

Technical Disclosure Commons

Defensive Publications Series

January 2021

ENHANCED ERROR DETECTION AND CORRECTION IN HIGH SPEED CHIP TO CHIP CONNECTIONS

Todd Lawson

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Lawson, Todd, "ENHANCED ERROR DETECTION AND CORRECTION IN HIGH SPEED CHIP TO CHIP CONNECTIONS", Technical Disclosure Commons, (January 17, 2021)
https://www.tdcommons.org/dpubs_series/3974



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

ENHANCED ERROR DETECTION AND CORRECTION IN HIGH SPEED CHIP TO CHIP CONNECTIONS

AUTHOR:
Todd Lawson

ABSTRACT

Chip-to-chip (CTC) connections often involve serializing parallel data. This serialized data is transmitted at a higher frequency than parallel data and is usually subject to a high bit error rate, compared to the error rate associated with normal flop-to-flop data movement within a single die. Error-Correcting Code (ECC) and Forward Error Correction (FEC) are often used as a detection and correction methods. However, each has limitations and neither error correction method allows for correction of a 100% bad link, due to a manufacturing defect (stuck at) or other defect. Techniques presented herein provide for interleaving multiple ECC payload and checksums across lanes of a 2.5D parallel CTC application, which allows for correction of any number of errors on a bad link, up to correcting all of the data on a completely bad link. Techniques presented herein may allow for a more robust communication channel that can potentially increase the yield of Application-Specific Integrated Circuit (ASIC) manufacturing processes, thus reducing cost.

DETAILED DESCRIPTION

Correcting the errors inherent in data serialization can be challenging and existing methods utilizing ECC or other FEC methods can present significant challenges. For example, ECC is often used as a high speed, high overhead, low latency method of correcting single bit errors and detecting multiple bit errors. However, serialization of the data removes many benefits of ECC-type correction methods, as multiple errors on a single serialized lane are the norm. Multiple errors on a single lane remove the ability of ECC to do anything other than detect errors.

Further, FEC methods are low overhead and high latency techniques that are typically used on a small number of Serializer/Deserializer (SERDES) -based interfaces for which latency is less of a concern. Examples of FEC methods may include Reed

Solomon FEC (RS-FEC), typically used in Ethernet applications for high-speed SERDES links. However, RS-FEC does not lend itself to use cases involving 2.5D packaging with CTC channels utilizing pico-SERDES and High Bandwidth Memory (HBM) chip technology. For example, RS-FEC is a store and forward method, resulting in high latency and logic complexity. In contrast, ECC provides single cycle error correction with no store and forward.

This proposal provides for the ability to use interleaving to eliminate the effect of serialization on the correction of data in 2.5D applications involving a large number of CTC interconnects.

In current designs involving error detection and correction, a payload that is to be sent to another die is broken in to manageable chunks of size 'N' bits, passed through an ECC generator for that size, and the resulting ECC checksum is concatenated with the original payload data. The payload and ECC checksums from multiple ECC generators are joined together in a single large bus and the data is sent across the CTC interface. Upon obtaining the data, the far side runs the payload and checksum through an ECC checker and any single bit errors are corrected in each ECC payload.

One problem with these operations is the data sent to a CTC channel is reduced from N bits at a clock rate 'Z' to 'N/M' bits at a clock rate of Z*M. For instance, consider an example CTC interconnect that has an 8x speed-up over the core clock rate and a core side payload capacity of bits (1600b). With this type of interconnect between dies, there will be 200 (1600/8) connections between the dies, each running at an 8x the clock rate of the core side of the interconnect in which bits 0-7 are serialized on die side lane 0, bits 15:8 are serialized down lane 1, etc.

With an example ECC payload width of 151b and a 9b checksum, bits 0-7 of the payload are serialized down link 0, bits 15:8 are serialized on link 1, etc. up to payload bit 150 and checksum bits 8:0 are serialized on link 9. If there is a problem on link 0, for instance, bits 0-7 of the ECC payload may be corrupted.

In accordance with techniques of this proposal, by interleaving the ECC bits in a pattern than allows only 1b per ECC group per C2C link, the issue of errors on the bad link causing uncorrectable ECC errors is prevented. This improvement will allow the C2C

interface to operate even if one of the lanes between the dies is completely bad or has a very high bit error rate (BER).

