ODI-2: Transport Layer

Optical Data Interface

PROVISIONAL Revision 3.0

November 11, 2018

Important Information

The Optical Data Interface family of specifications is authored by the AXIe Consortium, and is abbreviated as ODI. The ODI specifications are open to all organizations, whether a member of the AXIe Consortium or not.

For more information about ODI, go to http://axiestandard.org/odispecifications.html

For more information about the AXIe Consortium, go to http://axiestandard.org/home.html or contact execdir@axiestandard.org.

Warranty

Trademarks

VITA is the registered trademark of VITA, a trade association for standard computing architecture.

Interlaken is the registered trademark of the Interlaken Alliance

PICMG and AdvancedTCA are registered trademarks of the PCI Industrial Computers Manufacturers Group.

PCI-SIG, PCI Express, and PCIe are registered trademarks of PCI-SIG.

LXI is a trademark of the LXI Consortium Inc.

PXI is a trademark of the PXI Systems Alliance.

Other product and company names listed are trademarks or trade names of their respective companies.

No investigation has been made of common-law trademark rights in any work.

Table of Contents

List of Figures

ODI-2 Transport Layer Specification

Revision History

This section is an overview of the revision history of the ODI-2 specification.

Revision Number	Date of Revision	Revision Notes
1.0	October 2, 2017	Initial Version (Preliminary). In slide format.
2.0	April 20, 2018	In slide format (Preliminary). Incorporated numerous changes due to the adoption of V49.2 in place of V49.0. The key mandatory change for data packets is that bit 25 of the Signal Data Packet header is changed from “0” to “1”. This indicates that it is no longer a V49.0 packet, but a V49.2 (or higher) packet. This change is also implemented for Context Packets and the newly defined Command Packets. Revision 2.0 has adopted the new V49.2 Command Packet type. Command Packets are useful for sending command information to an arbitrary waveform generator or other exciters, specifically metadata about the signal (e.g. bandwidth, IF offset frequency, etc.). This was previously performed using Context Packets. As in the past, Context Packets and Command Packets remain optional capability. Since Context and Command packets have very similar properties, ODI-2 allows either to control a consumer. However, if a consumer can execute Command or Context packets, it must execute Command

		packets as the mandatory method. Execution of Context Packets is allowed if the device is explicitly enabled to do so through the control plane. This may be a useful feature for an AWG playing back recorded sample data, as only Context Packets would have been recorded. Only one Command Packet subtype, the Control Packet, is expected to be used in ODI systems. ODI-2 has also changed the default Timestamp Fields in the Prologue for devices that do not implement Timestamps. TSI/TSF has changed from 11/11 to 11/01. This is due to 11/11 already being a legal V49.2 combination for other functionality. It should be noted that ODI-2 has grown in size due to the VITA addition of Command Packets and various additional Context Fields. ODI-2 has documented how ODI devices should implement these if the developer finds them to be necessary in their application. However, many of the packet subtypes are never expected to be used in ODI systems. As with the first revision, only the Signal Data Packet subtype is needed for a fully functioning ODI system. ODI-2 Rev.2 has added a number of rules for Port Aggregation. The default Stream IDs on aggregated ports now increment by 1024 instead of 64. This allows more automatic assigning of Stream ID of downstream channels. Also, more details have been added for using Flow Control with Port Aggregation.

3.0	November 11, 2018	First formal textual specification. Moved requirements for packet length to ODI-1. Changed several packet subtypes from “Not Recommended” to “Permitted”. Corrected Context Packer Header Bit 25 to be set to “1”. Revision 2 had erroneously indicated 0. 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. Appendix added showing options to meet the ODI multiple of 32-byte length requirement. Added explicit reference and section for Extension Command Packets.

Table 1-1: Architectural Specification Revisions

1. ODI Specification Organization and Requirements

1.1 Introduction

ODI is the abbreviation for Optical Data Interface, a high-speed interface for advanced instrumentation and embedded systems. ODI breaks speed and distance barriers by relying on optical communication between devices, over a simple pluggable optical cable.

With single port speeds up to 20 GBytes/s, and system speeds up to 80 GBytes/s, ODI is designed to address challenging applications in 5G communications, mil/aero systems, high-speed data acquisition, and communication research. Though managed by the AXIe Consortium, ODI is not specific to AXIe modular systems, and works equally well with any product format, whether AXIe, PXI, VPX, LXI, or a traditional bench instrument design. Standard ODI ports enable communication between instruments, processors, storage, and embedded devices.

ODI-1 describes the Physical Layer requirements for ODI-1 ports and cables, including the mechanical, optical, line rate, and protocol requirements. ODI-1 is designed to work together with other ODI specifications as shown in the diagram below. ODI-2, the Transport Layer, describes how VITA 49 packets are utilized and deployed, and how ports are aggregated to increase throughput. ODI-2.1 documents a specific set of high-speed data and metadata formats to deliver plug and play interoperability. ODI-A describes a common API for test and measurement equipment.

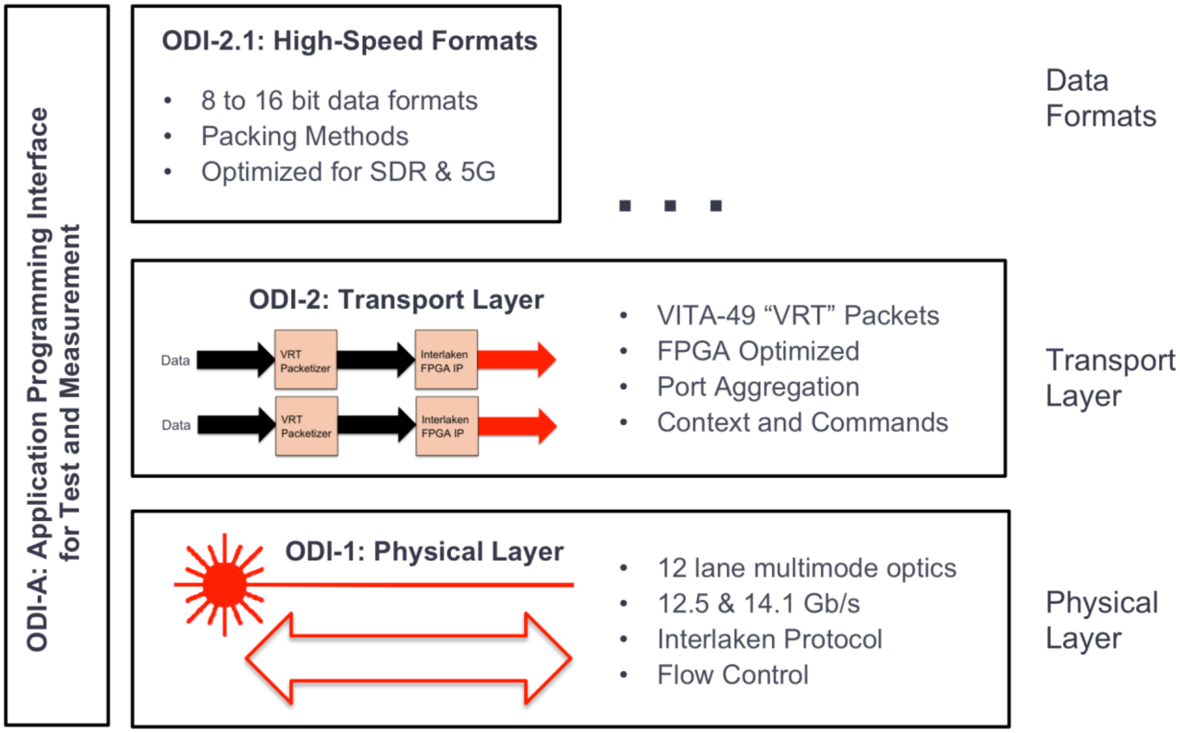


Figure 1-1: ODI Specification Structure

1.2 ODI-2 Compliance

ODI-2 defines the transport layer requirements for ODI-2-enabled products. Common products include instruments and instrumentation modules, RF embedded systems and modules, and storage and processing devices. For a device to claim conformance, it must comply with all applicable RULES in this document, including documentation requirements.

ODI-2 is designed for use with other ODI specifications. Compliance to each ODI specification is to be claimed independently.

1.3 Audience of Specification

This specification is primarily for the use by

- Instrumentation engineers designing ODI products
- Storage, processing, and similar design engineers creating ODI products
- Embedded system and module design engineers creating ODI products
- Component and cable suppliers offering ODI-compatible products

1.4 Organization of Specification

This specification consists of a system of numbered RULES, RECOMMENDATIONS, PERMISSIONS, and OBSERVATIONS, along with supporting text, tables, and figures.

RULEs outline the core requirements of the specification. They are characterized by the keyword “**SHALL**”. Conformance to these rules provides the necessary level of compatibility to support the multi-vendor interoperability expected by system integrators and end users. Products that conform to this specification must meet all of the requirements spelled out in the various rules.

RECOMMENDATIONs provide additional guidance that will help ODI equipment suppliers improve their users’ experiences with ODI systems. They are characterized by the keyword “**SHOULD**”. Following the recommendations should improve the functionality, flexibility, interoperability, and/or usability of ODI products. Products are not required to implement the recommendations.

PERMISSIONs explicitly highlight some of the flexibility of the ODI specification. They are characterized by the keyword “**MAY**”. The permissions generally clarify the range of design choices that are available to product and system designers at their discretion. They allow designers to trade off functionality, cost, and other factors in order to produce products that meet their users’ expectations. Permissions are neutral and imply no preference as to their implementation.

OBSERVATIONs explicitly highlight some of the important nuances of the specification. They help the readers to fully understand some of the implications of the specification’s requirements and/or the rationale behind particular requirements. They generally provide valuable design guidance.

All rules, recommendations, permissions, and observations must be considered in the context of the surrounding text, tables, and figures. Rules may explicitly or implicitly incorporate information from the text, tables, and figures. Although the authors of this specification have gone to significant effort to insure that all of the necessary requirements are spelled out in the rules, it is possible that some important requirements appear only in the specification's free text. Conservative design practice dictates that such embedded requirements be treated as rules.

The ODI specifications also rely on the Interlaken and VITA 49 specifications. The relevant Interlaken and VITA 49 requirements are explicitly referenced in ODI specifications' rules, recommendations, permissions, and observations. The ODI specifications do not duplicate the Interlaken and VITA 49 specifications, but reference them as appropriate. Successful implementation of ODI products and systems requires in-depth knowledge of Interlaken and VITA 49. Designers are encouraged to understand those specifications.

1.5 References

Several other documents and specifications are related to the ODI specifications. These include:

- Telecommunications Industry Association (TIA) standards:
  TIA-604-5-D, FOCIS 5 *Fiber Optic Connector Intermateability Standard- Type MPO,*
  TIA-568.3-D*, Optical Fiber Cabling Component Standard.*

  http://www.tiaonline.org
- Institute of Electrical and Electronics Engineers (IEEE)
  802.3-2016, *IEEE Standard for Ethernet*

  http://standards.ieee.org
- Interlaken Protocol Specification, v1.2. http://www.interlakenalliance.com
- VITA standards:
  VITA 49.2 *VITA Radio Transport (VRT) Standard for Electromagnetic Spectrum*, VITA 49A *Spectrum Survey Interoperability Specification*

  https://shop.vita.com/ANSI-VITA-Standards_c4.htm

1.6 Glossary

The ODI Glossary is organized into three sections:

- ODI Terms
- Interlaken Terms
- Packet Terms

1.6.1 Common ODI Terms

Here are the definitions of the common ODI terms:

- **Device –** A product or assembly that generates or receives data and has one or more optical ports
- **Port –** A single optical connector on a device, and the associated photonics and electronics.
- **Cable –** A multi-fiber optical cable that connects between two ports.
- **Link. -** A unidirectional connection between two ports, consisting of 12 lanes of multimode optical transmission. A bi-directional connection has two links, one in each direction.
- **Producer -** ODI device that generates data to be sent over one or more optical ports.
- **Consumer -** ODI device that receives data sent over one or more optical ports.
- **Transmitter –** Interlaken term for a producer.
- **Receiver –** Interlaken term for a consumer.
- **Emitter –** VITA 49 term for a producer.
- **Exciter –** VITA 49 term for an RF Signal Generator
- **Interlaken –** Interlaken is the name of a chip-to-chip interface specification that is used by ODI to transfer packets between two ODI ports. It is the primary communication protocol, and sits just above the optical layer. Interlaken does not define any structure to the packet at all, other than a SOP (Start of Packet) and EOP (End of Packet) signal. The ODI-1 Physical Layer specifications specify the use of Interlaken, but do not define the packet contents. Separately, the ODI-2 family of specifications defines packet contents and behaviors.
- **VRT –** VRT is an abbreviation for VITA Radio Transport, standardized in VITA 49.2, and enhanced by other VITA 49x specifications. VRT specifies the structure and behavior of VRT packets, which carry data, context, and control information about signals, and the data stream itself. VITA 49 may be abbreviated as V49, just as VITA 49.x may be abbreviated as V49.x
- **Channel –** "Channel" is used in two different contexts in ODI, Interlaken channel and signal channel. Channel is used by Interlaken to enable a completely different data stream with its own flow control. This is not envisioned to be widely used in ODI systems, but is permitted. ODI generally uses only a single Interlaken channel.

  Outside of the Interlaken context, ODI adopts the term "channel" to mean signal channel, and uses VITA 49 VRT packets to transmit one or multiple channels of digitized data. Synchronous signal channels are encoded into the VRT stream in a rotating sequence, and are referred to as a "sample vector" in VRT parlance. VRT Sample Vector Size field is the number of signal channels minus 1. This assumes synchronous channels, all at the same data rate and resolution.

- **Word –** An Interlaken Word is 8 bytes (64 bits). A VRT Word is 4 bytes (32 bits). ODI will specify the context if “Word” is used. ODI often uses “bytes” to avoid this confusion.

1.6.2 Interlaken Terms

ODI uses many Interlaken-specific terms. These include:

- **Burst –** In Interlaken, data is divided into data bursts, each delineated by one or more burst control words. One or more bursts are required to send a complete packet.
- **BurstMax –** An Interlaken parameter that determines the maximum number of data bytes sent for each burst. Consecutive bursts are used to stream data. ODI allows 256 and 2048 byte BurstMax.
- **BurstShort –** An Interlaken parameter that reflects the shortest burst allowed.
- **BurstMin –** An Interlaken parameter for the Optional Scheduling Enhancement that guarantees all packets are at least BurstMin in length, and no idle control words will be needed for long packets.
- **Packet –** A packet refers to the block of data sent between Interlaken SOP and EOP (Start of Packet and End of Packet) indicators. At the Interlaken layer, the format of the packet is unknown. ODI-2 has defined the packet to be VRT packets. The term “packet” within ODI may refer to either.

1.6.3 Packet Terms

ODI has adopted VITA 49 VRT packets as its standard packet framework. Packet terms include VRT terms and ODI terms related to packets:

- **Prologue** – The fields within a packet that precede the packet’s data payload. ODI defines a standard prologue for each VRT packet type.
- **Header** – The first field in the VRT prologue, 4 bytes in length. The header bits indicate packet type, optional fields within a packet, time stamp formats, a modulo-16 counter, and total packet size.
- **Trailer** – The fields within a packet that follow the data payload and conclude the content of the packet. In ODI, the trailer refers to the 4-byte field that follows the data payload within a VRT Data packet. There is no trailer associated with Context Packets or Command Packets.
- **Processing-efficient packing** – Processing-efficient packing refers to a data packing method within the VRT Packet data payload where the packed data is aligned to 32-bit boundaries.
- **Link-efficient packing** – Link-efficient packing refers to a data packing method within the VRT Packet data payload where the data is packed as tightly as possible, leading to the highest sample density and speed.
- **Stream** – A VRT term for a sequence of related packets. All packets of the same stream have the same Stream ID sent from the producer. A typical stream has consecutive Signal Data Packets, with optional Context Packets and/or Command Packets occasionally.

- **Signal Data Packets** – VRT term for a packet carrying digitized samples of one or more signals. This is the primary packet type of ODI. Many ODI systems will only include Signal Data Packets.
- **Context Packet** – VRT term for a packet carrying meta-data or “context” data related to the digitized signals in the same stream. This may include information such as reference level or sampling rate. Context Packets are optional in ODI, but a standard Context Packet is defined in ODI-2.1 if used.
- **Command Packet** – VRT packet type added in V49.2. Command Packets are used to control devices, and the control and acknowledgement process. The Control Packet is the only recommended Command Packet subtype in ODI, and has similar field types to the Context Packet, which are used for control. Control and other Command Packets are optional in ODI, but a standard Control Packet is defined in ODI-2.1 if used. The Control Packet is analogous to the Context Packet, as it is a way to send meta-data to a consumer, such as a signal generator, and has similar fields.
- **Extension Packet** – Extension Signal, Context, and Command packets are used when it is impossible to use the pre-defined data types. An example may be the sending of encrypted data.
- **Train** – For streaming applications, the Train refers to a series of packets, typically of the same size, sent sequentially from a producer, but not including the final packet, called the Caboose
- **Caboose** – For streaming applications, the Caboose refers to the final packet sent from the producer. It may or may not be the same size as the Train packets.
- **Sample Vector** – A Sample Vector is defined within V49.2 as a collection of synchronous Data Samples. This is the common method of transporting multi-channel sample data within the VRT data payload fields. Vector size describes the number of channels. However, the VRT Vector Size Field, used in V49.2 and ODI-2.1, is calculated as the vector size minus one. Therefore a two-channel stream has a vector size of two, but a Vector Size Field of 1.
- **V49.2** – Shorthand for VITA 49.2 specification.
- **ODI-2.1 Data Packet** – A standardized Signal Data Packet defined in ODI-2.1 with common data formats for interoperability.
- **ODI-2.1 Context Packet** – A standardized Signal Context Packet defined in ODI-2.1 with common context fields for interoperability.
- **ODI-2.1 Control Packet** – A standardized Control Packet subtype defined in ODI-2.1 with common control fields for interoperability.

2. Overview of the ODI Transport Layer Specification

ODI-2 defines the transport layer of the ODI specification.

The ODI-2 Transport Layer sits one level higher than the ODI-1 Physical Layer, and has two key deliverables. First, ODI-2 defines the packet structure for sending data between producers and consumers. The packet structure is based on the VITA 49.2 VITA Radio Transport (VRT) standard. ODI-2 specifies how general V49.2 functionality is translated into the ODI environment for each VRT packet type. Second, ODI-2 uses the packet structure to aggregate optical ports together for higher aggregate bandwidth.

Any data may be streamed using ODI-2, while the next layer higher, ODI-2.1, defines specific packet and data formats for high speed sample streaming.

The speeds delivered by ODI dictate that much of the data handling is done by FPGAs. ODI-2 defines a number of requirements to allow VRT packets to be processed efficiently by FPGA-based devices. Besides the ODI-1 requirement that all packets be a multiple of 32 bytes in length, ODI-2 includes mandated fixed-length “Prologue” and “Header” fields for each packet type.

The primary ODI data packet type is the VRT Signal Data Packet. Complete ODI systems may be created using only this packet type. Along with VRT Signal Data Packets, ODI-2 also describes the operations for VRT Extension Data Packets, Context Packets, and Command Packets. Context and Command Packets are typically used to describe or control metadata about the signal stream. ODI-2.1 will define specific Context and Command Packets for high-speed RF streaming.

ODI-2 specifies the rules for aggregating ports. ODI ports are aggregated by synchronizing the packet transmission from each of the ports being aggregated. Each port of the producer sends a VRT Signal Data Packet at approximately the same time. Interlaken SOP (Start of Packet) signals are combined with VRT Prologue fields to synchronize and re-aggregate the packets at the consumer.

2.1 Scope of ODI-2

ODI-2 defines the transport layer of the Optical Data Interface specification
It includes:

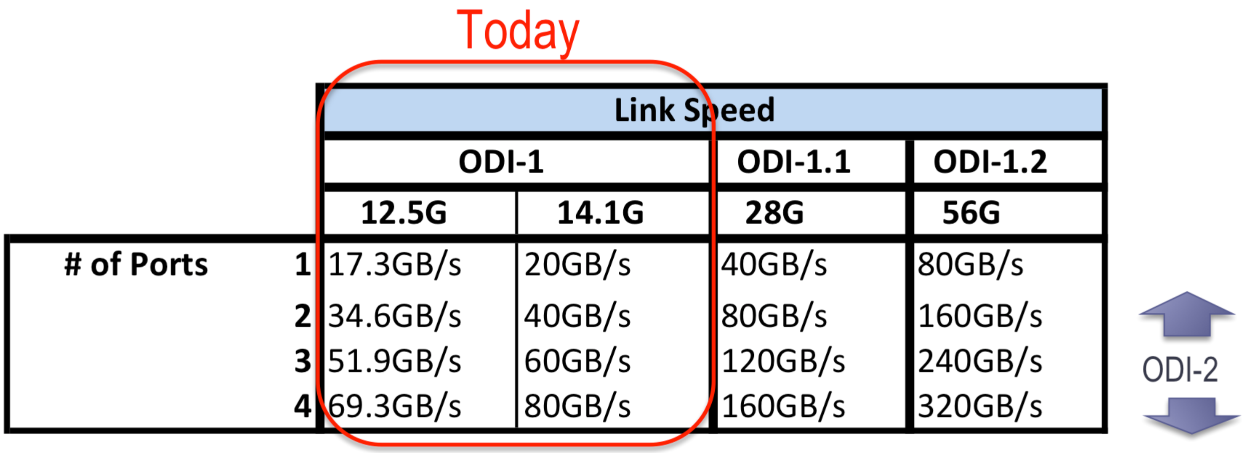
- Packet definitions based on VITA-49 VRT packets
- Specific methods and constraints of using VRT packets
    - Includes Data Packets, Context Packets, and Command Packets
- Port aggregation using VRT packet structure
- Recommendations to work well with ODI-2.1 devices
- Documentation requirements

2.2 ODI-2 Capability and Performance Summary

Packets are fundamental to ODI.

- Packets are bracketed by Interlaken SOP and EOP signals
- Packets contain single channel or multi-channel sample data
- Packets allow the inclusion of timestamps
- Packet boundaries allow for error recovery
- Packets allow port aggregation and synchronization
- Consecutive packets are sent to stream data
- All data is stored as packets
- Packets are independent of the underlying transmission method
- ODI-2 Packets are compliant with VITA 49.2 standard
- ODI-2 constrains the design of VRT packets in a way to make them easily executable by FPGAs with minimal limits on functionality.

Using ODI-2 port aggregation, speeds may be increased as shown in the figure below.



