Interlaken Technology: New-Generation Packet Interconnect Protocol

White Paper

08 March 2007

Mark Gustlin: Cisco Systems, Inc.

Fred Olsson: Cortina Systems, Inc.

Matt Weber: Silicon Logic Engineering, Inc.

Contents

Figures

1.0 Abstract

Serial link technology has multiplied device interconnect bandwidths in advanced communications equipment. Interlaken is an interconnect protocol optimized for high-bandwidth and reliable packet transfers. It uses bundles of serial links to create a logical connection between components with multiple channels, backpressure capability, and data-integrity protection to boost the performance of communications equipment. This white paper gives an overview of Interlaken's features and an implementation case study.

2.0 Design Goals

2.1 Protocol Description

Components with gigabit-scale throughput traditionally have data buses running about 100 Mbps per pin. Differential signaling technology has increased this bandwidth almost ten times, to about 800 Mbps per pin pair, which enables components with throughputs on the order of 10 Gbps. New serial technology with clock and data recovery has increased bandwidth another ten times to about 6 Gbps per pin pair, which enables components with multiple 10 Gbps streams. It has also enabled a 90 percent reduction in I/Os and board traces over previous generations.

The Interlaken protocol was created to take advantage of this latest serial technology for a high-speed, robust, versatile interface for packet transfers between components within communications systems.

2.2 Bandwidth Range

Interlaken has no inherent upper limit but it is primarily targeted for 10 Gbps to 100 Gbps connections. This wide range makes it suitable for many applications and allows generations of devices to be backward-compatible. Interlaken implementation would be suitable in MACs with many 10 Gbps ports, OC-768 SONET framers, next-generation 100 Gb Ethernet integrated circuits (ICs), and 100 Gbps switch fabrics and packet processors.

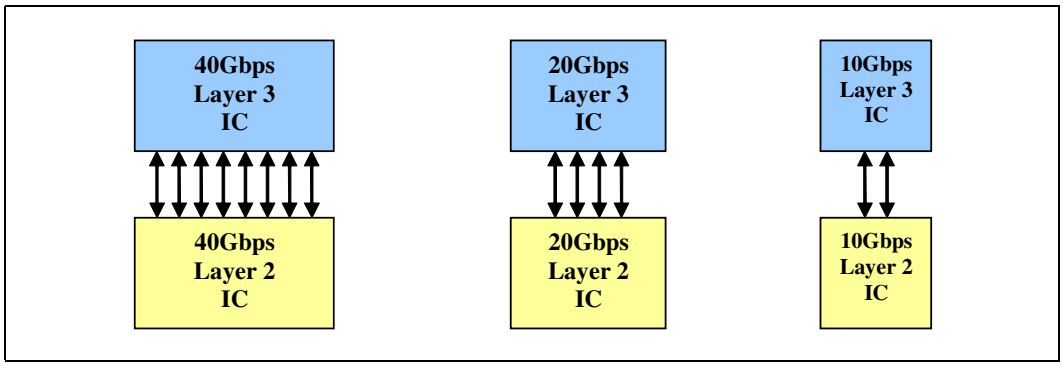
2.3 Scalability

Interlaken's scalability is achieved by its ability to run over a varying number of lanes. The following two parameters determine the connection bandwidth:

1. **Number of serial lanes in the interface**

   Any number of serial links (or "lanes") can be used in an Interlaken interface. Effective bandwidth corresponds directly with the number of lanes. For example, an eight-lane interface can carry twice the payload of a four-lane interface running at the same per-lane speed, as shown in Figure 1.

