Off-loading TCP/IP into hardware makes Gigabit Ethernet a reality for your application

TCP/IP Offload Engines (TOE) can deliver over 100MBytes/s data rate (in each direction), low latency and simple interfacing between Gigabit Ethernet and your embedded device

Introduction

From its origins as an experimental cable network developed by the Xerox Corporation, Ethernet has evolved over 3 decades to become a family of de facto standard technologies and protocols for modern communications networks. Ethernet is used for approximately 85 percent of the world's LAN-connected PCs and is being increasingly deployed in embedded and industrial networking.

Ethernet performance has increased from megabits per second (Mbits/s) to gigabits per second (Gbits/s) and its popularity reflects not only its status as an IEEE standard, but because the Ethernet protocol has a number of features and benefits that have proved attractive to designers and engineers:

- Long established and well understood technology;
- Allows low-cost network implementations;
- Provides extensive topological flexibility for network installation;
- By embedding Ethernet onto smart, connected devices they have the capability to communicate via Ethernet without using a computer; and,
- Ethernet can guarantee successful interconnection and operation of standards-compliant products, regardless of manufacturer.

Though an IEEE standard, there are many different flavors of Ethernet available. Within industrial networking, Ethernet has been aggressively promoted with more than 20 different variants of 'Industrial Ethernet' competing for the industrial applications market (e.g. PROFINET, Modbus/TCP). This, together with the widespread deployment of Ethernet technologies and ever increasing data rates has given rise to the need for increasingly low cost and high bandwidth interfaces that are simple to integrate and use.

There also remains a significant design and deployment issue with Ethernet, that of the high CPU over-head of running a full TCP/IP stack, and high latency when compared to other networking solutions. As bandwidths increase the processor spends more of its time handling incoming frames rather than running user algorithms.

In this paper we discuss how developers who are looking to introduce or optimize Gigabit Ethernet can defeat the TCP/IP overhead through offload, and accommodate the many different Ethernet standards (e.g. Industrial Ethernet, GigEVision) on a single, low cost universal platform such as the ZestETM1 (which we discuss later). We highlight the benefits of a TCP/IP Off-load Engine (TOE) that is easy to use, simple to design with and is supported with interfaces that improve productivity, lower cost and accelerate the deployment of Gigabit Ethernet.

Introduction to IP, TCP and UDP

Ethernet networks use a 'stack' of standards that include:
Application Layer HTTP· DHCP· DNS · FTP · GTP · BGP · IMAP · IRC · Megaco · MGCP · NNTP · NTP · POP · PTP · RIP · RPC · RTP · RTSP · SDP · SIP · SMTP · SNMP · SOAP · SSH · Telnet · TLS/SSL · XMPP
Transport Layer TCP · UDP · DCCP · SCTP · RSVP · ECN
Internet Layer IP (**IPv4**, IPv6) · **ICMP** · ICMPv6 · IGMP · IPsec
Link Layer ARP · **Ethernet** · RARP · NDP · OSPF · Tunnels (L2TP) · PPP · Media
Access Control MPLS · DSL · ISDN · FDDI · Device Drivers

Orange Tree Technologies Ltd
The board-level embedded hardware and software solutions company
www.orangetreetech.com
Tel +44 1235 838646
Email: info@orangetreetech.com

The protocols in bold form the core of the communications protocols across the vast majority of today's local area networks (LANs). In order for a device to connect to an Ethernet network, an implementation of each of the protocols is required. The italicised protocols (HTTP and DHCP) are not mandatory, but they are implemented by a large number of network devices for convenience.

The Transmission Control Protocol (TCP) in particular is one of the core protocols of the Internet Protocol Suite. TCP was one of the two original components (with Internet Protocol (IP)), of the suite, so that the entire suite is commonly referred to as TCP/IP. Whereas IP handles lower-level transmissions from computer to computer as a message makes its way across the Internet, TCP operates at a higher level, concerned only with the end systems, for example, a Web browser and a Web server.

Benefits of TCP offload

Traditionally TCP/IP is implemented in software and executed on a processor. With the advent of higher bandwidth networks this has become a major bottleneck in data transfer as the processor must spend more of its time handling incoming frames rather than running user algorithms. This performance degradation negatively impacts network efficiency and is inconsistent with real-time applications. Moreover as ever more powerful processors are used to improve performance, BOM costs increase as does the physical size of the host card or module.

To solve this bottleneck, more functions are now being offloaded into dedicated hardware. For example, most network cards will perform checksum offload (a task that the processor is particularly unsuitable for). By selectively offloading parts of the TCP/IP stack to hardware, vast improvements in transmission bandwidth can be achieved.