		Link Speed			
		ODI-1		ODI-1.1	ODI-1.2
		12.5G	14.1G	28G	56G
# of Ports	1	17.3GB/s	20GB/s	40GB/s	80GB/s
	2	34.6GB/s	40GB/s	80GB/s	160GB/s
	3	51.9GB/s	60GB/s	120GB/s	240GB/s
	4	69.3GB/s	80GB/s	160GB/s	320GB/s

Figure 2-1: Speed characteristics of ODI devices.

ODI ports deliver continuous device-to-device streaming up to 20 GBytes/s when using a 14.1 Gb/s line rate. 12.5 Gb/s devices can achieve 17.3 GBytes/s. These speeds are designed to increase as the technology permits.

ODI-2, as indicated by the arrows to the right, allows these ports to be aggregated to increase single-stream bandwidth by the number of the ports. ODI-2 does this by a specific use of the VRT packet structure sent from each port.

2.3 ODI Transport Layer Overview

ODI-2 relies heavily on the VRT packet structure. The figure below shows the VRT packet hierarchy used in ODI-2, and largely matches the taxonomy of VRT packet types shown in V49.2.

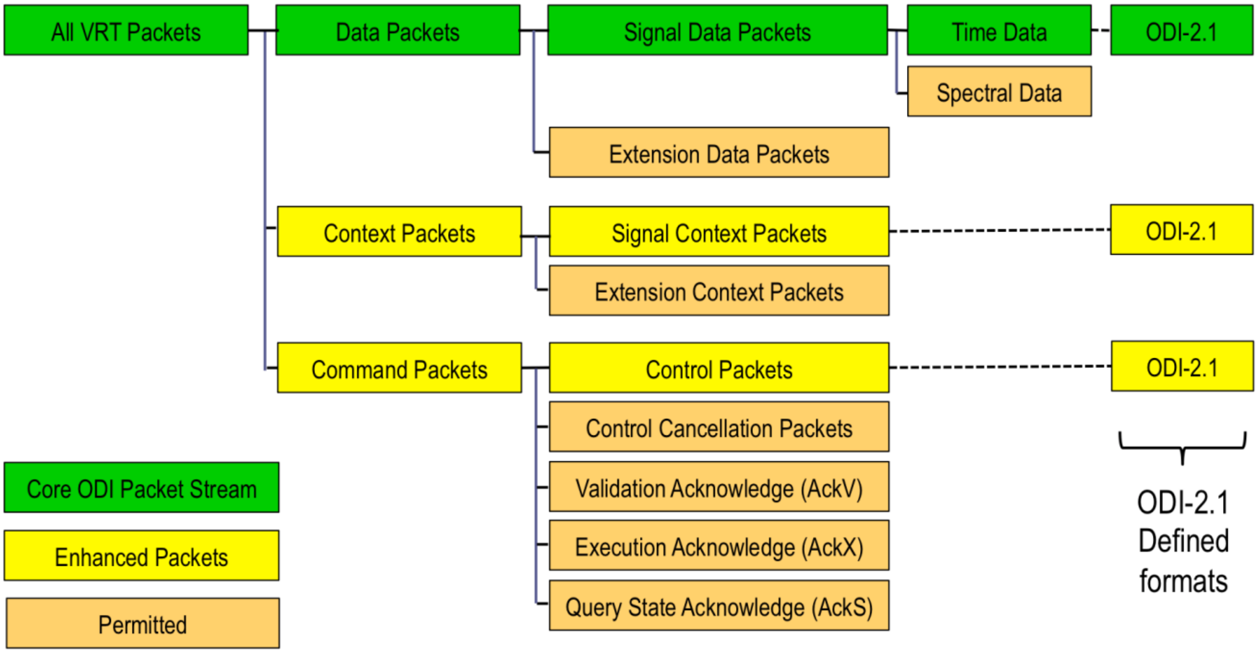


Figure 2-2: ODI Packet Hierarchy

Figure 2-5 shows the packet hierarchy for ODI-2 and ODI-2.1. The highest level includes all VRT packets, and each level to the right becomes more specific. ODI-2 will specify the packet requirements at each point in the hierarchy. For this reason, much of the ODI-2 specification is organized similarly to the packet hierarchy above. ODI-2.1 further defines specific data formats and packet structures, as shown to the far right.

The principal ODI Packet Stream, shown in green, consists of consecutive Signal Data Packets sending time sample data. No other packet streams are required. It is expected that many ODI systems will be built using only Signal Data Packets. ODI-2.1 further defines data formats for 8-bit to 16-bit multi-channel sample data.

Context and Command Packets may accompany a signal data stream and contain meta-data about a signal. This enhances the data stream with valuable information, and the major packet subtypes are coded in yellow, as enhanced packets.

Context Packets optionally send additional information about the signal. This is often meta-data that describes certain signal attributes such as bandwidth, reference level, or sample rate. ODI-2.1 further defines a standard Context Packet for interoperability.

Command Packets, and specifically the Control Packet subtype, are analogous to Context packets, but used for commanding a device to change meta-data parameters, such as its reference level. ODI-2.1 further defines a standard Control Packet for interoperability.

While Context Packets are used to report meta-data about a signal, and Control Packets are used to control similar meta-data attributes, ODI allows the use of Context Packets for control applications. This allows a system to record and store a signal, including all of its meta-data parameters indicated in the Context Packets, and subsequently play the recorded stream back without converting the Context Packets to Control Packets.

The ODI packet hierarchy shows a number of packet types in brown (permitted). These packet types will be rare, but are permitted if needed. There is no further standardization of these packet subtypes in ODI-2.1. They include:

- **Spectral Data**. This is not considered the core application of data streaming, as time series will be much more common. This is allowed if needed.
- **Extension packets for data and context**. These are used for data that isn't easily described in a time or spectral sample series, or has context values not defined in V49.2. Whenever possible, standard formats should be adopted. For that reason, Extension packets are not further defined, but they are allowed.
- **Non-Control Command Packets**. V49.2 includes robust capability to not only send commands to a device, but to also cancel those commands or query their execution status. These advanced features, beyond those of the Control Packet subtype, are very challenging to execute in an FPGA-based system. Because of this, Control Cancellation and the three Acknowledge Packet subtypes are not further defined. However, they are allowed if needed.

ODI eases FPGA implementation by requiring that all packets be a multiple of 32 bytes in length and begin with a standard Prologue for that packet type. The standard Prologue ensures all Prologue fields are in the same position each time.

Data Packets begin with a 28-byte Prologue that includes the Header, a Stream Identifier, Class ID, and Time Stamps. Data packets will also include a 4-byte Trailer at the end of the packet, summing to 32 bytes of overhead. Since ODI requires packet length to be a multiple of 32 bytes, the data payload within a Data Packet must also be a multiple of 32 bytes. Null data may be used to insure data packets are always a multiple of 32 bytes.

Context Packets contain the same 28-byte Prologue as the Data Packets, but have no Trailer. Therefore, the Context Data that follows must be a multiple of 32 bytes plus 4

bytes. The first 4 bytes are dedicated to the CIF0 (Context Indicator Field 0), so the remainder of the Context Data must be a multiple of 32 bytes.
Command Packets begin with a 36-byte Prologue. The first 28 bytes are the same as the Data and Context Prologue definitions, but 8 more bytes are added to include the CAM and Message ID fields. Therefore the Command Data must be a multiple of 32 bytes minus 4 bytes.

While timestamps are permitted, and the timestamp fields are always present in the Header, devices are not required to use timestamps. ODI-2 specifies the how to use the timestamp indicator bits in the Header to indicate the type of timestamps used, or if there are no timestamps at all.

Port aggregation is accomplished by sending Data Packets near simultaneously on each port, as indicated in the figure below.

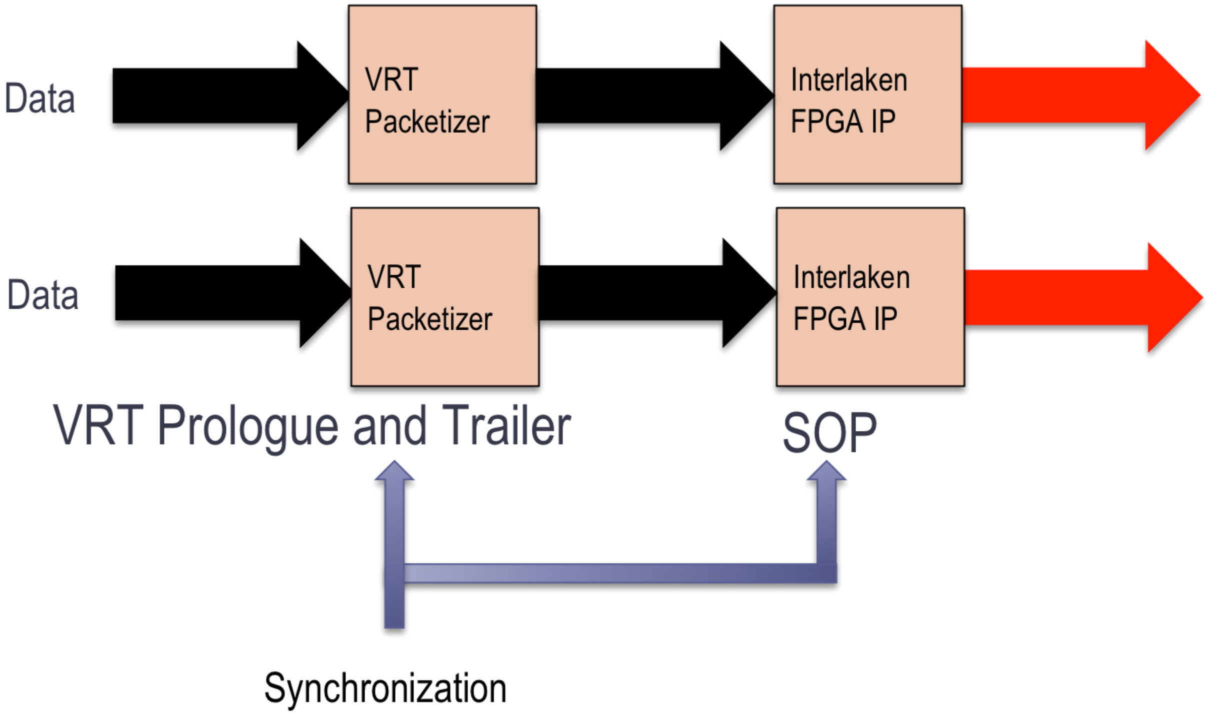


Figure 2-3: ODI Port Aggregation

The VRT Packetizer consists of the encapsulating the signal data with the VRT Prologue and Trailer. At the start of a packet, both ports are commanded to send out Interlaken SOP (Start of Packet) indicator. Data is streamed on each port until the end of packet, where the process is repeated for the next packet. Since most VRT packets are much longer than the variable latency of the FPGA, there is no ambiguity about which packets are time aligned, and they may be recombined at the receiving device.

Port aggregation may be used for single channel or multi-channel data. In all cases, the length of time represented by a single packet is the same. In the single channel case, data is sent to each port in a round robin fashion, with each subsequent sample sent on the next port in the rotation sequence. This can be recombined by the consumer into the original sequence.

For multi-channel data, each port handles a subset of the signal channels. Therefore signals A, B, and C may be on Port 1, while channels D and E may be on Port 2. As long as the signals are synchronous (have the same sample rate and same samples/packet for all channels), multi-channel data may be synchronized across ports.

Flow control is achieved by implementing flow control on each port. The state diagrams are updated to allow robust error recovery in the case of a temporary outage on one or more ports.

3. ODI Packet Specification

This section details the requirements of each ODI packet type and subtype. The ODI packet hierarchy is shown below.

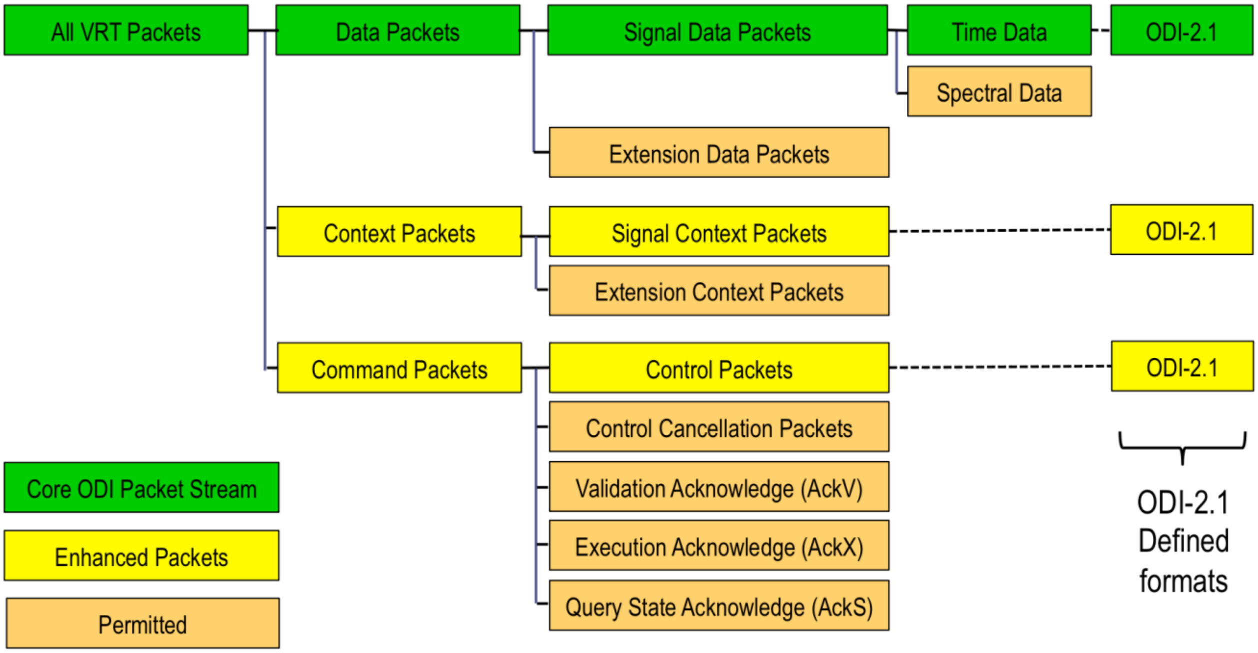


Figure 3-1: ODI-2 Packet Hierarchy

Figure 3-1 shows the ODI-2 packet hierarchy. The core ODI packet stream is shown in green at the top.

OBSERVATION 3.1: ODI-1 requires all packets to be an integer multiple of 32 bytes in length, and between 64 bytes and 262,144 bytes in length. This remains a requirement for all ODI-2 packets.

Packets are described as in Figure 3-2, as a succession of Interlaken words, 64 bits wide each. Interlaken SOP is sent, followed by the Prologue for that packet type. After the prologue, data is sent. At the end of the data may be an optional Trailer. The Prologue, Data Payload, and Trailer need not each be an integer number of Interlaken words. The figure shows a packet with the components each being integer number of 32-bit blocks, half the size of an Interlaken word. The packet end is indicated by EOP. As allowed in ODI-1, SOP and EOP may be indicated together in a single Interlaken word, and is commonly done so for streaming data with the highest efficiency.

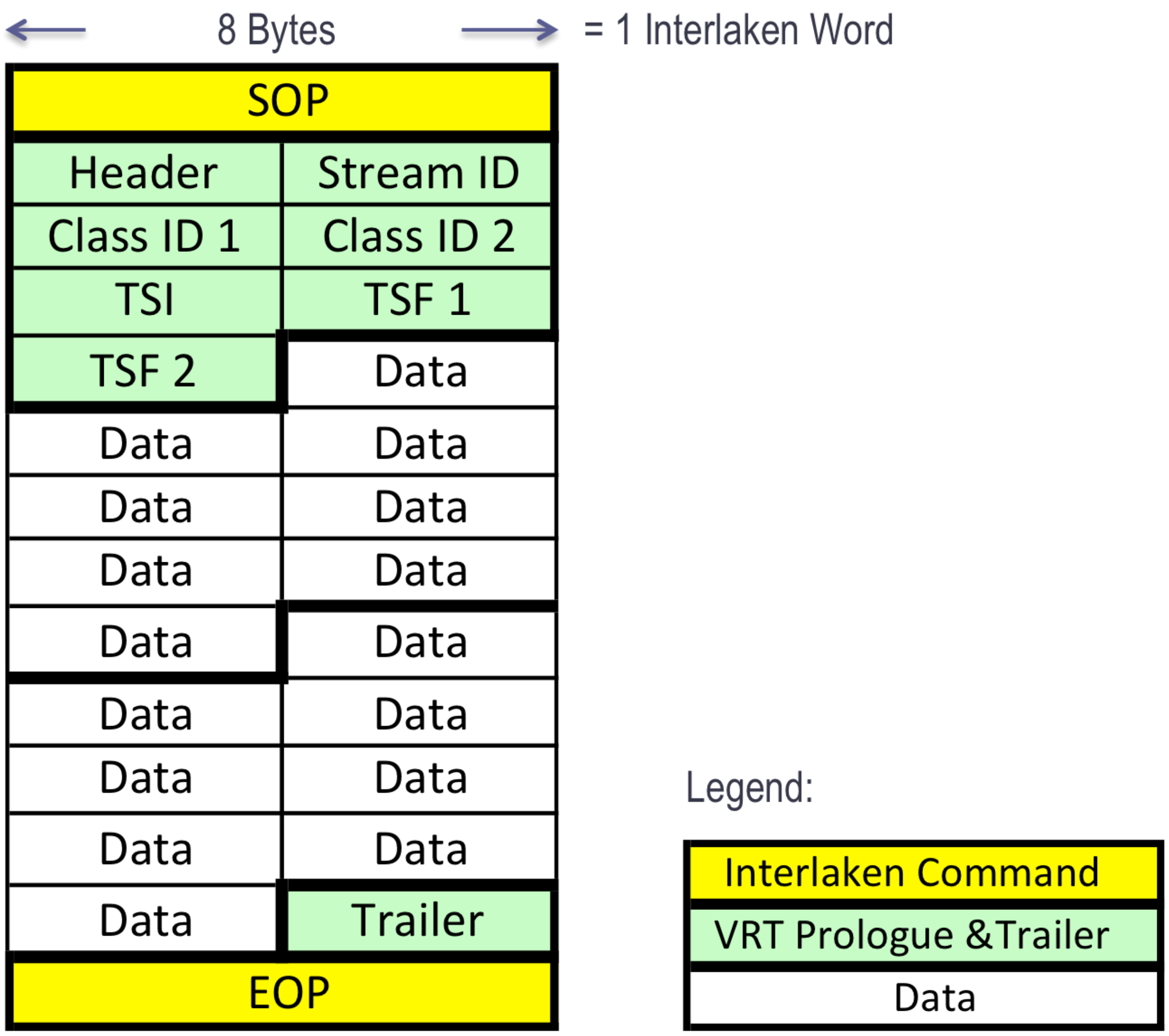


Figure 3-2: ODI-2 Packet Structure

Figure 3-2 shows the packet structure for an ODI-2 compliant VRT Packet. The first seven fields of the Prologue are common to all packet types. The Trailer only exists within VRT Data Packets.

RULE 3.1: All VRT Packets SHALL include the following VRT defined Prologue fields: Header, Stream ID, Class ID 1, Class ID 2, TSI, TSF 1, TSF 2.

3.1 VRT Data Packets

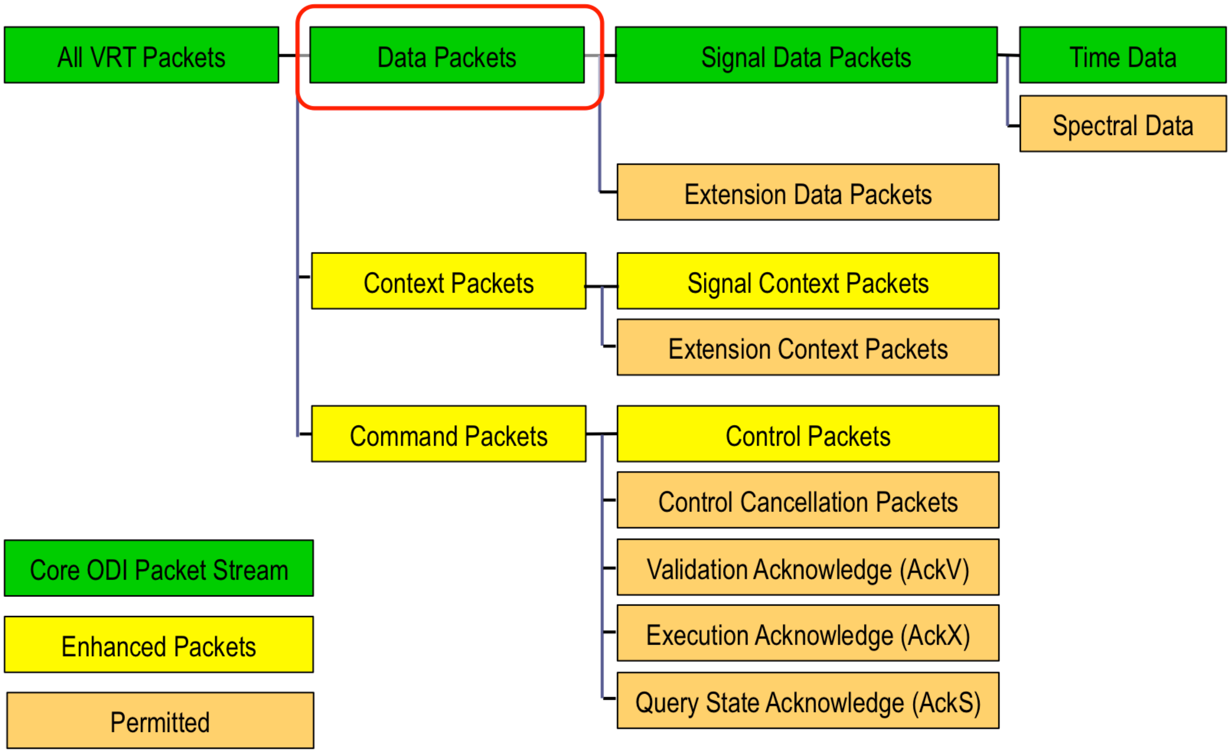


Figure 3-3: VRT Packet Hierarchy – Data Packets

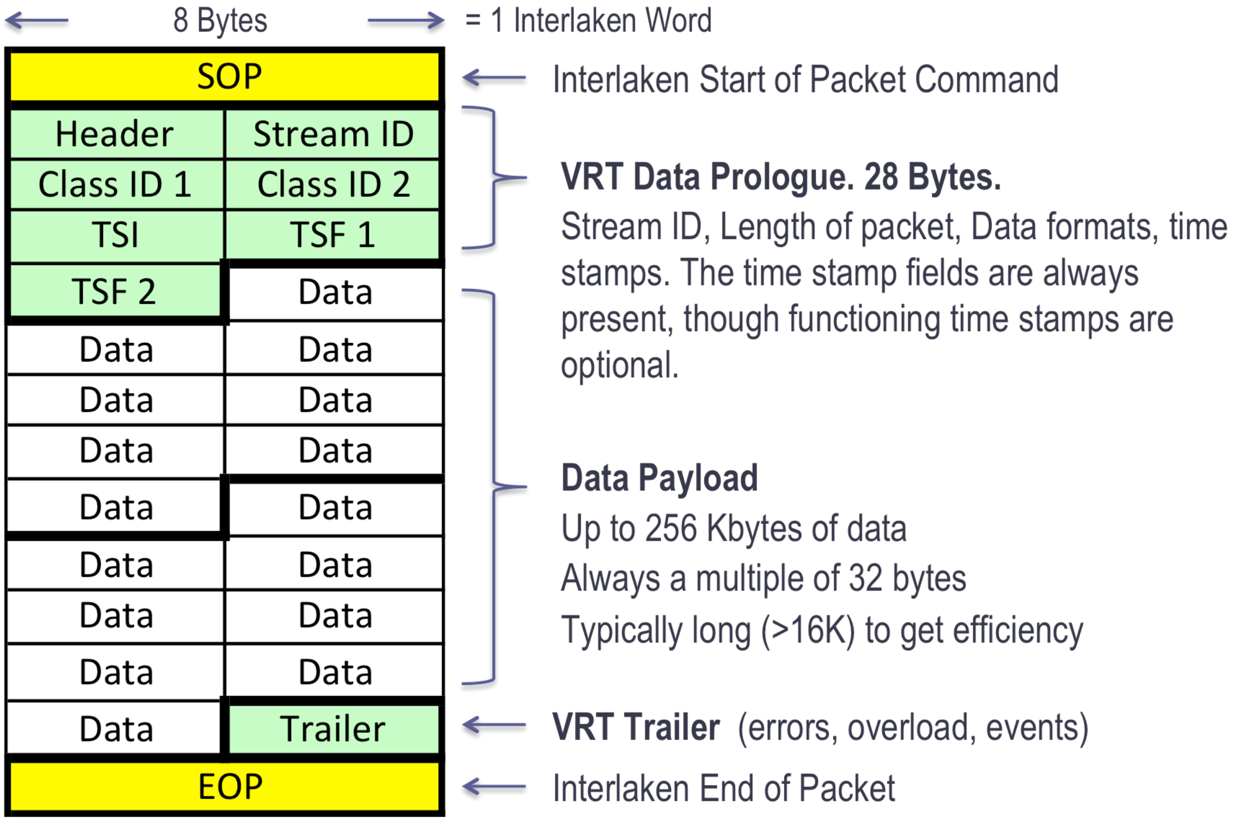
VRT Data Packets include Signal Data Packets and Extension Data Packets.