Figure 1 Effective Bandwidth Scales with the Number of Lanes

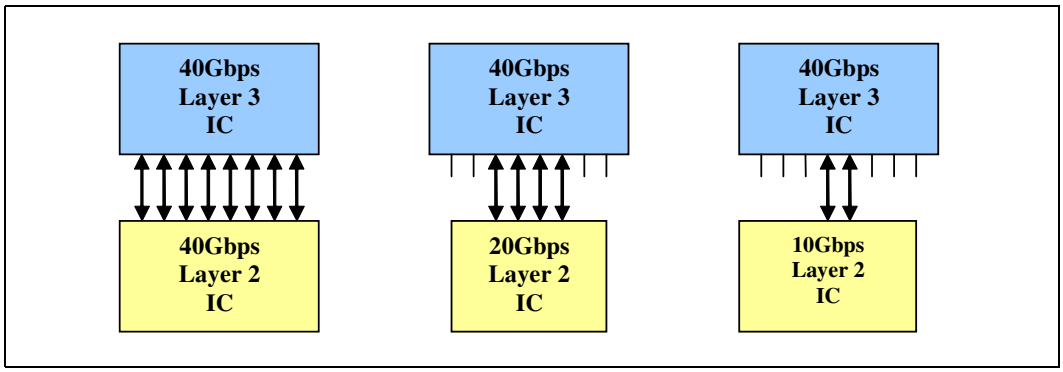


2. **Frequency of each lane**

   Effective bandwidth also scales directly with the per-lane bit rate. For example, a 3.125 Gbps port can carry half the payload of a 6.25 Gbps port using the same number of lanes.

Since bandwidth can be increased by either adding more lanes or by increasing the bit rate per lane, Interlaken is a very scalable interface. For example, as shown in Figure 2, an IC with a capacity of 40 Gbps can connect to other 40 Gbps ICs using eight lanes, to 20 Gbps ICs using four lanes and to 10 Gbps devices using two lanes. Thus, ICs of different capacities can be made to interoperate, which enables backward compatibility.

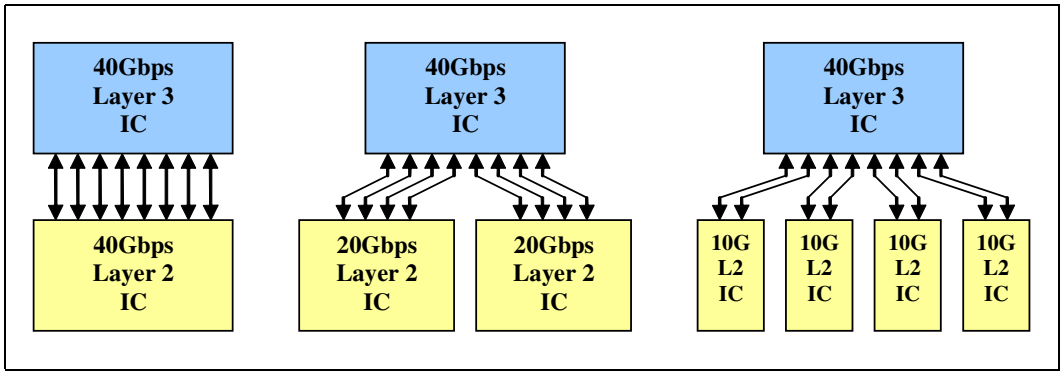
Figure 2 Flexibility of Interlaken Allows ICs of Different Capacities to Connect



2.4 Flexibility

Interlaken's ability to run over a varying number of lanes provides high flexibility in component interconnects. Different-capacity ICs in a single physical interface can be split up into multiple lower-speed physical interfaces. For example, as shown in Figure 3, eight physical lanes can be organized as a single 40 Gbps interface, two 20 Gbps interfaces, or four 10 Gbps interfaces. In this example, a higher-bandwidth IC can therefore connect to multiple lower-bandwidth ICs to provide an increased system port count.

Figure 3 Flexibility of Interlaken Allows Multiple Connection Options



2.5 Channelization

In many applications it is important to provide many logical channels across a physical interface. For example, different channels can be used to carry traffic destined for separate physical ports, logical SONET channels, or traffic priority flows.

Interlaken is designed to natively support 256 channels, extendable up to 64 K channels, by using a dual-use channel-field extension that meets the requirements of most applications.

