

ODI-1: Physical Layer Specification

Optical Data Interface

Revision 3.1

December 15, 2020

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

The AXIe Consortium and its member companies make no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, and its member companies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

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.

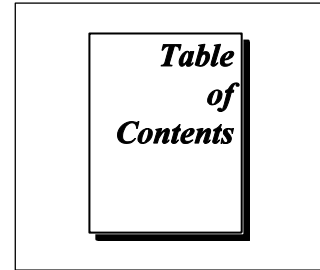
PCI-SIG, PCI Express, and PCIe are registered trademarks of PCI-SIG.

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.



Important Information	2
ODI-1 Physical Layer Specification	6
1. ODI Specification Organization and Requirements	8
1.1 Introduction.....	8
1.2 ODI-1 Compliance	9
1.3 Audience of Specification	9
1.4 Organization of Specification	9
1.5 References.....	10
1.6 Glossary	10
1.6.1 Common ODI Terms.....	11
1.6.2 Interlaken Terms	12
1.6.3 Packet Terms.....	12
2. Overview of the ODI Physical Layer Specification	14
2.1 Scope of ODI-1	14
2.2 ODI-1 Capability and Performance Summary.....	15
2.3 Example Operation	16
2.4 ODI Physical Layer Overview.....	19
3. ODI Optical Layer Specification.....	21
3.1 ODI Optical Cables	21
3.1.1 Mechanical Requirements	22
3.1.2 ODI Cable Pin Assignments	23
3.1.3 ODI Cable Optical Requirements	23
3.1.4 ODI Dual Cable Adapter	23
3.2 ODI Ports.....	25
3.2.1 Optical Port Placement.....	25
3.2.2 Optical Port Mechanical.....	27
3.2.3 Optical Port Pin Assignments	28
3.2.4 Optical Port Optical Requirements.....	30

4. ODI Protocol Layer Specification	31
4.1 Interlaken Operation	32
4.1.1 Interlaken Channels.....	32
4.1.2 Interlaken Bursts and Burst Parameters	33
4.1.3 Interlaken Rate Matching.....	34
4.1.4 Interlaken Meta Frame.....	34
4.1.5 Status Messaging	34
4.1.6 Error Detection and Correction.....	35
4.1.7 ODI Flow Control	35
4.1.8 Interlaken Property Summary.....	39
4.2 ODI Packet Operation.....	40
4.2.1 ODI Generic Packet.....	40
4.2.2 ODI Packet Requirements.....	41
4.3 ODI Data Streaming.....	41
4.3.1 Initiating Streaming Without Flow Control.....	42
4.3.2 Terminating Streaming Without Flow Control	43
4.3.3 Initiating Streaming With Flow Control.....	44
4.3.4 Terminating Streaming With Flow Control	46
4.4 State Diagrams	47
4.4.1 Overall Streaming State Diagram.....	48
4.4.2 Xmit Train Packets State Diagram	50
4.4.3 Xmit Caboose Packet State Diagram.....	52
4.4.4 Rcv Packets State Diagram.....	53
4.5 ODI Test Pattern	56
4.5.1 FPGA Implementation	58
5. ODI-1 Documentation Requirements.....	60
6. Appendix A: ODI Speed Calculations.....	61

List of Figures

Figure 1-1: ODI Specification Structure.....	8
Figure 2-1: Speed characteristics of ODI devices.....	15
Figure 2-2: Block Diagram of storage/playback instrument system.	16
Figure 2-3: Example storage operation	17
Figure 2-4: Example playback operation from storage	18
Figure 2-5: ODI-1 Overview Diagram.....	19
Figure 3-1: Typical ODI Optical Cables.....	21
Figure 3-2: ODI Cable Crossover Diagram	22
Figure 3-3: ODI Cable Pin Assignments	23
Figure 3-4: ODI Dual Cable Adapter	24
Figure 3-5: Recommended labeling for ODI Dual Cable Adapter	25
Figure 3-6: Front view of an ODI port.....	28
Figure 3-7: ODI port pin assignments.....	29
Figure 4-1: Interlaken Diagram for ODI	31
Figure 4-2: Flow Control – Analogy to a water tank.....	36
Figure 4-3: Consumer FIFO Input Buffer.....	37
Figure 4-4: ODI Generic Packet	40
Figure 4-5: Streaming by Sending Consecutive Packets	42
Figure 4-6: Time Diagram of Initiating Streaming Without Flow Control	43
Figure 4-7: Time Diagram of Terminating Streaming Without Flow Control	44
Figure 4-8: Example of Streaming With Flow Control to an AWG.....	45
Figure 4-9: Time Diagram of Initiating Streaming With Flow Control	46
Figure 4-10: Time Diagram of Terminating Streaming With Flow Control	47
Figure 4-11: Overall Streaming State Diagram.....	50
Figure 4-12: Xmit Train Packets State Diagram.....	52
Figure 4-13: Xmit Caboose Packet State Diagram	53
Figure 4-14: Rcv Packets State Diagram	55
Figure 4-15: Sequence order of least significant nibble of ODI Test Pattern	56
Figure 4-16: First packet of the ODI Test Pattern	57
Figure 4-17: Middle packets of the ODI Test Pattern.....	57
Figure 4-18: Final packet of the ODI Test Pattern	58
Figure 4-19: Example FPGA Implementation of first 64 bytes of ODI Test Pattern.....	59
Figure 4-20: Example FPGA Implementation of second 64 bytes of ODI Test Pattern...	59

List of Tables

Table 1-1: Architectural Specification Revisions.....	7
Table 3-1: Indicator lights	26
Table 4-1: Interlaken Property Summary	39

ODI-1 Physical Layer Specification

Revision History

This section is an overview of the revision history of the ODI-1 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). Updated state diagrams for flow control and non-flow control applications. Added support for dual uni-directional devices. Updated glossary.
3.0	January 3, 2019	First formal textual specification. Expanded Glossary. Added Recommendations for ODI connector orientation, numbering, and indicator lights. Added Recommendation for ODI Dual Cable Adapter labeling. Moved specific packet length requirements from ODI-2 to ODI-1. An ODI-1 packet shall be an integer multiple of 32 bytes in length, and between 64 bytes and 262,144 bytes inclusively. Changed some terms in state diagrams to match terms in ODI-A. Expanded documentation requirements to include flow control. Added Appendix for derivation of speed calculations.
3.1	December 15, 2020	Added Observation 4.8 that the sequencing of start-up between the producer and consumer is not defined. Therefore, the consumer must be tolerant of the timing of a valid producer

		<p>output, which may be before or after the consumer enters the Sync Links state.</p> <p>Added Section 4.5 ODI Test Pattern. This is a recommended but optional test pattern for testing the links between two devices. This test pattern may be used by developers of ODI devices, or it may be used by users while integrating an ODI system.</p>
--	--	---

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.

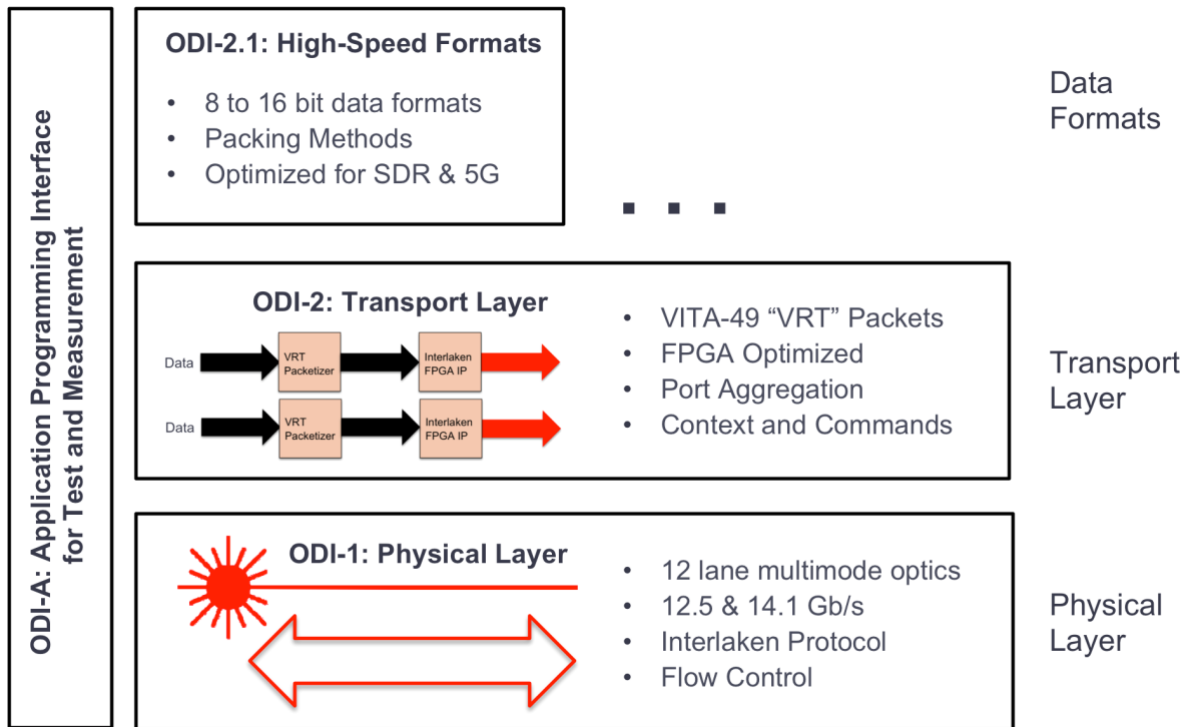


Figure 1-1: ODI Specification Structure

1.2 ODI-1 Compliance

ODI-1 defines the physical layer requirements for ODI-1-enabled products. Common products include instruments and instrumentation modules, RF embedded systems and modules, storage and processing devices, and the optical cables that interconnect them. For a device to claim conformance, it must comply with all applicable RULES in this document, including documentation requirements.

ODI-1 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 Physical Layer Specification

ODI-1 specifies the physical layer of a single optical interconnect for very high speed streaming applications between instruments, processors, and storage. It includes two line rates, one at 12.5 Gb/s, and one at 14.1 Gb/s. The latter allows 20 GBytes/s continuous streaming from a single port.

ODI-1 specifies the mechanical, optical, and timing requirements for each optical port. ODI ports may be deployed onto any electrical product or assembly. ODI-1 also specifies the mechanical and optical characteristics of the optical cable.

In general, optical ports are either uni-directional or bi-directional optical links of 12 lanes. Optical cables are 24 lanes, 12 lanes in each direction.

ODI-1 uses the Interlaken protocol to transfer data from a producer to a consumer. The Interlaken protocol sends arbitrary data over the link, separated into packets. ODI-1 does not specify the formats of the packets, though ODI-2 and ODI-2.1 specify VRT (VITA Radio Transport, also known as VITA 49.2) as the packet format.

ODI-1 is optimized for data streaming, and many ODI systems will consist entirely of signal data, while control is done through a separate interface.

ODI-1 supports flow control, where a consumer can modulate the speed of the incoming data. This is a critical feature for consumers such as signal generators.

2.1 Scope of ODI-1


ODI-1 defines the physical layer of the Optical Data Interface specification
It includes:

- Optical ports
- Optical cables
- Optical bit rates and lane widths
- Interlaken protocol use
- Flow control
- Packet framing and lengths
- State diagrams for streaming
- Documentation requirements

2.2 ODI-1 Capability and Performance Summary

Today

		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



 ODI-2




Figure 2-1: Speed characteristics of ODI devices.

ODI-1 devices can deliver continuous device-to-device streaming up to 20 GBytes/s when using a 14.1 Gb/s line rate, as shown in the figure above. 12.5 Gb/s devices can achieve 17.3 GBytes/s. Both line rates have advantages. FPGA vendors often include 12.5 Gb/s as hard IP with no licensing costs. 14.1 Gb/s devices offer higher speeds and achieve the cardinal value of 20 GBytes/s, or 160 Gb/s. 14.1 Gb/s devices can also work at 12.5 Gb/s speeds, so the devices may be mixed and match, defaulting to the lower speed.

As optical line rates increase, it is likely that ODI will adopt higher line rates, approximately doubling with each new specification. This has been indicated by the placeholder ODI-1.1 and ODI-1.2 columns.

ODI-1 is a single port specification. ODI-2 describes how ports are aggregated, which increases the aggregate throughput proportionally to the number of ports.

ODI-1 also includes the following features and capability:

- Compliance to IEC EN 60825-1:2007 Class 1 safety specification
- Ability to place ODI ports on any device, regardless of format
- Single standard fiber cable topology
- Fiber cable lengths up to 100 meters
- Uni-directional and bi-directional operation
- In-band and out-of-band flow control
- Packet encapsulation
- Streaming through consecutive packet transfers
- Ability to regain normal operation from a data outage

2.3 Example Operation

The following example shows an instrument system where data is being recorded from a digitizer, and then played back to an AWG (Arbitrary Waveform Generator), reproducing the original signal.

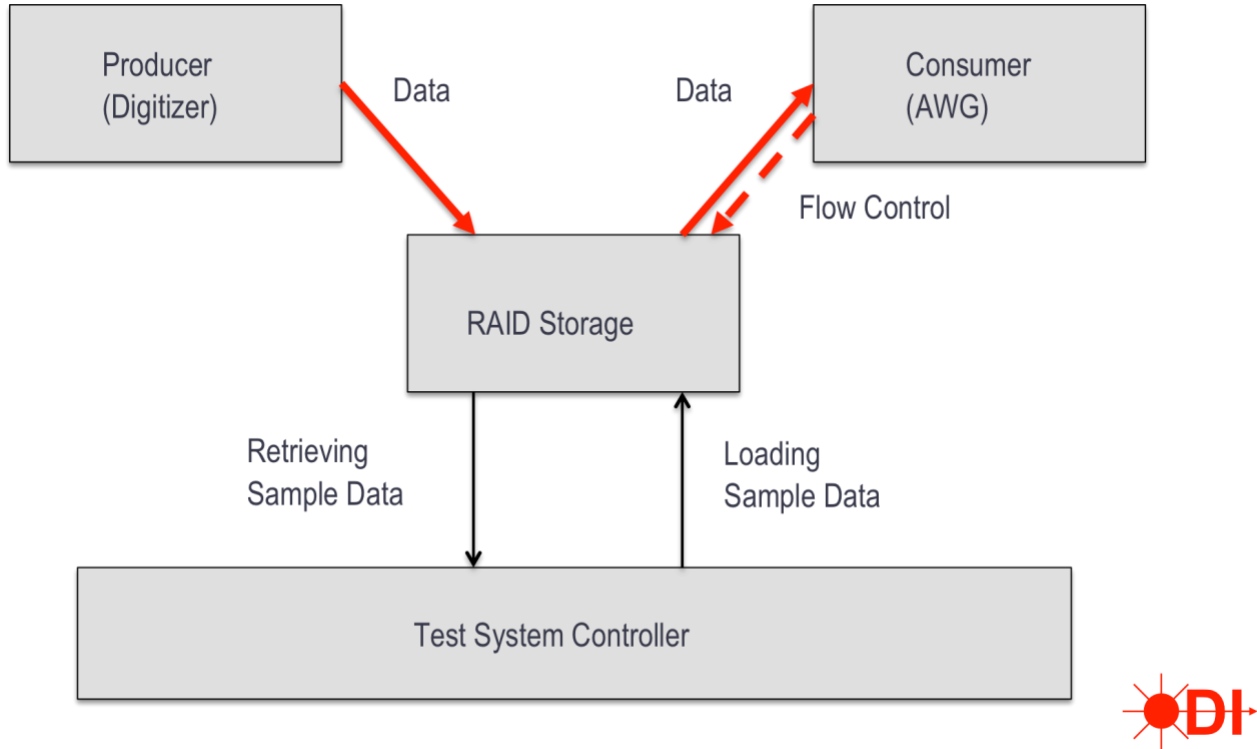


Figure 2-2: Block Diagram of storage/playback instrument system.

Figure 2-2 shows the basic components of a storage and playback instrument system. The test system controller communicates with the three ODI devices- the digitizer, the AWG, and the RAID storage system. The controller sets up each device, though full speed data is only sent over the optical data links. The intent is to record data from the digitizer, and play it back to the AWG at some future time.