RULE 3.2: All VRT Data Packets SHALL comply with the Data Packet and Streams section of VITA 49.2, Section 6.

OBSERVATION 3.2: VITA 49.2 specifies two data packet types, Signal Data and Extension Data, but they have similar Prologue and Trailer requirements.

RULE 3.3: All VRT Data Packets SHALL include all seven Prologue fields plus the Trailer field, as defined in VITA 49.2

RULE 3.4: All VRT Data Packets SHALL comply with the diagrams and definitions of each field as documented in VITA 49.2



Interlaken Start of Packet Command

VRT Data Prologue. 28 Bytes.
Stream ID, Length of packet, Data formats, time stamps. The time stamp fields are always present, though functioning time stamps are optional.

Data Payload
Up to 256 Kbytes of data
Always a multiple of 32 bytes
Typically long (>16K) to get efficiency

VRT Trailer (errors, overload, events)

Interlaken End of Packet

Figure 3-4: ODI-2 Packet Structure – Data Packets

3.1.1 Data Packet Structure – Header

The figure below shows the content of the mandatory Header for Data Packets. It is functionally equivalent to VITA 49.2 Figure 6.1-2.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				C	Indicators			TSI		TSF		Packet Count				Packet Size															
0	0	X	1	C	T	r	S	TSI		TSF		Pkt Count				Packet Size															

Figure 3-5: ODI-2 Data Packet Header Field

RULE 3.5: All VRT Data Packets SHALL include a Header that complies with the definitions given in Figure 3-5.

Packet Type (28-31). If X=0, the Header indicates a Signal Data Packet. If X=1, the Header indicates an Extension Data Packet.

Bit 28 of the Packet Type is set to 1. This indicates Stream ID field is present.

C bit (27) is set to 1. This indicates Class ID fields are present.

T Indicator bit (26) is set to 1. This indicates a Trailer is present after the data payload.

r Indicator bit (25) is set to 1. This indicates NOT VITA 49.0. The implication is that the packet definitions comply with VITA 49.2 or later.

S Indicator bit (24) is set to 0 for time domain data, and set to 1 for spectral domain data.

TSI bits (22-23) will be set to 01, 10, or 11, depending on the VITA timestamp method chosen. These non-zero values indicate that the TimeStamp-Integer field is present. If the device does not support timestamps, 11 will be used. See Timestamp section for more information.

TSF bits (20-21) will be set to 01, 10, or 11, depending on the VITA timestamp method chosen. These non-zero values indicate that the TimeStamp-Fractional field is present. If the device does not support timestamps, 01 will be used. See Timestamp section for more information.

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.

Packet Size (0-15) indicates how many VRT 32-bit (4-Byte) words are present in the entire data packet, including the mandatory 7 (seven) Prologue fields and the Trailer field. Therefore, the Packet Size indicates the data payload size plus 8 (eight). Maximum VRT size is 65535 4-Byte words. Since ODI-1 requires all packet lengths to be divisible by 32 Bytes, the maximum ODI size is 65,528 VRT words, or 262,112 Bytes.

3.1.2 Data Packet Structure – Stream ID

Stream ID is an abbreviation for Stream Identifier. The Stream ID is a 32-bit field, whose value is the same for all data, context, and command packets associated with that stream. All ODI-2 packets include a Stream ID field.

RULE 3.6: The Stream ID SHALL be programmable by the user

RULE 3.7: The default Stream ID for a single port device SHALL be 4096.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-6: Default Stream ID of 4096 for a single port device

RULE 3.8: In a multi-port device where the ports are being aggregated, each additional port's Stream ID SHALL be incremented by 1024 by default.

OBSERVATION 3.3: In a 4-port aggregation, the default Stream IDs are:
Port 1: 4096
Port 2: 5120
Port 3: 6144
Port 4: 7168

OBSERVATION 3.4: By incrementing by 1024 for each additional port, each port can be identified by the Stream ID. Incrementing by 1024 still allows downstream devices processing the data to increment the Stream ID by 1, as envisioned by VITA 49A, without causing duplication of a Stream ID.

3.1.3 Data Packet Structure – Class ID

Class ID is a required field of 64 bits, shown as two 32-bit words below. It is functionally equivalent to VITA 49.2 Figure 5.1.3-1.

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="5">Pad Bit Count</td><td colspan="3">Reserved</td><td colspan="24">Organizationally Unique Identifier (OUI)</td></tr>
<tr><td colspan="16">Information Class Code</td><td colspan="16">Packet Class Code</td></tr>
</table>

Figure 3-7: Contents of the Class ID Field

The purpose of Class ID is to identify the Information Class used for the application, and the Packet Class from which each packet was made. ODI-2.1 will further define the contents of the Class ID field to uniquely determine the number of signal channels, data formats, and other attributes algorithmically.

RULE 3.9: ODI-2 devices SHALL include a Class ID field as documented in Section 3.1.3.

The OUI (Organizationally Unique Identifier) identifies the organization that the subsequent Information Class Code and Packet Class Code relate to. Those codes will uniquely identify the data formats and other attributes of the data payload. A device provider may choose their own OUI and class codes, or use a standard OUI and class codes for multi-vendor interoperability. For ODI, the AXIe Consortium has obtained an OUI of 24-5C-CB, which is free for all device suppliers to use as long as they comply with the subsequent Information Class and Packet Class codes.

RULE 3.10: The OUI field SHALL be set to the AXIe OUI of 24-5C-CB or to the OUI of the organization defining the Information Class Code and Packet Class Code.

Reserved (24-26) is set to 0, per VITA 49.2.

Pad Bit Count (27-31) is set per VITA 49.2.

PERMISSION 3-1: IF a device supplier uses their own OUI, THEN they MAY define the Information Class Code and Packet Class Code as they wish, pursuant to VITA 49.2.

RULE 3.11: IF a device supplier uses the AXIe OUI, and they are implementing packets defined by an auxiliary ODI specification (such as ODI-2.1), THEN they SHALL set the Information Class Code and the Packet Class Code to the value specified by the auxiliary standard.

3.1.4 Data Packet Structure – Timestamps

Timestamp fields are mandatory in ODI-2, though operational timestamps are not. This keeps all Prologues for Data Packets the same length, regardless of timestamp implementation. Timestamp fields include TSI, TSF1, and TSF2, each four bytes wide as shown in Figure 3-4. Since the TSI and TSF bits in the Header are not set to 00, The TSI and TSF fields are always present.

The following table shows the timestamp functionality indicated by the TSI and TSF Header bits. They are consistent with definitions in VITA 49.2. The key change is specifying the “No Valid Timestamps” option, even when the fields are required.

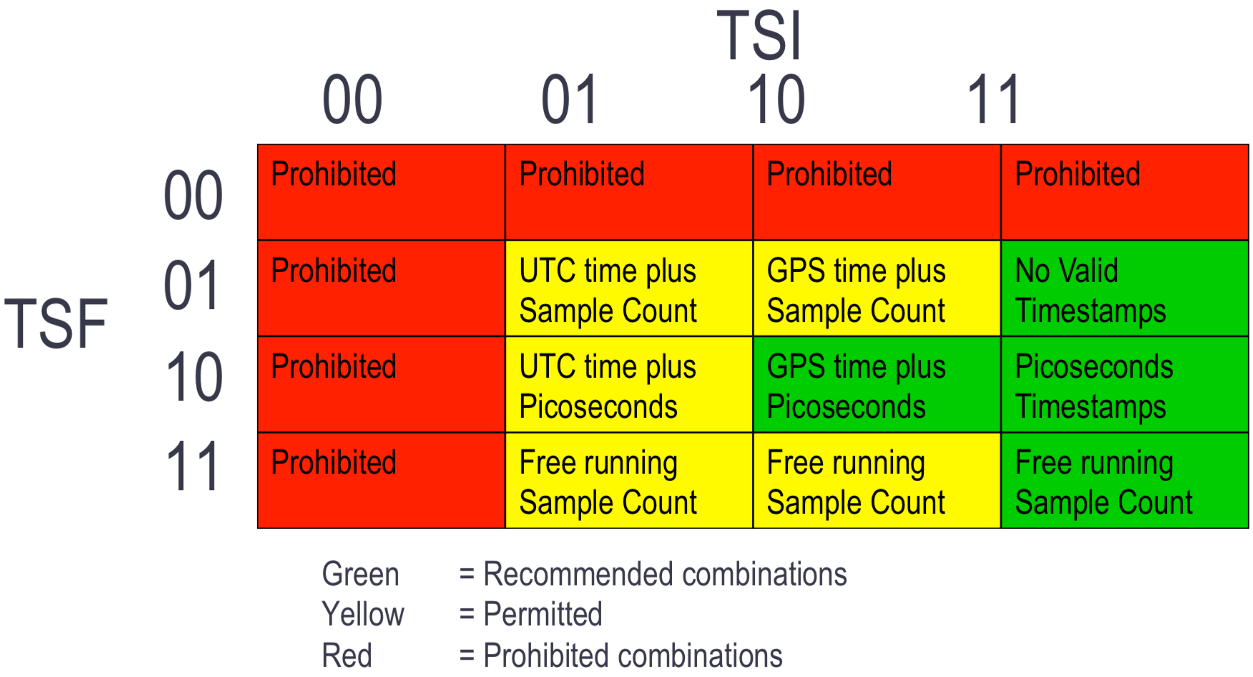
ODI-2 allows the use of all V49.2 timestamps, but recommends four in particular.

For absolute time timestamps, GPS is recommended over UTC. GPS is easier to implement with FPGAs than UTC. Neither of the absolute time plus Sample Count combinations is recommended.

Picoseconds Timestamps and Free running Sample Count are also recommended, and may be implemented with FPGAs.

In the above combinations there are no restraints on the accuracy of the timestamps. The accuracy is specified by the ODI device. The absolute time timestamp methods of GPS and UTC require the format be correct, but the absolute time need not be.

Finally, No Valid Timestamps is also a recommended combination.



TSF \ TSI	00	01	10	11
00	🟥 Prohibited	🟥 Prohibited	🟥 Prohibited	🟥 Prohibited
01	🟥 Prohibited	🟨 UTC time plus Sample Count	🟨 GPS time plus Sample Count	🟩 No Valid Timestamps
10	🟥 Prohibited	🟨 UTC time plus Picoseconds	🟩 GPS time plus Picoseconds	🟩 Picoseconds Timestamps
11	🟥 Prohibited	🟨 Free running Sample Count	🟨 Free running Sample Count	🟩 Free running Sample Count

Green	= Recommended combinations
Yellow	= Permitted
Red	= Prohibited combinations

Figure 3-8: Timestamp functionality indicated by TSI and TSF bits

RULE 3.12: The TSI and TSF codes in the Header of an ODI-2 packet SHALL accurately reflect the timestamp used, as shown in Figure 3-8.

OBSERVATION 3.5: The seven prohibited combinations of TSI and TSF Header bits eliminate any combination that does not include all timestamp fields.

OBSERVATION 3.6: A device that does not produce valid timestamps will set TSI=11 and TSF=01.

RECOMMENDATION 3.1: IF the ODI-2 device can execute timestamps, THEN it SHOULD execute GPS timestamps.

The above recommendation aligns with ODI-2.1, where the preferred timestamp method is GPS.

RULE 3.13: A consumer SHALL ignore timestamp fields IF it cannot operate on the timestamps.

The above rule makes the consumer a forgiving listener if timestamp data is sent to it. It allows a producer to generate timestamps without requiring the consumer to operate on them.

3.1.5 Data Packet Structure – Data Payload

The Data Payload occurs between the 28-byte Prologue and the 4-byte Trailer, as shown below.

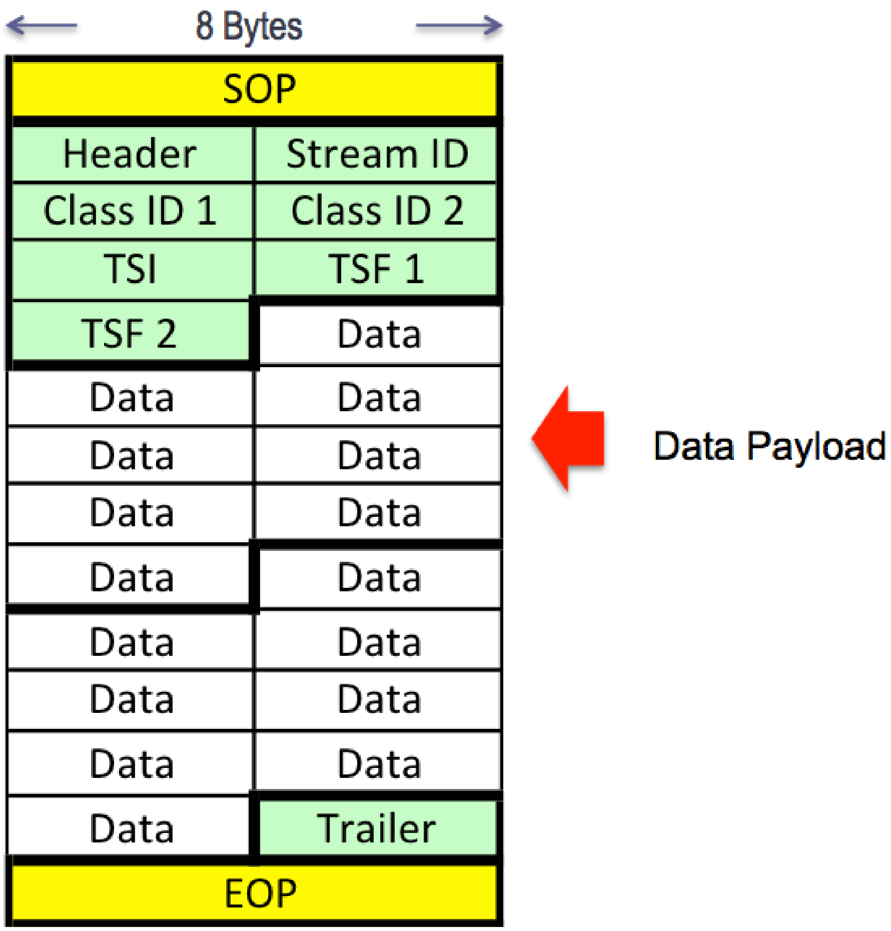


Figure 3-9: Data Payload Fields

RULE 3.14: The length of the Data payload SHALL be an integer multiple of 32 bytes.

OBSERVATION 3.7: ODI-1 requires all packets to be an integer multiple of 32-bytes. Since the Prologue and Trailer sum to 32 Bytes, the above rule forces the entire packet to be a multiple of 32 Bytes.

OBSERVATION 3.8: If the Data Payload is not naturally divisible by 32 bytes, null data may be appended to the end of the data payload to make it divisible by 32.

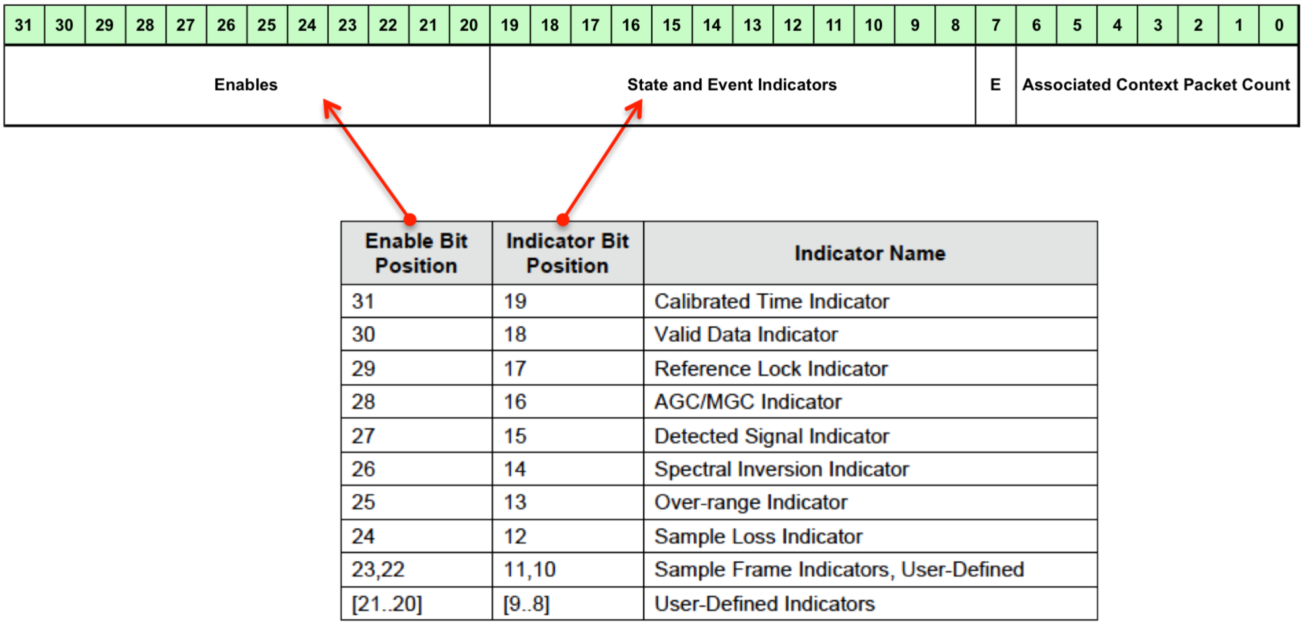
With Packet Length specified in the Packet Header and Pad Bit Count specified in the Class ID field, the valid data may be determined.

OBSERVATION 3.9: Most multi-channel sample packing combinations can be chosen to make the streamed Train packets divisible by 32 bytes. The simplest way to do this is for a packet to contain an integer multiple of 256 samples of each channel.

PERMISSION 3-2: A Data Packet MAY pack the Data Payload using either link-efficient packing or processing-efficient packing.

3.1.6 Data Packet Structure – Trailer

The Trailer is a mandatory field of 32 bits for all Data Packets. The figure below shows the fields comprising the Trailer



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Enables												State and Event Indicators												E	Associated Context Packet Count						

Enable Bit Position	Indicator Bit Position	Indicator Name
31	19	Calibrated Time Indicator
30	18	Valid Data Indicator
29	17	Reference Lock Indicator
28	16	AGC/MGC Indicator
27	15	Detected Signal Indicator
26	14	Spectral Inversion Indicator
25	13	Over-range Indicator
24	12	Sample Loss Indicator
23,22	11,10	Sample Frame Indicators, User-Defined
[21..20]	[9..8]	User-Defined Indicators

Figure 3-10: Data Packet Trailer

RULE 3.15: The length of the SHALL implement the Data Packet Trailer as defined in VITA 49.2 for all Data Packets.

RULE 3.16: The Calibrated Time Indicator, Valid Data Indicator, Reference Lock Indicator, AGC/MGC Indicator, Detected Signal Indicator, Spectral Inversion Indicator, Over-range Indicator, and Sample Loss Indicator SHALL be enabled and set in the trailer if their values are known.

PERMISSION 3-3: An ODI-2 device MAY use User-Defined Indicators in the Trailer.

PERMISSION 3-4: An ODI-2 device MAY use the E field and Associated Context Packet Count field, but is not required to do so.

RULE 3.17: If a consumer receives Trailer information that it can’t act on, it SHALL continue normal operation.

The above rule makes the consumer a forgiving listener with regards to the Trailer.