2.6 Resiliency

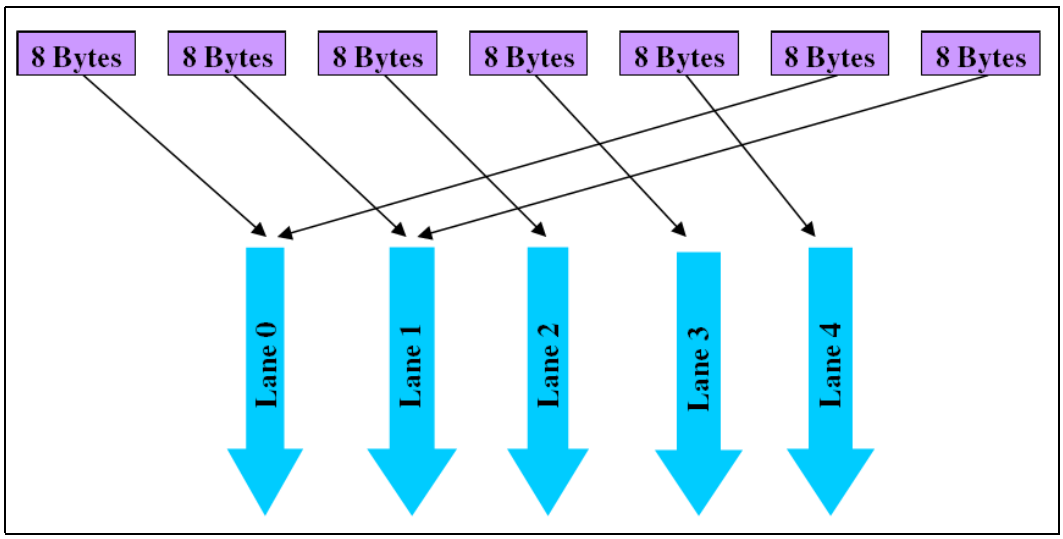
Bit errors can occur across any serial link. Interlaken minimizes the impact of bit errors by avoiding scrambling error multiplication and by using strong cyclic redundancy check (CRC) protection for each transfer. The health of each serial link is continuously and transparently monitored.

3.0 Functionality

3.1 Striping Data for Scalability

How data is striped across an interface determines how easily the interface can increase in bandwidth. Interlaken operates on 8-byte words that are distributed to all lanes. The more lanes, the more words transmitted during each interval. Since transmission scales in steps of 8 bytes, an interface can efficiently increase bandwidth by supporting a range of lanes.

Figure 4 Lane Striping



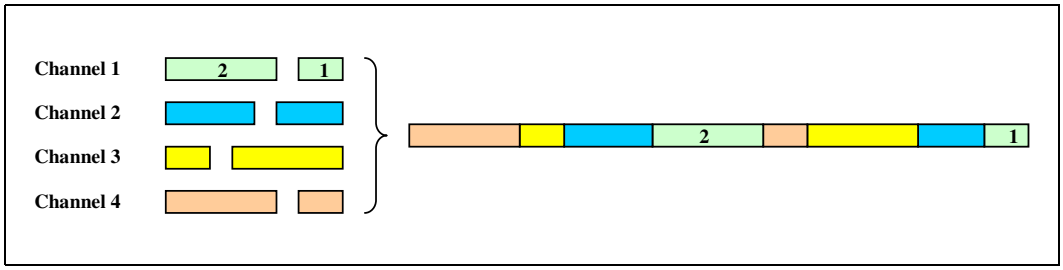
3.2 Burstable for Low Latency

There are two basic methods of sending packets across an interface; interleaved and non-interleaved.

- **Non-interleaved packet transmission**

  The transfer of a packet is always completed before transfers are started on another channel (see Figure 5).

Figure 5 Non-interleaved Transfers



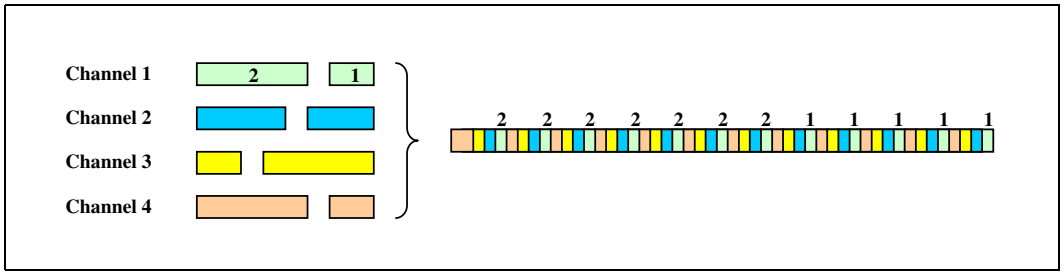
Because full-length packets are sent, buffers on each side of the interface must be capable of absorbing data on the other channels while a packet is transferred on one

channel. Since full packets are sent without fragmentation, no reassembly is needed on the receiving side.

- **Interleaved packet transmission:**

  Each channel transmits only a small fragment of a packet before moving to the next channel (see Figure 6).

Figure 6 Interleaved Transfers



Here data is transmitted in small bursts as it become available, which cuts buffer-capacity requirements to a minimum and therefore reduces latency through the interface.

