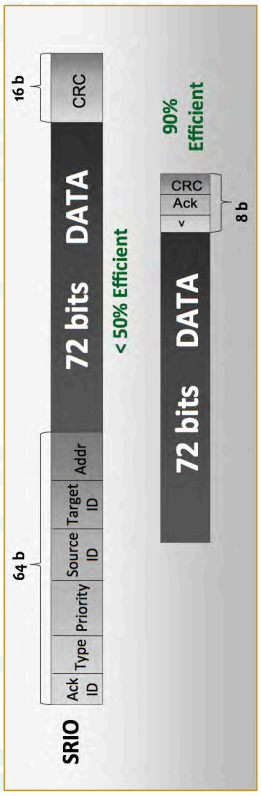
THE GIGACHIP INTERFACE:



Efficient Board Level Chip-to-Chip Look-Aside Communication

by Michael Miller
VP, Technology Innovation & Systems Applications
MoSys, Inc.

Efficient Board Level Chip-to-Chip Look-Aside Communication

INTRODUCTION:

Serial communication has progressed from its earliest implementations in telegraph (Morse Code), undersea cables, and microwave to Ethernet connected boxes on a network. New serial standards such as PCI Express, SRIO, XAUI, FDDI (for disk drives), Interlaken Look-Aside and SFI-S have facilitated widespread adoption on backplanes and for packet transactions. However, these protocols are only efficient with relatively large transmissions. When an ASIC or FPGA performs load/store transactions, the transmission to and from memory occurs in small synchronous transfers of data, such as 72 bits (64 bits + 8 bits of EDC). Using existing protocols results in poor utilization of the native bandwidth of the serial links. Solution: A new protocol optimized for short synchronous transactions that makes use of existing SerDes electrical standards.

The problem: Today's serial interfaces, such as those noted above, are optimized for large data packets. The protocol overhead for a transmission of 4 – 256 Bytes of data totals 64 bits for SRIO, either 96 or 128 bits for PCIe, plus 16 bits for CRC. However, for chip-to-chip communication in the look-aside direction, the same protocol overhead applies, but because the data size is 72 bits, even the most efficient protocol (SRIO) transmits the packet at less than 50 percent efficiency. Such inefficiency increases costs in the form of extra I/O, bigger packages, additional traces, increased board real estate and higher power requirements. To overcome this hurdle, MoSys developed the open standard GigaChip Interface (GCI), optimized to transmit 72 bit data words with 90 per cent efficiency.

CHANNELIZED PACKETS VS. DATA WORDS

Data transmissions from backplanes to or from ASIC or FPGA devices typically use "packet transfer protocols." Such transfers exhibit the following data transmission characteristics:

- Date rates of n x 1/10/100 Gbps
- Variable length packets from 64B to 1.5KB
- Asynchronous transfer mode
- Reach of 8 – 30 in.
- ASIC/FPGA to and from Network PHY or switch over back plane

The most common transmission protocols serve one of two categories: Packets and Load/Store transactions. For packets, SFI-S and Interlaken Look Aside represent the two most common protocols; for load/store transactions, SRIO and PCI Express are the most widely used. Packets are transported through multiple devices, one of which is often a switching device. Due to multiple end points and potential congestion in switches, packets face the reasonable possibility that they will be dropped. To address this problem, these protocols require more complex error checking and flow control mechanisms with multiple fields to communicate individual packet ID, priority levels, packet types, and end-to-end port ID fields.

By contrast, data word transmissions that take place between ASIC/FPGA devices to (and from) a microprocessor, ASSP, and memory rely on Look-Aside Data Word protocols and exhibit these transfer characteristics:

- Date is in fixed length frames of 32b, 64b or 72b
- Rate at greater than 1 billion accesses per second
- Synchronous transfer mode
- Reach of less than 8 in.

Data word transfers occur solely between two devices that are the end points, which eliminates switch related issues. Packet integrity is the only concern and is typically low (BER 10^{-15}) because Signal Integrity related issues are the source for information loss. Data loss can be managed by design best practices and error check protocols.

