TOEnK_2G : Datasheet

TCP/IP Offload Engine *2Gbit/second Embedded TOE core*

Overview

TOEnK_2G is a hardware state-machine based TCP/IP offload engine designed to provide embedded TOE functionality at 2Gbps rates, full-duplex

TOEnK_2G is targeted at embedded applications that require high bandwidth TCP/IP traffic without the requirement for a large processor, or applications requiring low CPU overhead

TOEnK_2G uses internal memory to store Connection Table data for up to 2k Channels. A dedicated channel is provided to allow a bypass to a software stack. The Connection Table can be programmed via a standard CPU-type interface.

When coupled with an Application Buffer Control function, a comprehensive Linux software solution is available. Acceleration of TCP sessions is 100% transparent to users, with the TOE driver detecting traffic levels and automatically moving control between the Linux stack and TOE.

Low latency solution, typically 0.2µs on Rx (lookup latency), 6µs on Tx (full pkts to calculate checksums), at 166MHz

Comprehensive VMM-based verification suite. Fully tested (hw and sw) on FPGA-based Linux platforms.

Specifications

TOEnK Core:

- Provides TCP termination over both IPV4 and IPV6 with a minimal amount of processor overhead.
- Supports up to 2k accelerated TCP sessions, storing Connection Table (CT) data per session in internal memory. Parameterisable for any number of connections up to 2k.
- TCP session setup and teardown controlled by external processor with hardware-assist by the TOEnK core. Control of sessions is passed seamlessly between the processor and TOE core.
- Transparent channels are provided to allow CPU access to the Tx and Rx GMAC interfaces.
- The Tx Application Buffer Control block generates requests to the transmit active queue (TxActiveQ), receives indications of amount of data transmitted, amount of transmitted data acknowledged and requests to provide retransmit data.
- Tx Data is transmitted based on active session on the TxActiveQ and transmit opportunities based on window sizes.
- Rx packets are demultiplexed to channels based on IP addresses and port numbers. Unidentified packets are forwarded to the transparent channel.
- The Rx Application Buffer Control handles received data and indicates receive buffer status to the TOE, to facilitate acknowledging data and updating the receive advertised window size per channel.
- The Rx and Tx engines update the CT based on packets transmitted and received. CT bandwidth allows both Rx and Tx table accesses at line rate.
- All major TCP functionality handled by hw state-machines, including congestion control, sliding window, Tx and Rx advertised window support, slow start, Tx and Rx silly window avoidance, delayed Ack, fast retry, RTT calculation per connection, window probe and persist timer support, MSS per connection, TCP and IP checksums. SACK is not supported; out of order Rx packets are filtered; IP fragmentation not supported.

Linux Support:

For a full solution, when coupled with an Application Buffer Control block customised to user requirements, a comprehensive Linux-based software suite is provided, consisting of a modified Linux stack (currently 2.6.38.2) and a TOEnK driver, fully tested in the application below. Acceleration of TCP connections is 100% transparent to user applications: the driver detects traffic levels on TCP connections and automatically moves traffic handling from Linux to the TOE core, and back when traffic levels subside.

Typical Application: The TOENk core is shown below in a typical application, where it interfaces two Gigabit Ethernet Ethernet channels to a 64-bit Axi system. The TOEnK core is independent from the system environment, with generic interfaces at both line and system sides. The TOE core is available separately, or can be combined with a customised version of the Application Buffer Control function to match the target application.