It is important for Interlaken to support *both* non-interleaved and interleaved transfers since either can be optimal depending on the application.

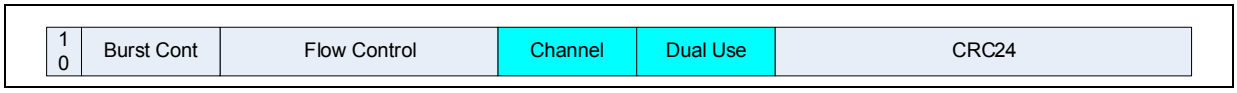
3.3 Channelization

Interlaken is designed to support multi-channel or multi-port applications with ease. The burst control word has a channel field that contains the ID of the channel or port being carried across the interface. Through this mechanism, many applications can be supported.

In a trivial but still common application, a single port or channel is carried across the interface, one burst at a time, with the channel field always set to the same value. A more typical application might be a 24-port Ethernet MAC. In this case, the traffic on each port would be sent with a unique channel ID on the Interlaken interface. At the extreme end, applications that support thousands of channels can be supported by using the dual-use field combined with the normal 8-bit channel field. In this way, up to 64 K channels can be supported. This is enough for even the most demanding applications such as highly channelized SONET/SDH interfaces.

The layout of the burst control word is shown in Figure 7 (with the channel and dual-use fields highlighted).

Figure 7 Burst Control Word with Channel Field



3.4 Flow Control

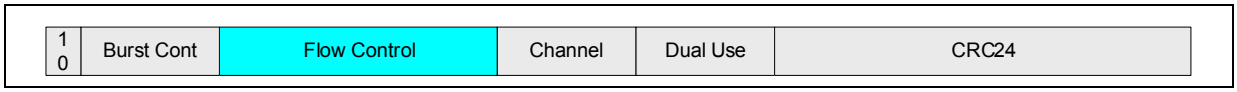
Another important tool required in a packet interface is backpressure or flow control. Since Interlaken normally runs asynchronously to any line interfaces, and it carries packets for many channels, some type of flow control is necessary to prevent buffer overflow and to allow for rate matching between devices on a board.

Interlaken provides for simple on and off indications (commonly called Xon/Xoff) that tell the transmitting side when to stop sending it. An Interlaken sink device will typically have a per-channel buffer with a programmable flow control threshold. When the buffer is filled above its threshold, the sink device indicates this by sending Xoff to the Interlaken source device. The source device then stops sending traffic for that channel. In a similar manner, once the buffer is drained below the threshold, the sink sends an Xon to the Interlaken source, telling the source to start sending traffic once again for that channel.

The buffer size and the thresholds must be set with channel rate, flow control latency, source scheduling responsiveness, and other factors in mind. If the thresholds and buffer depths are set correctly, then no packets are ever dropped in the sink and the line is always fully utilized.

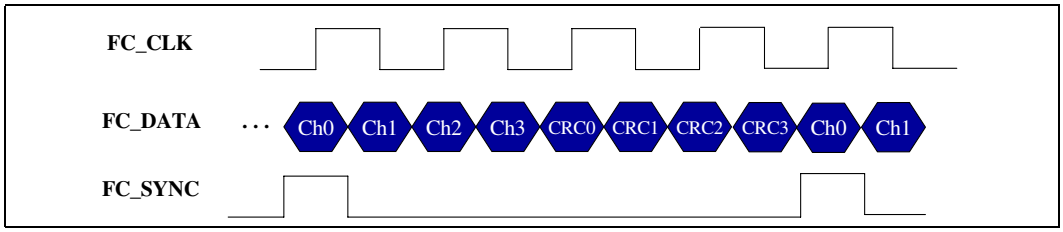
Interlaken has two methods for sending the Xon/Xoff flow control messages. In-band flow control is carried within the burst control word (see Figure 8) and is typically used for bidirectional applications where the source and sink are in the same device.

Figure 8 Burst Control Word with Flow Control Field



Out-of-band flow control is carried on a simple 3-bit bus. This is more appropriate when the applications are unidirectional or when the source and sink are not in the same device. Figure 9 shows the out-of-band flow control bus.