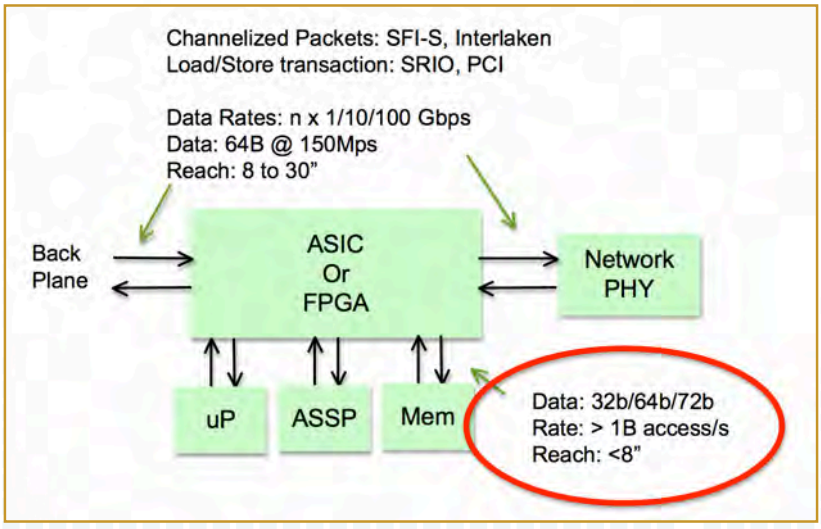


Fig. 1: Device-to-Device Data Word Transmission vs. Packet Transmission

Until now, no commonly available protocol using SerDes has been optimized for synchronous fixed length transfers in the look-aside path. Therefore, designers have utilized packet-oriented protocols over SerDes resulting in higher overhead in resources and latency. The GCI Protocol was developed specifically to streamline device-to-device data transmissions. In operation, the GCI overcomes the inefficiencies of existing protocols for the look-aside application. See Fig. 1 for a graphic representation of data word transmissions.

STRUCTURE OF THE GIGACHIP INTERFACE PROTOCOL

The GCI protocol provides for the reliable transport of 80 bit data words. Including CRC and positive Acknowledgement (Ack), GCI transports frames with 90 percent efficiency due to only 8 bits of overhead. See Fig. 2.