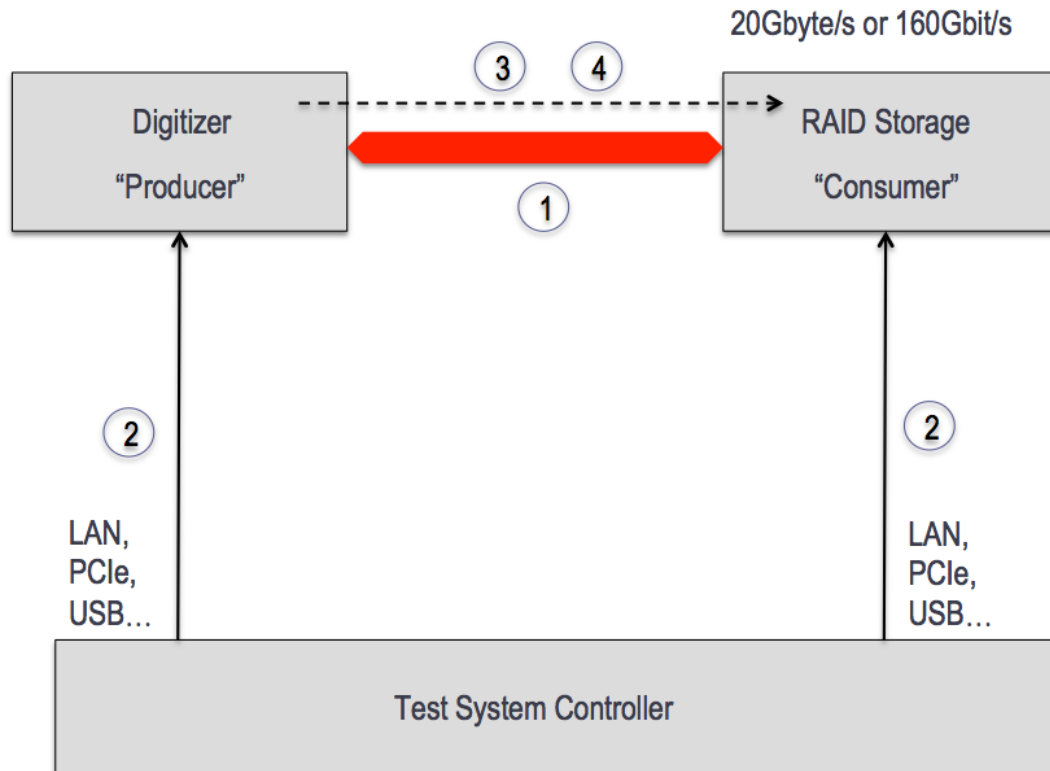


Figure 2-3: Example storage operation

To stream data into a storage device, the following four steps are executed:

1. The user connects the two devices together with a standard optical cable.
2. The user sends configuration commands to each device, arming them
3. After a hardware or software trigger event, transfer begins
4. Data transfer continues until a hardware or software “end” event

At this time, the recorded data is in the storage device. It may be accessed by the test system controller through the interface to the storage device, or it may be played back into another device. It may also be modified in some way, and then reloaded into the storage device for playback later.

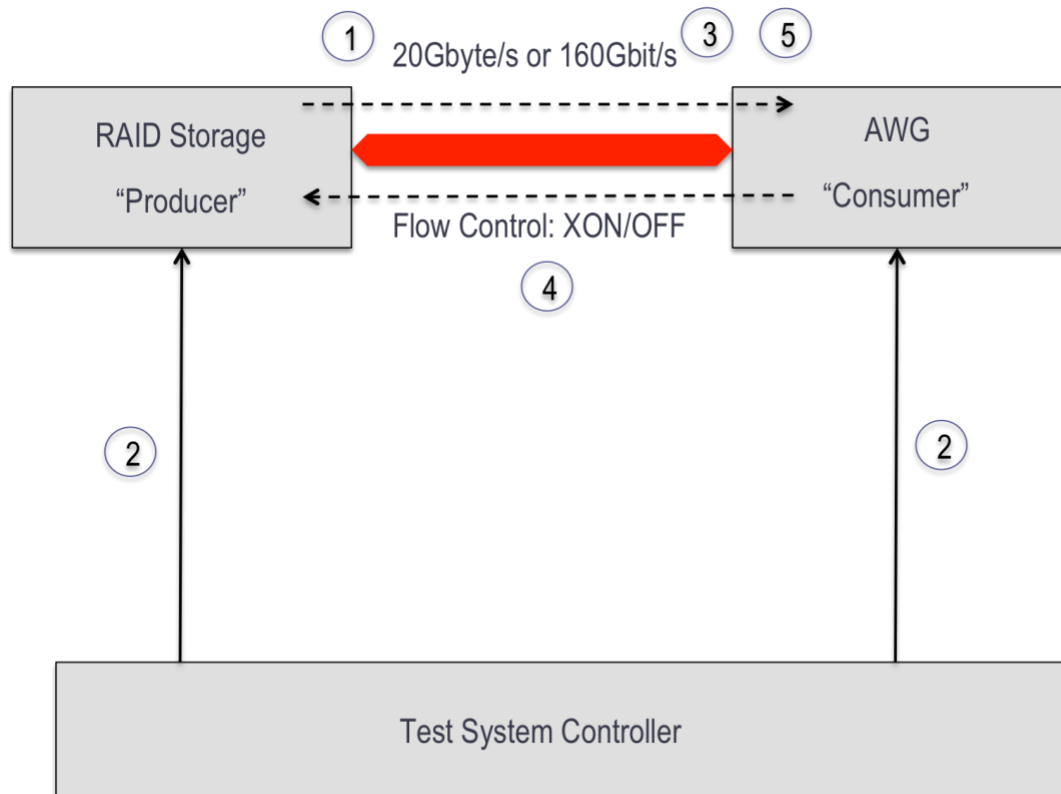


Figure 2-4: Example playback operation from storage

To play back recorded or computed data, data is entered into the storage device, either by recording as shown previously in Figure 2-3, or by an upload from the test system controller. The storage device, which was previously a data consumer in the recording scenario, is now a producer. An AWG or other signal generator is the consumer, which will recreate the signal. In this example, the device ports will be enabled for bi-directional flow, with one direction data and the other direction the flow control command words. The procedure is similar to that of data recording. However, since the producer will send the data at full rate, this rate will not match the precision rate of the AWG. Flow control is evoked by the AWG to turn transmission off and on to have just enough data in the AWG's input buffer to match to the AWG's sample rate. Not only are the rates matched, but the output may also be low jitter defined by the AWG's precision clock. The sequence of events is:

1. The user connects the two devices together with a standard optical cable.
2. The user sends configuration commands to each device, arming them
3. After a hardware or software trigger event, transfer begins
4. Flow control is evoked as needed from the consumer, an AWG
5. Data transfer continues until a hardware or software "end" event

2.4 ODI Physical Layer Overview

The following diagram shows an ODI link between two devices and identifies the key optical and protocol components.

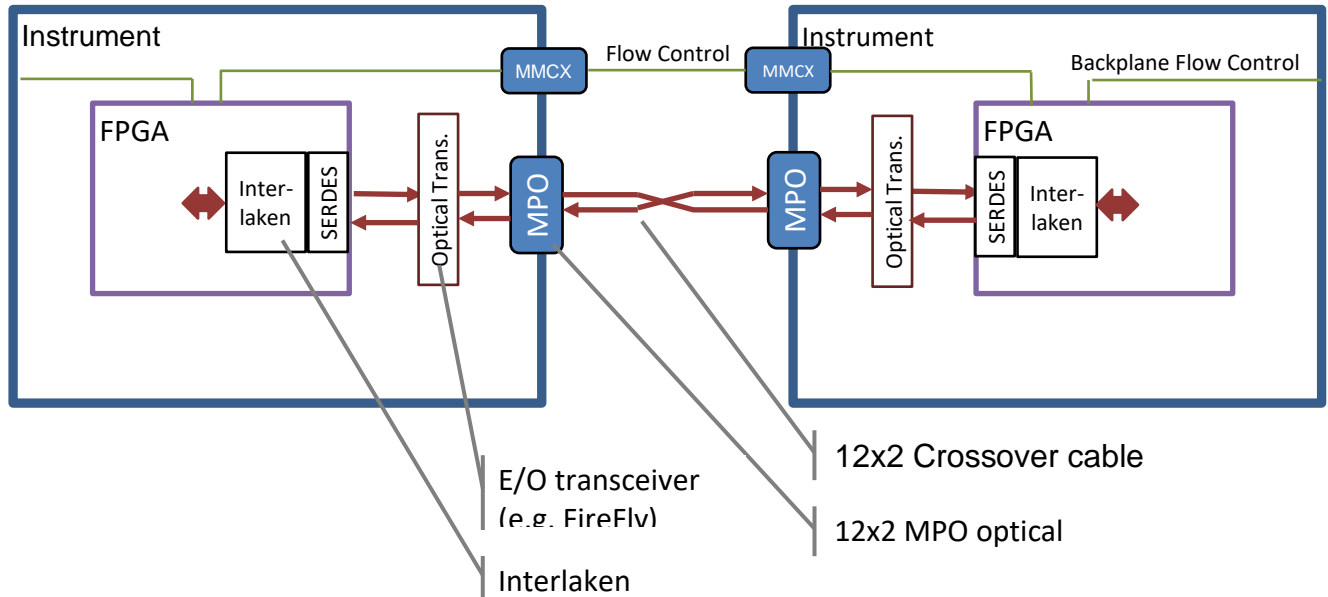


Figure 2-5: ODI-1 Overview Diagram

Figure 2-5 is a simplified overview diagram for ODI-1. Each device has a MPO (Multi-fiber Push On) connector, which houses two rows of 12 fibers each. The top row is for transmit functions, and the bottom half is for receive functions. The standard ODI cable is a 24-fiber crossover cable, which automatically connects transmitters to receivers.

Uni-directional devices connect their internal fibers to either the optical transmitters or receivers, depending which type of device they are. Bi-directional devices (shown) connect to both. Samtec Firefly optical engines meet the ODI specifications, but so do any optics that meet 802.3ba optical levels and the specified line rates. ODI-1 has two specified line rates, 14.1 GB/s and 12.5 Gb/s. ODI can achieve 160 Gb/s with the 14.1Gb/s rate, which equates to 20 GB/s.

ODI-1 uses common data center optics for the optical components. The optical transmitters are 850nm VCSELs, while the cable is created from standard multimode fibers.

For the protocol layer, ODI-1 uses Interlaken to stripe and de-stripe data over multiple lanes, provide packet framing, error detection, and flow control.

Both in-band (IB) and out-of-band (OOB) flow control are supported. In-band flow control uses the reverse path of a bi-directional optical link, allowing the consumer to pace the amount of data transferred into its input buffer. An advantage of IB flow control is that no other connections are required; the flow control commands are sent over the reverse link from the consumer. The commands are simple: XON requests the producer to transmit data, and XOFF requests the producer to stop transmitting data.

OOB flow control uses a 1-wire external connection, typically electrical, between the two devices. This is shown as a signal between the two MMCX connectors in the diagram above. The level of the signal indicates XON or XOFF. The advantage of OOB flow control is that uni-directional devices may be used, controlling costs. Devices that use a modular backplane, such as AXIe, PXI, or VPX, may use a backplane “trigger” signal for OOB flow control.

3. ODI Optical Layer Specification

This section details the optical layer requirements of an ODI-1 product or assembly.

3.1 ODI Optical Cables

The standard ODI cable is a 24-fiber crossover cable, which connects ODI producers to consumers by connecting the optical transmitters of one device to the optical receivers of the other. ODI cables consist of 12 lanes of OM3 850nm multimode fiber in each direction, and up to 100 meters in length. Each end of the cable is terminated with a female MPO (Multi-fiber Push On) connector. The crossover topology, along with the female MPO connectors, allows any end to be plugged into either device. There is no polarity of the cable. The crossover design also allows the same standard cable to be used with either uni-directional devices or bi-directional devices.



Figure 3-1: Typical ODI Optical Cables

3.1.1 Mechanical Requirements

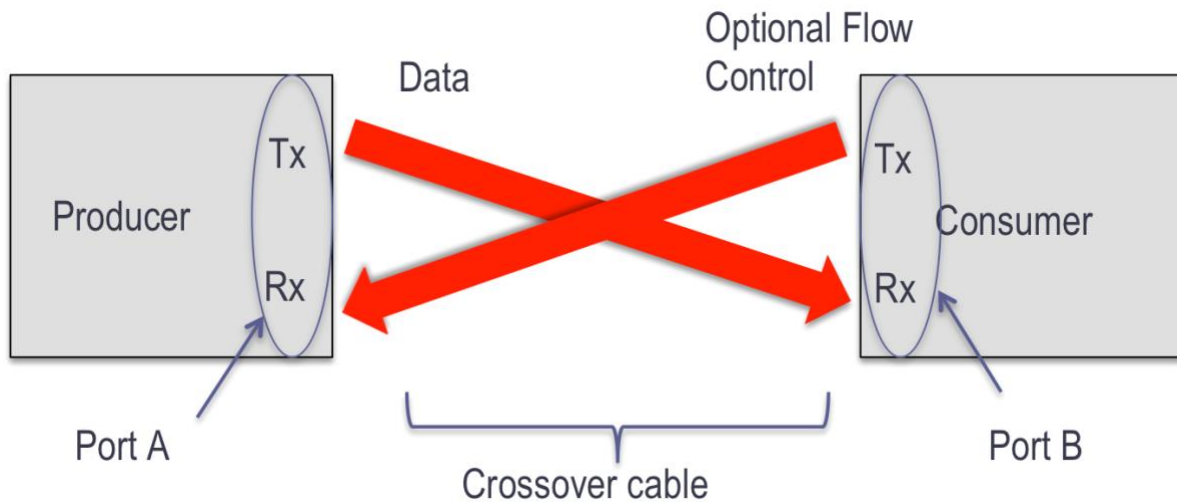


Figure 3-2: ODI Cable Crossover Diagram

Figure 3-2 shows the crossover topology of an ODI cable, and how transmitters (Tx) of one port are connected to the receivers (Rx) of another.

RULE 3.1: An ODI-1 cable SHALL implement 12 lanes of OM3 multimode fiber in each direction and in a crossover configuration as indicated in Figure 3-2.

RULE 3.2: An ODI-1 cable SHALL use a female MPO style connector on both ends, containing 2 rows of 12 fibers each.

PERMISSION 3-1: An ODI-1 cable MAY use MTP connectors, which are MPO connectors built to tighter tolerances.

RULE 3.3: An ODI-1 cable SHALL NOT include the two ferrule guide pins in the female MPO connectors

RULE 3.4: An ODI-1 cable SHALL NOT include the MPO Bulkhead adapter at either end.

RULE 3.5: An ODI-1 cable SHALL NOT exceed 100 meters in length.

3.1.2 ODI Cable Pin Assignments

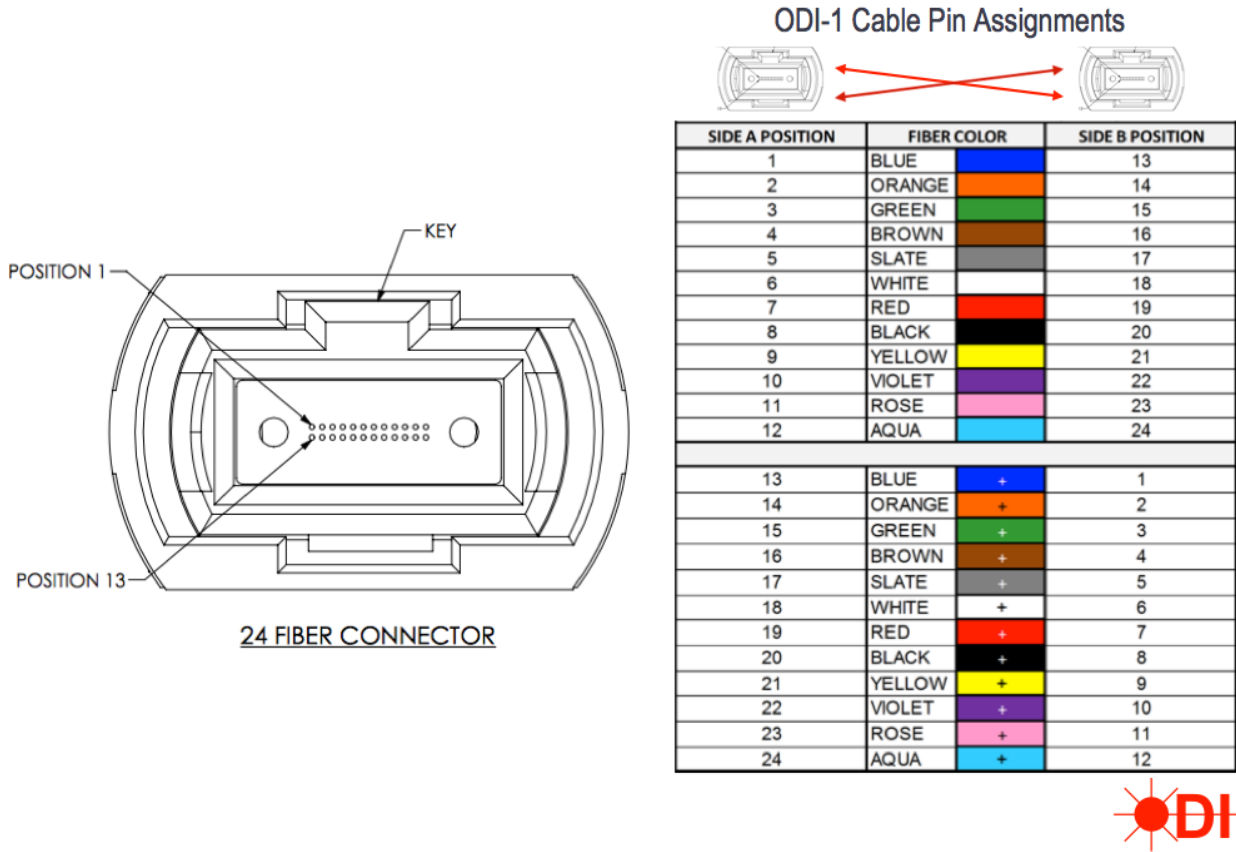


Figure 3-3: ODI Cable Pin Assignments

RULE 3.6: An ODI-1 cable shall implement a bi-directional crossover topology by complying to the pin assignments in Figure 3-3.