3.1.7 VRT Signal Data Packet

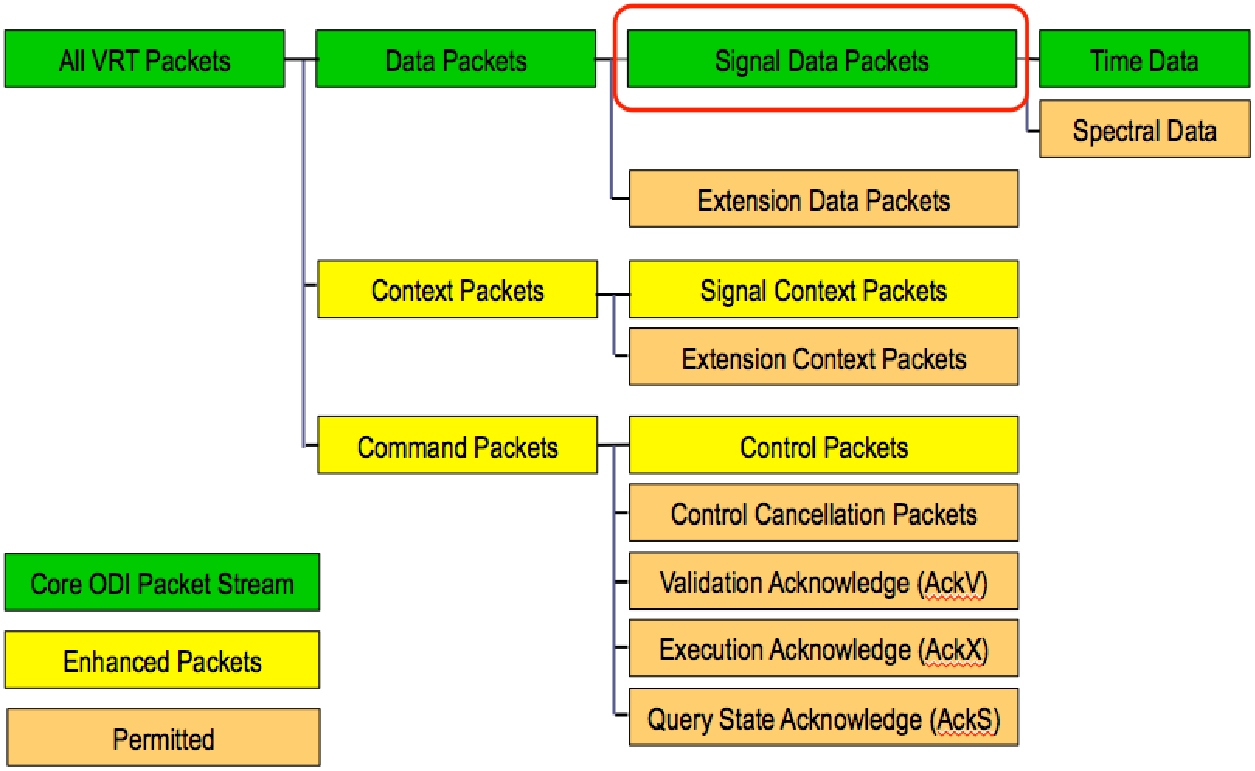


Figure 3-11: VRT Signal Data Packet

A VRT Signal Data Packet is indicated by Packet Type bits in the Header, as shown below. In this case, Packet Type is set to 0001. Note that bit 29 is set to 0, differentiating it from an Extension Data Packet.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				C	Indicators			TSI		TSF		Packet Count				Packet Size															
0	0	0	1	C	T	r	S	TSI		TSF		Pkt Count				Packet Size															

Figure 3-12: VRT Signal Data Packet Header

3.1.7.1 Time and Spectral Data

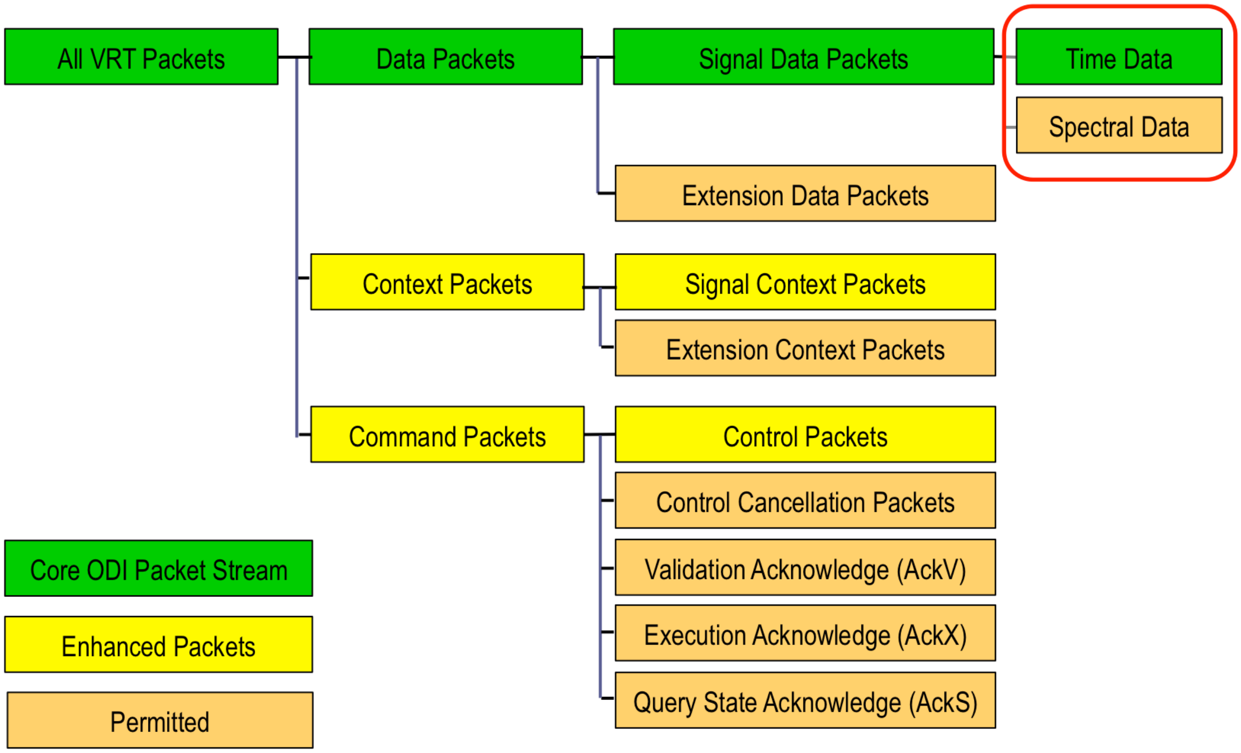


Figure 3-13: Time and Spectral Data

VRT Signal Data Packets can send either Time Data or Spectral Data. This is indicated by the "S" bit of the Signal Data Packet Header. As specified in VITA 49.2, if S=0, then the Data Payload represents Time Data. If S=1, then the Data Payload represents Spectral Data.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				**C**	**Indicators**			**TSI**		**TSF**		**Packet Count**				**Packet Size**															
0	0	0	1	C	T	r	S	TSI		TSF		Pkt Count				Packet Size															

"S" bit indicates Time or Spectral Data.

Figure 3-14: VRT Signal Data Header for Time and Spectral Data

3.1.8 VRT Extension Data Packet

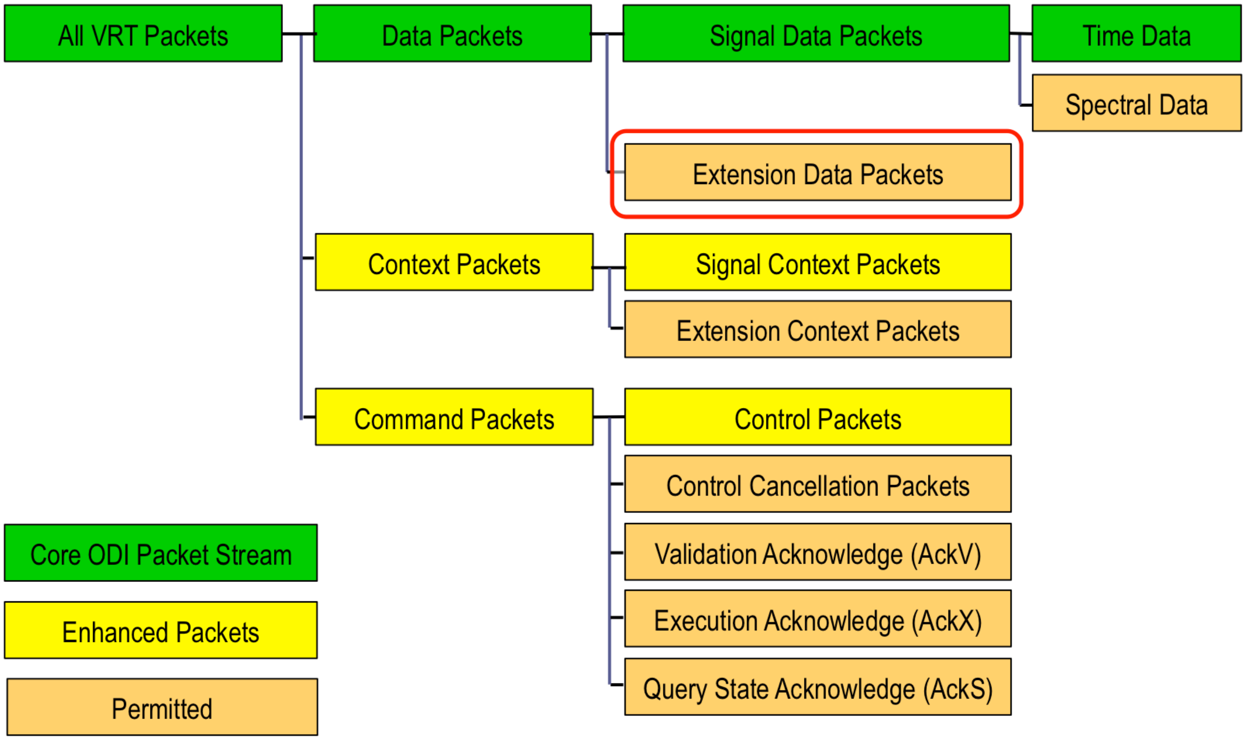


Figure 3-15: VRT Extension Data Packet

VRT Extension Data is used for data sequences that doesn't classify well as either time data or spectral data. Examples of Extension Data may include encrypted data or digital block data. Under VITA 49.2 rules, if the data payload is time or spectral data, an Extension Data Packet cannot be used.

A VRT Extension Data Packet is indicated by Packet Type bits in the Header, as shown below. In this case, Packet Type is set to 0011. Note that bit 29 is set to 1, differentiating it from a Signal Data Packet.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Packet Type</td><td>C</td><td colspan="3">Indicators</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="4">Packet Count</td><td colspan="16">Packet Size</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>C</td><td>T</td><td>r</td><td>S</td><td colspan="2">TSI</td><td colspan="2">TSF</td><td colspan="4">Pkt Count</td><td colspan="16">Packet Size</td></tr>
</table>

Figure 3-16: VRT Extension Data Packet Header

3.2 VRT Context Packets

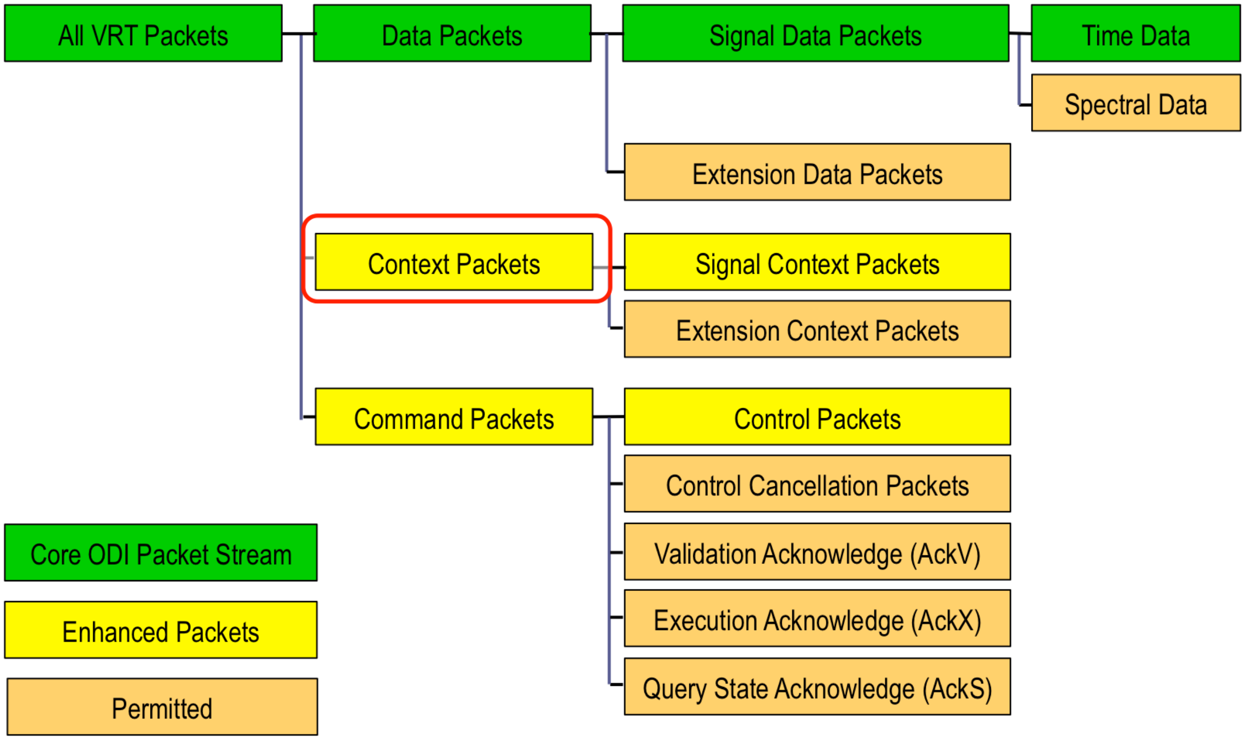


Figure 3-17: VRT Context Packet

Adopting the VRT packet definitions allows the use of VRT Context Packets in addition to VRT Data Packets. Context Packets convey metadata related to the signal, such as the Reference Level, RF Reference Frequency, or Sample Rate. V49.2 contains a long and robust list of standard metadata attributes to choose from. These standard attributes may be mixed and matched within a Signal Context Packet. Extension Context Packets may be used to communicate metadata not defined within the choices defined in V49.2.

The generation and handling of Context Packets is optional capability for ODI-2 devices. That is, a producer is not required to generate Context Packets, but may do so. Similarly, a consumer is not required to act on Context Packets, but may do so.

Context Packets and Command Packets (described in Section 3.3) have many similarities. Both communicate metadata about the signal stream. Starting with V49.2, VITA has specified Context Packets as the standard way to report metadata related to a signal, and Command Packets as the standard way to control metadata parameters related to the signal. That is, an RF digitizer may report attributes about the signal using Context Packets, while an RF signal generator would expect to receive Command

Packets to change metadata parameters. V49.2 and ODI use the Control Packet subtype of the Command Packet to relay metadata information.

ODI-2 allows consumers to also treat Context Packets as commands, allowing a recorded signal, including the metadata parameters, to be played back. However, if a device can execute Context Packets it receives, it must also execute Command Packets, the preferred method. In this case, the specific Command Packet subtype would be the Control Packet.

Consumers are forgiving listeners of Context Packets. If the consumer cannot operate on Context Packets, then it will ignore the packet and continue normal operation. This behavior is extended to any Context Packet parameter the consumer doesn’t understand or doesn’t have the capability to act on. The consumer will ignore those parameters and continue normal operation.

Context packets are fully described in the V49.2 standard. ODI-2 Context Packets have a standard 28 byte Prologue and no Trailer. Context packets, like all ODI packets, must be an integer multiple of 32 bytes in length. This can be achieved by adding null data after the last Context Field.

PERMISSION 3-5: An ODI-2 producer MAY generate Context Packets, but is not required to do so.

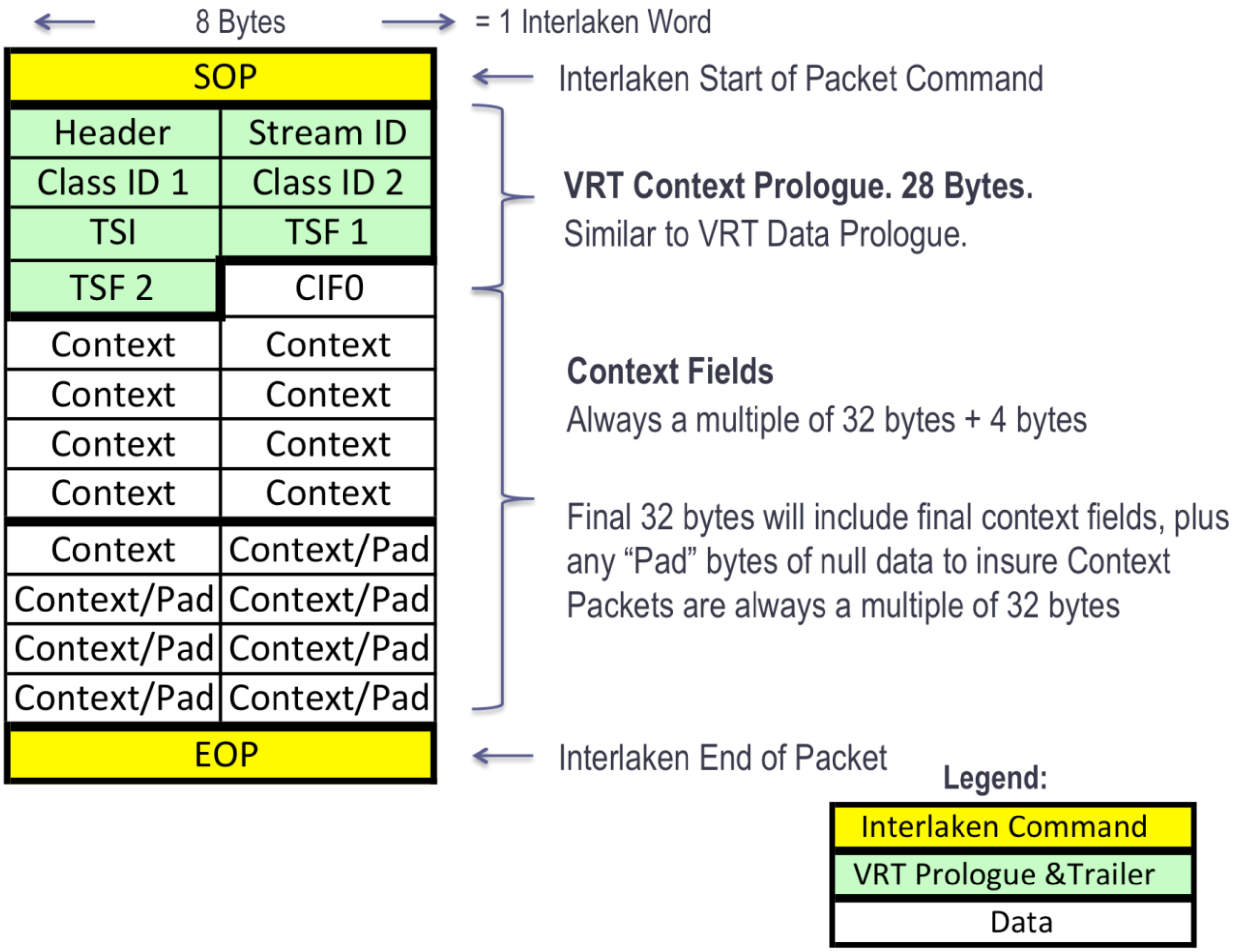
PERMISSION 3-6: An ODI-2 consumer MAY execute Context Packets, but is not required to do so.

RULE 3.18: If a consumer can execute Context Packets, then it SHALL also execute Control Packets for the same metadata parameters.

RULE 3.19: If a consumer cannot execute Context Packets or specific metadata parameters within those packets, then it SHALL continue normal operation without execution of the metadata.

OBSERVATION 3.10: Since an ODI-2 Context Packet is a VRT packet, it SHALL be an integer multiple of 32 bytes in length.

OBSERVATION 3.11: If a Context Packet is not naturally an integer multiple of 32 bytes in length, it can be made to be so by adding null data after the last Context Field. The amount of null data is chosen to meet the 32-byte requirement.



Interlaken Start of Packet Command

VRT Context Prologue. 28 Bytes. Similar to VRT Data Prologue.

Context Fields Always a multiple of 32 bytes + 4 bytes

Final 32 bytes will include final context fields, plus any "Pad" bytes of null data to insure Context Packets are always a multiple of 32 bytes

Interlaken End of Packet

Figure 3-18: VRT Context Packet Fields

The figure above shows the fields for an ODI-2 Context Packet.

RULE 3.20: AN ODI-2 producer that generates Context Packets SHALL include all Prologue fields specified for Context packets, as shown in Figure 3-18.

OBSERVATION 3.12: The mandated specified fields for the Context Packet Prologue are Header, Stream ID, Class ID 1, Class ID 2, TSI, TSF 1, and TSF 2.

3.2.1 Context Packet Structure – Header

The figure below shows the content of the mandatory Header for Context Packets. It is functionally equivalent to VITA 49.2 Figure 7.1.1-1. The header of the Context Packet is the same as the header of the Signal Data Packet with the exception of the omission of the T (Trailer) bit, and the addition of the TSM bit. The Packet Type is also changed to indicate the appropriate Context Packet.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				C	Indicators			TSI		TSF		Packet Count				Packet Size															
0	1	0	X	C	Res	r	TSM	TSI		TSF		Pkt Count				Packet Size															

Figure 3-19: ODI-2 Context Packet Header Field

RULE 3.21: All VRT Context Packets SHALL include a Header that complies with the definitions given in Figure 3-19.

Packet Type (28-31). If X=0, the Header indicates a Signal Context Packet. If X=1, the Header indicates an Extension Context Packet.

C bit (27) is set to 1. This indicates Class ID fields are present.

Res. Indicator bit (26) is set to 0. This is a Reserved bit.

r Indicator bit (25) is set to 1. This indicates NOT VITA 49.0. The implication is that the packet definitions comply with VITA 49.2 or later.

TSM Indicator bit (24) is set as described in V49.2 Section 7.1.1. Timestamp Mode (TSM) bit is used to indicate whether the Timestamp in the Signal Context Packet is being used to convey timing of Context events related to the Described Signal with fine or coarse resolution. TSM=0 indicates a precise resolution of when context changes occur. TSM=1 indicates a coarse resolution essentially equivalent to the time span of a packet.

TSI bits (22-23) will be set to 01, 10, or 11, depending on the VITA timestamp method chosen, as described in Section 3.1.4. When paired with a Data Packet stream, these bits match the TSI bits of the Data Packet Stream.

TSF bits (20-21) will be set to 01, 10, or 11, depending on the VITA timestamp method chosen, as described in Section 3.1.4. When paired with a Data Packet stream, these bits match the TSF bits of the Data Packet Stream.