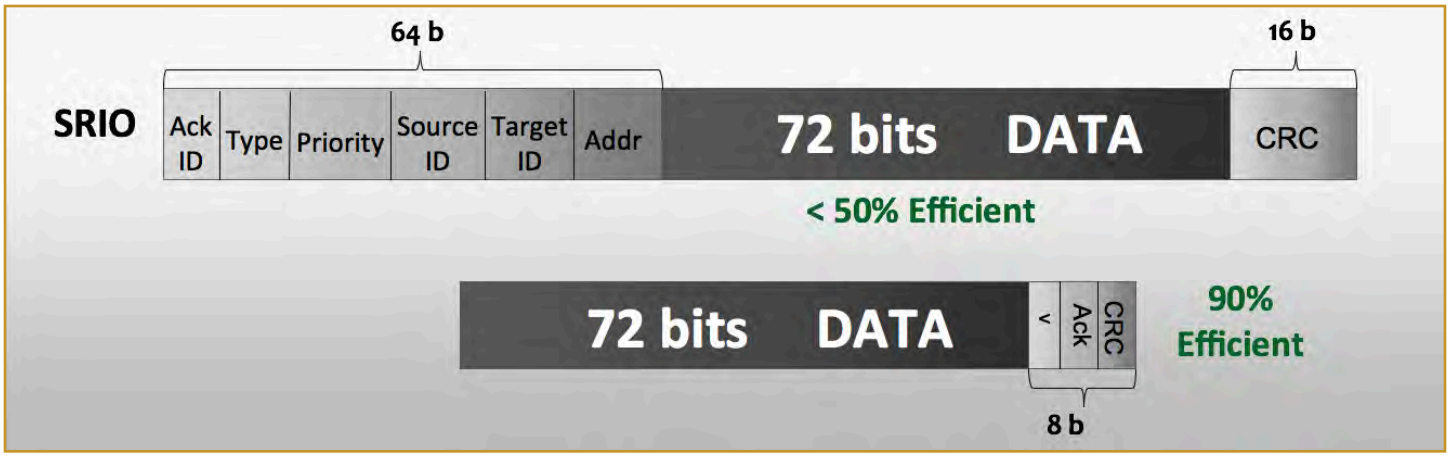


Fig. 2: Graphic representation of the inherent efficiency in the GCI protocol

Data transmitted via the GCI protocol exhibits these transfer characteristics:

- 72 bits of payload defined by a higher level application
- 6 bits of CRC error handling
- 1 bit for positive Ack channel
- 1 bit to flag a data link message or data in the payload

When the GCI system discovers an error, the built-in error handling system backs up and retransmits the first frame after the last acknowledged good transmission. The process operates much like a Ferris Wheel: As each new frame is transmitted, the ASIC/FPGA retains it in a sequential memory transmission Replay Queue of n units determined during system design. When the n+1 unit is transmitted, the earliest transmission unit drops out of the Replay Queue, or analogously, gets off the Ferris Wheel. In short, GCI provides bits for CRC error handling and the mechanism to actually perform error handling.

The protocol utilizes the Physical Coding Sublayer (PCS) in the FPGA/ASIC to ensure lane deskew as part of link initialization. At the PCS, the protocol initiates PRBS scrambling to minimize DC wander and works over various serial electrical links. Nominally, it uses the Common Electrical Interface 11+ G for electrical coding.

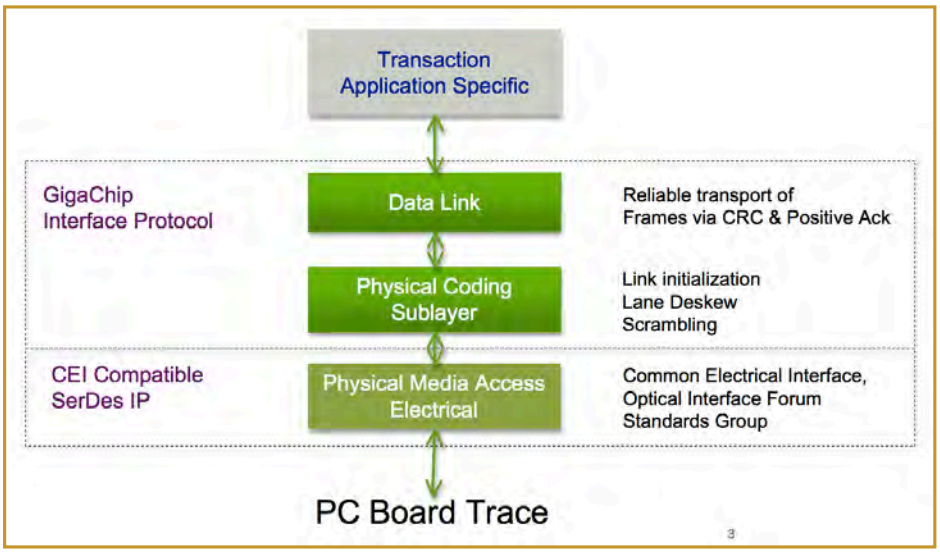


Fig. 3: Interaction between GCI and FPGA/ASIC

To summarize the interaction between the GCI and the FPGA/ASIC, designers utilize the following interface layers in the FPGA/ASIC to implement GCI; the data link to ensure reliable transport, the PCS to assure lane alignment, and the Physical Media Access Electrical Layer in the form of CEI compatible SerDes IP. See Fig. 3.