3.1.3 ODI Cable Optical Requirements

RULE 3.7: The performance of an ODI-1 cable shall meet or exceed the specifications of multimode OM3 grade cable.

3.1.4 ODI Dual Cable Adapter

ODI-1 allows the implementation of dual uni-directional devices. These are devices that are similar to bi-directional devices in that they have a Tx row and an Rx row, but those rows connect to separate devices in a daisy chain topology. ODI implements this by defining an ODI Dual Cable Adapter that then allows standard ODI cables to go to the two devices, as shown in the figure below.

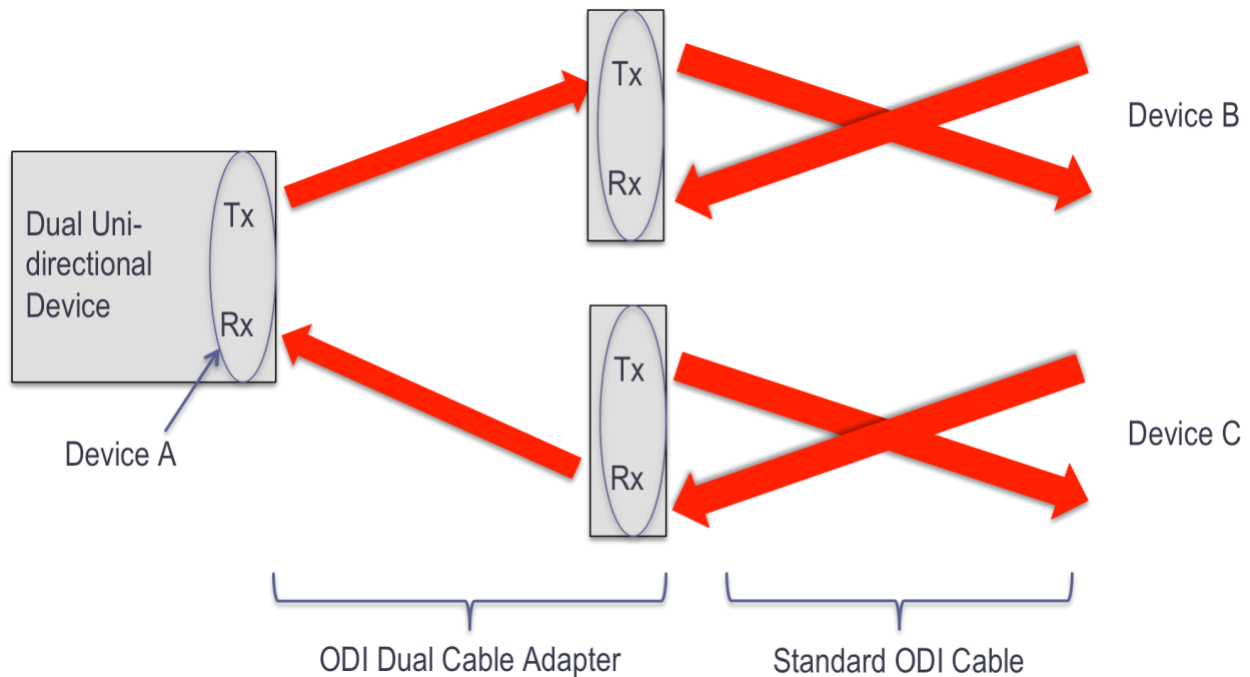


Figure 3-4: ODI Dual Cable Adapter

In Figure 3-4, Device A is a dual uni-directional device. That is, it receives data from one device (Device C), and transmits data to a different device (Device B). One example application is a Digital Signal Processor (DSP) processing data in real time.

ODI has defined a standard Dual Cable Adapter for this application. The adapter splits the rows at Device A, receiving from Device B and sending to Device C. The intent is that this is a simple compact universal adapter, and then standard ODI cables can be used to connect to Devices B and C. The Dual Cable Adapter has a single female MPO connector (connecting to Device A) and two male connectors that act like ODI ports- one with the Tx rows and one with the Rx rows. This way standard ODI cables can be used to extend the reach to the distance needed.

RULE 3.8: An ODI Dual Cable Adapter SHALL comply with the cable and connector topology shown in Figure 3-4.

PERMISSION 3-2: An ODI-1 cable assembly MAY add ODI cable functionality to an ODI Dual Cable Adapter on one or both of its ports.

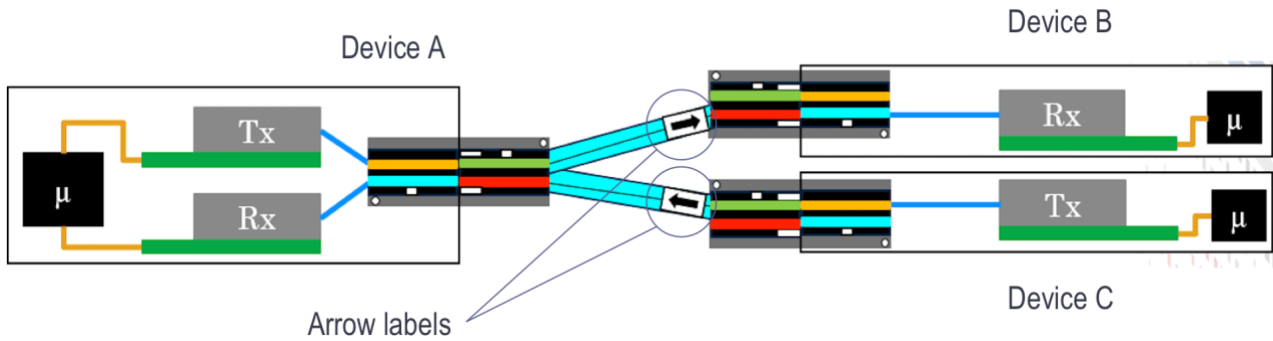


Figure 3-5: Recommended labeling for ODI Dual Cable Adapter

An ODI Dual Cable Adapter will have two ends, to go to Device B and to Device C, as indicated in Figure 3-4. There could be ambiguity to the user if those two ends are not properly labeled. ODI-1 recommends labeling the two ends with arrow symbols denoting the direction of data flow, as shown in Figure 3-5.

RECOMMENDATION 3.1: An ODI Dual Cable Adapter SHOULD label the direction of data flow as shown in Figure 3-5.

3.2 ODI Ports

An ODI device has one or more optical ports as shown in Figure 2-5. ODI-1 specifies the requirements for a single port. A producer is a device that transmits data, and a consumer is a device that receives data. A single port may be a producer, a consumer, or both. Additionally, a producer or a consumer may offer in-band flow control, where flow control commands are sent from the consumer to the producer, using the additional 12 lanes of fiber. Due to these options, an ODI port may be uni-directional or bi-directional. In all cases the ODI port consists of two rows of 12 fiber positions each. A uni-directional port simply makes one row inoperable.

Mechanically, ODI ports use MPO (Multi-fiber Push On) connectors, with a bulkhead placed around the connector.

3.2.1 Optical Port Placement

ODI ports may be placed anywhere on the device, including front panels, rear panels, or the backplane areas. Multiple ports will require that the ports be uniquely labeled. ODI-A references the ports starting with the number “1” (as in “ODI_1”), and each subsequent port is numbered consecutively. It is recommended that ports be similarly labeled. While there is no requirement that ports include an associated indicator light, it is a useful feature and recommended.

PERMISSION 3-3: An ODI device **MAY** contain one or more ODI ports.

PERMISSION 3-4: ODI ports **MAY** be placed on any surface of the ODI device.

RECOMMENDATION 3.2: ODI Ports **SHOULD** be numbered starting with the number “1” and incremented by 1 for each successive port.

OBSERVATION 3.1: Labeling each ODI port as “ODI_N”, where N is the ODI port number, will match the text used in ODI-A.

RECOMMENDATION 3.3: Each ODI port **SHOULD** include a nearby indicator light that indicates the status of the port.

Activity	State	Indicator
Port not activated	Off	Dark
Activated for Bidirectional or Dual Unidirectional use	Receiver not synchronized or transmitter not ready	Yellow
	Receiver synchronized and transmitter ready	Green
Activated as a Unidirectional data Consumer	Receiver not synchronized	Yellow
	Receiver synchronized	Green
Activated as a Unidirectional data Producer	Transmitter not ready	Yellow
	Transmitter ready	Green
Activated	Sending or receiving data	Blinking Green (recommended), or Green

Table 3-1: Indicator lights

RECOMMENDATION 3.4: ODI ports with an indicator light **SHOULD** follow the behavior indicated in Table 3-1.

RECOMMENDATION 3.5: ODI ports **SHOULD** be oriented horizontal with the keyway up.

The above recommendation encourages similar orientation of ODI ports for user ease of use when plugging in an ODI cable. It should be noted that many devices, particularly modular devices, do not have a fixed orientation themselves, and some judgment may be

needed. Other factors, such as density and available surface space, may preclude this orientation.

PERMISSION 3-5: An ODI port MAY include a spring-loaded dust cover.

A spring-load dust cover helps dust from accumulating in the port, which can cause a poor optical connection when an ODI cable is plugged in.

3.2.2 Optical Port Mechanical

RULE 3.9: An ODI port SHALL use a male MPO style connector, containing 2 rows of 12 fiber positions each.

PERMISSION 3-6: An ODI port MAY use MTP connectors, which are MPO connectors built to tighter tolerances.

PERMISSION 3-7: An ODI device MAY contain a higher density optical connector for density reasons, if the device supplier also offers an adapter to expand to multiple standard ODI connectors.

RULE 3.10: An ODI port SHALL implement either 12 lanes or zero lanes in each direction.

RULE 3.11: An ODI port SHALL NOT implement more than 12 lanes in a single direction, nor between 1 and 11 lanes.

OBSERVATION 3.2: A bi-directional port has 12 lanes in each direction. A uni-directional port either transmits using 12 lanes or receives using 12 lanes, but not both.

PERMISSION 3-8: An ODI-1 port MAY be configured as a dual uni-directional device. In this case, the Tx row and the Rx row will be connected to different devices.

3.2.3 Optical Port Pin Assignments

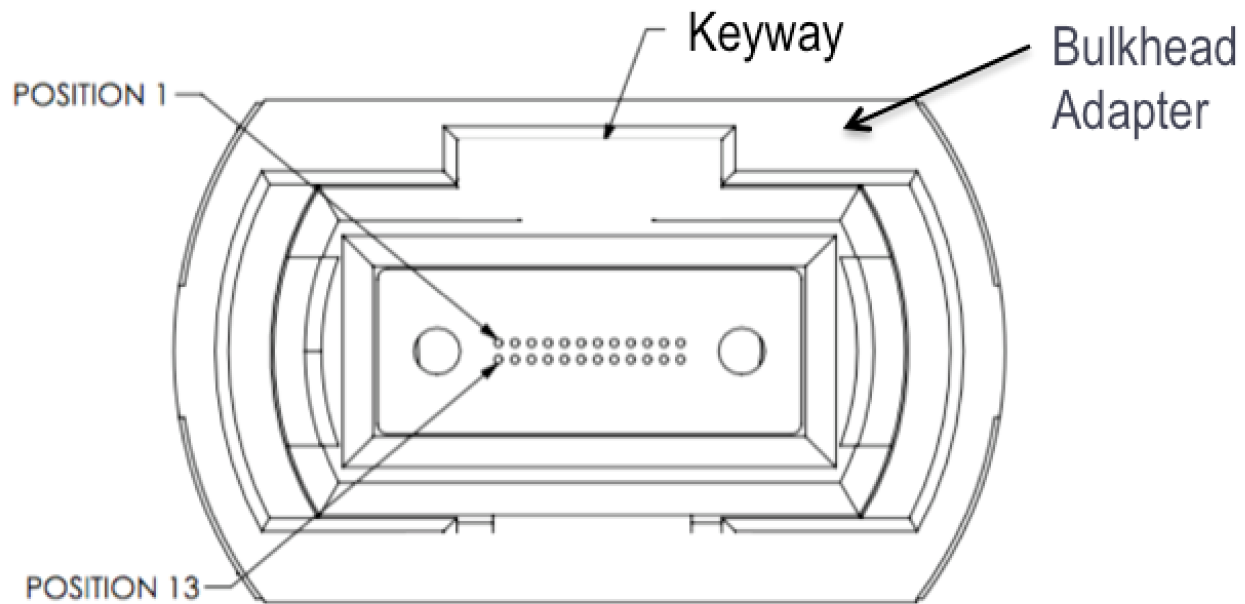


Figure 3-6: Front view of an ODI port.

RULE 3.12: An ODI-1 port SHALL include the two MPO ferrule guide pins.

RULE 3.13: An ODI-1 port SHALL include the MPO bulkhead adapter

OBSERVATION 3.3: The inclusion of the bulkhead adapter allows ODI cables to be easily and reliably snapped in.

OBSERVATION 3.4: The use of an opposing style bulkhead adapter allows the use of commonly available components to meet the port pin positions shown in Figure 3-6.

RULE 3.14: An ODI-1 port SHALL place the keyway adjacent to the row of Tx pins, positions 1 through 12.

Function	MPO Port Position
Tx11	1
Tx10	2
Tx9	3
Tx8	4
Tx7	5
Tx6	6
Tx5	7
Tx4	8
Tx3	9
Tx2	10
Tx1	11
Tx0	12
<hr/>	
Rx11	13
Rx10	14
Rx9	15
Rx8	16
Rx7	17
Rx6	18
Rx5	19
Rx4	20
Rx3	21
Rx2	22
Rx1	23
Rx0	24

Figure 3-7: ODI port pin assignments

RULE 3.15: An ODI-1 port SHALL comply with the ODI pin assignments as shown in Figure 3-7.

3.2.4 Optical Port Optical Requirements

ODI optical parameter requirements essentially match those of IEEE 802.3ba, driven at 12.5 Gb/s or 14.1 Gb/s line rate. This includes 850nm multi-mode transmitters and receivers. The transmitters typically deploy VCSEL technology.

RULE 3.16: An ODI-1 port **SHALL** comply with the optical levels specified by IEEE 802.3ba.

RULE 3.17: An ODI-1 port **SHALL** operate at 12.5 Gb/s line rate, unless such rate makes the device unusable.

PERMISSION 3-9: An ODI-1 port **MAY** operate at 14.1 Gb/s line rate.

OBSERVATION 3.5: The above **RULE** and **PERMISSION** essentially mandate a higher speed device to operate at the lower speed if at all possible. This enables upwards compatibility from 12.5 Gb/s to 14.1 Gb/s.

OBSERVATION 3.6: Automatically sensing and negotiating line rate is not required. Instead, the system software is expected to command each device to the proper rate. There is no time limit specified for a device to change rates.

RECOMMENDATION 3.6: Certain digitizers and signal generators may require 14.1 Gb/s for full performance. Those products **SHOULD** offer modes that operate at 12.5 Gb/s, perhaps by limiting the resolution, number of channels, or sampling rate.

OBSERVATION 3.7: 12.5 Gb/s allows the most economical implementation, particularly when FPGA IP costs are considered. 14.1 Gb/s allows higher speed implementations where 20 GB/s is a critical specification.

4. ODI Protocol Layer Specification

ODI uses Interlaken, a chip-to-chip protocol, as its protocol layer. Interlaken provides the data striping and de-striping across the 12 lanes, packet framing, error detection, and flow control. ODI specifies the use of Interlaken Protocol Definition Revision 1.2.

Interlaken has many advantages. It is a proven protocol supported by the major FPGA vendors, with readily available IP. It efficiently packs data across the 12 lanes, and supports uni-directional and bi-directional communication. It supports In-Band and Out-Of-Band flow control, critical for signal generation applications. In-Band allows flow control without additional cables, while Out-Of-Band offers a cost effective alternative by deploying an electrical signal in place of an optical link. Interlaken supports long bursts for minimal overhead. By streaming data as consecutive packets, Interlaken allows simple outage recovery methods. In ODI-2, Interlaken packets are sent synchronized across all ports for port aggregation.