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.

Packet Size (0-15) indicates how many VRT 32-bit (4-Byte) words are present in the entire Context Packet, including the mandatory 7 (seven) Prologue fields and any null data.

3.2.2 Context Packet Structure – Stream ID

Stream IDs are used similarly in Context packets as they are used in Data packets. This allows the Context Packet stream to be paired to a specific Data Packet Stream, or associated with another Context Packet stream.

RULE 3.22: The Stream ID for a Context Packet SHALL match the Stream ID for the related Data Packet stream.

OBSERVATION 3.13: ODI-2 specifies that Data Packet Stream IDs must be programmable by the user. Since related Context Packets must share the same Stream ID, they must also be programmable to the same value.

OBSERVATION 3.14: The default Stream ID is 4096, the same as for Data packets.

3.2.3 Context Packet Structure – Class ID

The Class ID identifies the Information Class and Packet Class to which the packet stream belongs.

RULE 3.23: The Class ID for a Context Packet SHALL match the Class ID for the related Data Packet stream.

3.2.4 Context Packet Structure – Timestamps

The Timestamp Fields of a Context Packet function in the same way as for Data Packets.

3.2.5 Context Packet Structure – Context Section

The last section of a Context Packet is the Context Section. This section is described in V49.2 Section 7.1.4. Each Context Section starts with the mandated Context Indicator Field #0 (CIF0). Up to seven additional CIFs (CIF1 through CIF 7) may be included, as indicated within CIF0. Together, the CIF fields indicate the existence or lack of existence of the related Context Fields. The Context Fields contain the metadata being sent.

Since ODI mandates that all VRT packets be integer multiples of 32 bytes in length, the length of the Context Section must be an integer multiple of 32 bytes plus four bytes. This is due to the Prologue being 28 bytes in length, and no Trailer. The length of the Context Section includes all CIFs, all Context Fields, and all null data appended to the end of the Context Section.

3.2.6 VRT Signal Context Packets

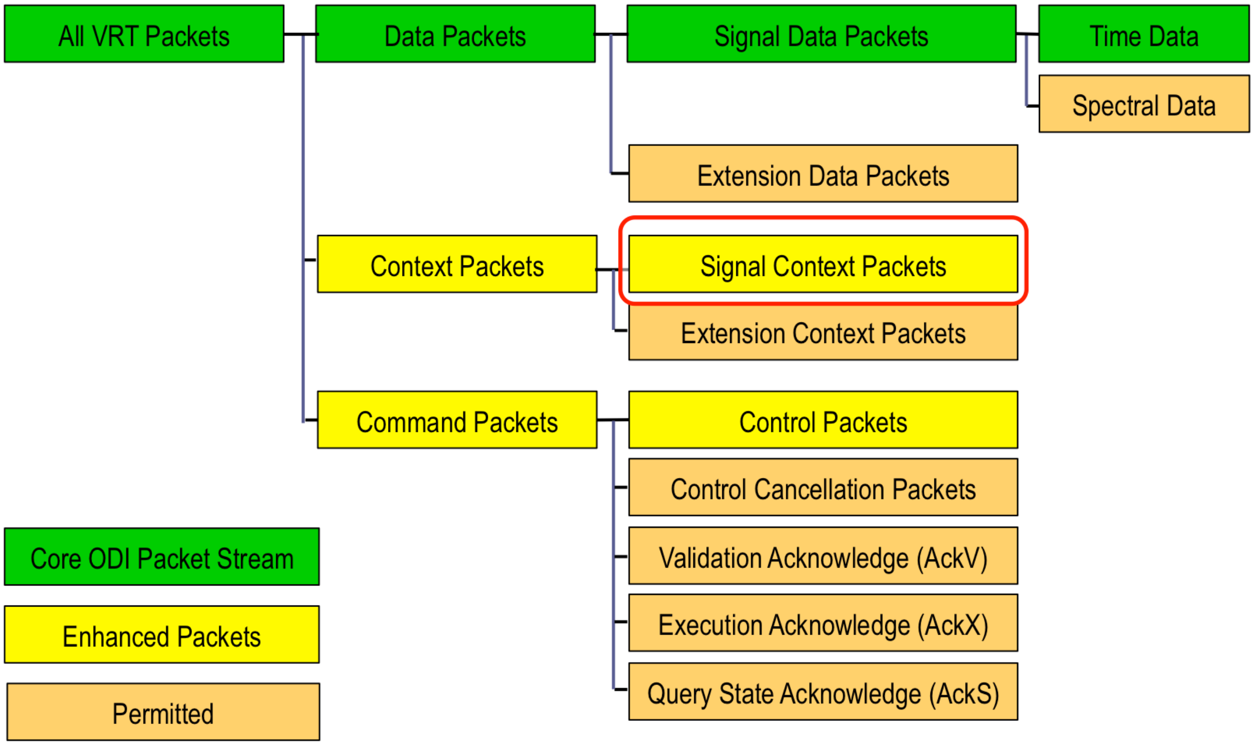


Figure 3-20: VRT Signal Context Packet

A VRT Signal Packet is indicated by the Packet Type bits in the Header, as shown below. In this case, Packet Type is set to 0100. Note that Bit 28 is set to 0, differentiating it from an Extension Context Packet.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				**C**	**Indicators**			**TSI**		**TSF**		**Packet Count**				**Packet Size**															
0	1	0	0	C	Res	r	TSM	TSI		TSF		Pkt Count				Packet Size															

Figure 3-21: VRT Signal Context Packet Header

ODI-2 essentially allows the complete menu of Context Fields specified in V49.2. Auxiliary ODI standards, such as ODI-2.1, specify the particular Context Fields used. Since ODI enables the recording and playback of RF signals, ODI-2.1 has chosen specific fields in Context Packets and Control Packets so the same field locations within a packet contain the same metadata between the two packet types. This is useful for FPGA-based devices that will have field specific logic. This allows a consumer to execute the same metadata whether it is created with a recording using Context Packets, or generated directly using Control Packets.

A method to align Signal Context Packet fields with Control Packet fields is the insertion of two additional CIFs in the Context Section of a Context Packet. This is needed because the Prologue length of a Control Packet is 36 bytes, while that of a Context Packet is 28 bytes. The two CIFs would contain all zeros, so no Context Fields are added other than the two CIFs, each 4 bytes in length.

OBSERVATION 3.15: ODI-2.1 includes CIF0, CIF1, and CIF2 fields in the Context Section of a Context packet, while a Control Packet only includes CIF0.

RECOMMENDATION 3.2: An ODI-2 device SHOULD comply with ODI-2.1 where possible to improve interoperability.

3.2.7 VRT Extension Context Packets

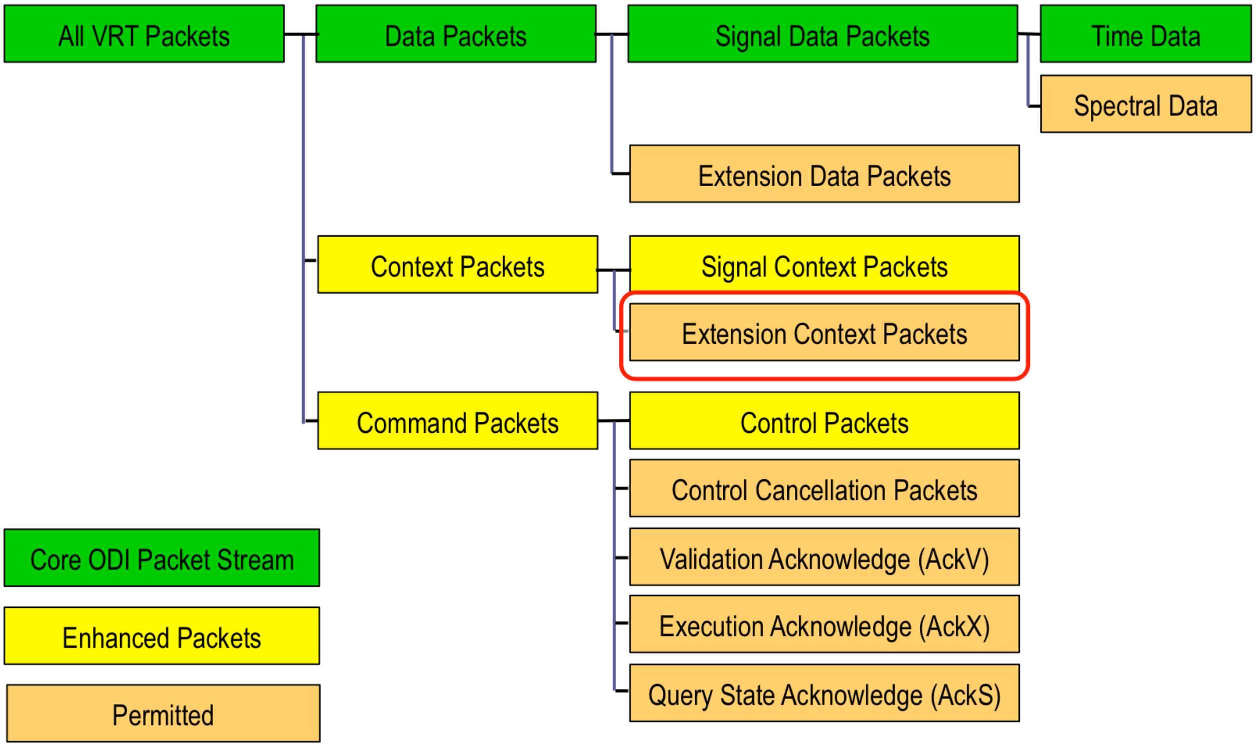


Figure 3-22: VRT Extension Context Packet

Extension Context Packets are intended to be used to communicate metadata for which no provision has been made. They are indicated by the Packet Type of 0101 in the

Header field, as shown in the figure below. Note that Bit 28 is set to "1", indicating an Extension Context Packet.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				C	Indicators			TSI		TSF		Packet Count				Packet Size															
0	1	0	1	C	Res	r	TSM	TSI		TSF		Pkt Count				Packet Size															

Figure 3-23: VRT Extension Context Packet Header

3.3 VRT Command Packets

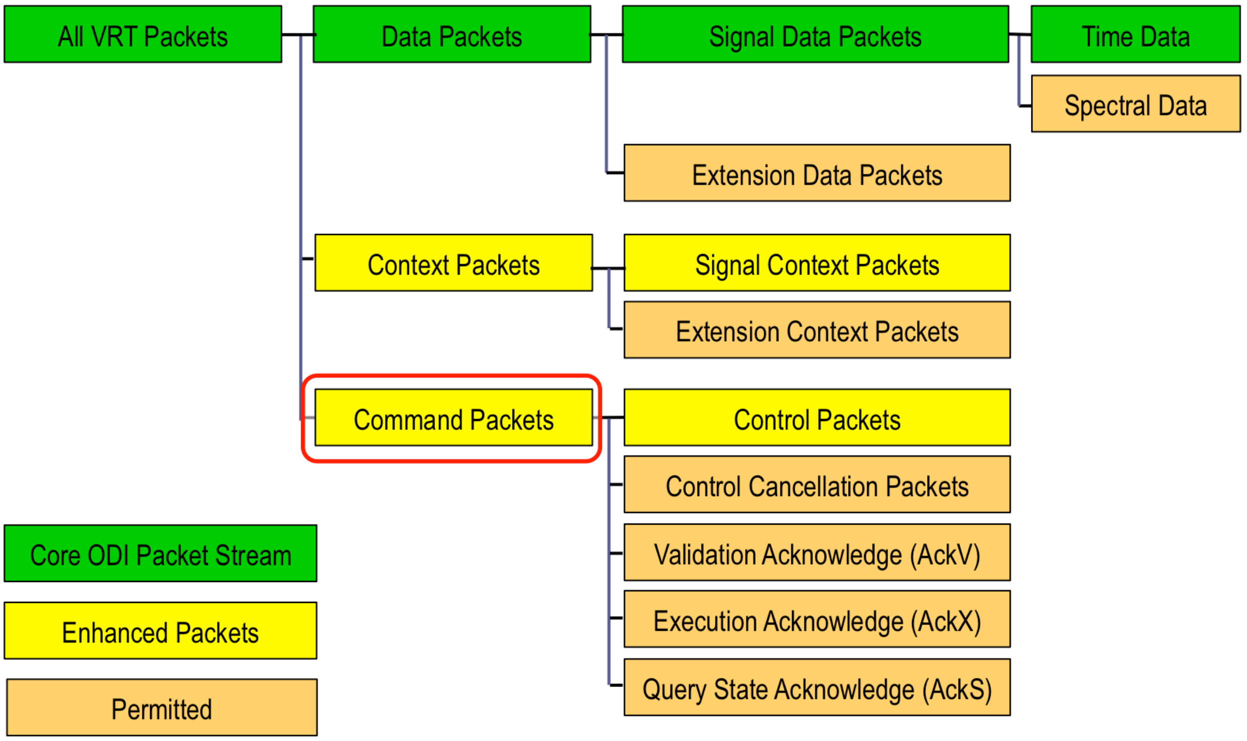


Figure 3-24: VRT Command Packet

Command Packets are the dual of Context Packets (described in Section 3.2) and have many similarities. While Context Packets are used to report metadata about a signal stream, Command Packets are used to control metadata related to a signal stream. That is, an RF digitizer may report attributes about the signal using Context Packets, while an RF signal generator would expect to receive Command Packets to change metadata parameters. V49.2 and ODI use the Control Packet subtype of the Command Packet to relay metadata information.

Command Packets leverage greatly from Context Packet design. The fields being controlled are indicated in the Control Indicator Field (CIF), which is the same structure and definition as the Context Indicator Field used in Context Packets. The CIF in the Control Packet indicates which fields are being controlled, while in the Context Packet it is used to represent which fields are reporting their values.

While ODI-2 permits the use of Context Packets for control applications, the preferred method for control is the Command Packet, and specifically the Control Packet subtype. ODI-2 mandates that a device must offer Control Packets to control any parameter that’s controllable through Context Packets. The only reason Context Packets are permitted for control at all is so recorded data may be “played back” without modification.

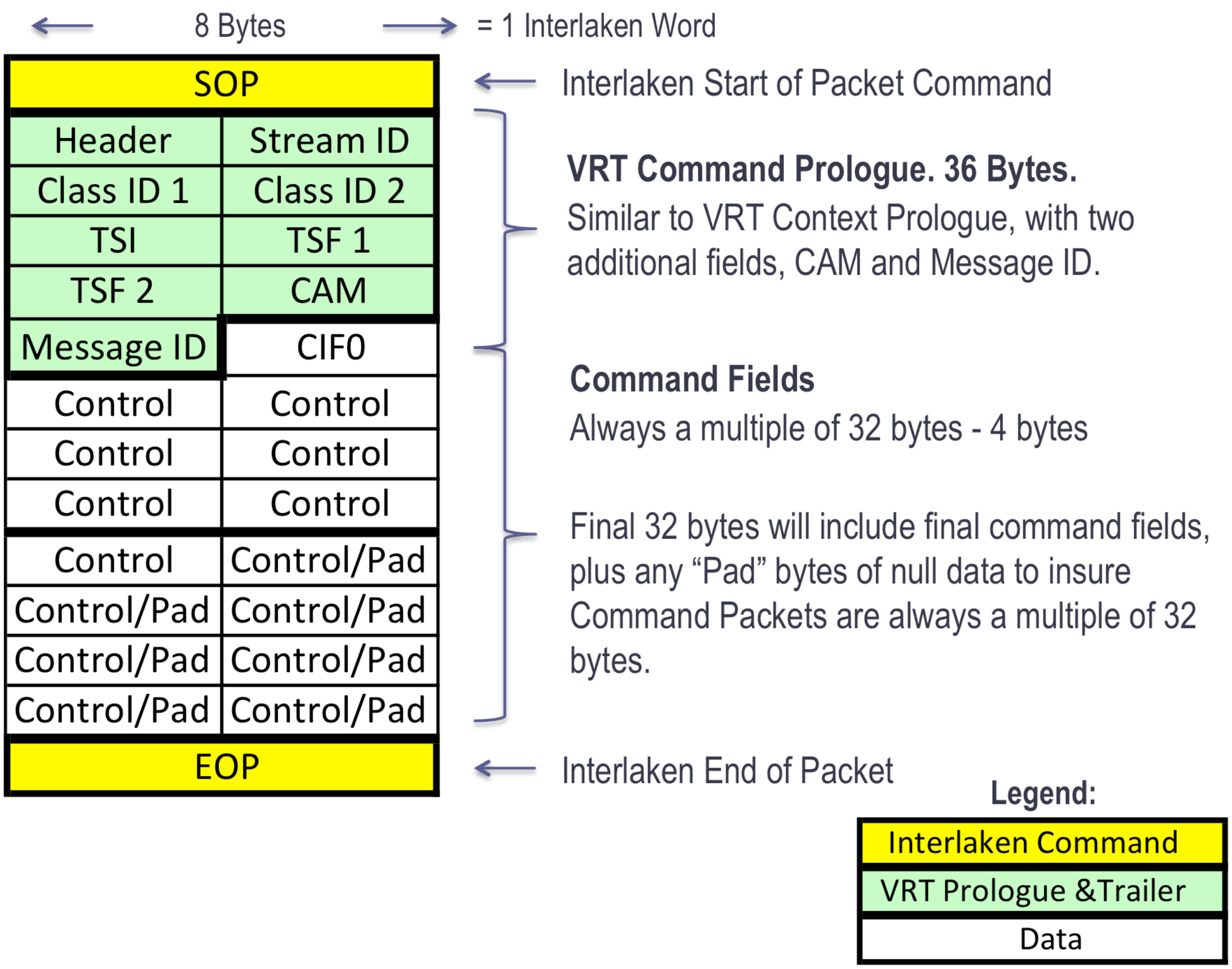
The Command Packet type is based on the notion of a Controller and a Controllee relationship. A Controller issues control messages to the Controllee, and a Controllee puts those controls into effect. V49.2 offers immense flexibility in the routing of these commands, by adding optional Controller and Controllee identification fields. ODI-2 omits those fields as all connections are point to point, and instead relies on Stream ID to distinguish Control Points within an ODI device.

V49.2 adds numerous capabilities to Command Packets not present or needed for Context Packets. Among these are the ability to cancel a Control Packet, and several “acknowledge” packet subtypes that allow a Controllee to report back warnings, errors, and the device state. ODI-2 documents the proper operation of each of these subtypes. However, because ODI devices typically include other means of control, ODI-2 recommends only the use of the Control Packet subtype when control must be embedded in the data stream, avoiding the complexity inherent in control cancellation and acknowledgements. In ODI-2.1, a standard Control Packet is specified for RF applications.

The Prologue for a Command Packet differs from that of a Context Packet by adding two fields, CAM (Control/Acknowledge Mode) and Message ID. The CAM field offers immense flexibility. It defines how and when the recipient of a Control Packet should react to the controls, whether Acknowledge packets are to be generated, and how warnings and errors are handled. Included is the capability to designate a Control Packet as a “dry run”, and settings on handling problems with the Control fields.

Much of this capability is beyond the scope of FPGA-based devices. Additionally, Acknowledge Packets require a reverse optical link to be present. For this reason, ODI-2 recommends producers send simple execution commands via Control Packets and consumers be forgiving listeners. A forgiving listener is one that continues normal behavior even when confronted with a command it doesn’t understand or can’t execute.

However, ODI-2 also specifies how the full capability of the Command Packets can be executed, if required.



← 8 Bytes → = 1 Interlaken Word

SOP	
Header	Stream ID
Class ID 1	Class ID 2
TSI	TSF 1
TSF 2	CAM
Message ID	CIF0
Control	Control
Control	Control
Control	Control
Control	Control/Pad
Control/Pad	Control/Pad
Control/Pad	Control/Pad
Control/Pad	Control/Pad
EOP	

← Interlaken Start of Packet Command

VRT Command Prologue. 36 Bytes.
Similar to VRT Context Prologue, with two additional fields, CAM and Message ID.

Command Fields
Always a multiple of 32 bytes - 4 bytes

Final 32 bytes will include final command fields, plus any "Pad" bytes of null data to insure Command Packets are always a multiple of 32 bytes.

← Interlaken End of Packet

Legend:

🟨 Interlaken Command
🟩 VRT Prologue &Trailer
□ Data

Figure 3-25: VRT Command Packet Fields

Figure 3-25 shows the fields for a Command Packet.

OBSERVATION 3.16: Since an ODI-2 Command Packet is a VRT packet, it SHALL be an integer multiple of 32 bytes in length.

OBSERVATION 3.17: If a Command Packet is not naturally an integer multiple of 32 bytes in length, it can be made to be so by adding null data after the last Control Field. The amount of null data is chosen to meet the 32-byte requirement.

RULE 3.24: AN ODI-2 device that generates Command Packets SHALL include all Prologue fields specified for Command Packets, as shown in Figure 3-25.

OBSERVATION 3.18: The mandated specified fields for the Command Packet Prologue are Header, Stream ID, Class ID 1, Class ID 2, TSI, TSF 1, TSF 2, CAM, and Message ID.

OBSERVATION 3.19: The difference between the Prologue of a Command Packet and that of a Context Packet is the addition of the CAM and Message ID fields.

OBSERVATION 3.20: The Prologue of a Command Packet does not include the Controller/Controllee ID/UUID fields.

3.3.1 Command Packet Structure – Header

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				C	Indicators			TSI		TSF		Packet Count				Packet Size															
0	1	1	0	C	A	R	L	TSI		TSF		Pkt Count				Packet Size															

Figure 3-26: VRT Command Packet Header

RULE 3.25: All VRT Command Packets SHALL include a Header that complies with the definitions given in Figure 3-26.

Packet Type (28-31). The Packet Type is set to 0110, indicating a Command Packet.

C bit (27) is set to 1. This indicates Class ID fields are present.

A bit (26) is set to 0 for Control Packets and Control Cancellation Packets. It is set to 1 to indicate an Acknowledge Packet

R bit (25) is reserved, and is set to 0 for all Command Packets.

L bit (24) is set to 1 to indicate a Control Cancellation Packet, and set to 0 otherwise.

TSI bits (22-23) will be set to 01, 10, or 11, depending on the VITA timestamp method chosen, as described in Section 3.1.4. When paired with a Data Packet stream, these bits match the TSI bits of the Data Packet Stream.

TSF bits (20-21) will be set to 01, 10, or 11, depending on the VITA timestamp method chosen, as described in Section 3.1.4. When paired with a Data Packet stream, these bits match the TSF bits of the Data Packet Stream.

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.

Packet Size (0-15) indicates how many VRT 32-bit (4-Byte) words are present in the entire Context Packet, including the mandatory 9 (nine) Prologue fields and any null data.