COMPARISON TO OTHER CANDIDATE CHIP-TO-CHIP PROTOCOLS

SRIO and PCIe share similar characteristics in that both are optimized for large packet (as vs. data word) transmissions. The headers vary in size: SRIO requires 64b while PCIe requires 96 (Generation One and Two) and 128b (Generation Three). Both require 16b for CRC. SRIO and earlier generations of PCIe required 8b/10b and PCIe Generation Three now requires 128b/130b for various PCS encodings. SRIO, the lower overhead of the two, exhibits less than 50% efficiency when transmitting 72 bits of data. Both protocols include a reliable transport mechanism to recover from bit errors. See Fig. 4.

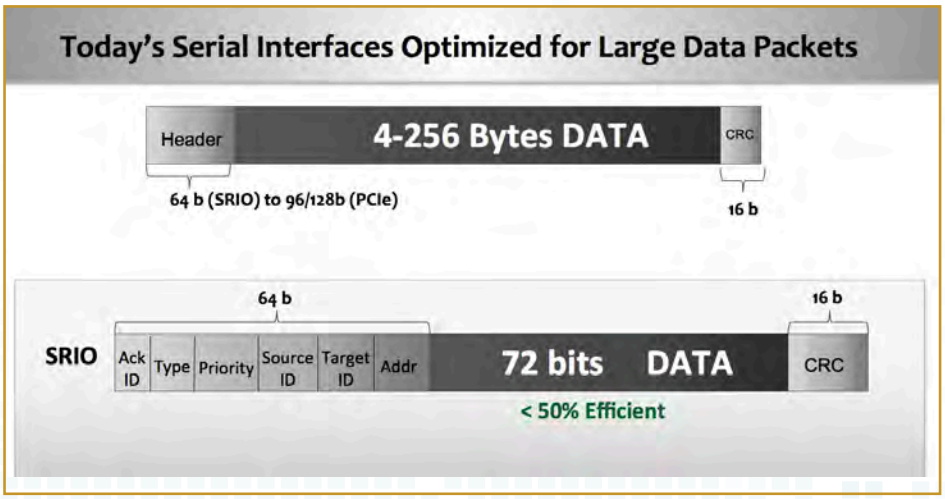


Fig. 4: Efficiency comparison between optimized SRIO and PCIe at 4-256 Bytes of data in a packet (top) to SRIO transmission of 72b data words (bottom)

The XAUI-like (pronounced “zowie”) class of protocols exhibits similar characteristics to other packet-oriented protocols. It too relies on 8b/10b encoding for DC balance. At the same time, XAUI-like protocols use K characters to communicate “start-of-frame” and “end-of-frame”. To better align packets in buffers and the interface, a one-time character pad creates even byte count payloads to keep data on 16b/20b boundaries. Unfortunately, for 72b messages, XAUI-like protocols lose 36% percent efficiency just in the encapsulation and another 20% in encoding (over 50% in total). Perhaps most crucially, XAUI provides no means to verify reliable transport – meaning no error checking mechanism. See Fig. 5.