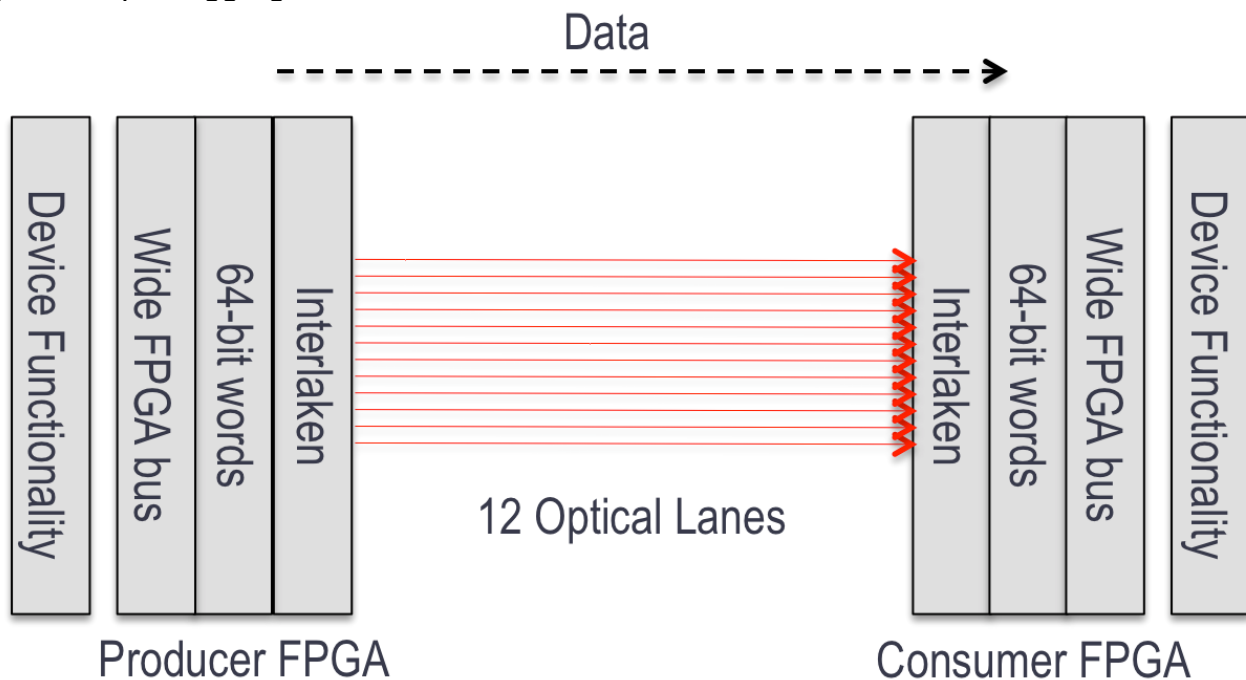


Figure 4-1: Interlaken Diagram for ODI

Interlaken is based on 8-byte (64-bit) words. Interlaken FPGA IP typically supports very wide buses that are a multiple of 64-bits on the device-facing side of the FPGA, and 12 high-speed SerDes on the optical-facing side of the FPGA, that connect to the appropriate E/O (Electrical to Optical) and O/E (Optical to Electrical) Converters. Interlaken words may either be Data Words or Control Words. Data is sent as a series of Interlaken Bursts, with each Burst bounded by an Interlaken Burst Control Word. The Interlaken Burst Control Word may also indicate SOP (Start of Packet) and/or EOP (End

of Packet). Interlaken Bursts may be up to 2048 bytes in length. Typically, ODI packets are much longer, requiring multiple Interlaken Bursts to transfer a packet.

Although Interlaken also contains the Interlaken channel number in the Burst Control Word, these are distinct from the concept of signal channels. ODI is designed so that many applications can be deployed over a single Interlaken channel. In this case, multiple synchronous signal channels are packed into a single packet structure within a single Interlaken channel. ODI permits the use of multiple Interlaken channels, but the performance of the combined system must be determined by the system integrator. Additional Interlaken channels may be used to indicate asynchronous measurements such as temperature or a signal stream at a different rate. For many applications, only a single Interlaken channel is deployed.

Interlaken operation is controlled by setting numerous parameters defined within the Interlaken specification. Much of the ODI Protocol Layer specification is detailing the requirements and constraints of those parameters.

4.1 Interlaken Operation

RULE 4.1: An ODI device SHALL comply with Interlaken Protocol Definition, Rev 1.2.

4.1.1 Interlaken Channels

Signals may have several channels. ODI does not use the Interlaken channel number to identify the signal channel number. ODI is designed so that only a single Interlaken channel is needed for streaming multiple synchronous signals. ODI relies on the signal channel information to be woven into the data stream and packet structure. A typical implementation would be to send the multi-channel data in a round-robin fashion.

RULE 4.2: An ODI device SHALL implement Interlaken Channel 0.

PERMISSION 4-1: An ODI device MAY use more than one Interlaken channel.

OBSERVATION 4.1: If an ODI device uses more than one channel, then performance is unspecified by the ODI specification.

OBSERVATION 4.2: Since synchronous multi-channel data may be encoded into the data stream, multiple Interlaken channels are not necessary to implement a multi-signal channel system. However, there may be a case where asynchronous data, such as temperature, may be sent on a second Interlaken channel.

When using multiple Interlaken channels, there are two Interlaken modes- packet mode or segment mode. Packet mode requires packets to be transferred one full packet at a time before switching Interlaken channels, while segment mode allows partial packets to be interleaved. ODI specifies packet mode if multiple Interlaken channels are used.

RULE 4.3: If an ODI device deploys multiple Interlaken channels, then that device SHALL use packet mode as the packet transfer method.

4.1.2 Interlaken Bursts and Burst Parameters

Interlaken sends data as packets, each packet being a series of data bursts. In most cases, the length of the data burst is a parameter labeled BurstMax. The longer the BurstMax, the more efficient the data transfer, as the same overhead is shared by longer data patterns. The IP for a 256-byte BurstMax is prevalent in the data center industry, and is often available at no additional cost. ODI defines two BurstMax options:

- 256 bytes at 12.5 Gb/s line rate for economy applications
- 2048 bytes at 14.1 Gb/s for performance applications

Interlaken BurstShort is a parameter that sets the minimum size of a data burst, and thus the interval between Burst Control Words. This minimum size for a burst prevents successive small bursts from overloading the processing of the device. ODI sets BurstShort to 64 for all devices, regardless of line rates.

RULE 4.4: A 12.5 Gb/s ODI producer SHALL transfer data using 256 byte BurstMax.

RULE 4.5: A 14.1 Gb/s ODI producer SHALL transfer data using 2048 byte BurstMax.

RULE 4.6: An ODI producer SHALL transfer data using 64 byte BurstShort.

RULE 4.7: A 12.5 Gb/s ODI consumer SHALL support receiving burst lengths between 64 and 256 bytes.

RULE 4.8: A 14.1 Gb/s ODI consumer SHALL support receiving burst lengths between 64 and 2048 bytes.

The above rules result in compatibility between devices of the same line rate. Since Section 3.2.4 compels 14.1 Gb/s devices to also operate at 12.5 Gb/s if possible, the BurstMax requirements will change to match the requirements of the configured line rate.

A final Interlaken burst parameter is BurstMin. A simple Interlaken scheduling procedure is to send a packet through successive bursts of BurstMax, until the remainder (packet length modulus BurstMax) is sent in the final burst. If the remainder is less than

BurstShort, extra Idle words are inserted to enforce the BurstShort guarantee. This is unused bandwidth that can lower the usable bandwidth of the link. To avoid this inefficiency, Interlaken offers Optional Scheduling Enhancement (OSE) documented in Section 5.3.2.1.1 of the Interlaken specification. In this case, Interlaken looks ahead to the EOP (End of Packet) indicator within the Interlaken Control Word. If it detects that the final burst will be less than another parameter, BurstMin, it then shortens the length of the second-to-last burst so that the final burst does not require Idle words. ODI permits the Optional Scheduling Enhancement and sets BurstMin to 64 bytes.

PERMISSION 4.1: An ODI producer MAY use the Interlaken Optional Scheduling Enhancement.

RULE 4.9: If an ODI producer uses the Interlaken Optional Scheduling Enhancement, it SHALL set BurstMin to 64 bytes.

RECOMMENDATION 4.1: An ODI producer SHOULD use the Interlaken Optional Scheduling Enhancement.

While it is possible to meet the required data rate not using OSE in some cases, a good practice it to deploy OSE and the associated 64 byte BurstMin.

4.1.3 Interlaken Rate Matching

ODI does not use Interlaken Rate Matching. Rate Matching is a mechanism where the effective data rate is limited by insertion of Idle words. In a signal-oriented system, an equivalent effect is achieved by the setting of the sample rate.

4.1.4 Interlaken Meta Frame

The Interlaken Meta Frame is defined in Interlaken Section 5.4.3 as the per-lane set of the Synchronization, Scrambler State, Skip, and Diagnostic words, along with the payload data (burst data and control information) carried on each lane.

The size of the Meta Frame is a single programmable parameter, MetaFrameLength, that applies to all lanes of the bundle. The Meta Frame structure is orthogonal to the data transmissions, and Meta Frame control words may occur at any point within a data burst. ODI specifies a MetaFrameLength of 2,048 words (16,384 bytes).

RULE 4.10: An ODI producer SHALL set MetaFrameLength to 2,048 words.

4.1.5 Status Messaging

Interlaken Status Message is not required.

4.1.6 Error Detection and Correction

Errors are detected by means of a mismatch in the Burst/Idle Control Word CRC. The CRC24 covers all data in the previous burst and bits [63:24] of the Burst/Idle Control Word.

There is no mechanism for error correction. ODI-1 does not specify the use of FEC (Forward Error Correction) nor data re-transmission.

4.1.7 ODI Flow Control

The ODI optional flow control allows a consumer to modulate the rate of data it receives by sending a signal XON/XOFF (Transmit ON, Transmit OFF) to the producer. This is also known as backpressure. This technique manages the status of the consumer's input buffer, avoiding underflow and overflow situations. This method is often deployed by signal generators, which then create the desired signal at the desired rate and phase fidelity.

Consumers such as AWGs and other signal generators are likely to implement flow control to keep the incoming data patterns matched with their sampling speed and at the required frequency and phase fidelity.

Producers such as digitizers are more likely to **not** implement flow control, since they must generate and transmit data at their sampling speed.

Memory, other storage devices, and processors are likely to implement **both**, flow control and no flow control, to interface with all instrument types.

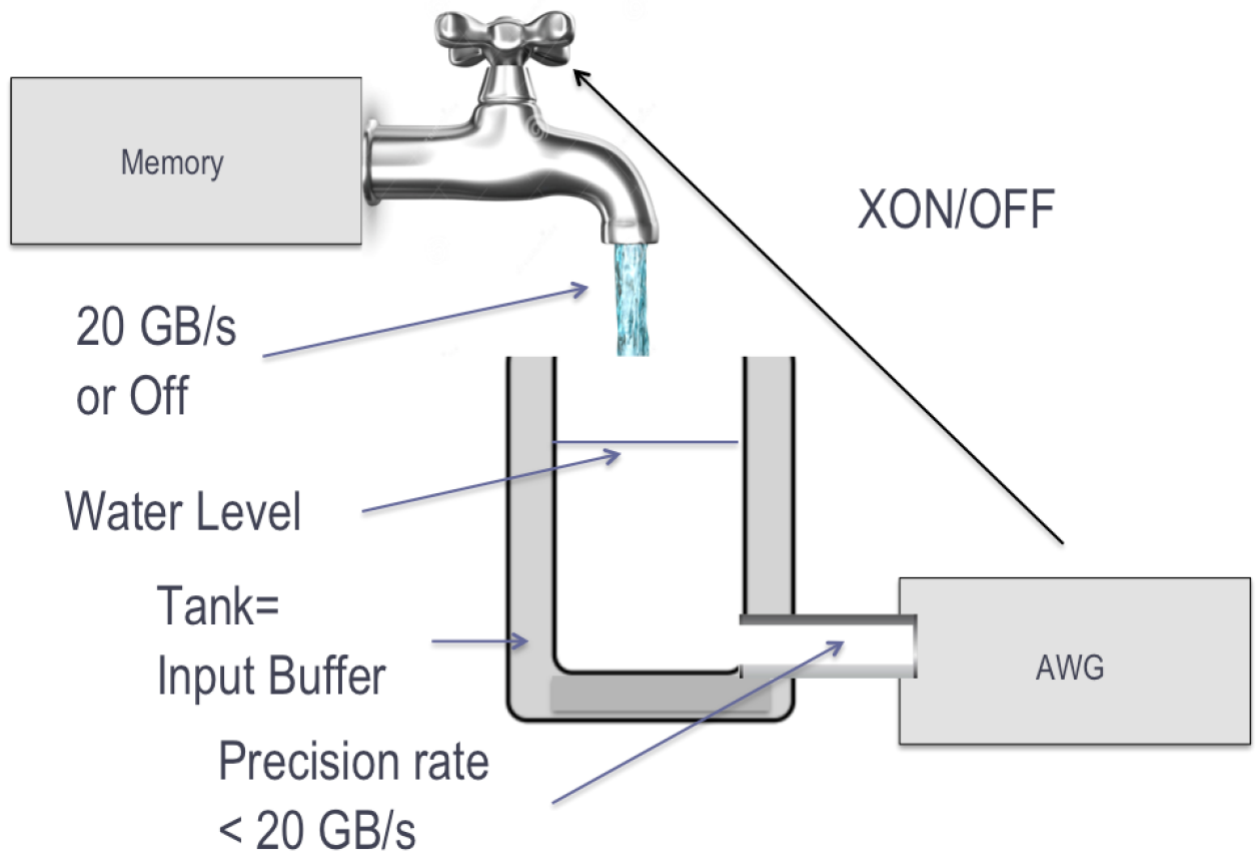


Figure 4-2: Flow Control – Analogy to a water tank.

Figure 4-2 shows an analogy to a water tank. The water tank represents the consumer's input buffer. It is filled by the faucet, which represents the producer. The faucet is either full on or off. This is controlled by the consumer, a precision AWG (arbitrary waveform generator), by indicating XON or XOFF. By doing so, the AWG can sample data at a precision rate, while keeping the water level within the limits of the vat.

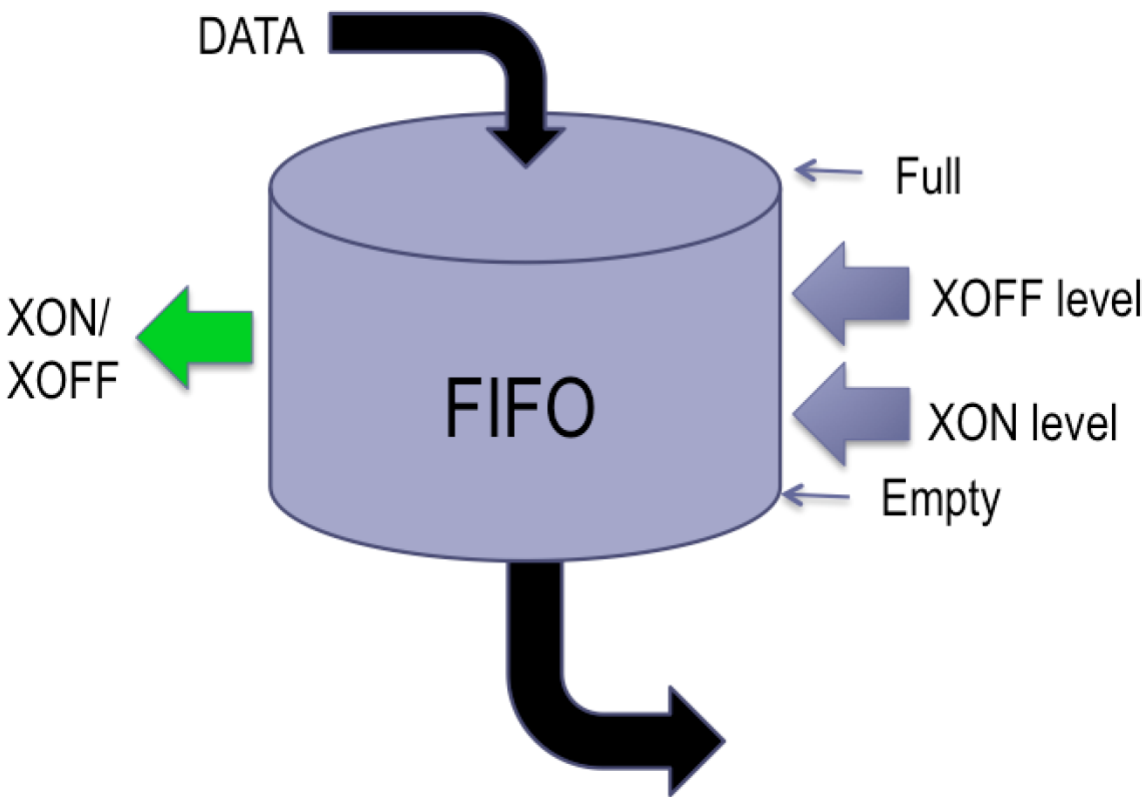


Figure 4-3: Consumer FIFO Input Buffer

Figure 4-3 shows the standard FIFO (First In First Out) Input Buffer inside a consumer deploying flow control. Data enters the FIFO at full rate at the top, and exits at the bottom at the data rate of the consumer. The green arrow is the XON/XOFF indicator sent back to the producer. The value of XON/XOFF is determined by the “level” of the FIFO, that is, how much valid data is in the buffer yet to exit. Below “XON level”, and the indicator is set to XON. The buffer will fill until at some future time the “XOFF level” is exceeded, and an XOFF indicator is set. The buffer will then empty until the XON level is reached again, and the cycle repeats.

