Freescale royalty and NRE free to registered users is shown in Figure 7.

Application use cases for wireless base station network interface cards, wide area network (WAN) gateways and DSP aggregation cards in media gateways are provided to minimize customer development time. Additional driver use cases are provided to illustrate the API usage. The driver test application suite provides exhaustive tests of the driver functionality. To deliver the highest quality and reliability for customer development, rigorous testing and verification of microcode, drivers and use cases is performed on the interworking packages.

Figure 7—QUICC Engine™ Technology Supported Protocols

Application

- Demo applications (Node B, DSLAM, DSP aggregator, WAN GW)
- Drivers use cases
- Drivers test application suite
- Demo manuals (use cases, how to's, tests reports)

Drivers

- Termination drivers (ATM, Ethernet, TDM, etc.)
- Interworking drivers
- Driver API documentation

Microcode

Layer 3+	**Layer 3+ (IP/Ethernet)** • L3 control packets filtered to host CPU • Parsing and multi-field classification • Hierarchical lookups • IP Sec* **Traffic management (IP/Ethernet)** • Combined SP + WFQ scheduling • Rate limiting/shaping • Lossless flow control • DiffServ + WRED • UDP/IP aggregation • IP fragmentation and reassembly*	**Traffic management (ATM)** • TM 4.1 UBR, CBR, GFR, VBR • Per-flow ATM scheduler for 64K VCs • Hierarchical frame and cell-based scheduling • Policing • Congestion control
Layer 2.5	**Autonomous Interworking** • Ethernet to Ethernet interworking (including IP forwarding) • PPP/TDM to Ethernet interworking • ATM to ATM switching • AAL2 CPS switching and SSSAR • ATM AAL5 to Ethernet interworking	• ATM AAL2 to Ethernet interworking • Multi-link PPP, multi-class PPP, PPP-mux • IPv4/UDP header compression/decompression • IMA over TDM and UTOPIA • VLAN
Layer 2	• ATM AAL 0/1/2/5 • 10/100/1000 Ethernet • L2 (fast Ethernet) switch • Serial ATM (ATM TC sublayer) • PWE3 for ATM pseudowire *	• HLDC • Asynchronous HDLC • SS7 • BISYNC • PWE3 for iTDM pseudowire*
Layer 1 (Physical)	• UTOPIA—L2, POS-PHY-L2 • Ethernet—MII, RMII, GMII, RGMII (for 10/100/1000) • TDM—T1/T3, E1/E3 • USB	• SPI • UART • MDIO

*Available in the future

Learn More: For current information about Freescale products and documentation, please visit **www.freescale.com**.

Document Number: QUICCENGINEFS
REV 0