3.3.2 Command Packet Structure – Stream ID

Stream IDs are used similarly in Command Packets as they are used in Data Packets. This allows the Command Packet stream to be paired to a specific Data Packet Stream, or associated with another Command Packet stream. V49.2 also introduces the concept of a Control Point. A Control Point is the point in the architecture to which control is being sent. Each unique Control Point has a unique Stream ID.

RULE 3.26: The Stream ID for a Command Packet SHALL match the Stream ID for the specific Control Point being controlled or the Stream ID for the related Data Packet stream.

OBSERVATION 3.21: ODI-2 specifies that Data Packet Stream IDs must be programmable by the user. Since related Command Packets must share the same Stream ID, they must also be programmable to the same value.

OBSERVATION 3.22: The default Stream ID for a device is 4096, the same as for Data packets.

3.3.3 Command Packet Structure – Class ID

The Class ID identifies the Information Class and Packet Class to which the packet stream belongs.

RULE 3.27: If a Command Packet is paired with a Data Packet stream, then the Class ID for a Context Packet SHALL match the Class ID for the related Data Packet stream.

3.3.4 Command Packet Structure – Timestamps

The Timestamp Fields of a Command Packet function in the same way as for Data Packets.

RULE 3.28: If a Command Packet is paired with a Data Packet stream, then the Timestamp fields of the Context Packet SHALL match the timestamp fields for the related Data Packet stream.

RECOMMENDATION 3.3: The Class ID fields of an Acknowledge packet SHOULD match that used in the corresponding Control packet.

3.3.5 Command Packet Structure – CAM (Control/Acknowledge Mode)

V49.2 summarizes the CAM field in the paragraph below:

“The Control/Acknowledge Mode field bits defines how and when the recipient of a Control packet should react to the controls, whether or not an Acknowledge packet or packets should be generated, and how warnings and errors should be handled. This includes the ability to designate a Control packet as a “dryrun” and settings on how to handle potential problems with the packet Control fields.”

Much of this capability is beyond the scope of FPGA-based devices. Though ODI-2 documents the full capability of the Command Packet type, ODI-2 recommends only the use of the Control Packet subtype, and that consumers be forgiving listeners.

The Rules and Recommendations of the CAM field can be found in the CAM section under each packet type.

3.3.6 Command Packet Structure – Message ID

Message ID provides a numeric value for directly associating an Acknowledge Packet response with its associated Control Packet. Also, Message ID together with Stream ID provides a unique identifier for cancelling controls. Though acknowledgements and cancellations are not expected to be common in ODI systems, the Message ID field remains mandatory, allowing their use.

RULE 3.29: Each new Control Packet SHALL have a unique Message ID.

OBSERVATION 3.23: A unique Message ID may be created by a counter that is incremented for each Control Packet.

3.3.7 VRT Control Packets

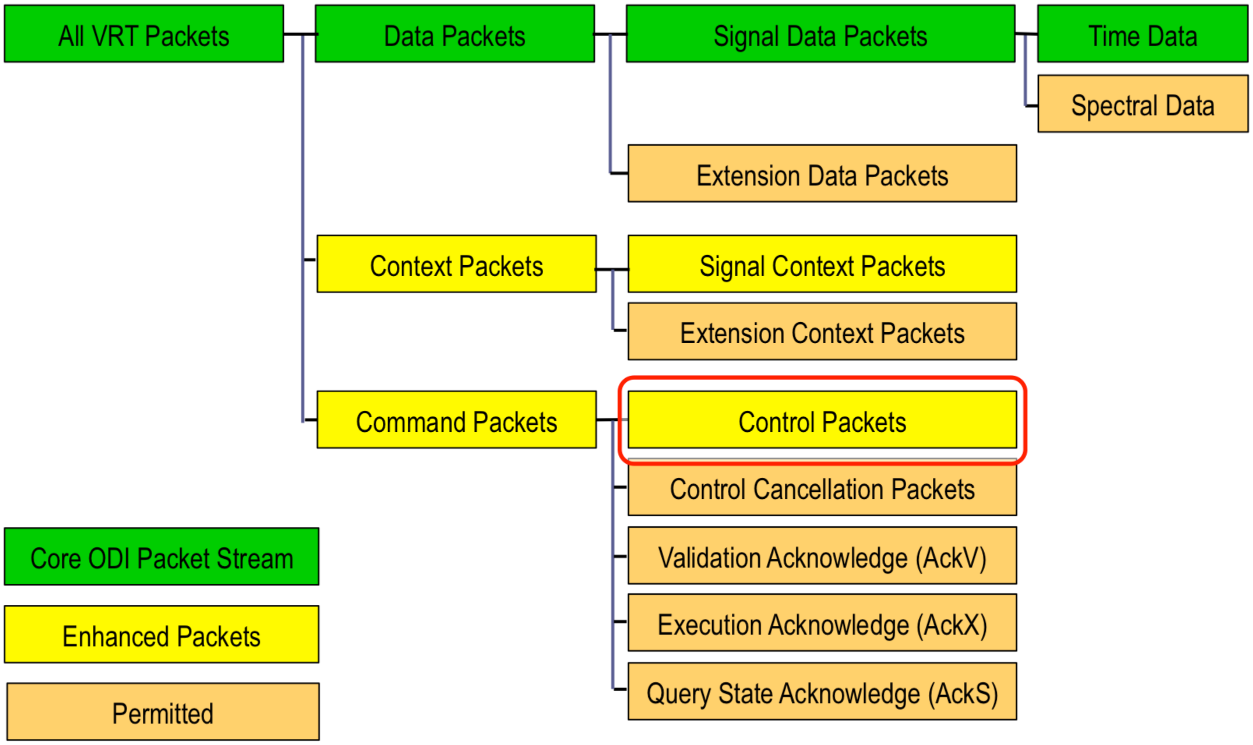


Figure 3-27: VRT Control Packet

The Control Packet is the most basic of the Command Packet subtypes, and commands a consumer to execute metadata control fields. It may be the sole Command Packet subtype in a system, and is the only Command Packet subtype that can be used alone.

Control Packets are indicated by the setting of Command Packet Header bits A (26) and L (24) to zero.

Like the Context Packet, the Control Packet includes indicator fields (designated as Control Indicator Field(s), or CIF(s)) to the packet body to indicate which control fields are present in the packet. For the most part, the Context Indicators Fields and the Control Indicator Fields have equivalent metadata descriptions.

The Control Packet uses the Control/Acknowledge Mode (CAM) field to designate how and when a Controllee is to put the Controls into effect, and how a Controllee is to acknowledge the controls sent from the Controller. Through the mode field, a Controller can specify how a Controllee executes controls, how it responds to query, how it reports the state of execution, or simply its evaluation of the Control Packet for correctness without any execution at all. Much of this capability is beyond the scope of FPGA-based devices. Additionally, Acknowledge Packets require a reverse optical link to be present. For this reason, ODI-2 recommends Controllers send simple execution commands via Control Packets and consumers be forgiving listeners. ODI-2 recommends specific CAM settings within the Control Packet that matches this simplified behavior.

3.3.7.1 Control Packet – CAM field

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Ctrl											Ctrl & Ack									Ack Only				Reserved							
	CE	IE	CR	IR	P	W	Er	A1	A0	Nack	Res	ReqV	ReqX	ReqS	ReqW	ReqEr	Res	CtrlT2	CtrlT1	CtrlT0	0	0	0	0	0	0	0	0	0	0	0	0
RULE:	0	0	0	0	x	x	x	x	x	x	0	x	x	x	x	x	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0
RECOMMENDATION:	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-28: VRT Control Packet – CAM field

The CAM field of Control Packet is shown in the above figure, along with the ODI-2 rules and recommendations of each bit setting. Rules show the allowable values for the CAM bit parameter, while the recommendations show the settings for a simple control environment that consists of only the Control Packet subtype along with forgiving Controllees.

RULE 3.30: All VRT Control Packets SHALL include a CAM that complies with the definitions given in Figure 3-28.

RULE 3.31: All VRT Control Packets SHALL include a CAM that complies with the values given in the RULE row in Figure 3-28.

CE (31) ControlleE enable. This bit is set to zero in ODI, indicating there is no optional Controllee identifier field in the Prologue.

IE (30) Identifier format – controlleE. This bit is set to zero in ODI-2. It is used to identify the format of the Controllee identifier field, which is not used in ODI-2.

CR (29) ControlleR enable. This bit is set to zero in ODI, indicating there is no optional Controller identifier field in the Prologue.

IR (28) Identifier format – controlleR. This bit is set to zero in ODI-2. It is used to identify the format of the Controller identifier field, which is not used in ODI-2.

P (27) Partial packet implementation permitted. Setting the P bit to 1 indicates that the Controllee should execute any fields that don't have any errors or warnings. Setting this bit to 0 indicates not to execute any fields at all if there are any issues that cannot be overcome by modifying problematic fields. ODI-2 recommends setting P to 1 as part of a forgiving Controllee behavior.

RULE 3.32: The P (27) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.4: The P (27) bit of the CAM field SHOULD be set to 1, allowing partial execution.

W (26) Warnings. Setting the W bit to 1 allows the execution of fields that generate warning, while setting the W bit to 0 does not permit the execution fields that generate warnings. ODI-2 recommends setting W to 1 as part of a forgiving Controllee behavior.

RULE 3.33: The W (26) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.5: The W (26) bit of the CAM field SHOULD be set to 1, allowing the executing of fields that generate warnings.

Er (25) Errors. Setting the Er bit to 1 allows the execution of fields that generate errors, while setting the Er bit to 0 does not permit the execution fields that generate errors. ODI-2 recommends setting Er to 1 as part of a forgiving Controllee behavior.

RULE 3.34: The Er (25) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.6: The Er (25) bit of the CAM field SHOULD be set to 1, allowing the executing of fields that generate errors.

A1 and A0 (24, 23) Action Bit Field. The Action bits are set to indicate three modes: No-Action, Dry Run, or Execute Mode. Execute Mode is the operational mode of the device, and is recommended in ODI-2. Its value is 10.

RULE 3.35: The A1 and A0 (24, 23) bits of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.7: The A1 and A0 (24, 23) bits of the CAM field SHOULD be set to 10, directing the Controllee to implement the commands.

Nack (22) Not-acK only. When Nack is set to 1, the Controllee provides AckV and/or AckX variants of Acknowledge Packet ONLY when warnings or errors have occurred, as per the AckV and AckX bits. When Nack is set to 0, the Controllee provides AckV and/or AckX Packets in all cases when requested per the AckV and AckX bits.

RULE 3.36: The Nack (22) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.8: The Nack (22) bit of the CAM field SHOULD be set to 0.

Reserved (21) bit. This bit is Reserved and set to 0.

ReqV (20) Request Validation Acknowledge Packet. Setting the ReqV bit to 1 requests a Validation Acknowledge Packet from the Controllee, while setting it to 0 does not request the Validation Acknowledge Packet. ODI-2 recommends setting the bit to 0 to simplify Controllee design.

RULE 3.37: The ReqV (20) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.9: The ReqV (20) bit of the CAM field SHOULD be set to 0.

ReqX (19) Request Execution Acknowledge Packet. Setting the ReqX bit to 1 requests an Execution Acknowledge Packet from the Controllee, while setting it to 0 does not request the Execution Acknowledge Packet. ODI-2 recommends setting the bit to 0 to simplify Controllee design.

RULE 3.38: The ReqX (19) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.10: The ReqX (19) bit of the CAM field SHOULD be set to 0.

ReqS (18) Request Query-State Acknowledge Packet. Setting the ReqS bit to 1 requests a Query-State Acknowledge Packet from the Controllee, while setting it to 0 does not request the Query-State Acknowledge Packet. ODI-2 recommends setting the bit to 0 to simplify Controllee design.

RULE 3.39: The ReqS (18) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.11: The ReqS (18) bit of the CAM field SHOULD be set to 0.

ReqW (17) Request Warning Acknowledge Packet. Setting the ReqW bit to 1 requests a Warning Acknowledge Packet from the Controllee, while setting it to 0 does not request the Warning Acknowledge Packet. ODI-2 recommends setting the bit to 0 to simplify Controllee design.

RULE 3.40: The ReqW (17) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.12: The ReqW (17) bit of the CAM field SHOULD be set to 0.

ReqEr (16) Request Error Acknowledge Packet. Setting the ReqEr bit to 1 requests an Error Acknowledge Packet from the Controllee, while setting it to 0 does not request the Error Acknowledge Packet. ODI-2 recommends setting the bit to 0 to simplify Controllee design.

RULE 3.41: The ReqEr (16) bit of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.13: The ReqEr (16) bit of the CAM field SHOULD be set to 0.

Ctrl T2-T0 (14-12) Timing Control. These bits govern the timing behavior of controls in the presence of timestamps. V49.2 offers very robust capabilities. Unless the timing controls are needed, ODI-2 recommends the “Ignore Timestamp” mode. In this case, it is at the discretion of when the control fields are executed. The “Ignore Timestamp” mode is indicated by the T2, T1, T0 bits being set to 000.

RULE 3.42: The Ctrl T2-T0 (14-12) bits of the CAM field SHALL be set as prescribed in V49.2.

RECOMMENDATION 3.14: The Ctrl T2-T0 (14-12) bits of the CAM field SHOULD be set to 000, indicating “Ignore Timestamp” mode.

Acknowledge Bits (11-8). These bits are set to zero in Control Packets. They are only used in Acknowledge Packets.

Reserved (7-0). These bits are reserved and set to 0.

3.3.8 VRT Control Cancellation Packets

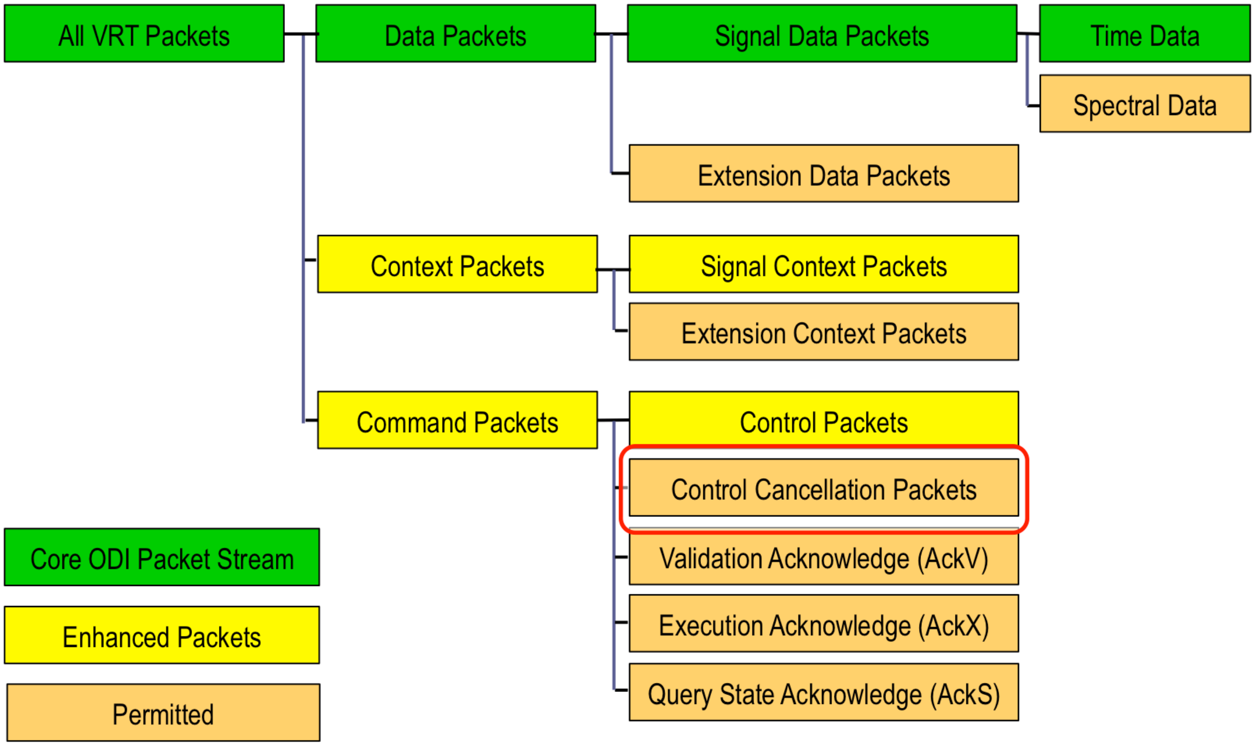


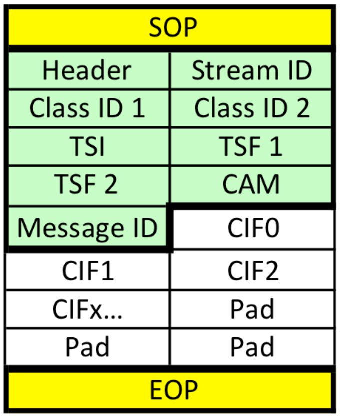
Figure 3-29: VRT Control Cancellation Packet

The Control Cancellation Packet subtype is used to interrupt the implementation of scheduled but not yet executed controls. Due to the speeds involved and widespread implementation of ODI devices using FPGAs, Control Cancellation is not recommended.

Control Cancellation Packets are indicated by the setting of Command Packet Header bits A (26) to 0, and L (24) to 1.

RULE 3.43: The Control Cancellation Packet SHALL behave as prescribed in V49.2.

RECOMMENDATION 3.15: Due to the complexity and speed involved, Controllers SHOULD NOT use Control Cancellation Packets.



SOP	
Header	Stream ID
Class ID 1	Class ID 2
TSI	TSF 1
TSF 2	CAM
Message ID	CIF0
CIF1	CIF2
CIFx…	Pad
Pad	Pad
EOP	

- The structure of the Control Cancellation Packet is the same as the Control Packet EXCEPT it does NOT have fields after the CIFs.
- In the diagram to the left this means only CIFx fields are in the Control portion, and any remaining fields will be null to meet the multiple of 32-byte length requirement.

Figure 3-30: VRT Control Cancellation Packet Structure

The figure above shows the structure of the Control Cancellation Packet subtype. Control Cancellation Packets have the same structure as Control Packets except that they do not have any fields after the CIFs in the Control Section. Due to the multiple of 32-byte packet length rule, null data will be inserted into any remaining fields necessary to bring the entire packet to a length divisible by 32 bytes.

3.3.9 VRT Control Acknowledge Packets

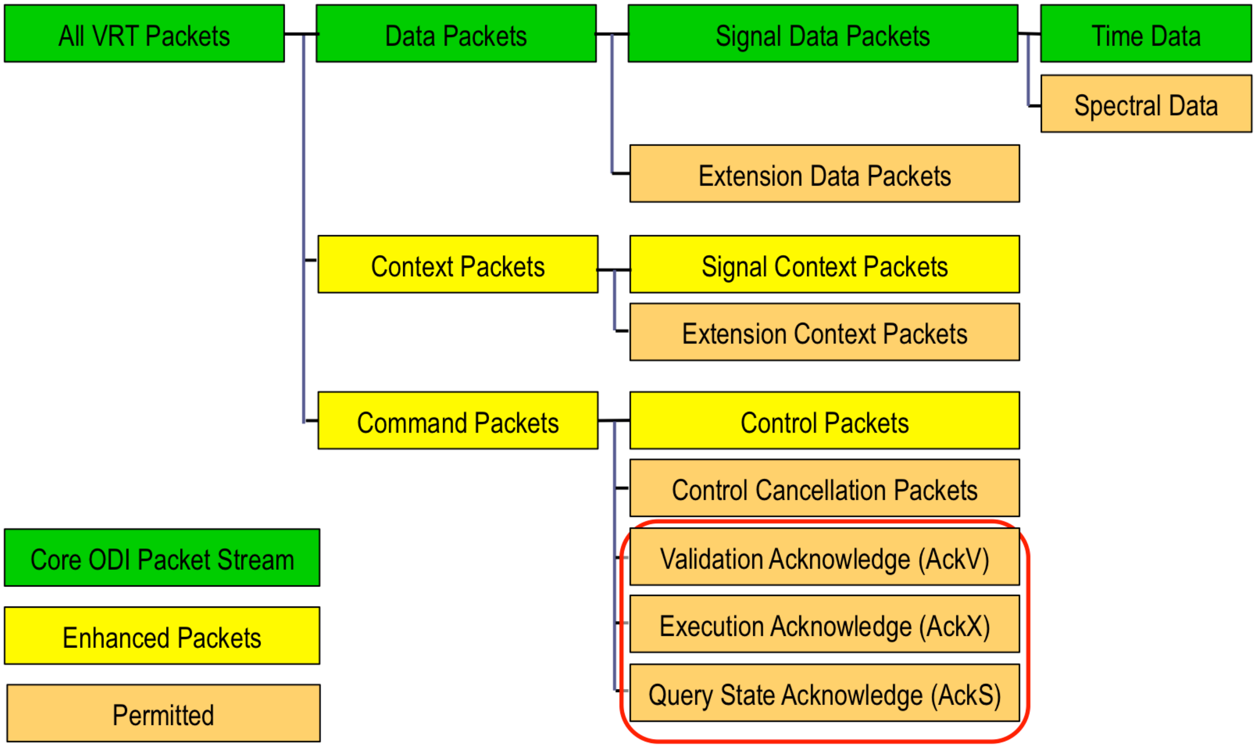


Figure 3-31: VRT Control Acknowledge Packets

Control Acknowledge Packets are sent from the Controllee to the Controller. This may be implemented in an ODI system by a bi-directional link. Due to the complexity involved, ODI-2 recommends avoiding Control Acknowledge Packets unless necessary for the application.

Acknowledge Packets are indicated by the setting of Command Packet Header bit A (26) to 1. An Acknowledge Packet may be only of subtype, a Validation Acknowledge, an Execution Acknowledge, or a Query-State Acknowledge. This is indicated by having only one bit out of the three AckV, AckX, and AckS bits set to 1. Acknowledge Packets use the Message ID field to indicate which controls are being reported, matching its Message ID to the Controller's Message ID field for the commands sent.

RULE 3.44: The Control Acknowledge Packet SHALL behave as prescribed in V49.2.

RECOMMENDATION 3.16: Due to the complexity and speed involved, Controllees SHOULD NOT use Control Acknowledge Packets.

