TCP Offload Engine IP (TOE)
For Latency Critical, FPGA-based
Embedded Networking Applications

Features
- FPGA TCP Offload Engine (TOE) IP for networking applications requiring minimum latency and deterministic latency
- Supplied as encrypted .ngc (Xilinx) or optional verilog source
- Integrated PCIe bridge (required) provided in encrypted .ngc format
- Complete simulation models and text fixtures
- Host CPU NOT involved in payload data transfer
  - 0% CPU load during middle of TCP session
  - TCP data packets handled by TOE not passed to CPU
- Full 10Gbe line rate
  - No Ethernet pause frames generated
- CPU required only for High complexity/low importance network features:
  - Setup/teardown of TCP session
  - ARP, ping, DHCP, SMTP, et. al.
  - Linux driver with ‘C’ source included
- Layers 2, 3, 4, 5 (datalink, network, transport, and session)
- Layers 6, 7 (presentation, application) is user’s responsibility in FPGA
- MTU of 1536 bytes
- CRC validation and checksum validation
  - Ethernet CRC validation
  - IP and TCP checksum validation
- Reordering of out-of-order packets
- Nagle algorithm
- Fast retransmit
- Congestion avoidance
- Packet retransmission upon error/lost/out of order packet reception
- 1 TCP/IP session per instantiated TOE
- Additional TOEs can be cascaded to support multiple sessions
  - Limited only by FPGA resources
  - Client or server mode
  - Configurable TX and RX replay buffer
  - 4KB → 64KB
  - Protection Against Wrapped Sequences (PAWS)
  - Configurable port number
  - IPv4 with future upgrade paths to IPv6/IPng
  - Initial version targeted to the DINI Group DNPCIe_10G_HXT_LL with Virtex-6 HXT
  - Cost reduced Kintex-7 version coming in early Q2’12
  - Altera Stratix-5 in Q3’12
  - Direct interface to the Xilinx 10 Gigabit Ethernet Media Access Controller (10GEMAC) (required)
  - 64-bit bus interface:
    - Synchronous FIFO clocked at 156.25Mhz
    - Optional asynchronous FIFO interface with 4-6 clocks cycles of added latency

Block Diagram

Description
Overview
TCP Offload (TOE) is FPGA-based IP that receives and transmits Ethernet/IP/TCP packets on Ethernet networks. TOE delivers payload data, in order, to the user’s application with:
- Extra TCP/IP packet fields removed
- No missing data
- Verified by appropriate CRCs and checksums
- Flow control
The purpose is to offload the TCP/IP function from the CPU and perform it directly in FPGA-based hardware. TOE dramatically reduces the input to output response time and jitter by eliminating the need for host processor intervention when analyzing data packets. This IP is designed to be utilized in FPGA-based high frequency, low latency Wall Street trading applications. Input to output packet latency of less than 1us can be achieved. Assuming a 100-byte payload (164-byte packet), the theoretical minimum input to output latency is about 500 ns.
The TOE works at the full 10 Gbe line rate and was developed internally at DINI Group. TCP Offload is a required function in low latency networking application. Data critical functions are executed directly in the FPGA. Infrequent, non-data TCP/IP functions such as setup/teardown, ARP, ping, DHCP, et al) are passed through to a standard Linux driver. Other, software based TCP sessions run normally with no changes required. At the intended target frequency of 156.25 MHz, the TOE operates at the full 10GbE line rate, generating no Ethernet pause frames.
**What basic functions are required?**

### 10 GbE Media Access Controller from Xilinx

In minimum latency approaches, it is necessary to avoid using external PHYs since they add significant latency. This IP assumes an FPGA PHY is used. The Xilinx PHY needs a MAC, and the **10 Gigabit Ethernet Media Access Controller (10GEMAC)** is required to use this TOE IP. You purchase this separately from Xilinx as it is not included. Note that Xilinx has a free version that disables itself after a few hours. This free version contains all of the functionality of the full version and can be used for evaluation.

The TOE IP can connect to the slower 1 GbE MAC and we can make modifications here in La Jolla to interface the TOE to different MACs. Contact DINI sales for more information.

This Xilinx core is compatible with the Virtex-6 HXT FPGAs and works fine on both Virtex-7 and Kintex-7. Features of the Xilinx 10GEMAC include:

- Designed to *IEEE 802.3-2005* specification
- Configured and monitored through an independent microprocessor-neutral interface
- Optional Statistics counters
- Configurable flow control through MAC Control pause frames; symmetrically or asymmetrically enabled
- Generate customized core using the CORE Generator™ technology
- Cut-through operation with maximum buffering for maximum flexibility in 64-bit client bus interfacing
- Ability to generate core with no physical interface to allow users to connect the PHY-side interface of the core to user logic
- Powerful EtherStats-based statistics gathering
- Programmable Interframe Gap
- Custom preamble preservation mode
- Supports Deficit Idle Control (DIC) for max. data throughput
- Maintains minimum IFG under all conditions and line rate performance
- Remote Fault/Local Fault signaling at the Reconciliation Sublayer

We use the *AXI4-S* bus interface option. Our testing and debug was performed using the unrestricted version of this core.