Figures 1–6, below, illustrate various example details associated with the interleaving techniques of this proposal as compared to current implementations. For example, Figure 1, below, illustrates a current implementation involving clean transfers in which ECC is generated and added to an ASIC data path payload. The ECC and data payload are sent across CTC links in an N to 1 reduction, with the bits in each lane serialized.

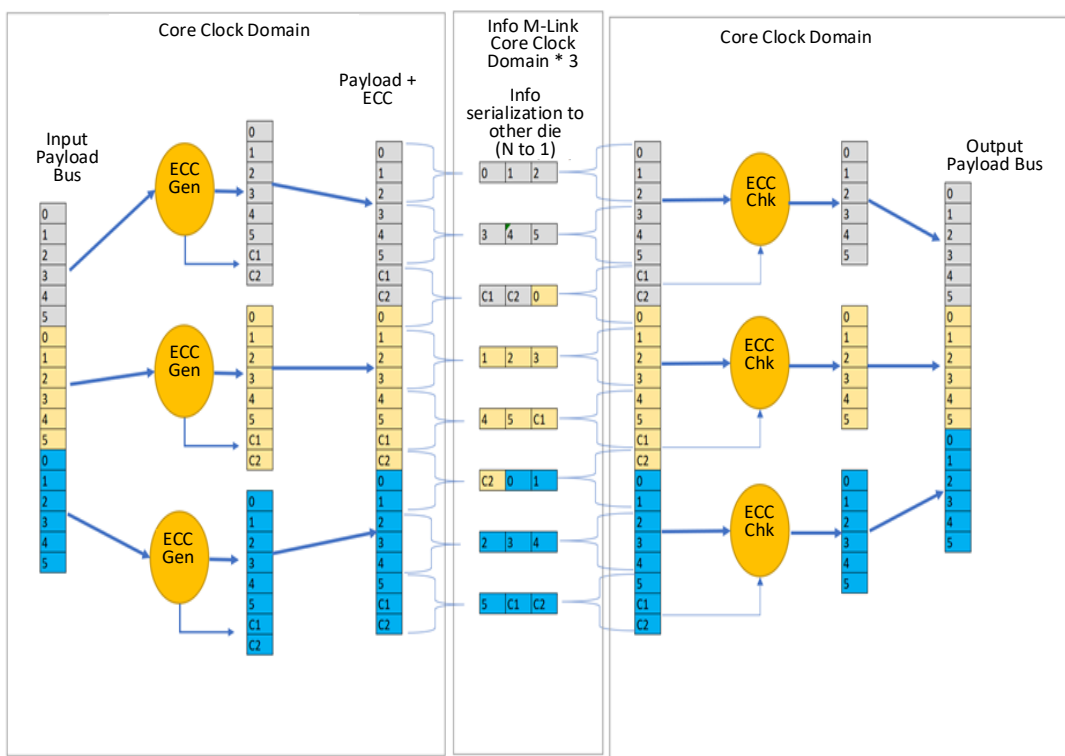


Figure 1: Current Example Implementation

Figure 2, below, illustrates an example in which a single-bit error occurs and is corrected as expected for the current implementation.