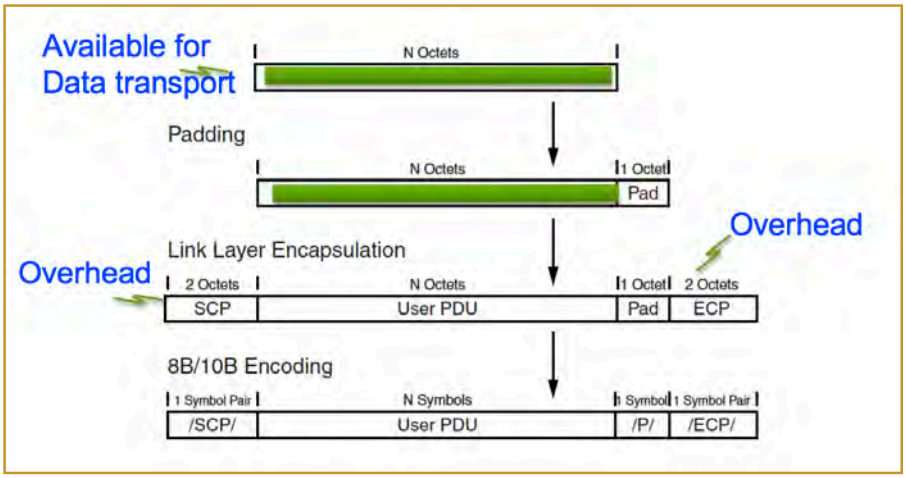


Fig. 5: Transmission Procedures for XAUI and XAUI-Like Encapsulation

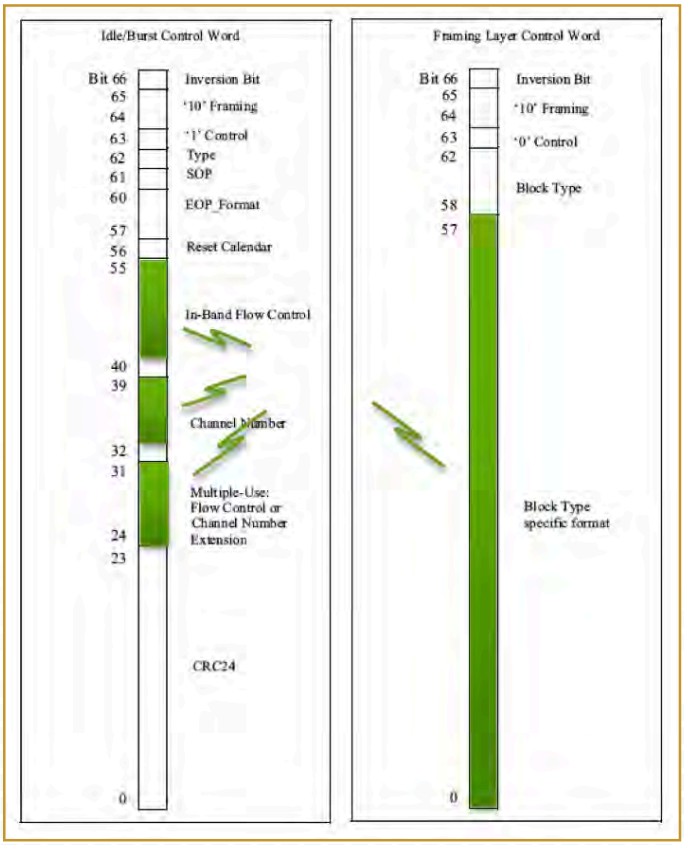


Fig. 6: The combined white space in the graphic above represents overhead in the Interlaken Look-Aside

Another popular protocol, the Interlaken Look-Aside uses 64b/67b encoding, which reduces the encoding overhead of 8b/10b. However the 64b/67b “framing” adds to latency which is not a problem in the packet forwarding case, but can be critical in the look-aside path. As demonstrated in Fig. 5 at right, the Interlaken protocol is approximately 40% efficient when transferring 72b. The relevant overhead factors include channelizing packets, 24b CRC, packet delineation, and multiple block format handling. Although the protocol includes 24 bits allocated for error checking, in use, it has no explicit error handling mechanism.

SERIAL CONNECTIONS – THE MOST LIKELY FAILURES

Random bit errors are the most likely failures that can occur in serial connections. Residual PLL jitter and/or quantum effects cause these errors and are guaranteed to occur less than 10^{-12} and practically, only occur less than 10^{-15} transmissions. By using a short CRC per operation, system designers can implement ongoing mitigation that will result in less than one (1) failure in time (FIT).