3.3.10 VRT Extension Command Packets

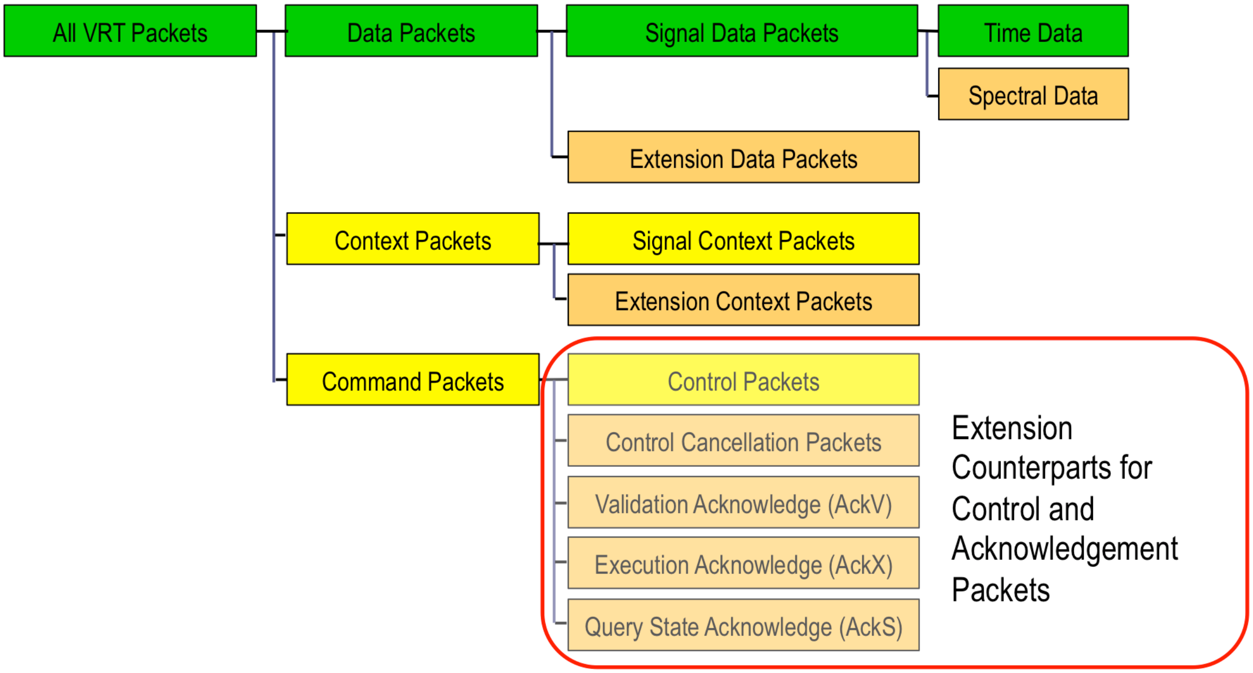

All VRT Packets
Data Packets
Signal Data Packets
Time Data
Spectral Data
Extension Data Packets
Context Packets
Signal Context Packets
Extension Context Packets
Command Packets
Control Packets
Control Cancellation Packets
Validation Acknowledge (AckV)
Execution Acknowledge (AckX)
Query State Acknowledge (AckS)
Extension Counterparts for Control and Acknowledgement Packets

Figure 3-32: VRT Extension Command Packets

Extension Command Packets are intended to be used to convey both controls and acknowledge packets for which no provision has been made in the defined Control Packet and Acknowledge Packet structure and/or fields. They are indicated by the Packet Type of 0111 in the Header field, as shown in the figure below. Note that Bit 28 is set to “1”, indicating an Extension Context Packet. It is not recommended to use Extension Command Packets unless it is required by the application.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Packet Type				C	Indicators			TSI		TSF		Packet Count				Packet Size															
0	1	1	1	C	A	R	L	TSI		TSF		Pkt Count				Packet Size															

Figure 3-33: VRT Extension Command Packet Header

RULE 3.45: AN ODI-2 device that generates Extension Command Packets SHALL comply with all Rules specified for Command Packets, as documented in section 3.3, other than the Packet Type value.

OBSERVATION 3.24: The mandated specified fields for the Command Packet Prologue are Header, Stream ID, Class ID 1, Class ID 2, TSI, TSF 1, TSF 2, CAM, and Message ID. These are also the mandated fields for an Extension Command Packet.

4. ODI Port Aggregation

ODI-2 Port aggregation is a method that uses multiple ports to send synchronous data at higher speeds than can be achieved with a single port. It is an optional capability. Port aggregation does not include using multiple ports to send independent asynchronous data streams. That ability is already allowed in ODI-1, and does not need any further specification.

There are two key use cases for port aggregation:

1. Transporting single channel data whose aggregate bandwidth is beyond the single port bandwidth.
2. Transporting multi-channel data whose aggregate bandwidth is beyond single port bandwidth.

ODI-2 uses the uses the VRT packet structure, coupled with Interlaken SOP (Start of Packet command), to synchronize data across multiple ports. There is no theoretical limit to the number of ports that can be aggregated, but four ports is a feasible number.

ODI-2 uses a per-port flow control method to aggregate ports where flow control is needed.

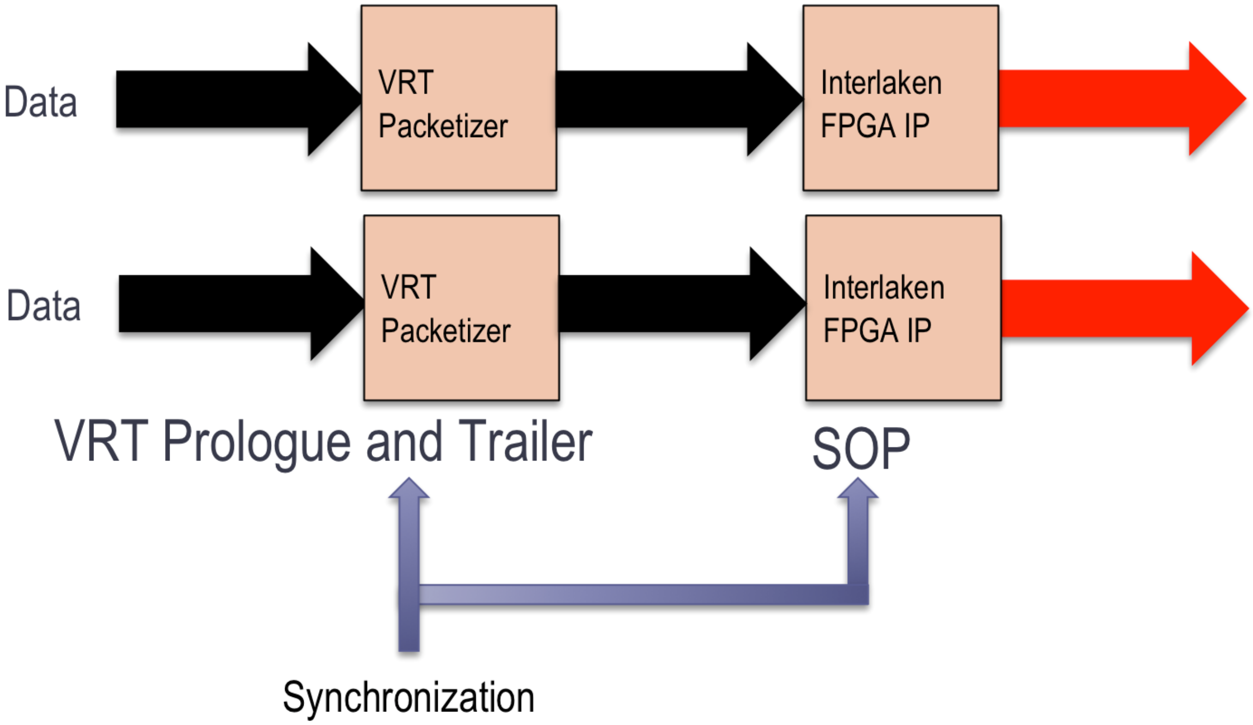


Figure 4-1: Port Aggregation and Synchronization

Figure 4-1 shows a simplified diagram of an ODI device performing port aggregation on two ports. Synchronization occurs by sending equal-sample-length VRT packets simultaneously. The transmission of each packet starts at the same time, indicated by an Interlaken SOP command. Using this method EOP (End of Packet) may not occur on all ports simultaneously, but often does. The VRT packetizer consists of inserting the 28-byte Prologue and 4-byte Trailer around the payload data, as described earlier in ODI-2. This method guarantees that the sample data from the same period of time is transported across all ports.

Equal-sample-length Data Packets are packets where the number of samples per channel are the same. For example, if Port 1 has two channels, each with 1024 samples in a Data Packet, and Port 2 has three channels, each with 1024 samples in a Data Packet, then Port 1 will have 2048 total samples in each Data Packet, and Port 2 will have 3072 samples in each Data Packet. However, they each report the values over the same time period. Port 1 merely reports two channels and Port 2 reports three.

While the ideal situation is that the Interlaken SOPs and packet transmission are perfectly simultaneous, this will not be the case. Designers interface with the Interlaken IP on each of the FPGAs, but the FPGAs themselves have significant latencies, and the differential latency may be significant. Because of this, ODI-2 specifies the timing requirements for interfacing with the FPGA IP for the producers, and takes differential latency into account when specifying the timing requirements for the consumers.

RULE 4.1: ODI-2 producers that implement port aggregation SHALL send Interlaken SOP signals to the FPGA IP on all aggregated ports within 5 nanoseconds of each other.

OBSERVATION 4.1: RULE 4.1 specifies the internal timing requirement of sending SOP to each port. Feasible implementations will use hardware signals sent to all ports nearly simultaneously, which will be well under 5 nanoseconds. Due to the differential latency of each port, the actual SOPs may occur more than 5 nanoseconds apart at the ODI port outputs.

Section 3.1.2 previously required that ports being aggregated each have a unique Stream ID, and each Stream ID is incremented by 1024 for each subsequent port. Therefore, the default Stream IDs for a 4-port aggregation are:

Port 1:	4096
Port 2:	5120
Port 3:	6144
Port 4:	7168

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 on each port, and increment the Packet Count for each subsequent packet from that port.

OBSERVATION 4.2: The above rule allows the consumer to align packets from one port correctly to those from another port.

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.

4.1 Single signal channel port aggregation

Single signal channel port aggregation is defined as aggregating ports to support the transport of a single channel of data. This may occur with high-speed digitizers, as shown in the figure below.

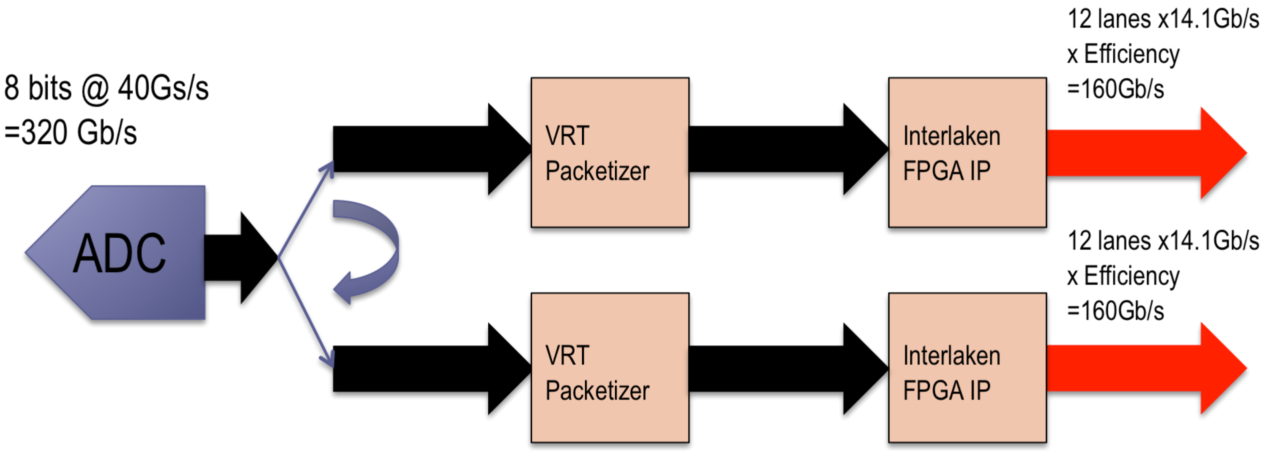


Figure 4-2: Single-channel port aggregation example - Transmission

Figure 4-2 shows a hypothetical example of a high-speed digitizer requiring the bandwidth of two ports. In the example, the digitizer is based on a single-channel 8 bit ADC (analog-to-digital converter) operating at 40Gsamples/s. The aggregate data bandwidth is 80 GBytes/s, requiring two ODI ports.

Samples from a single channel are sent in a round robin fashion to each port, and packaged in a VRT packet at each port. Interlaken SOP is sent on each port before the beginning of the VRT packet. The very first sample goes to Port 1, the next to Port 2, and so forth before starting at Port 1 again. This is performed until entire equal length packets are sent from each port, complete with Interlaken EOP commands, and then the next simultaneous packets are sent. The example shows two-port aggregation, but any number of ports may be aggregated.

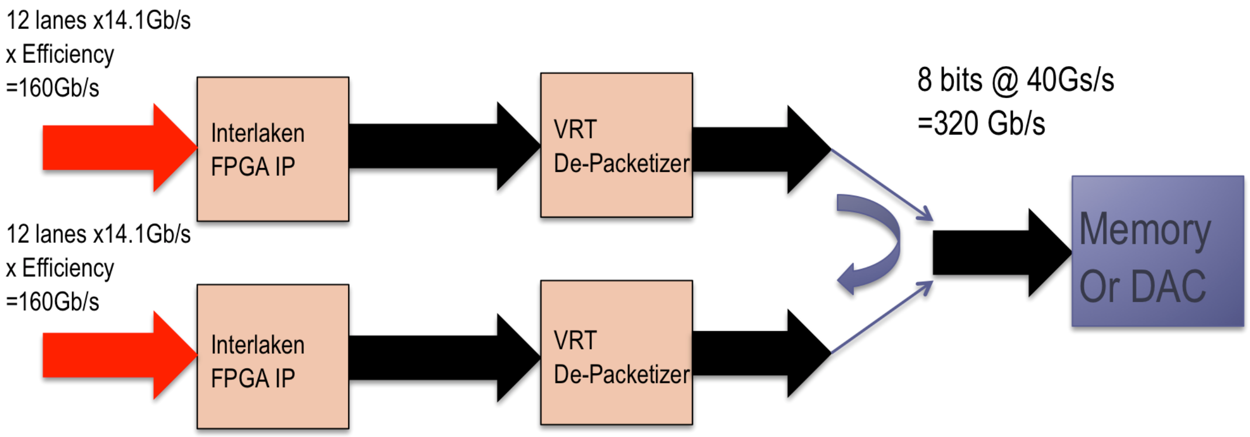


Figure 4-3: Single-channel port aggregation example – Reception

Figure 4-3 shows the reception of the same data stream transmitted in Figure 4-2. Essentially, the Data Payload of each VRT packets is extracted from each port, and the data interleaved again to form the original high-speed stream. Here, the original data stream is created again by starting with the first sample from Port 1, then interleaving the next sample from Port 2, and so forth, in a similar round robin fashion to transmission. This is performed until entire equal length packets are received from each port, complete with Interlaken EOP commands, and then the next simultaneous packets are received. The example shows two-port aggregation, but any number of ports may be aggregated.

RULE 4.4: ODI-2 producers that implement port aggregation for a single signal channel SHALL send the same number of data samples per packet across all ports.

OBSERVATION 4.4: The above rule ensures straight-forward interleaving of packets and their Data Payloads.

4.2 Multiple signal channel port aggregation

Multiple signal channel port aggregation is defined as aggregating ports to support the transport of multiple synchronous signal channels. This may occur when each port has the capacity to support one or more channels on each port, but not enough capacity to support all signal channels combined. In this case, some signal channels are sent with Port 1, some with Port 2, and so forth. The figure below shows an example.

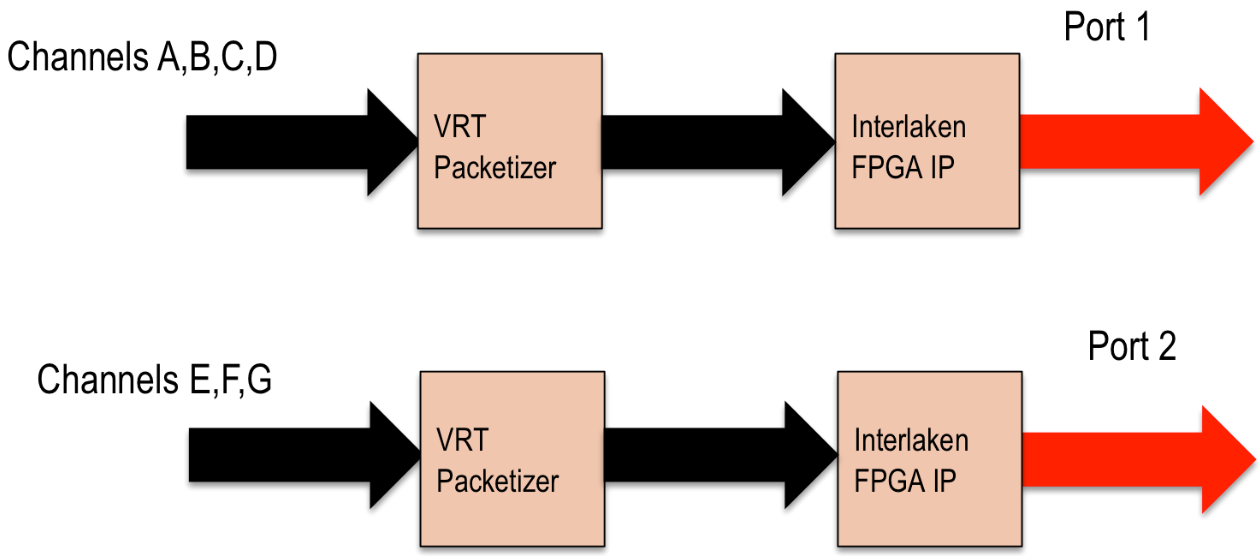


Figure 4-4: Multi-channel port aggregation example – Transmission

In the figure above, four signal channels are assigned to Port 1, and three signal channels are assigned to Port 2. This example shows that the packet sizes don't have to be the same on all ports, but the number of samples/channel within each packet do. Since Port 1 has four channels, and Port 2 has three channels, there will be 33% more data within a packet from Port 1 than from Port 2. However, there will be the same number of samples per channel on all ports and within all packets.

The beginning of all packets start at roughly the same time, indicated by an Interlaken SOP command. EOP may not occur simultaneously, but must occur before or coincident with the next SOP.

RULE 4.5: ODI-2 producers that implement port aggregation for a multiple signal channels SHALL send the same number of data samples per channel per packet across all ports.

OBSERVATION 4.5: The above rule ensures that the number of samples for each channel will be the same across all packets being sent simultaneously. By definition, it also ensures that each simultaneous packet covers the same time period.

PERMISSION 4-1: An ODI-2 producer MAY include more signal channels on one port than another.

OBSERVATION 4.6: The above permission reflects the fact that the number of channels may not be evenly divisible by the number of ports.

RECOMMENDATION 4.1: If the number of channels is evenly divisible by the number of ports in a multiple channel port aggregation application, then the producer SHOULD assign the same number of channels to each port.

OBSERVATION 4.7: The above permission reflects the fact that the number of channels may not be evenly divisible by the number of ports.

OBSERVATION 4.8: All ports can be made to support the same number of channels by adding "dummy channels" containing null data to the ports that have fewer channels.

4.3 Port aggregation with flow control

Flow control is needed in any application where the consumer is pacing the timing of the samples. This is most common when an AWG (arbitrary waveform generator) or other type of signal generator uses flow control to keep the average rate of data from the producer, which could be a storage device, to match its own sampling rate. Flow control has been defined earlier in ODI-1 on a single port basis. There are two methods, In-Band and Out-of-Band. Both rely on XON/XOFF signals being sent from the consumer to the producer, either by a reverse Interlaken link, or an explicit electrical signal.

Port Aggregation uses a per-port method of flow control. That is, each port is controlled with separate XON/XOFF signals as with the single port flow control model.

RULE 4.6: During port aggregation all ports using flow control SHALL use either In-Band flow control or Out-of-Band flow control, but not a mixture of the two.

OBSERVATION 4.9: Out-of-Band flow control requires an electrical signal for each port. Therefore a 4-port device would require four electrical signals. PXI and AXIe support 8 and 12 trigger lines respectively, and these lines can be designated to be the OOB flow control lines. It is possible to define an ODI system that requires more flow control lines than the number of backplane trigger lines in a modular instrument system.

RECOMMENDATION 4.2: Devices that support OOB flow control SHOULD include an explicit OOB signal for each port.

OBSERVATION 4.10: The above recommendation allows direct connections between devices without the use of modular backplane signals. This is useful for connecting non-modular devices together, or for expanding the OOB flow control capacity of a modular system.

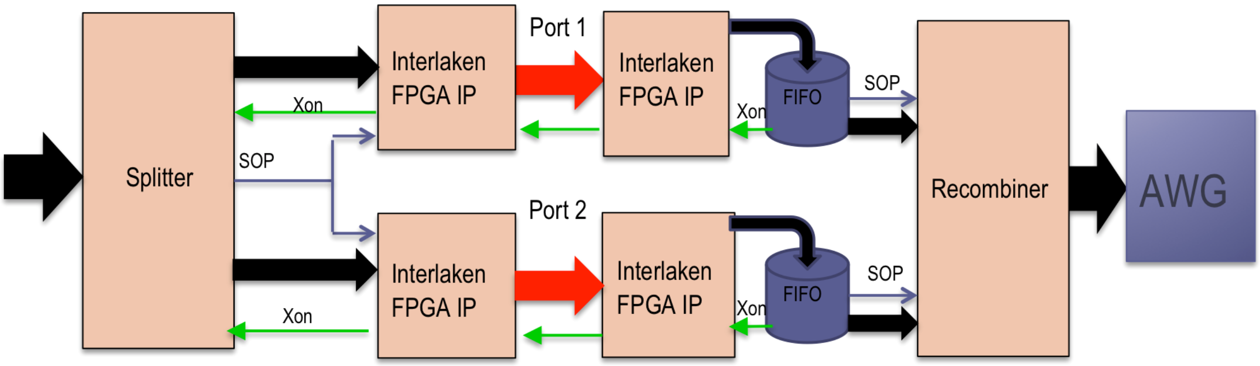


Figure 4-5: Port Aggregation – Flow Control per port

In the figure above, the entire data path from one device to another is shown. The example is a two-port device using flow control. The data is generated at the far left. It is then divided into the two port streams to be transmitted. The data itself may just be storage, or it could be real-time generated data. Since both streams are VRT packets, the data may be already stored as two parallel streams.