Figure 9 Out-of-Band Flow Control

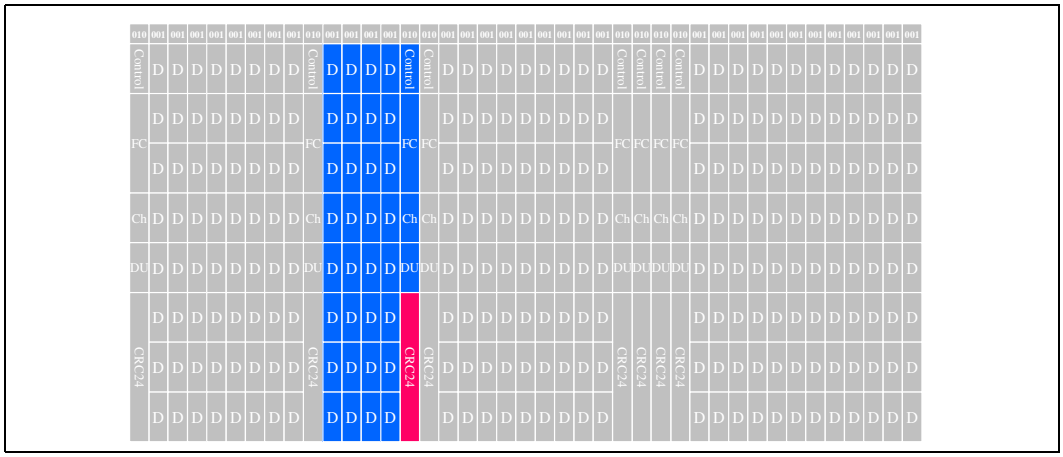


3.5 Data Integrity

Errors that might be introduced because of the underlying serializer/deserializer (SerDes) error rates must be detected to prevent corrupted packets from being passed by the system.

A CRC24 polynomial was chosen for Interlaken to protect each packet burst. The chosen polynomial can detect all single, double, triple, quadruple and all odd errors for Interlaken bursts up to 256 bytes. The CRC24 can also detect all burst errors up to 24 bits in length. Figure 10 shows the CRC24 coverage for a single burst (note that a burst will be striped across many lanes).

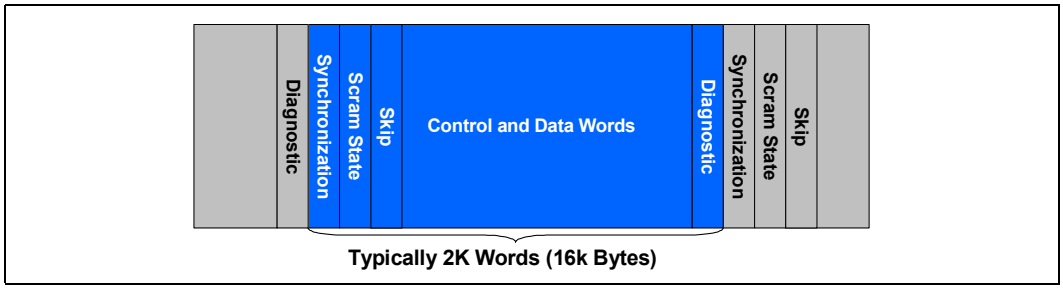
Figure 10 Burst CRC24 Coverage



3.6 Metaframes

As the control and data words are striped across the available serial lanes, each lane encapsulates those words into its own “metaframe.” A metaframe includes a synchronization word, scrambler-state word, skip words, and a diagnostic word as shown in Figure 11.

Figure 11 Metaframe



3.7 Synchronization Word for Lane Alignment

Data is striped eight bytes at a time across all of the lanes within an Interlaken interface. To align the data on the receive side of an interface, a synchronization word is sent on all lanes at the same time. The synchronization word, which is part of the metaframe, is a unique, unscrambled pattern that allows the receiver to find it and then deskew all lanes. This marks a common alignment point for all lanes that allows the receiver to deskew the lanes. How often the metaframe synchronization word is inserted is programmable.

3.8 Scrambling

Interlaken uses a scrambler to provide sufficient clock transitions for the receiver to recover the transmitted clock. A set/reset scrambler was chosen instead of a self-synchronous scrambler to prevent error multiplication at the receiver. The combination of error multiplication and the striping of data across many SerDes lanes would make it difficult to ensure that errored packets could be adequately detected.