Both in-band (IB) and out-of-band (OOB) flow control are supported. In-band flow control uses the reverse path of a bi-directional optical link, allowing the consumer to pace the amount of data transferred into its input buffer. An advantage of IB flow control is that no other connections are required; the flow control commands are sent over the reverse link from the consumer.

RULE 4.11: An ODI-1 consumer using in-band flow control SHALL use Interlaken Control Word Bit 55 to signal XON/XOFF.

OOB flow control uses a 1-wire signal, typically electrical, between the two devices. The signal may be present on a connector, or on the device's backplane. The level of the signal indicates XON or XOFF. The advantage of OOB flow control is that uni-directional devices may be used, lowering costs. Devices that use a modular backplane, such as AXIe, PXI, or VPX, may use a backplane "trigger" signal for OOB flow control.

RECOMMENDATION 4.2: A device that implements flow control SHOULD offer out-of-band flow control with an external electrical connector that indicates XON/XOFF for each optical port.

RECOMMENDATION 4.3: AXIe and PXI modular devices implementing ODI OOB flow control SHOULD implement electrical flow control using the backplane trigger lines, one line per optical port controlled.

RULE 4.12: If Out of Band flow control is implemented on a PXI or AXIe device, each flow control signal SHALL be programmable to use any of the backplane trigger lines.

RULE 4.13: When implementing XON/XOFF using a digital electrical signal, XON SHALL be indicated by a Logic High and XOFF SHALL be indicated by a Logic Low.

A producer cannot change between XON and XOFF until the completion of the current burst. Due to FPGA latencies, the size of the output buffer, and other pipeline processing, there can be a significant delay between when XON/XOFF is indicated and the FIFO sees the change in data transmission. ODI assumes a producer latency 450ns to accommodate pipeline processing. Once these latencies have been more thoroughly characterized, the required response times may change.

RULE 4.14: A producer with ODI flow control SHALL respond to an XON/XOFF indicator within 450 ns.

RULE 4.15: A consumer with ODI flow control SHALL set the XON level to accommodate up to 900 ns flow control reaction time.

RULE 4.16: A consumer with ODI flow control SHALL set the XOFF level to accommodate up to 900 ns flow control reaction time.

PERMISSION 4.2: A consumer with ODI flow control MAY choose a single level for XON and XOFF if the level meets the requirements for both.

OBSERVATION 4.3: The minimum FIFO size is 2 x 900 ns, multiplied by the data rate. AT 14.1 Gb/s line rate, this equates to 36,000 bytes.

OBSERVATION 4.4: In ODI-2, packet size is recommended to be larger than the minimum FIFO size of 36,000 bytes when using port aggregation.

4.1.8 Interlaken Property Summary

The Interlaken Alliance has published a document titled Interlaken Interoperability Recommendations v.1.11. It is available at

http://www.interlakenalliance.com/interlaken-interoperability-recommendations_v1.11.pdf

Section 2.2 of that document includes a table of recommendations for each Interlaken property. The following table lists each property, the Interlaken Alliance recommendations, and the ODI recommendations or requirements previously documented.

Property	Interlaken Recommendation	ODI Recommendation or Requirement
Backpressure Method	In-band.	In-band or Out-of-band. Single Interlaken channel only.
Channel count	Not specified, application dependent.	One Interlaken channel minimum.
Packet transfer method	Not specified, application dependent.	Packet mode.
Packet Mode Stop Boundary	For link level backpressure: Burst end. For channel backpressure: Packet end.	Burst End.
Burst Mode Stop Boundary	Burst.	Burst.
BurstMax / BurstMin / BurstShort	256 bytes / 64 bytes / 32 bytes	For 14.1 Gb/s line rate: 2048 bytes / 64 bytes / 64 bytes For 12.5 Gb/s line rate: 256 bytes / 64 bytes / 64 bytes
MetaFrameLength	2,048 Interlaken Words.	2,048 Interlaken Words.
Multiple use field	Not used.	Not used.
Rate matching	Yes, 1 Gb/s steps	Not used.
Status messaging	Not required.	Not required.
Retransmission	Optional, default is disabled	No retransmission.

Table 4-1: Interlaken Property Summary

4.2 ODI Packet Operation

ODI-1 defines a method for streaming data, regardless of packet format. Data is streamed through the transmission of consecutive packets. Packets are bracketed by Interlaken SOP (Start of Packet) and EOP (End of Packet) indicators. ODI-2 will add specific packet definitions based on VITA 49.2.

Transmitting data as a series of packets has several advantages. The packet itself may contain single channel or multi-channel data. The packet boundaries allow for error recovery. Data may be stored as packets, and packets are independent of the underlying transmission method. In ODI-2, VITA 49.2 packets allow port aggregation, synchronization, and time stamps, among other features.

4.2.1 ODI Generic Packet

A generic ODI packet is assumed to be packet data encapsulated by an optional prologue before the data and an optional trailer after the data. This is equivalent to specifying that the prologue and trailer may be any number of bytes, including zero. The packet itself is encapsulated by Interlaken SOP and EOP as shown in the diagram below.

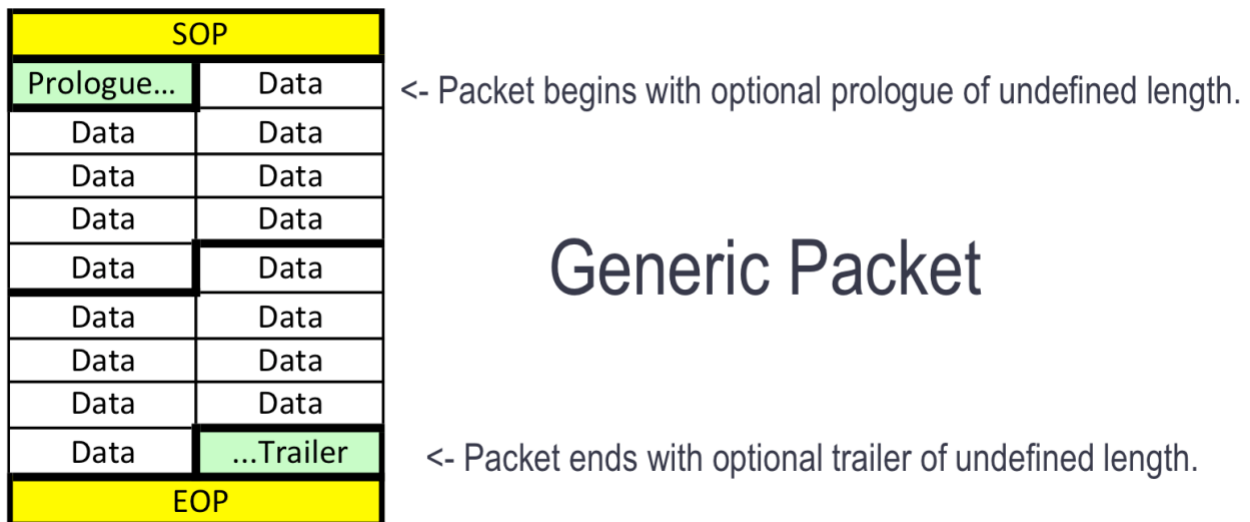


Figure 4-4: ODI Generic Packet

Packets are often described as in Figure 4-4, as a succession of Interlaken words, 64 bits wide each. Interlaken SOP is sent, followed by the optional prologue. 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. SOP and EOP may be

indicated together in a single Interlaken word, and is commonly done so for streaming data with the highest efficiency.

PERMISSION 4.3: A producer MAY indicate EOP for the previous packet and SOP for the following packet simultaneously within a single Interlaken Control Word.

4.2.2 ODI Packet Requirements

ODI-1 has two simple requirements for packets. The length must be between 64 bytes and 262,144 bytes, and an integer multiple of 32 bytes. The 32-byte requirement equates to 256 bits, and matches well with the device-facing bus of an FPGA, which is commonly a multiple of 256 lanes. The 64-byte requirement mandates a minimum packet length, which eases processing when multiple short packets are sent consecutively. The maximum length offers high efficiency while being long enough to transport the longest VRT packets, as specified in ODI-2.

RULE 4.17: An ODI packet SHALL be between 64 bytes and 262,144 bytes in length.

RULE 4.18: An ODI packet SHALL be an integer multiple of 32 bytes in length.

OBSERVATION 4.5: The maximum length of a VRT packet documented in VITA 49.2 and ODI-2 is 262,140 bytes. Therefore, the longest VRT packet may be transported over ODI-1.

OBSERVATION 4.6: It may be necessary to insert null data in order to meet the integer multiple of 32 bytes requirement.

4.3 ODI Data Streaming

Streaming is performed by transmitting a series of packets consecutively in a “train”. These packets are typically tens of thousands of bytes in length for maximum efficiency, and each is typically the same size as the other train packets.

When streaming ends, the final packet is called the caboose. It may be the same size as the previous packets, or it could be smaller.

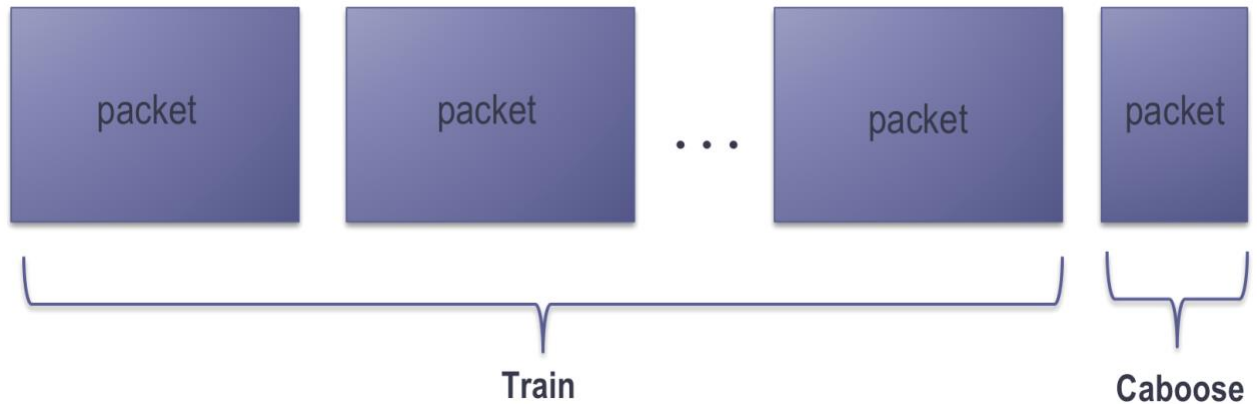


Figure 4-5: Streaming by Sending Consecutive Packets

OBSERVATION 4.7: An ODI-1 recording device can record a stream without knowledge of the format of the data packets, either for later analysis or for playback to another ODI device.

The producer has considerable flexibility in the length of each packet. It is up to the producer to choose packet lengths that will allow the full streaming speed to be supported over an ODI link. Typically, this is done by choosing packets sufficiently long so that the non-data overhead (e.g. packet prologue and trailer or Interlaken Control Words) is minimized.

Often, the producer can choose to make the Caboose the same length as the Train packets. A digitizer, for example, may send a Caboose packet of the same exact length as the others once it has received a “Stop Streaming” command.

The producer may also signal Interlaken SOP and EOP simultaneously between packets to reduce Interlaken overhead to a single control word.

4.3.1 Initiating Streaming Without Flow Control

The ODI specifications assume that certain sequences of commands are occurring at the system level. To begin streaming, the consumer is armed first. Once the consumer is armed, the producer is then armed. Finally, the producer is triggered, and streaming begins.

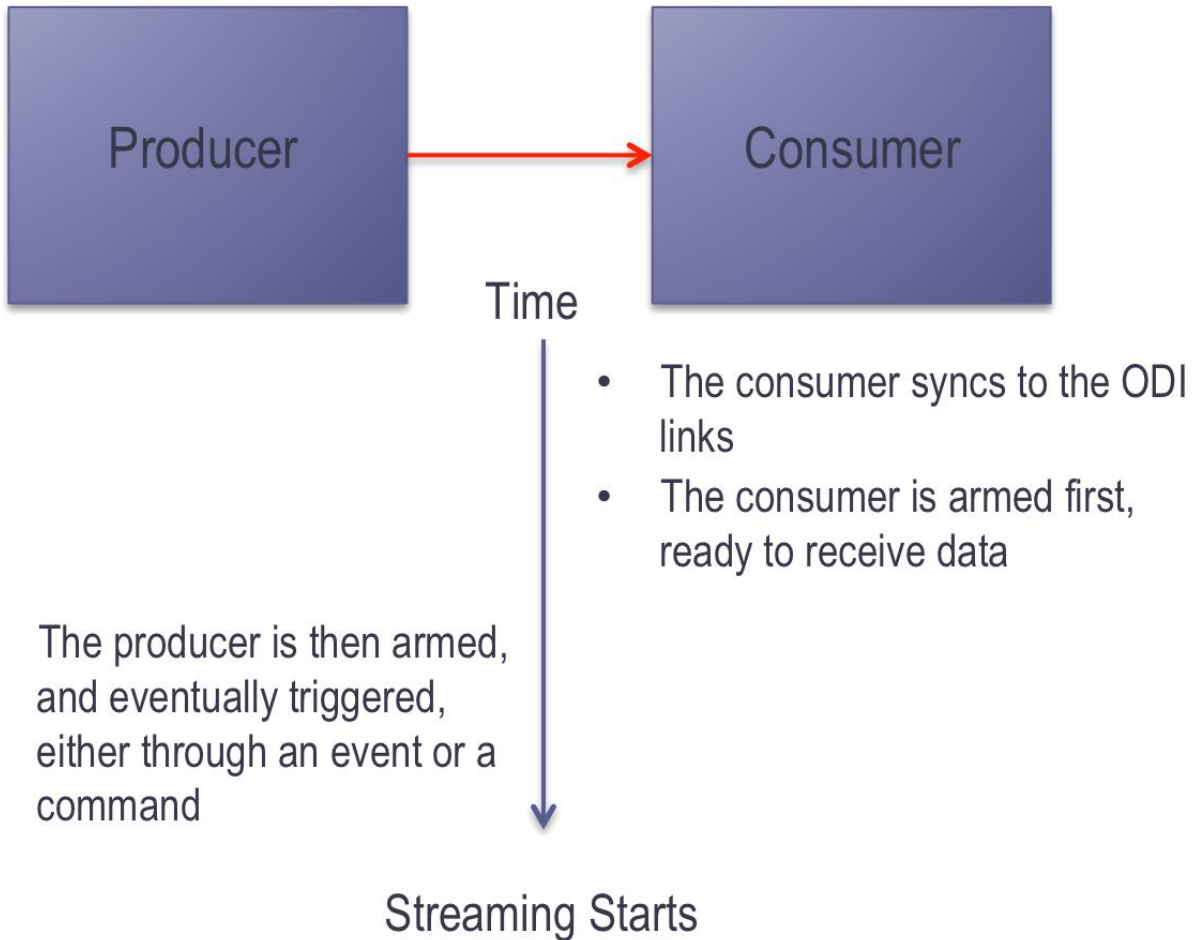


Figure 4-6: Time Diagram of Initiating Streaming Without Flow Control

4.3.2 Terminating Streaming Without Flow Control

To terminate streaming, the reverse of initiating streaming occurs.

A “stop event” is sent to the producer at the system level. This is typically a command, but could also be an electrical trigger signal or a value in the data. The producer sends the Caboose, and returns to the “Unarmed” state. Once confirmed, the system sends a Stop command to the consumer. The producer and consumer end in the unarmed state.