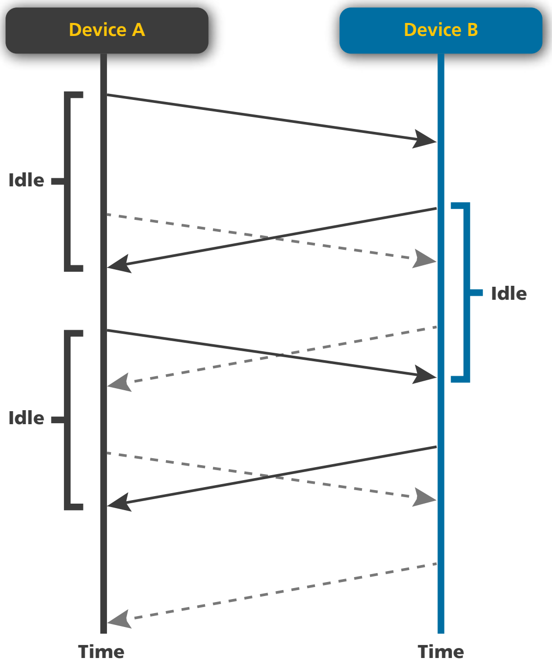
TCP achieves its robustness by forcing the receiver to acknowledge receipt of data. If either the data or the acknowledgement is lost in the network then the sender will detect this and re-transmit the data. In a naïve implementation, this means the sender is idle while waiting for acknowledgement from the receiver (see Figure 1). In actual fact, TCP allows the sender to send further data (represented by dotted lines in figure 1) before receiving an acknowledgement, but the amount it can send is limited.

Minimizing the round-trip time between sender and receiver is critical for improving the bandwidth. Since the delay through the network is outside the device's control, this comes down to minimising the delay between a receiver receiving data and sending an acknowledgement, and the delay between a sender receiving an acknowledgement and sending the next piece of data.

By offloading these parts of the TCP stack into dedicated hardware, such as Orange Tree's GigExpedite (GigEx) TCP/IP Off-Load Engine (TOE), it is possible to saturate the bandwidth of a gigabit network and minimize the delay, or latency, between the receipt and acknowledgement of data. Importantly, the GigEx TOE also contains a standard processor to handle the irregular parts of the TCP algorithm that are not good candidates for hardware acceleration. This means that the remaining system does not need high levels of intelligence or processing power to be able to connect to the network.

Tel +44 1235 838646
Email: info@orangetreetech.com
www.orangetreetech.com

Figure 1: TCP Receipt and Acknowledgement of Data



GigExpedite Architecture

The GigExpedite, or GigEx device mounted onto the Orange Tree's ZestETM1 and ZestET1 modules integrates hardware components optimised for acceleration and a conventional 32 bit processor to provide a complete implementation of a TCP/IP stack including application layer HTTP and DHCP, transport layer UDP and TCP, internet layer IPv4 and ICMP, and ARP and Ethernet from the link layer.

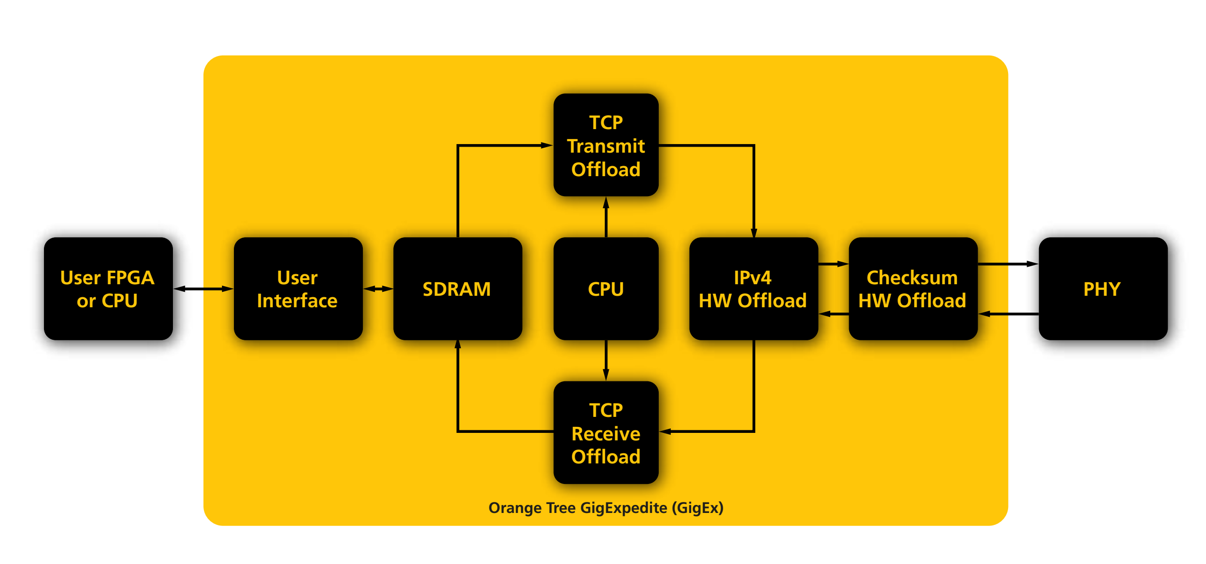
The external components are a Gigabit PHY and magnetics pair, SDRAM for buffering data and flash memory containing the processor firmware.