### PCIe Bridge (GEN1/GEN2)

A host interface is required to handle a number of functions related to the TOE with configuration being the most important. A PCI bridge is supplied in encrypted net list format (.ngc) for this purpose. The PCIe Bridge has 4-lanes of GEN1/GEN2, and is a full function PCIe core. Configuration, BARs (base address registers), and master-moding DMA engines are included. Drivers with ‘C’ source for Linux are included.

### TOE

The TOE implements the TCP function directly in FPGA gates. No external FPGA memory is required. TOEs can be cascaded to support multiple sessions. The TOE IP is intended to be clocked at the standard Ethernet interface frequency of 156.25 MHz, allowing fully synchronous and lowest latency data exchange with the MAC. At 156.25 MHz, the TOE operates at the full 10GbE line rate, generating no Ethernet pause frames. The IP is supplied either as an encrypted .ngc netlist for implementation in Xilinx-based FPGAs or as verilog source to do with as you see fit. Altera Stratix-5 and ASIC versions will follow shortly. A host interface is required and this IP package includes an integrated 4-lane GEN1/GEN2 PCIe bridge. Simulation models and test fixtures are included.

This IP is optimized for low latency: the host CPU is **NOT** involved in payload data transfer. Not all TCP functions are handled in the IP. High complexity/low importance network features such as setup/teardown, ARP, ping, DHCP, et. al. are passed to a Linux driver via the PCIe interface. ‘C’ source for this driver is included, allowing customization.

All of the functions associated with TCP/IP layers 2, 3, 4, 5 (datalink, network, transport, and session) are implemented. The user is responsible for presentation layer 6 and application layer 7 and can be implemented in the FPGA or elsewhere. The maximum transmission unit (MTU) is 1536 bytes. CRC validation and checksum validation and reordering of out-of-order packets are done directly in the FPGA, along with packet retransmission upon error/lost/out of order packet reception. The TX and RX replay buffers are configurable: 4KB →64KB. Protection Against Wrapped Sequences (PAWS) is handled in the FPGA.
FPGA/ASIC Resources Required

<table>
<thead>
<tr>
<th>Module</th>
<th>slices</th>
<th>FFs</th>
<th>LUTs</th>
<th>lut-ff-pairs</th>
<th>BRAMs</th>
<th>bufg</th>
<th>mmcm</th>
<th>gtx</th>
<th>% of HXT585</th>
<th>% of HXT380</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOE1 (8kb)</td>
<td>2650</td>
<td>3889</td>
<td>6875</td>
<td>8245</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.0%</td>
<td>4.4%</td>
</tr>
<tr>
<td>TOE1 (64kb)</td>
<td>2650</td>
<td>3899</td>
<td>6885</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.7%</td>
<td>4.4%</td>
</tr>
<tr>
<td>NIC</td>
<td>274</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.1%</td>
<td>1.3%</td>
</tr>
<tr>
<td>PCIe</td>
<td>8757</td>
<td></td>
<td></td>
<td></td>
<td>78</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>9.9%</td>
<td>14.7%</td>
</tr>
<tr>
<td>10GEMAC</td>
<td>1036</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.2%</td>
<td>1.7%</td>
</tr>
</tbody>
</table>

TOE IP Distribution Model

The TOE IP is distributed in two different ways:

- encrypted .ngc file
- complete verilog source

Model 1: Xilinx .ngc file

An .ngc file enables integration at the place and route stage into the Xilinx FPGA tools. Source is not provided, but full simulation libraries are supplied. You get this version when you get the FIX support package for our FPGA boards: **DN_FBSP**. Required operating system driver functions and APIs are supplied, with source, in ‘C’ for Linux.

The TOE IP, supplied as part of the **DN_FBSP**, is restricted to DINI products and will not operate on other FPGA-based boards. You are welcome to deploy this IP free of royalties or restrictions on DINI Group products. A single **DN_FBSP** license is required for your company and allows your company to use it worldwide in any number of DINI boards and any number of applications.

Model 2: Verilog Source

Verilog is our native language. This second distribution option gets you the complete source. You are not allowed to redistribute the source. The license agreement has all the details and the information in the license agreement supersedes what is written here.

Under extreme duress and only under extreme duress, we will convert to VHDL. Should we do this conversion, please note that new features and bug fixes will be first available in Verilog. We really don’t like VHDL and all reputable synthesis tools accept mixed language RTL anyway.

A maintenance contract, for bug fixes and feature enhancements is probably a good idea. 1 year is required at the time of purchase, with optional extensions sold on a yearly basis. Contact **DINI sales** for more details.