With a set/reset scrambler, errors are not multiplied on the receiver side and therefore are more likely to be detected. Since Interlaken uses a set/reset scrambler, there must be a methodology to synchronize the receiver to the scrambler-state. As part of the metaframe's scrambler-state word, the scrambler state is fed forward to the receiver. The receiver uses the recovered scrambler-state to synchronize its scrambler and then de-scramble the data stream.

3.9 Skip Words

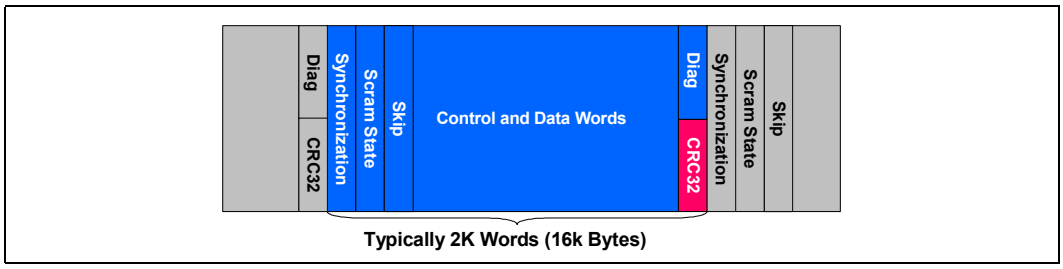
In an Interlaken repeater, the transmit and receive interfaces can run at slightly different speeds. To accommodate this, the metaframe includes one or more skip words. If the transmit interface is running slightly slower than the receive interface, these skip words may be deleted. Alternatively, if the transmit interface is running slightly faster than the receive interface, extra skip words may be added to the metaframe. This allows Interlaken to compensate for clock differences within a system.

3.10 Debugging & Diagnostics

Having an interface composed of many high-speed SerDes links can present a number of challenges both during the initial startup and when debugging a faulty interface. It is difficult if not impossible, to probe the high-speed SerDes in order to debug errors. Therefore it is critical that an interface based on SerDes be designed with integral debugging capabilities.

Interlaken has built-in test pattern and pseudo-random bit sequence (PRBS) capability on each SerDes lane to facilitate per lane testing and debugging. In addition, the protocol has a per-lane CRC32 calculated over the data contained within a metaframe. This allows errors to be isolated to an individual SerDes lane. The Interlaken protocol is flexible enough to allow for a persistently bad lane to be removed from the bundle. This is all in addition to the burst-level CRC24 (which protects data striped across all lanes). Figure 12 shows the CRC32 coverage for a metaframe (note that this is on a single lane).

Figure 12 Diagnostic CRC32 Coverage



3.11 AC Coupling

High-speed AC-coupled SerDes interfaces pose a number of electrical challenges. Among the challenges is maintaining a DC balance on the line so that the receiver can correctly decode the high-speed stream. Some codes such as 8B/10B maintain DC balance over a very short duration (several 10B symbols). Other codes such as 64B/66B maintain DC balance only statistically by scrambling the data. If a code is not balanced then at the receiver there will be an offset (called baseline wander). This voltage offset, depending on the link budget, can cause errors on the line. Simulations show that with a code like 64B/66B there can be a DC imbalance of thousands of bits. For some links the DC offset created by this is not a problem, but for more demanding links such an offset might not be tolerable.

Since Interlaken is designed for many different applications, an extra inversion bit in the code (hence the 64B/67B) has been added to the protocol to better control DC balance. This extra inversion bit allows the protocol to maintain DC balance within a disparity of plus or minus 65 bits.

3.12 Intellectual Property Cores

Critical to Interlaken's success as an open industry standard is the availability of third party intellectual property (IP) cores that can be easily integrated into customer designs. Compared with System Packet Interface Level 4, Phase 2 (SPI4.2), Interlaken IP cores will be easier to integrate, easier to prototype, and easier to extend to future products.

Integration of third party IP cores is eased by several features of the Interlaken protocol. A common issue caused by the SPI4.2 protocol is a short-term overwhelming of receiver bandwidth caused by short end-of-packet fragments and wide internal data paths. Interlaken avoids this issue by defining a "burst short" parameter and a "scheduling enhancement" that together ensure a minimum time between control words without loss of bandwidth efficiency. Interlaken also introduces a rate-matching function that can reduce the amount of buffering needed in devices that bridge a high-bandwidth Interlaken interface to lower-speed interfaces or applications.