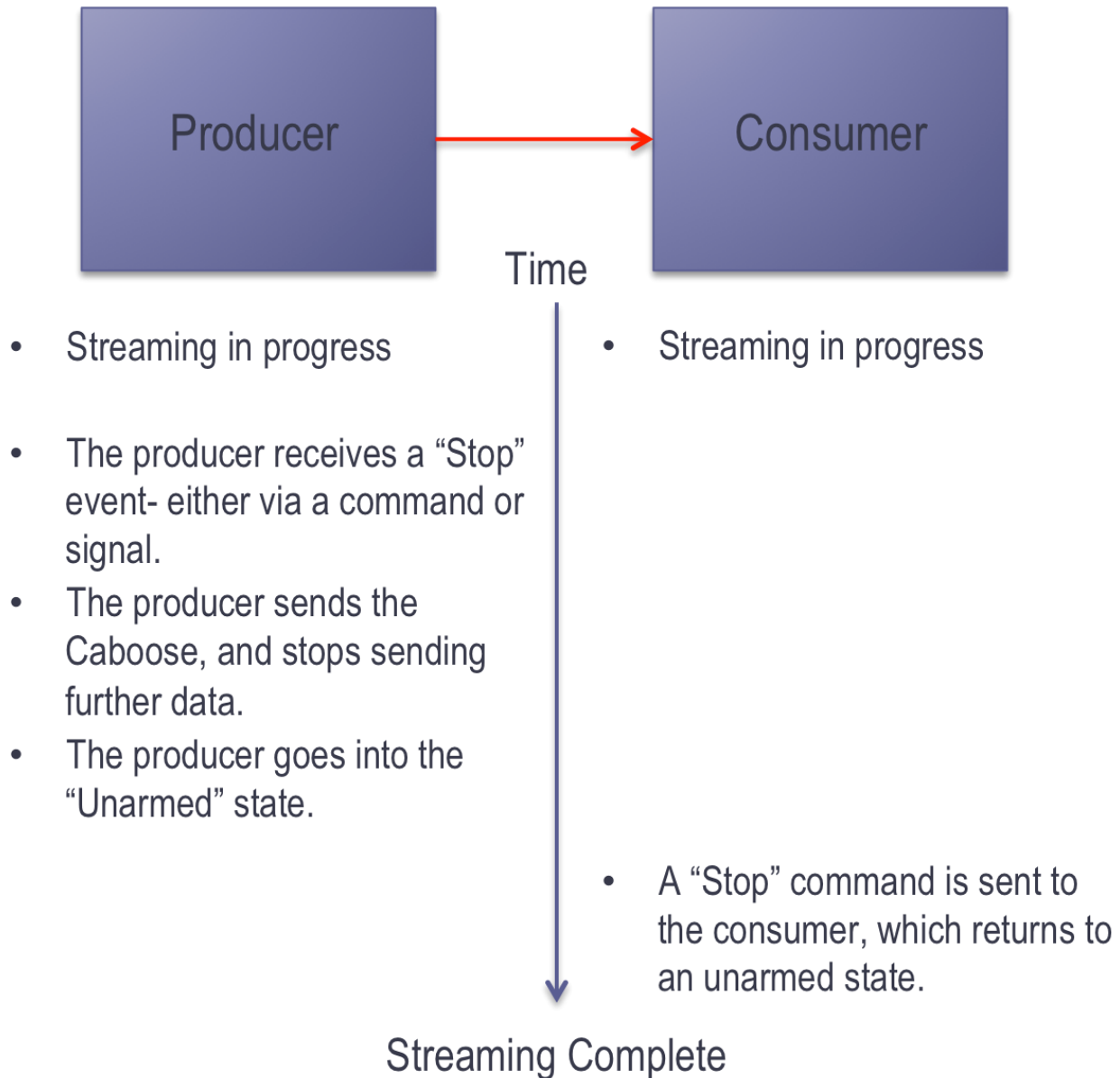


Figure 4-7: Time Diagram of Terminating Streaming Without Flow Control

4.3.3 Initiating Streaming With Flow Control

Flow control is performed by the Consumer indicating to the producer whether data is to be sent, or not. This is done by indicating XON (transmit) and XOFF (halt transmit) either through the reverse optical connection (in-band) or an external signal (out-of-band).

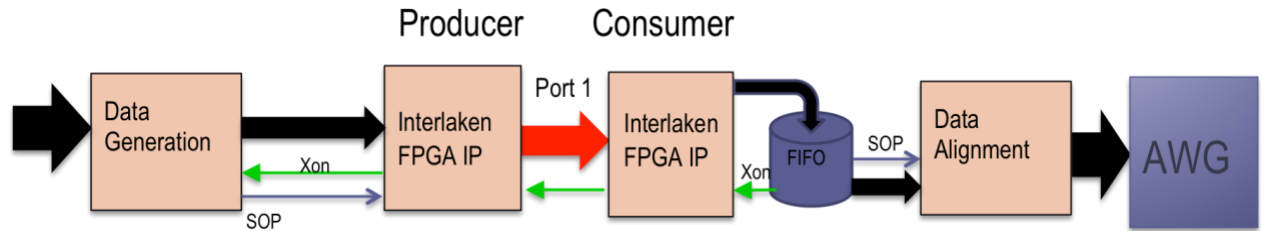


Figure 4-8: Example of Streaming With Flow Control to an AWG

In the above example, the producer is delivering data to the consumer, an AWG (Arbitrary Waveform Generator). The producer sends SOP to align the data freshly for every new packet. Data follows until the end of packet, where SOP is sent again for the next packet. The consumer, by monitoring the amount of data in the FIFO, indicates XON/XOFF to the producer. The Producer halts the sending of data when an XOFF is indicated, and resumes the sending of data when XON is indicated.

For streaming with flow control, the time sequence is similar to streaming without flow control, but the consumer's control of XON and XOFF determines when data is transferred. The four key events are:

- The consumer syncs to the ODI links, and indicates XOFF.
- The producer is enabled to send data whenever it sees XON.
- The consumer loads the data into its buffer, then indicates XOFF.
- The consumer will operate on the data when it is triggered.

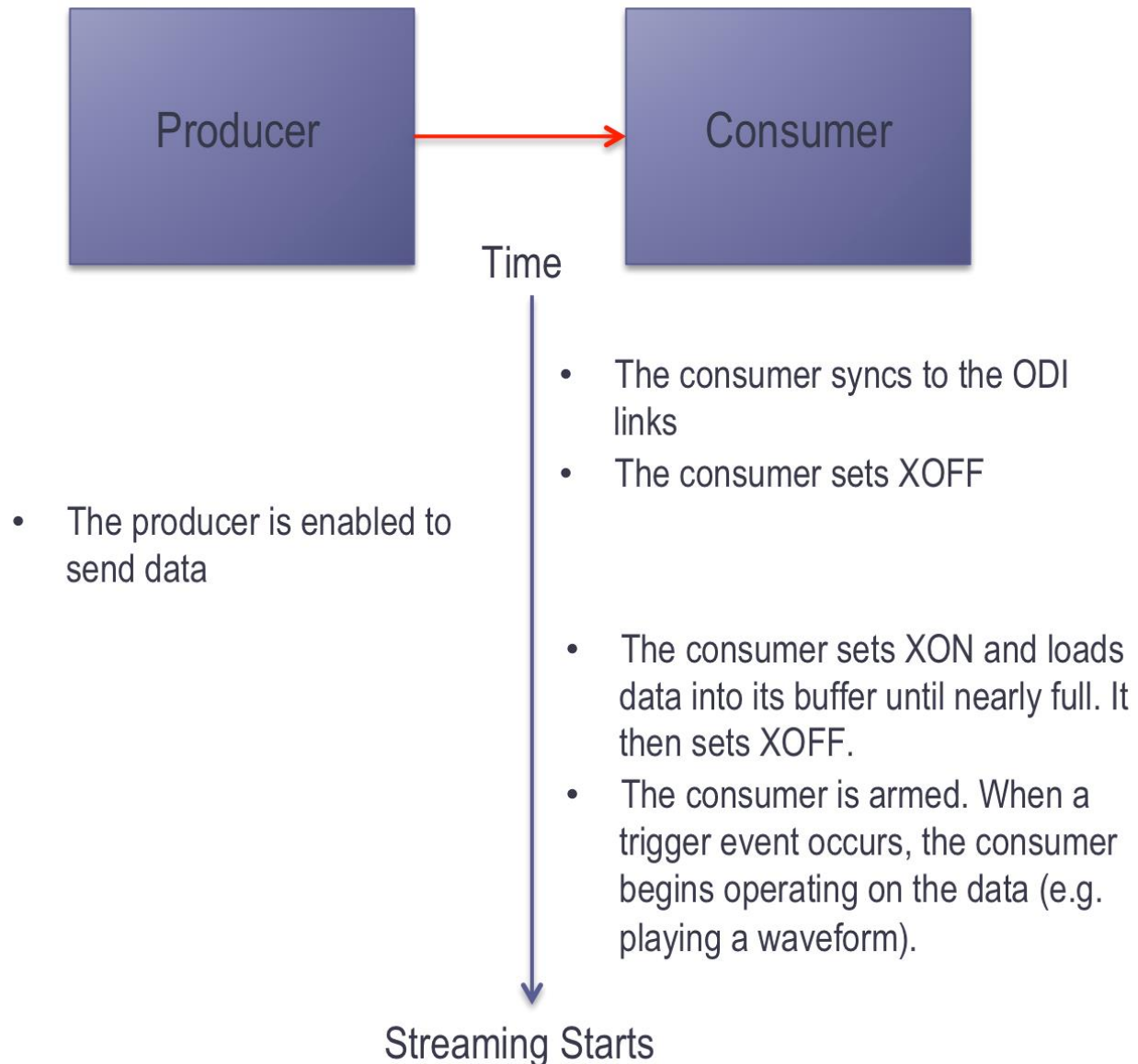


Figure 4-9: Time Diagram of Initiating Streaming With Flow Control

4.3.4 Terminating Streaming With Flow Control

During the streaming process, the consumer is managing its input buffer as shown in the time diagram below. The consumer is continuously processing the data in the buffer. The process can loop indefinitely until there is a STOP command to the consumer. At a convenient stopping point, the consumer will indicate one more "XOFF", and go into an unarmed state, completing the streaming process.

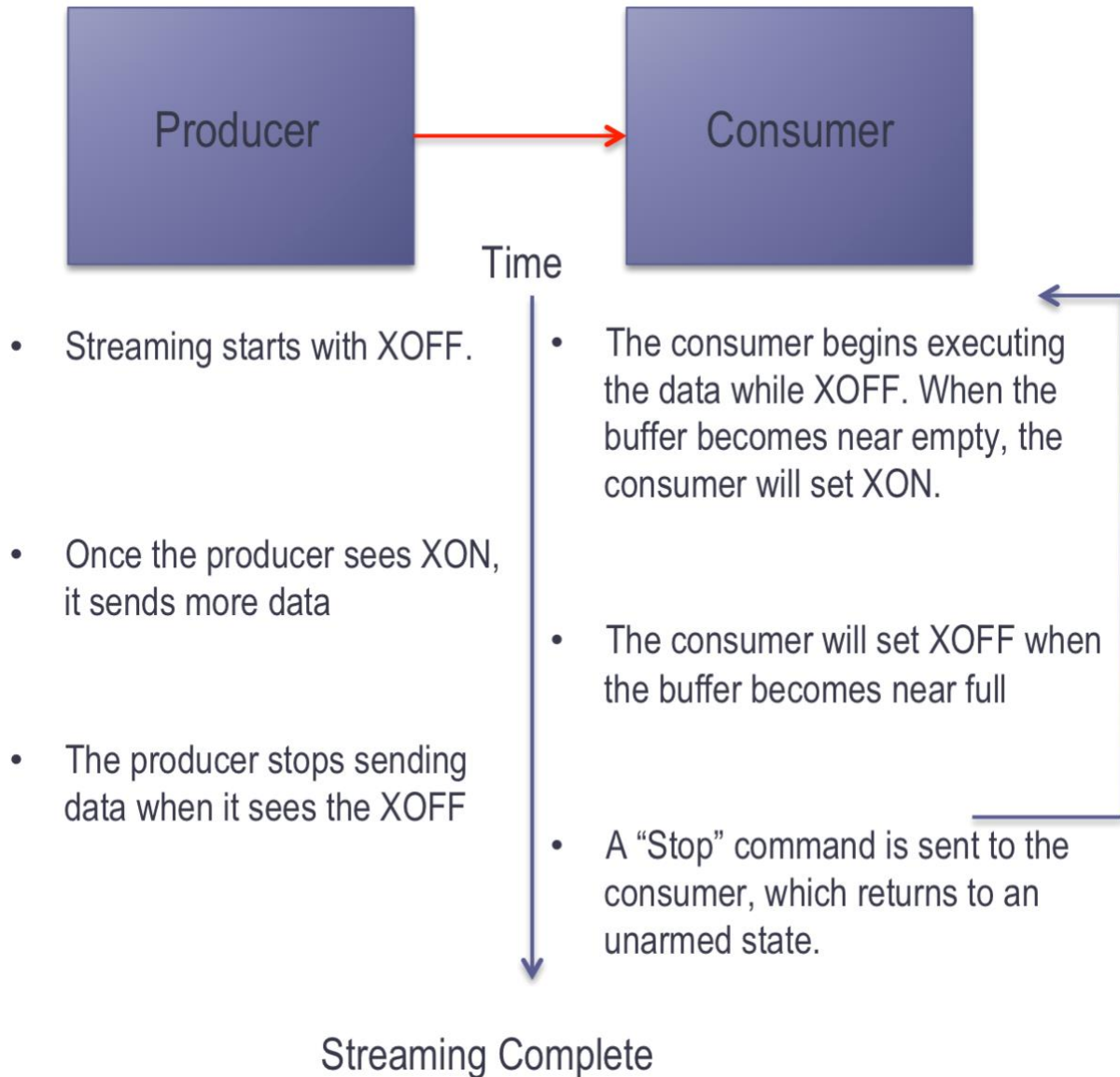


Figure 4-10: Time Diagram of Terminating Streaming With Flow Control

4.4 State Diagrams

ODI-1 state diagrams articulate the roles of the producer and consumer while streaming. The following terms are used to describe the events and processes during streaming.

- **Power/Reset** A Power-on event or Reset command
- **Stop** A Stop command unarms the consumer, typically at end of streaming

- **Tx Ready** If flow control is not being used, Tx Ready indicates the producer's transmitter is ready to transmit. If flow control is used, Tx Ready indicates the producer's transmitter is ready to transmit and the producer has received an XON indication.
- **Not Sync'd** Not Sync'd is a consumer state when the ODI links are not synchronized to the producer's
- **Rx Ready** Rx Ready is a consumer state when all lanes of the ODI link are synchronized and aligned to the producer's.
- **Arm** An Arm command is used to bring either the producer or the consumer into the Armed state. The consumer is ready to accept data once in the Armed state. The producer, in an Armed state, will send data once a Begin Stream event occurs.
- **Armed** The consumer is ready to accept data once in the Armed state. The producer, in an Armed state, will send data once a Begin Stream event occurs.
- **Begin Stream** Begin Stream is an event that initiates the producer to begin sending data. Begin Stream may be a command, a trigger event, or other event.
- **Xmit Train Packets** Xmit Train Packets is a process documented in the state diagrams where the producer sends (typically long) Train packets, one after the other, to the consumer. It will do so until an End Stream event occurs.
- **Xmit Caboose Packet** Xmit Caboose is a process documented in the state diagrams where the producer sends the final packet in a stream, called the Caboose. The caboose is typically the same length or shorter than the Train packets.
- **Rcv Packets** Rcv Packets (Receive Packets) is a process documented in the state diagrams where the consumer receives the packets in a stream. The consumer enters Rcv Packets mode after receiving the first SOP. It exits Rcv Packets mode only via the Stop command, usually sent after the stream is complete.
- **End Stream** End Stream is an event that signals the end of streaming. It can be a command from a controller, a trigger event, or generated by a storage device recognizing there is only one packet left to be transmitted.
- **SOP** Interlaken Start of Packet
- **EOP** Interlaken End of Packet

4.4.1 Overall Streaming State Diagrams

The overall sequence of events for streaming is documented in the state diagrams below. First, the ODI links between the producer and consumer are synchronized. When the links are synchronized, the consumer transitions to the Unarmed state. The producer will transition to the Unarmed state when Tx Ready occurs, defined by the Tx being ready to transmit for the non-flow control situation, or by the Tx being ready to transmit and it has received an XON indication for the flow control situation. At this point the producer and consumer are both in the unarmed state.

The subsequent Arm command to each will bring each to the Armed state. At this point ODI is interacting with the state diagram of the device, which may be the instrument state machine for a measurement instrument.

Once Armed, the consumer is ready to receive packets. Once Armed, the producer waits for a Begin Stream event, which may either be command or some sort of trigger event.

When the Begin Stream event occurs, the producer enters the Xmit Train Packets state and sends consecutive packets indefinitely (until an End Stream event). The producer will begin the packet transmission with an Interlaken SOP. When the consumer detects the SOP, it enters the Rcv Packet state, and receives packets continuously until a Stop command.

When the producer receives an End Stream event, it sends the final packet, called the Caboose. When the Caboose is sent, the producer returns to the Unarmed state. The consumer receives packets as long as it is in the Rcv Packets, without distinguishing the Caboose from the Train packets. It remains in Rcv Packet state even though no further packets are sent. It will return to the Unarmed state once it receives a Stop command. Streaming is complete.