The GigEx device presents a generic processor interface and register map to the external user device. The ZestETM1 module exposes this interface directly to an external user device whereas the ZestET1 incorporates a 1.4 million gate Xilinx Spartan-3A XC3S1400A companion FPGA. The ZestETM1 allows convenient bridging to systems already incorporating a processing engine whereas ZestET1 is a standalone system in its own right.

Internally, the GigEx device integrates Ethernet MAC, checksum offload, IPv4 (including reassembly), UDP processing and TCP flow control hardware blocks. The internal processor implements TCP session control and the higher level protocols including DHCP, AutoIP, UPnP and HTTP. This high level of integration means that the user device can be very simple – it needs to perform very little initialisation and its main task is streaming data to and from the network.

The GigEx device also incorporates a web server for simple configuration from a remote host using a conventional web browser. In addition, ZestETM1 provides a user web server allowing customisation of the web configuration interface for user applications.

Figure 2: GigExpedite Integrated Hardware UDP & TCP/IP Offload Engine (TOE) Block Diagram



Using the GigEx Device

The GigEx device has been expressly designed for high performance and simplicity of design with no detailed networking knowledge required. For the developers working at the board level this makes the ZestETM1 and ZestET1 modules easy to use and highly productive in terms of time-to-market and flexibility.

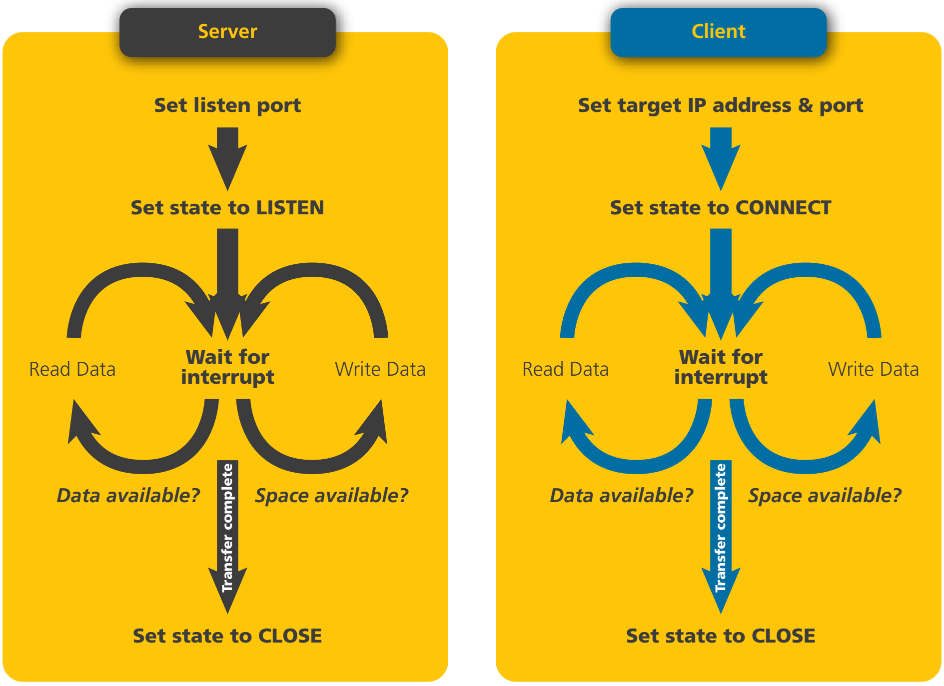
Unlike conventional systems that require complex integration of existing TCP/IP stacks or even complete operating systems, GigEx enabled devices require only the intelligence to program registers. In some simple applications, even this requirement can be removed enabling networking support for dumb devices without additional processing engines.

User devices are freed up to process data and the developers are freed to focus their development effort on algorithm development and optimisation.

