ERRATA – Identified changes to the current ODI specification revisions
April 13, 2020

ODI-1

There have been some implementations where it appeared that the start-up of an ODI link is sequence dependent. This should not be the case if the Consumer state diagram is designed to match that required in Figure 4-11. For that reason, we will add an OBSERVATION below Figure 4-11 that states:

OBSERVATION: The sequencing of start-up between the Producer and the Consumer is not defined. Therefore, the Consumer must be tolerant of timing of a valid Producer output which may be before or after the Consumer enters the Sync Links state. It is possible that the Producer could temporarily source invalid outputs as it is starting up, and the Consumer needs to be tolerant of this situation. Also, it is possible that the link could be interrupted during operation. Again, the Consumer should detect this and return to the Sync Links state.

ODI-2

In ODI-2 Rev. 3.0 the Packet Count behavior of a VRT Packet Header was described to act as a modulo-16 counter of all VRT packets. This is inconsistent with VITA 49.2, where the Packet Count is defined as a separate modulo-16 counter for each packet type and Stream ID. This will be changed in Rev. 3.1. The following paragraphs show how these changes will be incorporated into the next release:

Page 8 states: (in Revision history)

Packet Count will behave as a modulo-16 counter of all VRT packets, and not specific to the count of a particular VRT packet type.

Revision 3.1 in Revision History will say:

Packet Count will behave as a modulo-16 counter of all VRT packets with the same Stream Identifier and packet type.

Page 25 states:

Packet Count (16-19) is a modulo-16 counter that counts the number of VRT packets sent. Bit 16 is the LSB. Packet Count will increment for each packet sent.

Change to:

Packet Count (16-19) is a modulo-16 counter that counts the number of VRT packets sent having the same Stream Identifier and the same packet type. Bit 16 is the LSB. Packet Count will increment for each packet sent, rolling over from “1111” to “0000” as the count increments through the 4-bit value. This is identical behavior for all packet types.

Page 37 states:

Packet Count (16-19) is a modulo-16 counter that counts the number of VRT packets sent. Bit 16 is the LSB. Packet Count will increment for each packet sent. This is identical behavior for all packets types.

Change to:

Packet Count (16-19) is a modulo-16 counter that counts the number of VRT packets sent having the same Stream Identifier and the same packet type. Bit 16 is the LSB. Packet Count will increment for each packet sent, rolling over from “1111” to “0000” as the count increments through the 4-bit value. This is identical behavior for all packet types.

Page 45 states:

Packet Count (16-19) is a modulo-16 counter that counts the number of VRT packets sent. Bit 16 is the LSB. Packet Count will increment for each packet sent. This is identical behavior for all packets types.

Change to:

Packet Count (16-19) is a modulo-16 counter that counts the number of VRT packets sent having the same Stream Identifier and the same packet type. Bit 16 is the LSB. Packet Count will increment for each packet sent, rolling over from “1111” to “0000” as the count increments through the 4-bit value. This is identical behavior for all packet types.

Page 57 says:

RULE 4.2: ODI-2 producers that aggregate ports SHALL set the Packet Count field in the Header to 0 for the first packet sent transmitted on each port, and increment the Packet Count for each subsequent packet from that port.

Change to:

RULE 4.2: ODI-2 producers that aggregate ports SHALL set the Packet Count field in the Header to 0 for the first packet transmitted of the same Stream Identifier and the same packet type on each port, and increment the Packet Count for each subsequent packet with that same Stream Identifier and the same packet type from that port.

Page 58 says:

RULE 4.3: All ports being aggregated SHALL send the same Packet Count for each synchronized packet across all ports.

OBSERVATION 4.3: The above rule allows recovery from a line outage, perhaps caused by an electrostatic discharge event. Since Packet Count is a modulo-16 counter, it can be used to unambiguously align the beginning of a packet on an ODI port with the correct packet on another ODI port.

No changes needed.

Page 64 says:

Port aggregation is principally used to send Data Packets when higher bandwidth is needed. There may be Context and Command Packets coupled to the same packet stream. In this case, the same packet subtype, typically a Context Packet or a Control Packet, is sent across all ports. This forces all ports to have the same number of packets transmitted, including the same number of Context Packets and Command Packets on each port. The Packet Count field of the Header is incremented for each of these packet subtypes, as defined in single port applications. For this reason, all aligned Context Packets will have the same Packet Count, and all aligned Command Packets will have the same Packet Count as well. The same alignment process of aligning SOP, the packets, and Packet Count is used when aligning Context and/or Command Packets as with Data Packets.

RULE 4.9: During port aggregation if a producer sends a Context Packet or a Command Packet on one port, it SHALL send the same packet type on all ports, while matching the Packet Count field across the packet type.

OBSERVATION 4.12: The above rule forces all ports to have the same number of packets of each packet type.

No changes needed.

Page 64 also says:

RULE 4.10: During port aggregation if a producer sends a Context Packet or a Command Packet on one port, it SHALL send the same packet type on all ports, while matching the Packet Count field across the packet type.

Rule 4.10 is redundant and will be removed or replaced with null text.