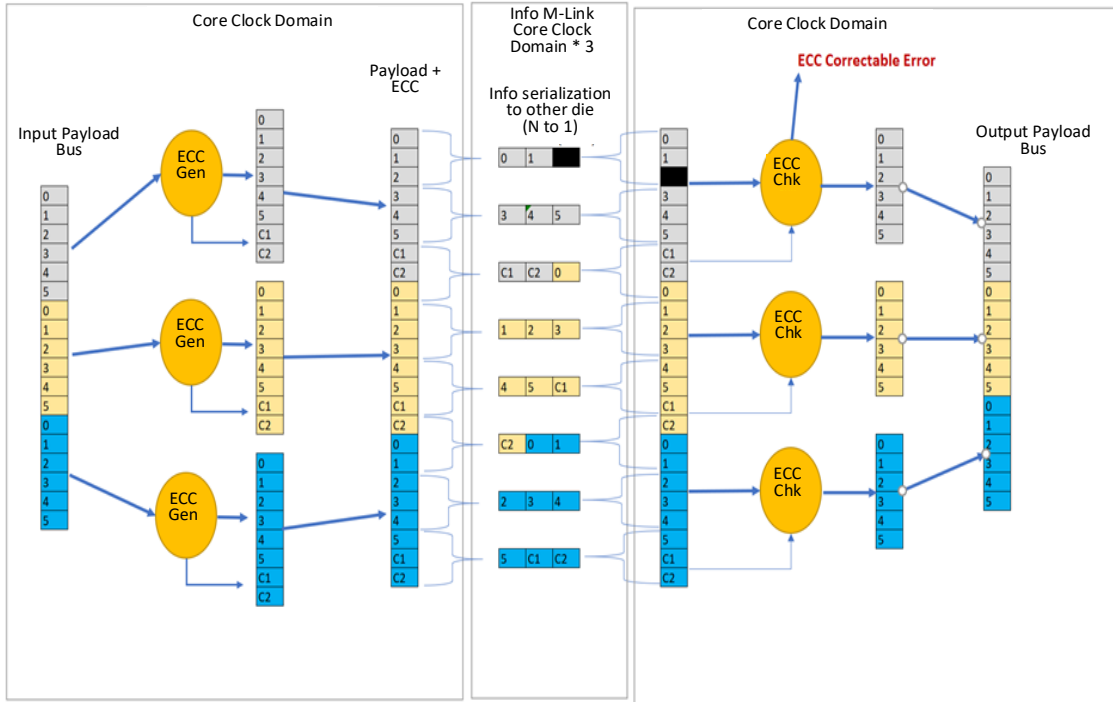


Figure 2: Single-bit Error Correction

In contrast, Figure 3, below, illustrates an example in which an entire CTC lane is corrupted (e.g., due to a bad redistribution layer (RDL) trace or other multi-bit error) that results in an uncorrectable error and a corrupted payload.

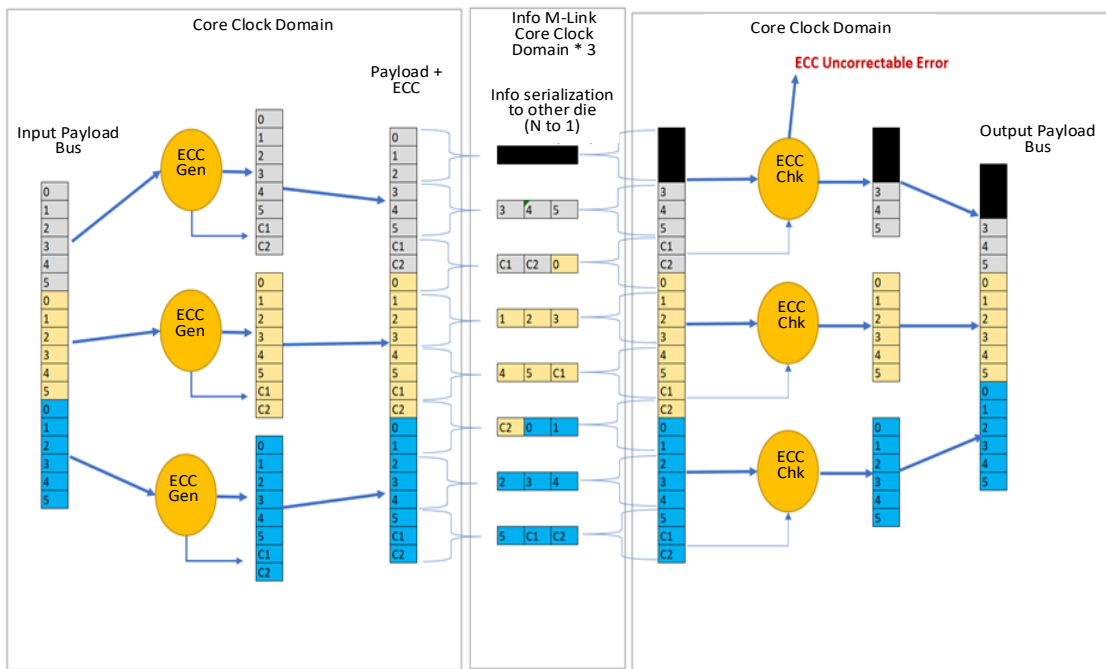


Figure 3: Example Multi-bit Error that is Uncorrectable

Figure 4, below, illustrates example details associated with the techniques of this proposal in which interleaving of the ECC is provided with clean transfers.