The Splitter function manages the port aggregation on the producer side by sending an Interlaken SOP command to both FPGAs as it manages the data to each FPGA. Each FPGA, using the Interlaken IP, serializes the data over the 12 high-speed SERDES that lead to the optics. It is then transmitted across standard ODI cables.

On the receiving end are two FPGAs that deserialize the optical data, and rebuild the Data Payload of the packet for each port. This data is fed into a FIFO buffer, as described in ODI-1 flow control. The FIFO acts the same as it does in single port mode. When it reaches the XOFF level, it sends the XOFF command over the flow control medium to stop the transmission of data. When it drops to the XON level, it will send XON to start data flowing into the FIFO again.

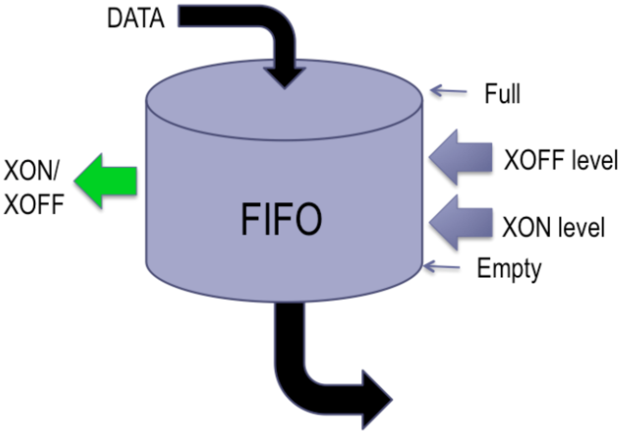


Figure 4-6: Consumer FIFO Input Buffer

The XON/XOFF signal is sent back to the producer to modulate the data on a per-port level. This may either be done with an In-Band signal or an Out-of-Band signal, but must be done the same way across all ports. This is shown as the green arrows leading back to the producer.

The Recombiner manages the port aggregation on the consumer side. It brings the two packet streams together. By searching for the SOP signal from each stream, it is able to align the packets. At this time the Recombiner may either interleave the data for single channel operation, or synchronize the channels for multi-channel operation.

In order to remain aligned, even in the face of a temporary line outage, both the Splitter and the Recombiner must obey certain procedures.

For the Splitter, the procedure is:

1. Wait for XON from all ports before sending out packets.
2. Send SOP on all ports, followed by the packet data.
3. Wait for all ports to finish sending packets.
4. Go to Step 1.

For the Recombiner, the procedure is:

1. On each port, read and discard FIFO words until the next word on all ports has SOP.
    a. Discard any unexpected words without SOP.
    b. After SOP occurs on any single port, allow up to Ts seconds for SOP to appear on all ports.
    c. If SOP words occur on all ports before Ts, Go to Step 2.
    d. If SOP words do not occur on all ports before Ts, Return to Step 1.
2. Read SOP word from all ports. Verify Packet Count field in the V49 Header is the same for all ports. If not, Return to Step 1.
3. Continue reading words from all ports, recombining data, until EOP is received on all ports. If any port reports error (CRC) or SOP appears before EOP, then mark that packet as bad.
4. Go to Step 1.

Ts, which is the time allowed for synchronization. Ts is chosen to be long enough to compensate for the variances in latency, but short enough to detect an error in synchronization. It is executed by the consumer, which chooses a default value typically shorter than the Data Packet length of a Train packet, but longer than the maximum time skew between ports.

RULE 4.7: During port aggregation, the Splitter function of the producer SHALL execute the 4-step procedure outlined in Section 4.3.

RULE 4.8: During port aggregation, the Combiner function of the consumer SHALL execute the 4-step procedure outlined in Section 4.3.

RECOMMENDATION 4.3: Consumers implementing port aggregation with flow control SHOULD make Ts programmable.

OBSERVATION 4.11: The above recommendation allows the adjustment of Ts in different system implementations.

4.4 Port aggregation with Context and Command Packets

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.

The ODI port aggregation rules compel Context and Command Packets to be sent across all ports due to the above rule. What is contained in those packets varies between single-channel port aggregation and multi-channel port aggregation.

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.

Packet being sent across all ports will contain the same content. That is, though the Stream IDs are different to match the Stream ID on each port, the remainder of the fields are the same.

RULE 4.11: During single-channel port aggregation if a producer sends a Context Packet or a Command Packet on one port, it SHALL send the same packet on all ports, with only the Stream ID being different.

For multi-channel port aggregation, the procedure is different. In this case, each port is referencing a different signal stream and different signals altogether, and the Context or Command Packets need to be tailored to the specific channels on each port. In this case, a similar packet is sent to all ports, but the fields may be different, reflecting the Context or Control related to those signals. This is specified in RULE 4.9.

OBSERVATION 4.13: The required field of CIF0 in all Context and Command Packets includes the Context Field Change Indicator bit 31. This allows Context or Command Packets being sent to explicitly mark that whether there is a change being reported or requested. This is a useful field during multi-channel port aggregation because it allows aligned Context and Command Packets to be sent over ports without initiating action.

PERMISSION 4-2: During port aggregation of multi-channel signals, an ODI-2 producer may send Context or Command Packets with the Context Field Change Indicator set to 0 in order to meet Rule 4.9.

4.5 Port aggregation state diagrams

Producer port aggregation without flow control.

Below is the state diagram for a producer executing port aggregation without flow control. The left series of states are for Port 1, the right series of states is for Port 2. Functions in the center are the combined functions of the device, called the Splitter.

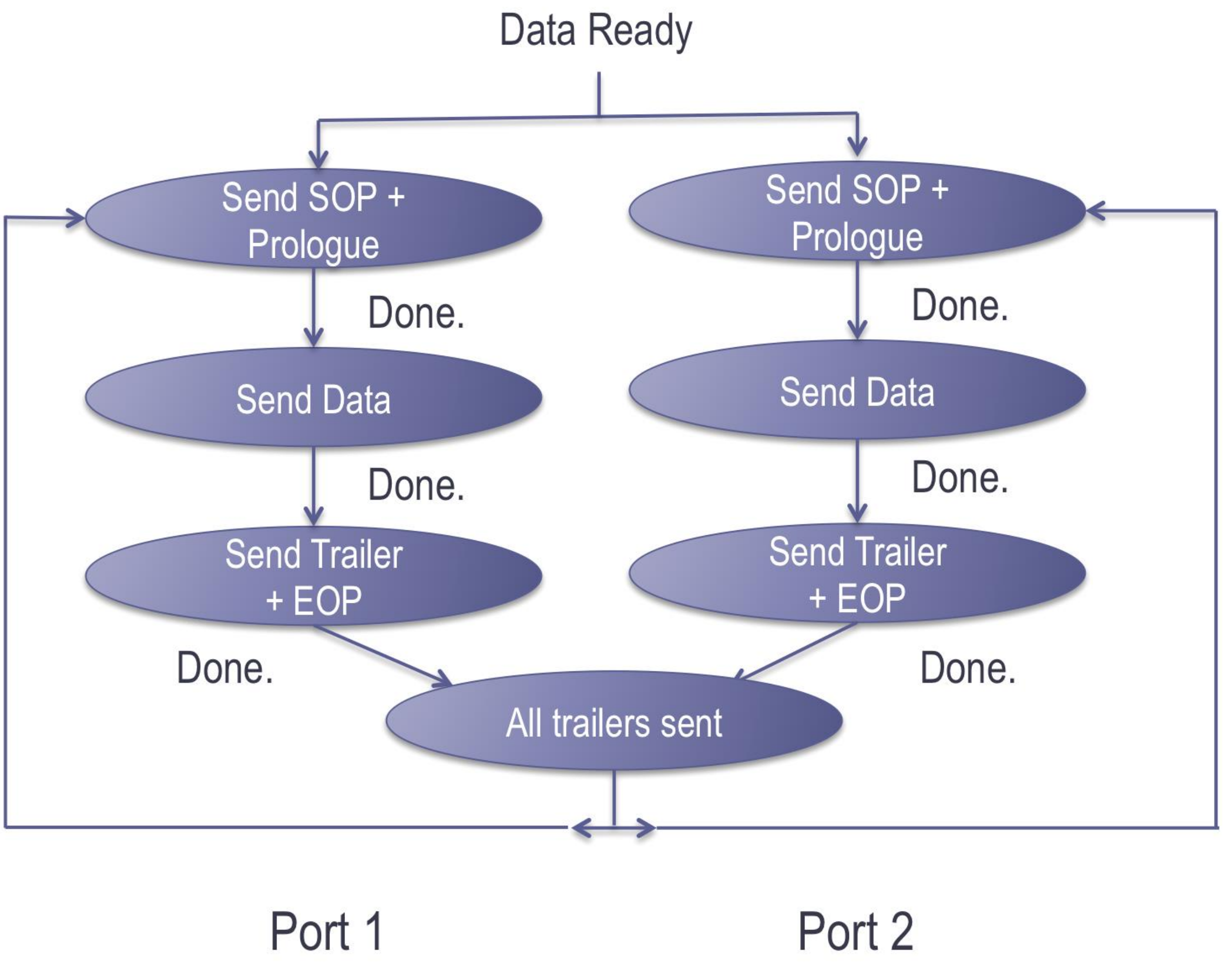


Figure 4-7: State Diagram: Producer port aggregation without flow control.

As described earlier, the SOP and Prologue are sent nearly simultaneously on all ports. The producer waits until all EOP is commanded to be sent on all packets before starting again. The process loops indefinitely until a command to stop the acquisition is given.

Producer port aggregation with flow control.
Below is the equivalent state diagram for the producer, but with flow control. The key difference is that the producer Splitter functionality does not initiate a new packet unless each port's flow control is indicating XON. Once all ports indicate XON, the producer will initiate a new packet.

Not shown is the flow control functionality during a packet transmission. This works the same way as in the single port case of ODI-1, with XON/XOFF modulating whether the producer is sending data or not.

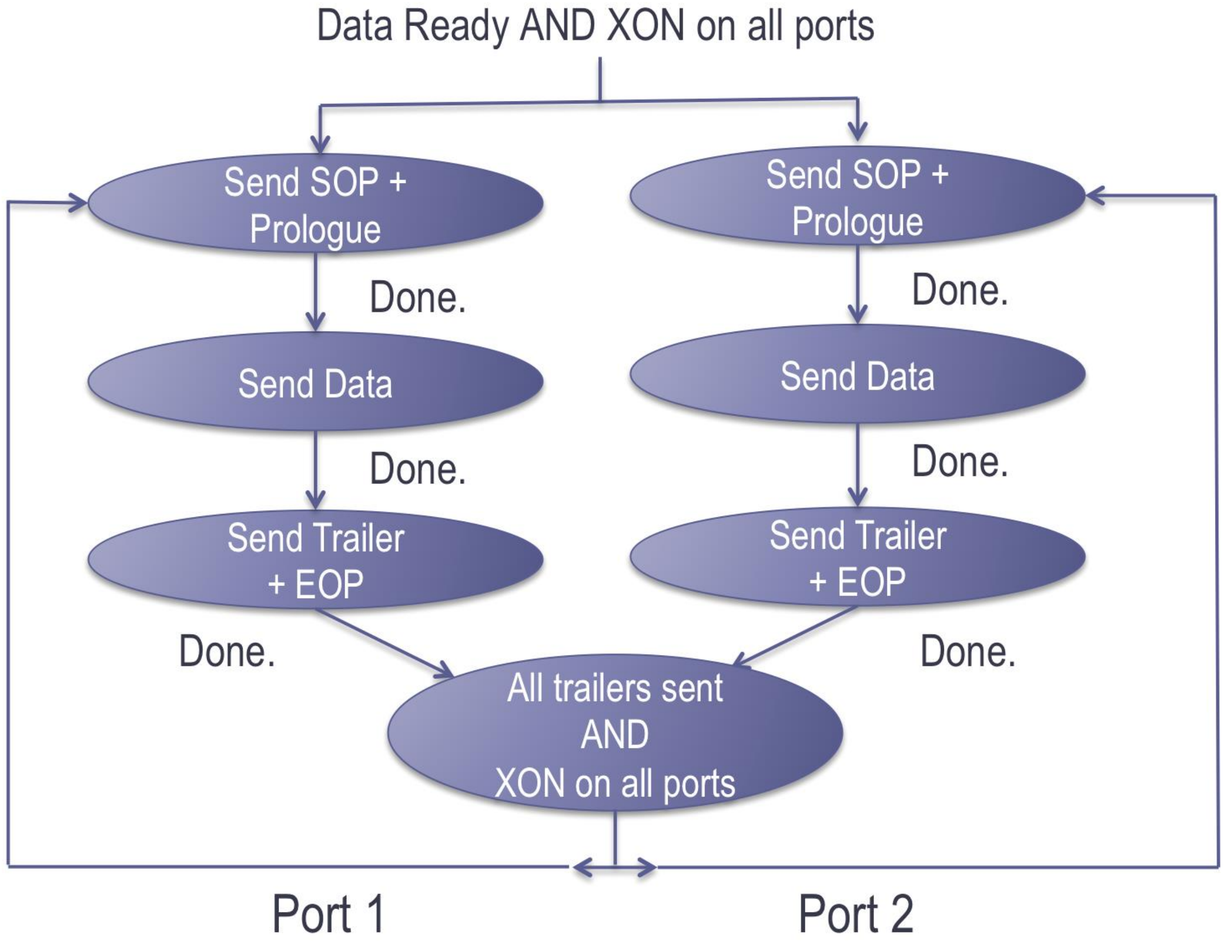


Figure 4-8: State Diagram: Producer port aggregation without flow control.

Consumer port aggregation.
Below is the state diagram for a consumer executing port aggregation. The left “Handle Packet Data” is for Port 1, the right “Handle Packet Data” is for Port 2. Functions in the center are the combined functions of the device, called the Combiner.

The flow chart is irrespective of with flow control or without flow control. The consumer generates XON/XOFF flow control signals as specified in ODI-1. The “Handle Packet Data” is the equivalent to the state with the same name in the ODI-1 Rcv Packets State Diagram. As specified in ODI-1, if while handling packet data an SOP occurs without an EOP occurring earlier or coincident than the SOP, then an error has occurred. In this case, the Combiner goes to the initial state to restart the process.

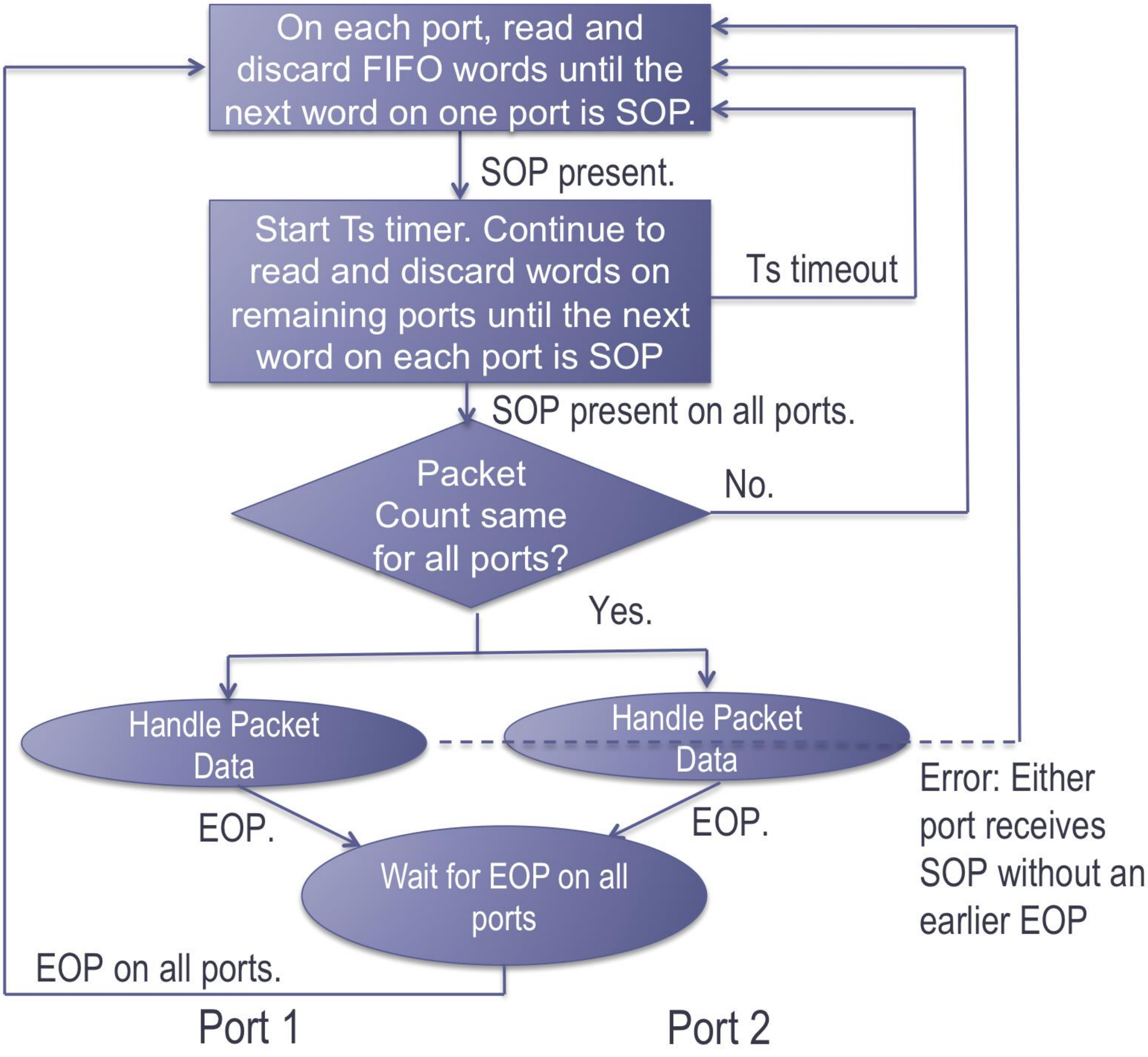

On each port, read and discard FIFO words until the next word on one port is SOP.
SOP present.
Start Ts timer. Continue to read and discard words on remaining ports until the next word on each port is SOP
Ts timeout
SOP present on all ports.
Packet Count same for all ports?
No.
Yes.
Handle Packet Data
Handle Packet Data
Error: Either port receives SOP without an earlier EOP
EOP.
EOP.
Wait for EOP on all ports
EOP on all ports.
Port 1
Port 2

Figure 4-9: State Diagram: Consumer port aggregation.

5. ODI-2 Documentation Requirements

ODI-2 includes documentation requirements for interoperability. This increases the chances of integrating compatible devices. Key among these are the ODI specification and revision numbers, line rates and associated BurstMax, and the flow control capabilities. These are required by the rules below.

RULE 5.1: An ODI-2 device SHALL document which ODI specifications and associated revision it complies to.

RULE 5.2: An ODI-2 device SHALL document all packet types used, and the contents of each of the packet fields.

OBSERVATION 5.1: RULE 5.2 may be met by using the templates published in VITA 49.2 Appendix A.2 Documenting Packet Classes

RULE 5.3: An ODI-2 device SHALL document what port aggregation capabilities it has, if any. This includes the number of ports that can be aggregated, and which In Band and Out of Band flow control methods used, if any.

RULE 5.4: An ODI-2 device SHALL document which In Band and Out of Band flow control methods are used, if any.

6. Appendix A: Meeting the 32-byte packet length requirement

ODI mandates that all packets be an integer multiple of 32 bytes in length, or eight VRT words. ODI permits the appending of null data in a packet's payload section in order to meet this requirement. Null data is also referred to as pad data, as the data payload is "padded" to meet the 32-byte requirement.

V49.2 defines a five-bit Pad Bit Count field within the Class ID field to indicate the number of pad bits (0…31) at the end of the Payload. It is an optional field, which must be set to zero if unused. This technique enables a developer to pad bits so the packet length will be evenly divisible by 32 bits, or one VRT word. However, three more bits would be needed to extend this technique to make all packets evenly divisible by 32 bytes, or 256 bits. There is no standard method in V49.2 to do this. Instead, this section will highlight allowable alternatives to meet the 32-byte rule.

For most cases, it is straight-forward to meet the 32-byte rule. For Context and Command Packets, null data is added to extend the length to the next 32-byte boundary. CIFs detail where each field is, distinguishing valid data from null data. In this case, neither the Pad Bit Count field, nor any additional fields, are needed.

For Data Packets streaming single-channel or multi-channel RF data, the solution is also straight-forward. As long as each packet contains an integer multiple of 256 samples of each channel, the 32-byte requirement will be met. This can be met by proper device design.

The issue of additional indicators arises when the Data Payload isn't naturally divisible by 32 bytes, and can't be made to be. This may arise when communicating data of finite length, such as spectral data or digital test vectors. In order for a Data Packet to meet the 32-byte requirement, the Data Payload also must be divisible by 32 bytes. This is due to the Prologue (28 bytes) and Trailer (4 bytes) summing to 32 bytes. The Pad Bit Count field of Class ID only allows the developer to pad up to 31 bits. Three more bits are needed to pad up to 256 bits, or 32 bytes. Below are three options for developers facing this situation.

1. Define three bits in the Class ID field to represent Pad Word Count. This is a technique that is specific to a device developer's OUI. Three bits would be defined in the Class ID field that indicate how many words are "Pad Words" to make a Data Payload divisible by 32 bytes. ODI-2.1 Rev. 3 uses this technique for the AXIe OUI. Since Pad Bit Count is already in the Class ID field, the combination unambiguously indicates which bits are valid, and which are pads.

2. Use the user-defined bits in the Trailer to indicate "Pad Word Count". This would indicate the number of Pad Words in the last 8 words sent. This, along with the Pad Bit Count field in the Class ID can determine precisely where valid data ends. The developer would assign a unique Class ID to this data type that defines the Trailer user-defined bits in this way.

A developer may be tempted to assign three of the four user-defined bits as Pad Word Count. However, Rule 5.1.6.4 says that the two user-defined Enable bits must be used as Enable bits, with the two user-defined Indicator bits as Indicator bits. A violation would occur if the Indicator bits were being set without the Enable bits.

A scheme that meets this rule would be to define the two user-defined Enable bits as PWCE36 and PWC12, meaning Pad Word Count Enable 36, and Pad Word Count Enable 12. They would enable the related Indicators PWCI36 and PWCI12 respectively. PWCI36 would add 3 or 6 Pad Words to the Pad Word Count, while PWC12 would add 1 or 2 words. If not enabled, neither indicator adds any words. Between the enabling and the indicator, any Pad Word Count from 0 to 8 can be generated, though 8 would never be used. The Class ID of the developer's OUI would indicate the use of these Trailer bits.

The disadvantage of this scheme is that the user-defined bits will be taken for this packet class, preempting other uses.