The GigEx device is capable of acting as a network server or client and supports up to 16 simultaneous network connections and is controlled from the User device using a simple set of registers. The User device performs simple initialisation of the GigEx device as shown in Figure 3 overleaf.

Tel +44 1235 838646
Email: info@orangetreetech.com

Figure 3: Initialisation of the GigEx device



Once initialised, the GigEx device handles all TCP session setup and tear down leaving the User device to process interrupts when data is received from the network or when data needs to be sent to the network. The ZestETM1 device includes additional interfaces which simplify connecting the GigEx chip to non-processor user devices.

GigEx Performance

Unlike software based TCP/IP stacks that are implemented into the host CPU, the GigEx device off-loads the TCP/IP protocols into its dedicated silicon. This frees the host processor or companion FPGA to run applications, rather than handle network traffic.

Latency	6µsec	
Throughput	User interface –> PC	PC -> User interface
(max sustained rate)	115MBytes/s	115MBytes/s

With no external processing the GigEx device is capable of saturating a Gigabit Ethernet network with data >100Mbytes/s in each direction. Its hardware acceleration increases bandwidth and reduces transfer latency to help designers meet the specifications of demanding real-time applications.

Tel +44 1235 838646
Email: info@orangetreetech.com

ZestETM1

The ZestETM1 board integrates Orange Tree's GigEx device with a minimal set of peripheral components in a small footprint module (30mm x 25mm) for integration into a larger user system. Figure 4 shows a block diagram of the ZestETM1 module. The user system must include Ethernet magnetics and RJ45 connectors and will normally have an application processor or FPGA on the carrier board but can also connect devices directly to the module.

As a drop-in networking solution, ZestETM1 reduces design costs and speeds time-to-market. The overhead of integrating software networking stacks into existing processor applications or the creation of custom networking hardware designs in FPGAs is removed. ZestETM1's flexible interfaces simplify connectivity with external devices and remove the need for glue logic between a CPU bus and transceivers.

ZestETM1's embedded web server brings a familiar user-friendly control and status mechanism to remote systems and allows customisation and branding of connected devices.

In a world where Ethernet is gaining ground in industrial networking applications, ZestETM1's high precision timing control hardware and software enables synchronisation of devices across a network using standard Synchronous Ethernet and IEEE 1588 PTP protocols.

Figure 4: ZestETM1 GigE TOE Module Block Diagram

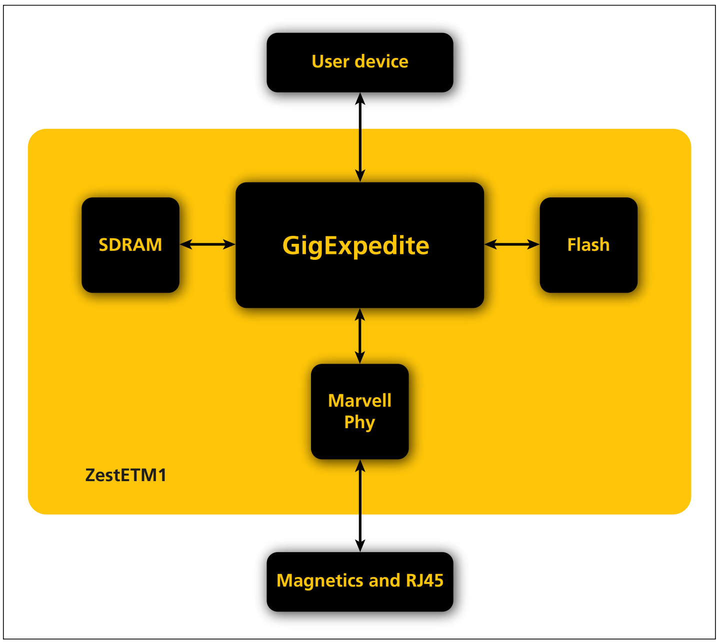
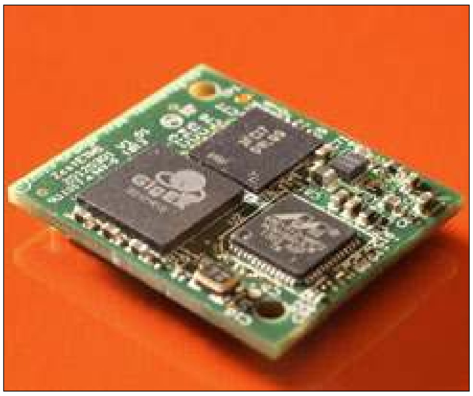


Figure 5: ZestETM1 GigE TOE Module



ZestET1

The ZestET1 module combines the power of Orange Tree's GigEx device with the versatility of a user programmable FPGA. With its low price point, ease of use and compact form factor (50mm x 75mm), the module is ideally suited to integration in embedded systems and OEM equipment. It features a user programmable Xilinx Spartan-3A FPGA with up to 1.4M system gates that are completely free for user programming. The FPGA is supported with 64MBytes DDR SDRAM, DDR333 speed, 16 bits data bus.