Also key to making Interlaken IP easy to integrate is the standardization of recommended error detection, statistics counters, and latency parameters. Finally, by cleanly separating the protocol IP from the SerDes blocks, it becomes possible to migrate the Interlaken design to a new application-specific integrated circuit (ASIC) technology with an evaluation of the available SerDes technology, rather than a redesign of the IP core.

Interlaken can be built on top of high-speed SerDes from either ASIC vendors or FPGA vendors. This enables field programmable gate array (FPGA) prototyping using the same Interlaken IP core that the eventual ASIC product will employ. Also, systems that are built using both FPGAs and ASICs will be able to use the same IP core in all their devices, improving the reuse of both the IP and its associated firmware.

4.0 Case Study

A common design for current 25 Gbps products uses two SPI4.2 interfaces running in parallel. The primary disadvantage of this approach is that it uses more than 150 I/O pins and board traces for a bi-directional interface. Using a similar chip area and only 16 I/O pins, an Interlaken interface can provide the 25 Gbps bandwidth with four bi-directional SerDes lanes running at 6.25 Gbps. At even higher bandwidths, Interlaken has the advantage in both area and pin count, making it the obvious choice for new product designs.

The scalability of the Interlaken protocol is well matched to current CMOS technology. Some of the logic is associated with per-SerDes lane functionality. This includes the 64/67 code, metaframe creation, receiver synchronization, and of course the SerDes itself. The logic for one lane can run parallel to, and independent of, the other lanes. Therefore, this section of logic can run at roughly the same clock speed regardless of whether it is for a four-lane 25 Gbps interface or a 20-lane 125 Gbps interface. A 32-bit data path through this portion of the design can run less than 200 MHz and still support a 125 Gbps design bandwidth. This low clock rate in turn yields benefits in easier timing closure and lower power.

The logic blocks that insert control words to create Interlaken bursts, calculate the burst CRC24, and stripe the data across the available SerDes lanes do need to scale up for higher-bandwidth products. Some designers may choose to run a narrower internal pipeline at a high clock rate while others may run a fat pipeline at a slower clock rate. For example, a 25 Gbps interface may calculate a single 64-bit Interlaken word every clock cycle running at 400 MHz, or it may calculate two words every clock cycle at 200 MHz. At 125 Gbps choices for this section of logic include four words at 500 MHz and six words at 333 Mhz. Current CMOS technology appears capable of implementing any of these options.

The scalability and features of Interlaken make it an attractive interconnect protocol for both current and future chip designs. These characteristics, however, do increase the difficulty of the verification effort. Fortunately, new verification methodologies can be used to ease the burden. Object-oriented verification languages such as System Verilog* make it easier to handle complex data types such as packets and metaframes. A constrained random verification methodology is useful for creating a wide variety of input traffic, as well as verifying the large number of configuration parameters inherent in the design. Assertions can be used in many areas, from checking requirements on the Interlaken interface such as the running disparity limit, to checking assumptions on the user interface to the Interlaken IP core, as well as internal logic structures in the register transfer language (RTL). Similar to assertions, coverage properties can be used to ensure that potential corner cases and other interesting scenarios can be covered by a verification suite.

5.0 Conclusion

When compared to available interconnect protocols, Interlaken offers many advantages in scalability, reduced pin count, and data integrity. Its channelization, flow control, and burst interleaving features make it appropriate for a wide variety of applications. Finally, the availability of a third party IP core minimizes the cost of adopting the new technology and makes Interlaken the obvious choice for next-generation communications equipment.

6.0 Contact Information

More information and the Interlaken specification can be accessed from:

Cisco Systems, Inc.

c/o Interlaken Inquiries,

Cisco Systems, Inc.

170 Tasman Drive

San Jose, California 95134

Phone: 408 - 526 - 4000

E-Mail: Interlaken@external.cisco.com

Web Site: www.cisco.com

Cortina Systems, Inc.

c/o Interlaken Inquiries,

Cortina Systems, Inc.

840 West California Avenue, Suite 100

Sunnyvale, California 94086

Phone: 408 - 481 - 2300

E-Mail: Interlaken@cortina-systems.com

Web Site: www.cortina-systems.com

Silicon Logic Engineering, Inc.

c/o Interlaken Inquiries,

Silicon Logic Engineering, Inc.

7 South Dewey Street

Eau Claire, WI 54701

Phone: 715 - 830 - 1200

E-Mail: Interlaken@siliconlogic.com

Web Site: www.siliconlogic.com