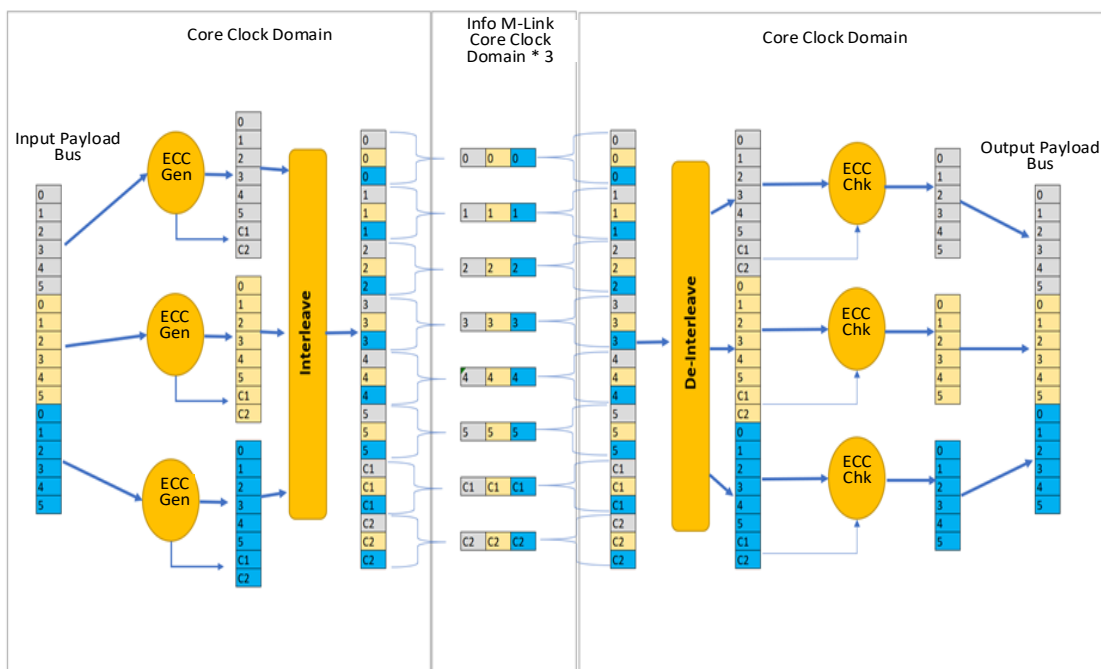


Figure 4: ECC Interleaving with Clean Transfers

Next, consider ECC interleaving for a use-case involving a single-bit correctable error, as shown in Figure 5, and for a multi-bit error use-case in which all bits on the CTC link are bad, can be corrected, and the lane can be identified, as shown in Figure 6.