A Power/Reset event will return the producer and consumer to the Sync Links state regardless of which state the devices previously were in. Once synchronized, each will transition to the Unarmed state. If the links lose synchronization at any time, this is detected by the consumer, which enters the Sync Links state.

The process of Xmit Train Packets, Xmit Caboose, and Rcv Packets are documented in further state diagrams.

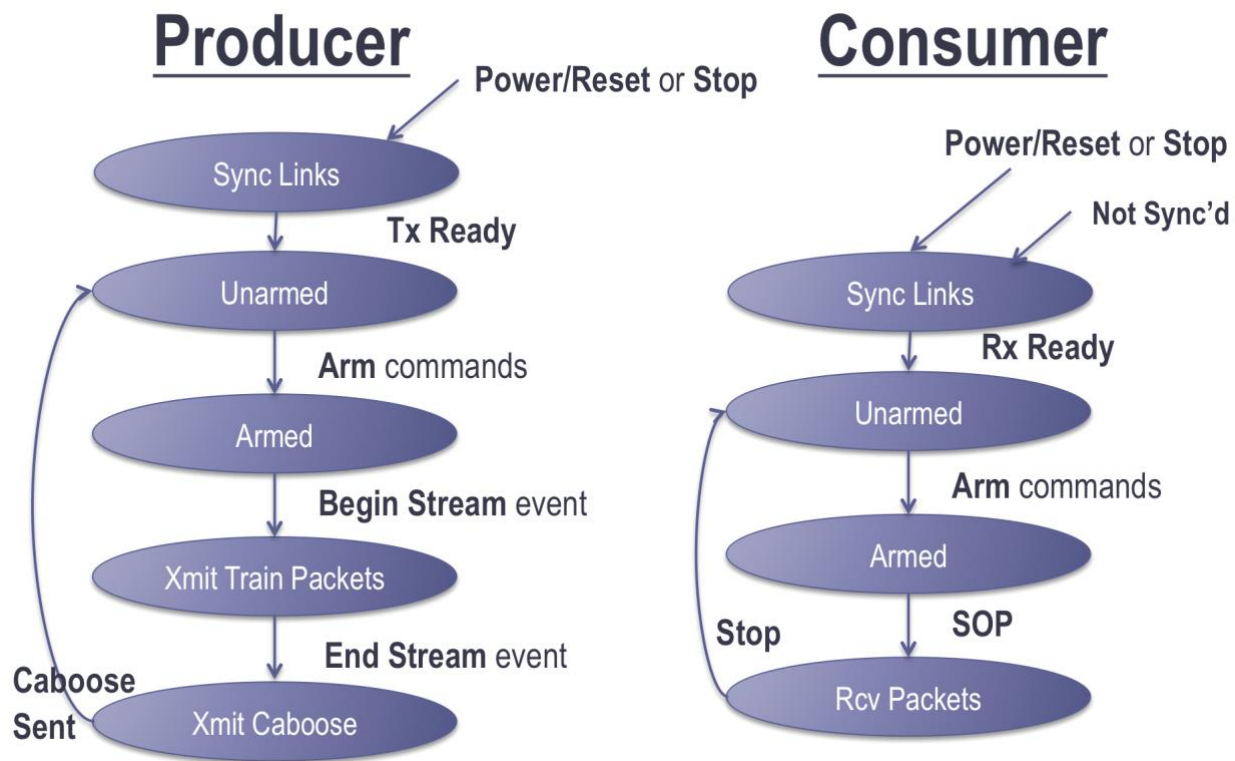


Figure 4-11: Overall Streaming State Diagram

RULE 4.19: An ODI producer or consumer SHALL comply with the state diagram documented in the Figure 4-11 Overall Streaming State Diagram.

OBSERVATION 4.8: The sequencing of start-up between the Producer and the Consumer is not defined. Therefore, the consumer must be tolerant of timing of a valid producer output which may be before or after the consumer enters the Sync Links state. It is possible that the producer could temporarily source invalid outputs as it is starting up, and the consumer needs to be tolerant of this situation. Also, it is possible that the link could be interrupted during operation. Again, the consumer should detect this and return to the Sync Links state.

4.4.2 Xmit Train Packets State Diagram

The Xmit Train Packets process is executed by the producer and is documented in the state diagram below. It shows three states for sending a single packet- the Prologue, the data payload, and the Trailer. The process is repeated for each packet.

The producer enters the Xmit Train Packets process after it receives a Begin Stream event, as documented in the Overall Streaming State Diagram.

Once the data becomes ready, the producer is ready to transmit a packet. It sends SOP and the optional Prologue to the packet. If the packet does not contain a Prologue, it will send the data payload instead. While the diagram shows SOP + Prologue is sent, this is actually the interaction with the Interlaken IP of the FPGA. The IP will sequence SOP first, followed by the optional Prologue or data.

The process continues to send data within the data payload until it nears the end of the data payload.

At the end of the packet, the producer sends the optional Trailer along with EOP. If there is no Trailer, the final data sent is the last elements of the data payload. While the diagram shows Trailer + EOP is sent, this is actually the interaction with the Interlaken IP of the FPGA. The IP will sequence the optional Trailer or end of data payload first, followed by the EOP indicator.

While SOP and EOP are shown as independent actions, the two may be indicated together within a single Interlaken Control Word to decrease overhead.

To meet the multiple-of-32-bytes requirement, the producer may add null data.

The state diagram is valid for both, flow control and no flow control, applications.

Xmit Train Packets State Diagram

Producer

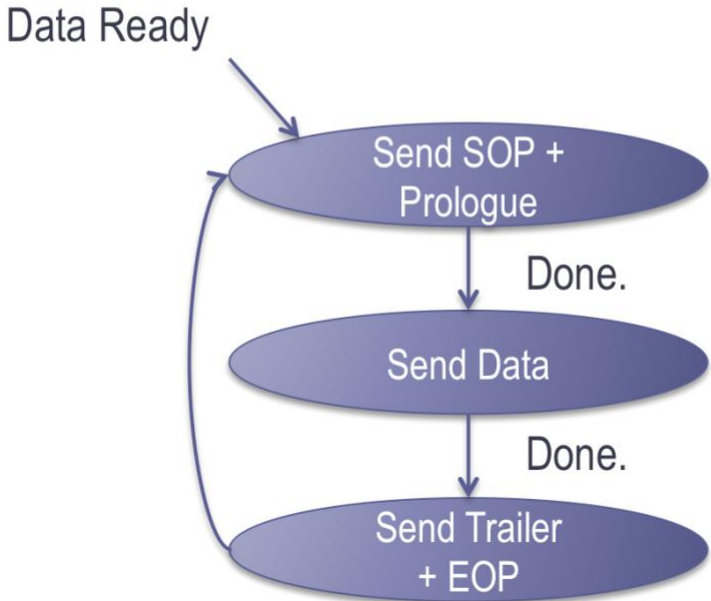


Figure 4-12: Xmit Train Packets State Diagram

RULE 4.20: An ODI producer SHALL comply with the state diagram documented in the Figure 4-12 Xmit Train Packets State Diagram

4.4.3 Xmit Caboose Packet State Diagram

The Xmit Caboose Packet process is executed by the producer and is documented in the state diagram below. It is the process for sending the final packet in a stream. It is identical to the Xmit Train Packets process, except that it is executed only once, and then the producer returns to the Unarmed state as documented in the Overall Streaming State Diagram.

Xmit Caboose Packet State Diagram

Producer

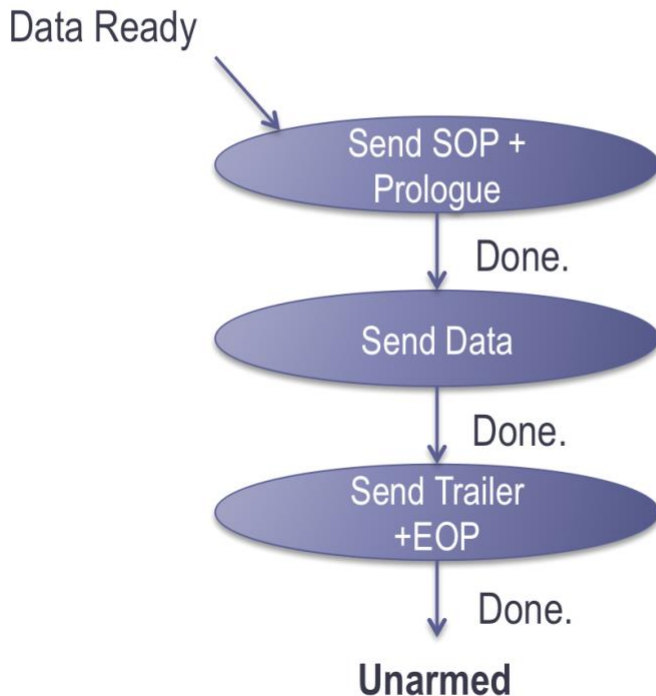


Figure 4-13: Xmit Caboose Packet State Diagram

RULE 4.21: An ODI producer SHALL comply with the state diagram documented in the Figure 4-13 Xmit Caboose Packet State Diagram

4.4.4 Rcv Packets State Diagram

The Rcv Packets process is executed by the consumer and is documented in the state diagram below.

The consumer enters the Rcv Packets process when it is armed and receives an SOP indicator as documented in the Overall Streaming State Diagrams. At that time it treats any data as the beginning of a packet, and proceeds to handle the data. Handling the data may include storing the data, processing the data, or creating one or more signals from the data. The handling of the data is specific to the device.

When a single packet is complete, the producer will send the EOP indicator. This may be coincident with SOP if the producer intends to send a successive packet. In that case, EOP and SOP are sent simultaneously in the Interlaken Control Word, and the consumer proceeds to handle a new packet.

If only the EOP is sent, as indicated in the EOP and NSOP (Not Start of Packet) event, the consumer completes handling the previous packet and enters the Wait state. If the producer sends another SOP, the consumer proceeds to handle the new packet. If no SOP is sent, the consumer will stay in the Wait state indefinitely, until a Stop command is received. At that point, the consumer returns to the Unarmed state. This is the method for completing the streaming process.

If the consumer receives an SOP indicator but without EOP (indicated by NEOP, Not End of Packet), then an error has occurred. In this case the consumer begins the new packet. The specifics of how the consumer handles the error is not specified in ODI-1, other than beginning a new packet.

Rcv Packets State Diagram

Consumer

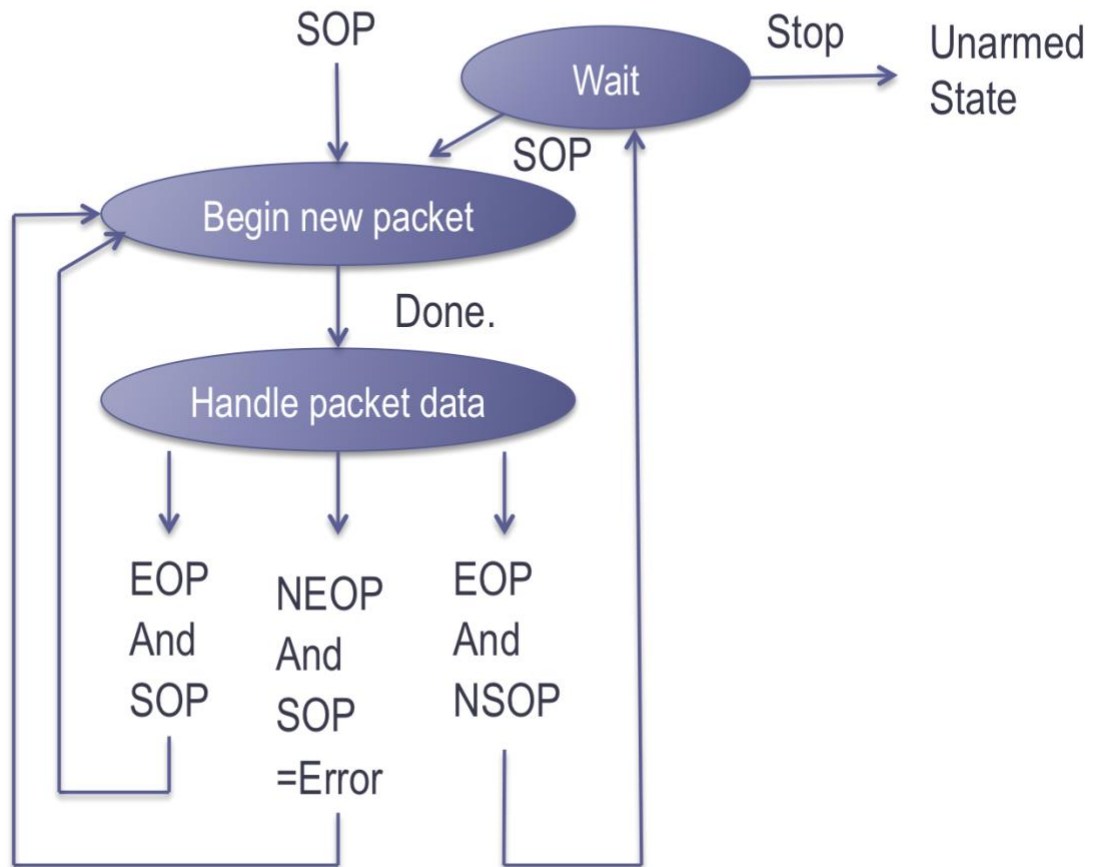


Figure 4-14: Rcv Packets State Diagram

RULE 4.22: An ODI consumer SHALL comply with the state diagram documented in the Figure 4-14 Rcv Packet State Diagram

4.5 ODI Test Pattern

RECOMMENDATION 4.4: An ODI device SHOULD implement the ODI Test Pattern.

The ODI Test Pattern is a standardized test pattern used to verify operation between a producer and a consumer. It delivers value in two situations- development and user integration. During development a producer being developed can be tested against a known good consumer, or a consumer can be tested against a known good producer. During integration, a user can verify the operation of the links between a producer and a consumer. A link may be the 12-lane connection transferring signal data from the producer to the consumer, or it may be the flow control connection from the consumer to the producer. The ODI Test Pattern enables these tests without disturbing the optical cabling.

The pattern is effective in testing that:

- All lanes sync up and are operable
- All lanes are oriented in the right order
- Interlaken words can be transferred at speed

The pattern itself is a 32-bit counting pattern where the least significant nibble (nibble equals 4 bits) sequences through 16 hexadecimal values in the order prescribed below. The more significant nibbles sequence in counting order.

xxxxxxx3	xxxxxxx2
xxxxxxx1	xxxxxxx0
xxxxxxx7	xxxxxxx6
xxxxxxx5	xxxxxxx4
xxxxxxxB	xxxxxxxA
xxxxxxx9	xxxxxxx8
xxxxxxxF	xxxxxxxE
xxxxxxxD	xxxxxxxC

<- One 64-bit Interlaken word consists of two 32-bit counting patterns.

Each counting pattern consists of the least significant nibble sequencing through 3-2-1-0-7-6-5-4-B-A-9-8-F-E-D-C while the seven most significant nibbles sequence through numerical counting order, starting with 0000000 and ending with FFFFFFFF.

Figure 4-15: Sequence order of least significant nibble of ODI Test Pattern

The unique sequence of the least significant nibble is an artifact of a simple implementation in commonly available FPGA designs. Essentially, the two LSBs count down, while the more significant bits count up. This will be examined later in this section.

The ODI Test Pattern is transmitted as a series of 16384-byte payload Interlaken packets. This equates to 2048 Interlaken words. The first packet is sequenced as shown in the figure below. The pattern begins with the value 00000003 and finishes with the value 0000FFC.

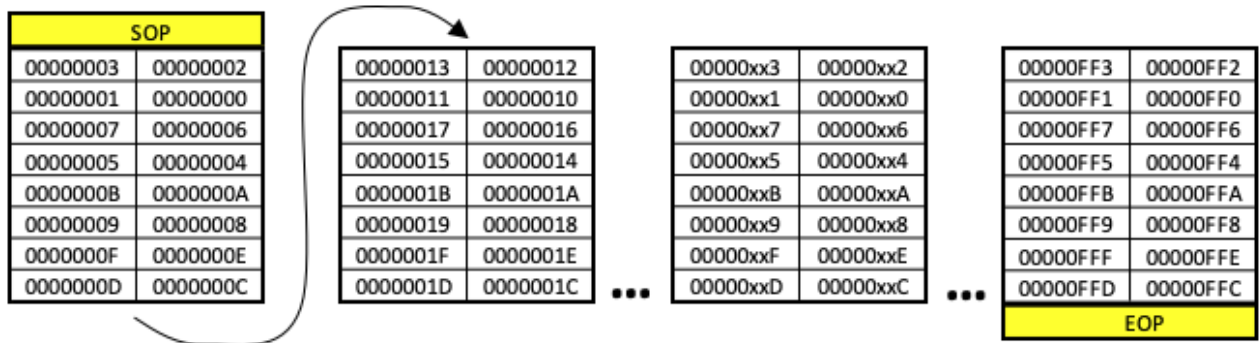


Figure 4-16: First packet of the ODI Test Pattern

In the figure above, the counting sequence becomes evident. The value of the seven most significant nibbles is incremented by one each time the least significant nibble sequences through all 16 values. 2048 Interlaken words equates to 4096 values and 16384 bytes for the data payload of each packet. Within a packet the values xx begin with 00, are incremented each time the least significant nibble sequences through all 16 values, and finally end with FF.