Single bit errors are typically the main failure mode. However, multiple back-to-back bit errors are possible and result from one of three causes: a poor power supply that produces ripples, droops, transients, aging, etc.; poor routing; and poor connections. System designers can identify these causes during system design, validation, and test processes and remove or remediate them accordingly. By implementing periodic in-service checks utilizing the housekeeping functions of the ASIC/FPGA, system designers can provide ongoing mitigation in several ways: At the system level, monitor long-term 32-bit CRC per SerDes lane and/or note multiple occurrences of "random bit errors" in a "short" period of time. If either of the foregoing occur, system software can trigger an alarm and the board is removed from service.

CRC ERROR HANDLING IN GCI

The PCI Express standard utilizes a mesosynchronous clocking methodology, a return to the original idea of the source synchronous method – eliminating the requirement for constant frequency adjustment. Both source and destination have the same clock source and must have a connection wire between them. Of course, this approach is not very practical for macro applications like an undersea cable, but it is practical at the board and device level.

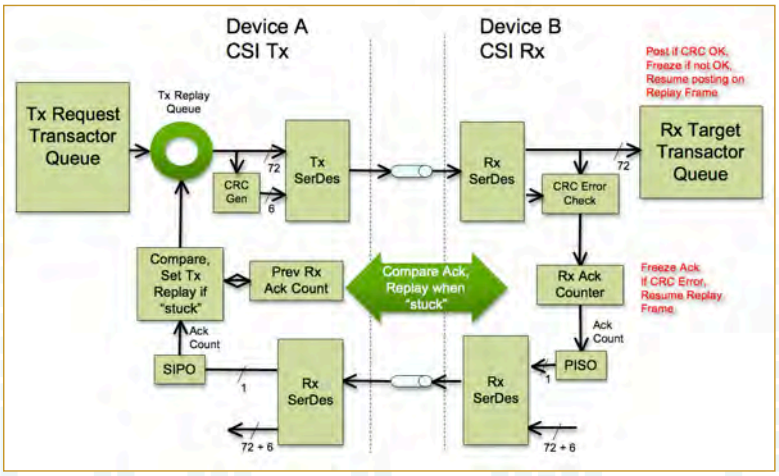


Fig. 7: CRC Error Handling with Positive Ack in GCI

SERIAL LATENCY

To handle bit errors, the GCI protocol includes a complete error handling mechanism using 6 bit CRC to check the frames with Positive Acknowledgement (Ack) (See Fig. 7 at left). The chart at left graphically presents the error handling mechanism in the GCI and is self-explanatory. The key points to note are:

- The large double-headed green arrow in the center of the chart above highlights the process of comparing the Ack count on the receiving side and the previously received Ack count on the transmission side.
- If “stuck”, meaning an error occurred, the system sets the transmission replay queue to the last “good” transmission and retransmits until the error clears.
- If no error occurs, the system transmits the Positive Ack and the receiving device posts the frame.

SUMMARY OF INNOVATIONS & RESULTS USING THE GCI PROTOCOL

The GCI protocol operates as a standard electrical layer compatible with OIF CEI 11 SR. The protocol supports high bandwidth density with differential serial links. Current design implementations running eight (8) links, each at 10 Gbps, achieve over 1 Billion 72b transfers per second with 1 ns latency.

Because the GCI protocol is oriented to transmit data words, it exhibits a consistently high transmission efficiency of 90% with low latency in fixed 80b frames. GCI provides system designers with a highly scalable serial protocol suitable for 1, 2, 4, 8, and 16 link configurations, which scales with the OIF roadmap.

Besides extraordinary speed and efficiency performance, the GCI features built-in high reliability with 6b CRC and 1 FIT. The protocol includes CRC protection for each frame transmitted and a built-in frame replay mechanism. As a result, less than 1 undetected error occurs in every 10^{25} frames transferred. This translates to less than one (1) FIT or one (1) undetected error in one (1) billion hours (150,000 years) of operation.

By comparison to other protocols, the GCI protocol is lightweight. It requires only 20,000 ASIC gates vs. Interlaken, which requires 100,000 or more ASIC gates. In other words, less than 20% of the gates needed for Interlaken.

Written with the assistance of Lee Stein, Stein & Associates, Inc.