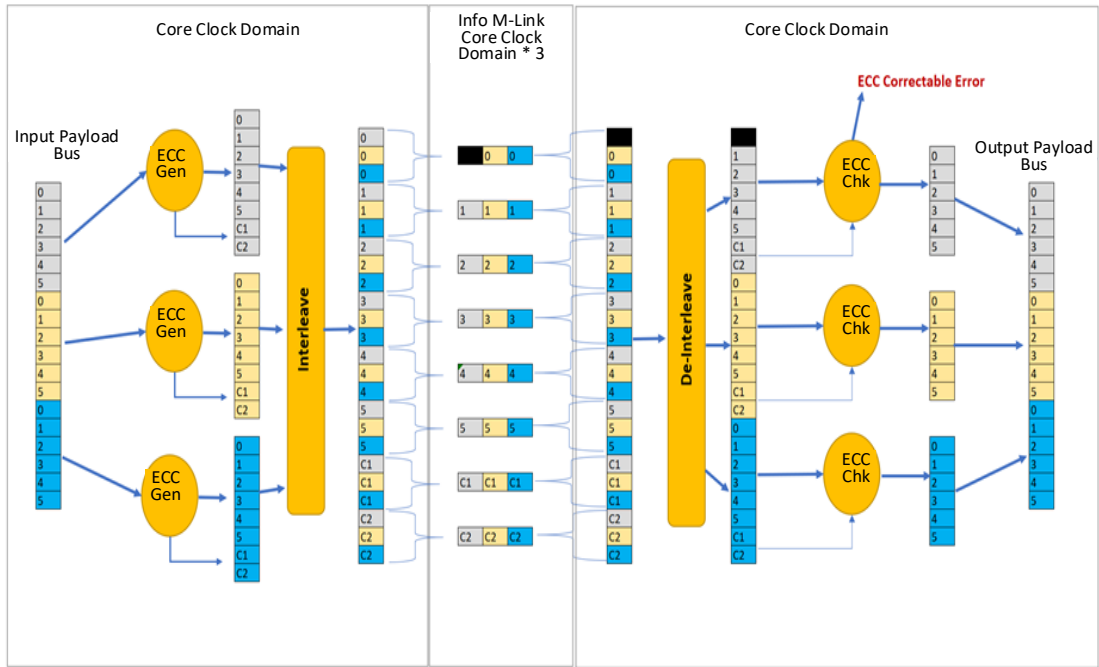


Figure 5: ECC Interleaving with a Single-Bit Correctable Error

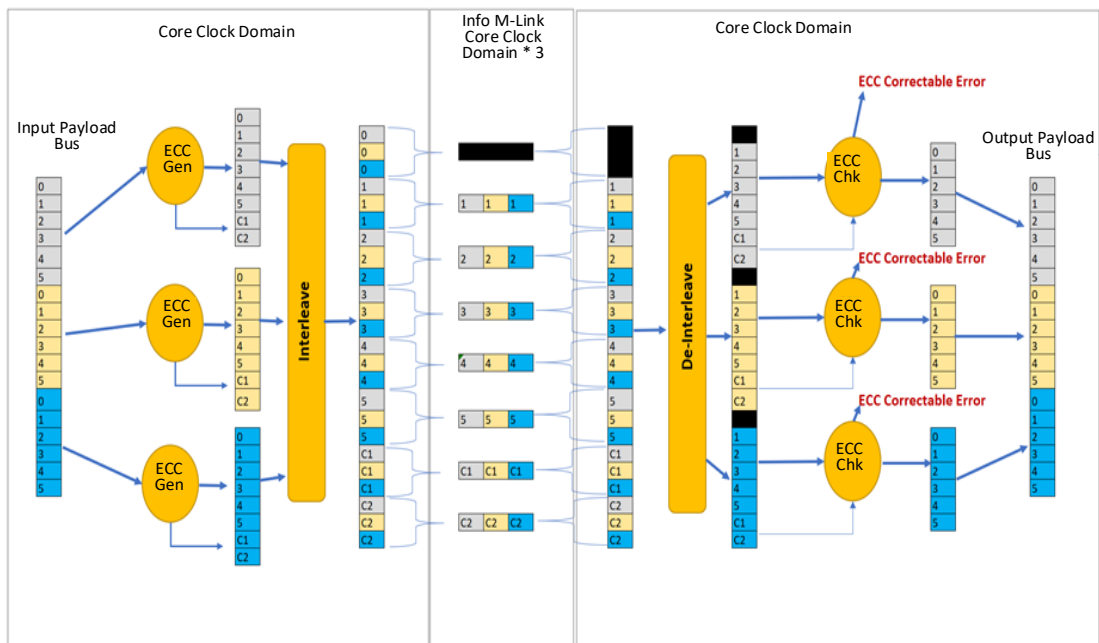


Figure 6: ECC Interleaving with Multi-Bit Correctable Errors

The ECC interleaving described herein may be useful in high-BER conditions involving a CTC interconnect where a non-zero BER is possible. For example, the ECC interleaving can be used to prevent a bad CTC interconnect from rendering a CTC channel unusable (e.g., due to manufacturing defects or any other cause that reduces a normally negligible BER rate to one that is unusable). Accordingly, techniques herein may allow for correction of any number of errors on a bad link, potentially providing for the ability to correct all data on a completely bad link.