The companion FPGA can be programmed from on-board Flash, Ethernet or JTAG and is capable of running soft-core processors and higher-level protocols such as GigE Vision and Industrial Ethernet.

Orange Tree Technologies Ltd
The board-level embedded hardware and software solutions company
www.orangetreetech.com
Tel +44 1235 838646
Email: info@orangetreetech.com

The versatility of the FPGA to run soft-core processors such as MicroBlaze enables the developer to implement processor functionality entirely in the general-purpose memory and logic fabric of the FPGA. No external or independent processor is needed (unless specified in the design) and the ZestET1 can offer a complete embedded solution. The FPGA can be used to build upon the core communications protocols provided by the GigEx device (IPv4, TCP, UDP, DHCP Client, Auto IP, UPnP, HTTP, ARP) and be quickly and cost-effectively extended to implement application layer protocols such as GigE Vision and the Industrial Ethernet standards. This unique capability offers the developer a common platform that can be applied to multiple projects and standards.

The FPGA also provides a programmable interface to external devices via the 80 pins of user IO and can be used for processing and formatting of data to be streamed over the Ethernet interface. In particular the ZestET1's use of a standard socket interface and simple register interface make it incredibly easy to use and quick to deploy. It avoids the cost and integration headache of PCI type interfacing and allows more of the FPGA logic to be dedicated to processing tasks.

Conclusion

The ubiquity of Ethernet and the relentless growth of higher bandwidth networks have driven the need for solutions to the processor overhead that degrades network and application performance.

TOE's are becoming the preferred solution and offer compelling benefits over processor-only approaches. As TOE's become more mature they can be integrated into easy-to-use, compact form factor modules to deliver more functionality to the developer.

As Ethernet enters new markets, designers without detailed networking knowledge and experience face the challenge of implementation. A flexible, easy to use interface closely coupled with TOE technology is an important feature that will help scale Ethernet adoption in non-core markets.

Glossary

Ethernet: *Low-level protocol for local area networks including definition of cabling, electrical signalling and framing of data.*

ARP: *Address Resolution Protocol. Used to determine physical addresses of devices on the network.*

IPv4: *Internet Protocol. Defines frame format and checksum to ensure correct delivery of single packets of data. However, due to variable routing paths, electrical interference and the lossy nature of networks it does not guarantee delivery of data or the order or delivery of data.*

ICMP: *Internet Control Message Protocol. Used by devices on the network to communicate control and status data including error information.*

UDP: *User Datagram Protocol. Lightweight user level protocol for transferring data between 'ports' on devices. Allows multiple streams of data to run over a single network between two devices. UDP is lightweight and simple but unreliable and does not guarantee data reception or order of reception.*

TCP: *Transmission Control Protocol. Heavier user level protocol for transferring data between 'ports' on devices. Allows multiple streams of data to run over a single network between two devices. Guarantees data reception and order of reception.*

DHCP: *Dynamic Host Configuration Protocol. Allows devices to configure their own addresses on a network with a suitable DHCP server.*

AutoIP: *Allows devices to choose their own address on a network without a DHCP server.*

HTTP: *Hypertext Transfer Protocol. Protocol sitting above TCP used to transfer HTML web pages.*

UPnP: *Universal Plug-and-Play. Allows devices to search for and query the capabilities of other de-vices on a network.*

Tel +44 1235 838646
Email: info@orangetreetech.com

Port: *TCP and UDP use the concept of ports to multiplex multiple data streams across the same net-work. Data is transferred between a port on one device and a port on a second device. Data transfer sessions between port pairs are kept separate.*

MAC: *Media access controller. Component that transmits and receives packets on a network.*

Phy: *Electrical interface to network cable.*

About Orange Tree Technologies

Orange Tree Technologies is a board level embedded hardware and software company specializing in high-speed embedded device interconnect and FPGA technologies.. Used by some of the world's leading technology companies our products and services help address the challenges of convergence in the defense, industrial, scientific and consumer electronics markets.

Orange Tree Technologies has been providing high-speed embedded device interconnect solutions since 2001. OEM engagements are supported through customization via Orange Tree's dedicated design services function. Headquartered in Oxfordshire, UK, Orange Tree Technologies is a privately held company and operates internationally.

For more information visit

www.orangetreetech.com

Tel +44 1235 838646
Email: info@orangetreetech.com