For the pattern to continue in subsequent packets, the fourth least significant nibble is incremented, and the three least significant nibbles repeat the sequence executed in the first packet. This is merely a continuation of the counting pattern in subsequent packets.

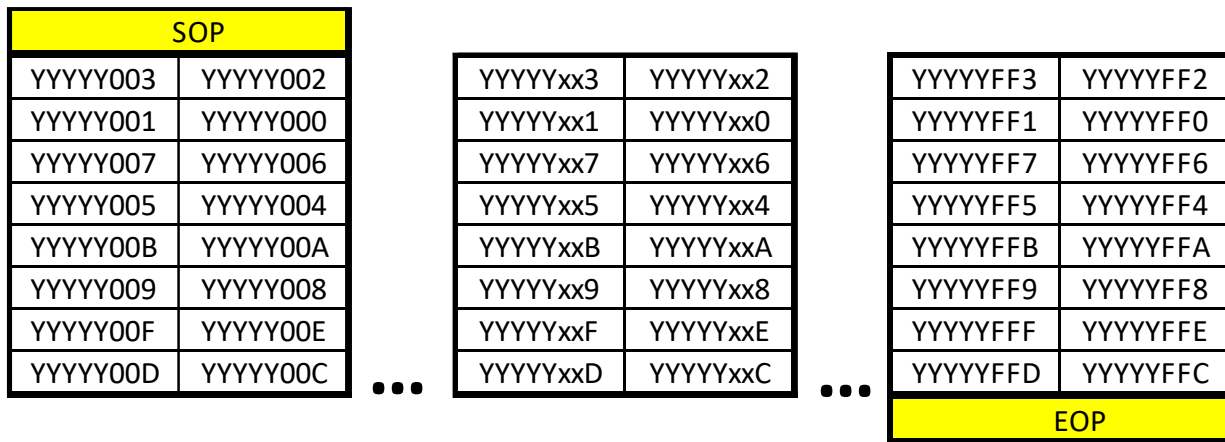


Figure 4-17: Middle packets of the ODI Test Pattern

Any packet between the first and the final packet of the ODI Test Pattern will have the five most significant nibbles constant, as indicated by the YYYYY value in the figure above. The values xx begin with 00, are incremented each time the least significant nibble sequences through all 16 values, and finally end with FF. The next packet will begin with the five most significant nibbles set to YYYYY+1, and the pattern repeats.

The complete ODI Test Pattern has 1,048,576 packets. The final packet begins with a value of FFFFF003 and ends with a value of FFFFFFFC, as shown in the figure below.

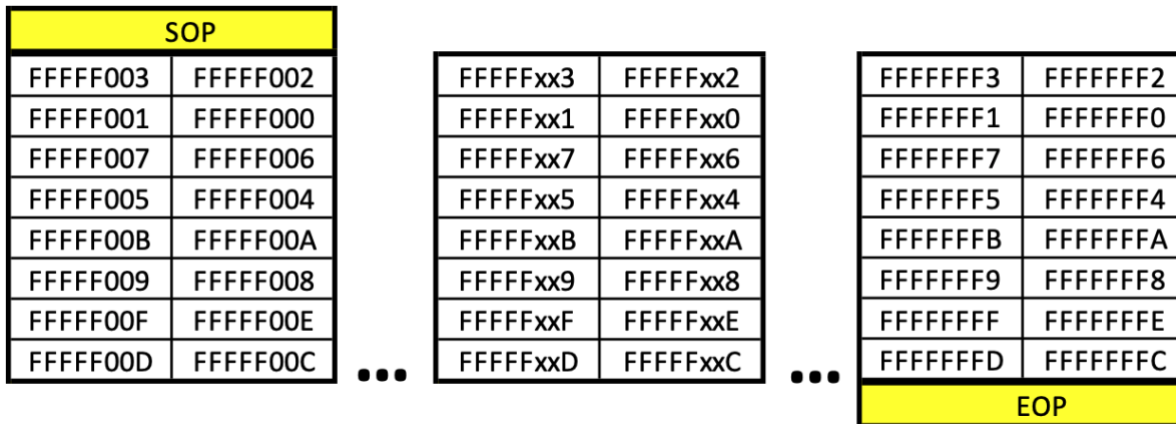


Figure 4-18: Final packet of the ODI Test Pattern

The complete ODI Test Pattern has approximately 17.18 GBytes of payload data. It requires slightly more than one second to complete at the 12.5Gb/s line rate, and slightly less than one second at 14.1Gb/s line rate.

To use the test pattern, the user commands the receiving device to prepare for the pattern. The user then commands the transmitting device to send the pattern. As the default option an ODI device must be able to execute the pattern as a single occurrence, but it may be programmed to repeat the pattern continuously.

4.5.1 FPGA Implementation

FPGAs are often characterized by deploying a wide internal bus, also known as a Local Bus or LBUS. These wide internal buses are clocked at a rate lower than the ODI line rates, but the width of the LBUS leads to the same aggregate data rate. These local buses are typically a power of two in width (e.g. 256 bits, 512 bits, or 1024 bits). These local buses are typically arranged as multiple segments of smaller buses, e.g. four segments of 128 bits each for a 512-bit wide bus.

Since the test pattern is essentially a 32-bit counter, several pattern values are loaded onto each local bus and each segment simultaneously and clocked in all together during a single bus cycle. The pattern itself is logically optimized for a 512-bit Local Bus organized as four 128-bit segments, though any combination of local bus width and segment width may be accommodated.

The figure below shows implementation of the ODI Test Pattern on a 512-bit Local Bus comprising four 128-bit segments.

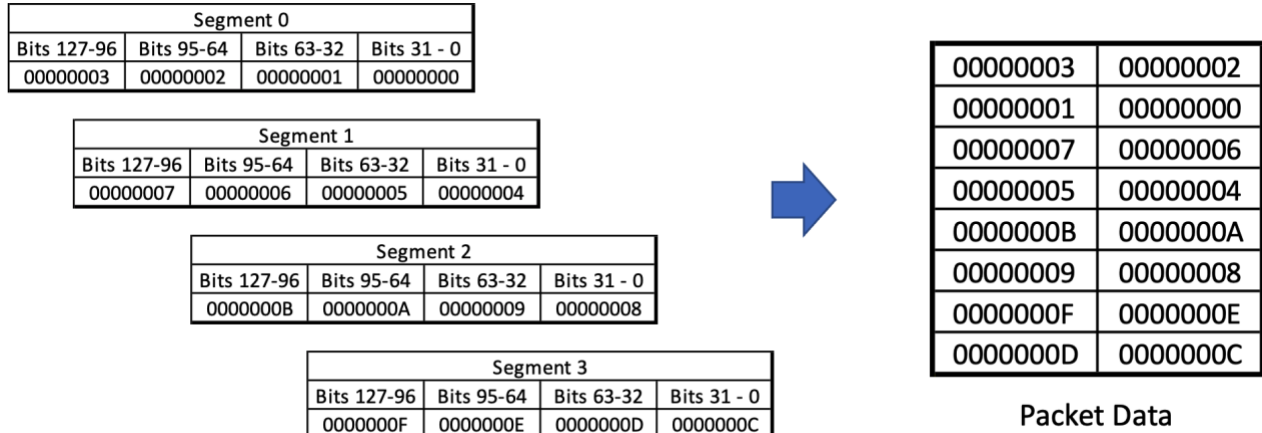


Figure 4-19: Example FPGA Implementation of first 64 bytes of ODI Test Pattern

In the figure above Segment 0 is transmitted first, then Segments 1, 2, and 3. The first byte sent consists of bits 127-119. Due to the sequence of transmitting bytes from the higher-numbered bits of each segment first, but in the order of the segments, the least significant nibble sequence of 3-2-1-0-7-6-5-4-B-A-9-8-F-E-D-C is realized. 64 bytes are transmitted in the figure above.

While the least significant nibbles must be loaded into the segments at the same location for each clock cycle, the seven most significant nibbles are loaded based on a counter that increments after each loading of these 64 bytes. The second 64 bytes would be loaded with the values below:

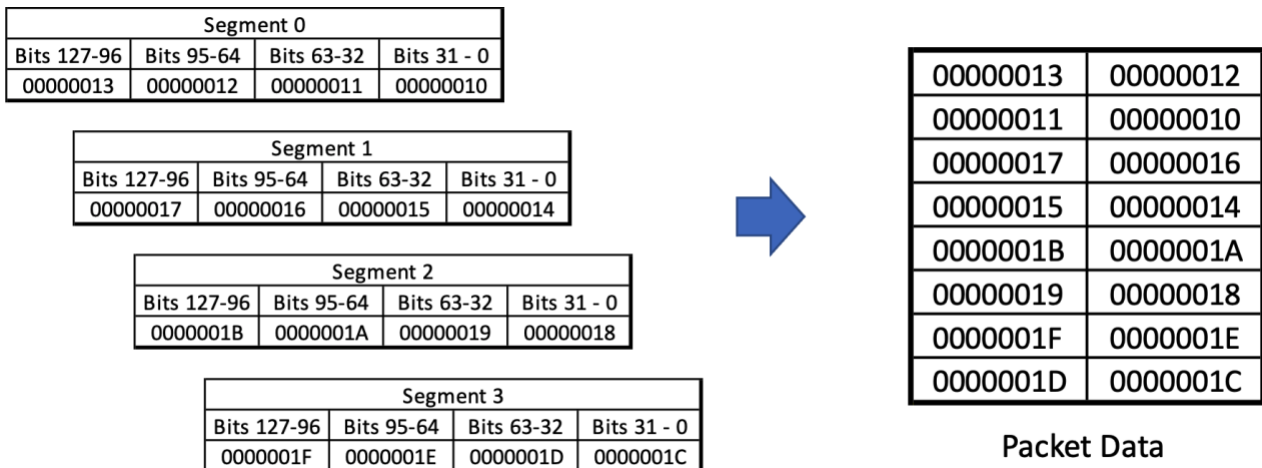


Figure 4-20: Example FPGA Implementation of second 64 bytes of ODI Test Pattern

Similarly, all subsequent loading of 64 bytes would follow the same pattern.

5. ODI-1 Documentation Requirements

ODI-1 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 device SHALL document which ODI specifications and associated revision it complies to.

RULE 5.2: An ODI device SHALL document the ODI-1 line rates and Interlaken BurstMax that are used.

RULE 5.3: An ODI device SHALL document which In Band and Out of Band flow control methods are used, if any.

ODI uses a simple speed calculation to determine speed compatibility between devices. It does this by requiring each device to document both, its requirements and its capability in terms of speed per port, measured as GB/sec (GBytes/sec). ODI uses a term called equivalent GB/sec. For devices that handle data as bytes, equivalent GB/sec is no different than the raw GB/sec required. For devices that handle data on non-byte boundaries, equivalent GB/s is merely the required Gb/sec divided by eight. Essentially, equivalent GB/sec measures the total speed of the ODI link, regardless how the data is packed.

For example, a 4-channel 16-bit digitizer operating at 1 Gsample/sec would specify 8 GB/sec required. A storage device may be capable of receiving data at 10GB/sec. In this case, it would have sufficient speed for this specific digitizer. For multi-port devices, speed is specified by the per-port requirements and capabilities. If that same digitizer above uses two ports, it would require just 4 GB/sec per port.

The rules for documentation of speed capabilities and requirements are below:

RULE 5.4: An ODI device SHALL document its data bandwidth requirements and capabilities per port in units of equivalent GB/sec for each operating mode.

RULE 5.5: An ODI device SHALL document any other requirements in order to reach the quoted aggregate bandwidth.

6. Appendix A: ODI Speed Calculations

For 14.1 Gb/sec line rates, the maximum speed calculation follows:

14.1 Gb/sec x 12 lanes / 8 bits/byte => 21.15 GB/sec raw channel speed

64/67 coding => 95.52% efficiency => 20.20 GB/sec coded speed

256 Word (2K byte) burst framing => 256/257 words => 99.61% efficiency
=> 20.12 GB/sec framed speed

Alignment efficiency = 100% => 20.12 GB/sec aligned framed speed

2048 Word Metaframing => 2044 Words/2048 Words => 99.8% efficiency
=> 20.09 GB/sec total speed.

Packet inefficiency must be added to this result.

For 12.5 Gb/sec line rates, the maximum speed calculation follows:

12.5 Gb/sec x 12 lanes / 8 bits/byte => 18.75 GB/sec raw channel speed

64/67 coding => 95.52% efficiency => 17.91 GB/sec coded speed

256 byte burst framing => 256/264 bytes => 96.97% efficiency
=> 17.37 GB/sec framed speed

Alignment efficiency = 100% => 17.37 GB/sec aligned framed speed

2048 Word Metaframing => 2044 Words/2048 Words => 99.8% efficiency
=> 17.33 GB/sec total speed.

Packet inefficiency must be added to this result.

Packet inefficiency

Packet inefficiency may exist due to the packet structure. There are two sources of packet inefficiency: Packet overhead and EOP. Below are the methods for calculating Packet Efficiency, which can then be used to de-rate the above speeds.

Packet Overhead inefficiency is due to the finite overhead of the Prologue and Trailer of the packet, when compared to the data payload.

POE (Packet Overhead Efficiency) =
 Data Payload Length / (Data Payload Length + Prologue Length + Trailer Length)

EOP (End of Packet) inefficiency occurs when the Packet Length is not an integer multiple of Interlaken BurstMax. In this case, the final Burst is shorter than BurstMax, so that Burst was not delivered as efficiently as the others. In the worst case scenario, an additional Interlaken Control Word is sent for a single byte. Below is the calculation of the efficiency for sending an extra control word with 0 additional bytes, bounding the worst case inefficiency:

EOP0 (EOP efficiency for extra control word with 0 additional data payload bytes) =>

$$EOP0 = \frac{(BurstMax + 8)}{BurstMax + 8 + (8 * BurstMax/PL)}$$

where PL = Packet Length in bytes, and BurstMax is in bytes.

Divide numerator and denominator by (*BurstMax* + 8) to get:

$$EOP0 = \frac{1}{1 + \frac{8 * BurstMax}{PL (BurstMax + 8)}}$$

Since $1/(1+x) \approx 1-x$ when x is small,

$$EOP0 \approx 1 - \frac{8}{PL * \left(1 + \frac{8}{BurstMax}\right)} \approx 1 - \frac{8}{PL} \left(1 - \frac{8}{BurstMax}\right) \approx 1 - \frac{8}{PL}$$

This is the worst case calculation assumes an extra Interlaken Control Word of 8 Bytes over one entire packet with no additional data. As the packet length increases, the last burst becomes more efficient, until the last burst reaches the size of BurstMax. At that point there are no End of Packet inefficiencies. An implementer may choose a packet length such that the packet length is an integer multiple of BurstMax, eliminating EOP inefficiencies:

IF $PL \bmod BurstMax = 0$ **THEN** $EOP = 1$

If the packet length is not an integer multiple of BurstMax, then the efficiency becomes:

$$\text{IF } PL \bmod BurstMax \neq 0 \text{ THEN } EOP \approx 1 - \frac{8}{PL} \left(1 - \frac{PL \bmod BurstMax}{BurstMax}\right)$$

An implementer may calculate the inefficiencies for a particular packet length using the formula above. Alternatively, an implementer may merely calculate the worst case EOP for a range of packets with a minimum packet length. In that case, the worst case efficiency is EOP0:

$$EOP0 \approx 1 - \frac{8}{PL}$$

Packet Efficiency Example Calculation

The calculations for POE and EOP0 above show that efficiency is increased by using longer packet lengths. Below are example calculated efficiencies of using 32 bytes of overhead with a packet length of approximately 32K Bytes, without knowing the exact length of the packet.

$$\begin{aligned} POE &= 32K / (32K + 32) = 99.9\% \\ EOP0 &= 99.975\% \end{aligned}$$

These efficiencies will de-rate the speeds calculated for the 14.1 Gb/s and 12.5 Gb/s line rates. The example speeds become:

For 14.1 Gb/sec line rates, packetized speed:

$$\Rightarrow 20.09 \text{ GB/s} * POE * EOP0 = 20.09 \text{ GB/s} * 0.999 * 0.99975 = 20.065 \text{ GB/s}$$

For 12.5 Gb/sec line rates, packetized speed:

$$\Rightarrow 17.33 \text{ GB/s} * POE * EOP0 = 17.33 \text{ GB/s} * 0.999 * 0.99975 = 17.31 \text{ GB/s}$$