



Flash Memory Summit

Overview of TL and DL Layer specifications

Bruno Mesnet, IBM Client Engineering
OpenCAPI - OMI enablement



Just Simple and Open

Overview



Flash Memory Summit

OpenCAPI in POWER10

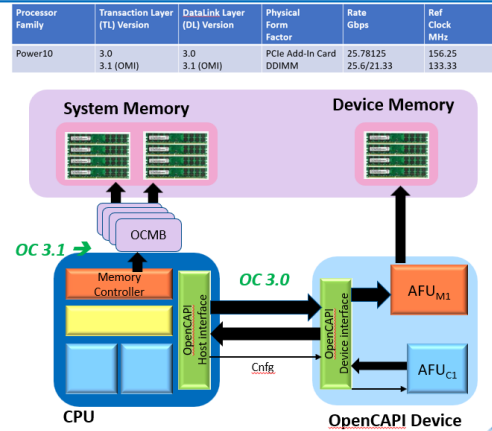


OpenCAPI attach capabilities are broken into two subclasses

- Compute (AFU_C) for function acceleration using a more traditional IO model with DMAs mastered by the device
 - Memory (AFU_M) for attaching various memory technologies using Loads / Stores mastered by the host
- OpenCAPI 3.0 – 25 Gbps (P9+P10)
- AFU_{C1}, AFU_{M1}
 - AFU_{C1} can perform non-cacheable DMA Reads and Writes to system memory
 - AFU_{M1} is accelerator device memory that is part of the overall coherent system address map.

Power 10: OpenCAPI 3.1 @ 25.6 Gbps

- OpenCAPI Memory Interface (OMI)



4 | ©2022 Flash Memory Summit. All Rights Reserved.

Specifications

<https://openmemoryinterface.org/>



The OMI architecture specification is a subset of the OpenCAPI bus and includes:

- OpenCAPI Transaction Link (TL) Architecture Specification 3.1
- OpenCAPI Data Link (DL) Architecture Specification.

The TL Architecture Specification 3.1 is highly tuned based on decades of experience in making a 'clean sheet architecture' by making tradeoff decisions to focus the architecture for a direct link to memory. The specifications can be downloaded here [OpenCAPI Consortium: Official Site](https://openmemoryinterface.org/).

2

PHY SIGNALING SPECIFICATIONS

The OMI bus protocol will be running over a 32Gbps PHY. Equally important to the OM

- OpenCAPI 32Gbps PHY Signaling Specification

This PHY Signaling Specification will provide you further guidance in how the TL will int here [OpenCAPI Consortium: Official Site](https://openmemoryinterface.org/).

Please download the standards below:

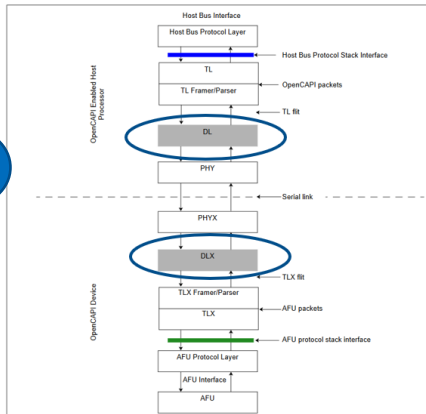
- OpenCAPI 3.0 Transaction Layer Specification
- OpenCAPI 3.1 Transaction Layer Specification
- OpenCAPI 4.0 Transaction Layer Specification
- OpenCAPI Data Link Layer Specification**
- OpenCAPI 32Gbps PHY Signaling Specification
- OpenCAPI 32Gbps PHY Mechanical Specification

7 | ©2022 Flash Memory Summit. All Rights Reserved.

DL Architecture specification (DL3.0/3.1/4.0)



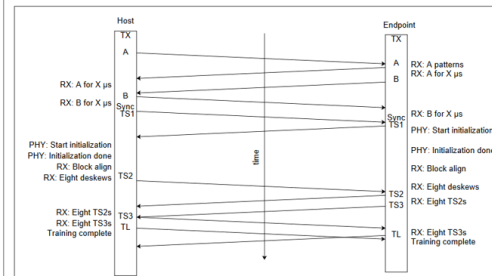
Figure 1-1. OpenCAPI stack



Same spec for 3.0/3.1/4.0 but with some "categories"

Same for host and device: DL=DLX

Figure 2-1. Training exchange between the host and endpoint



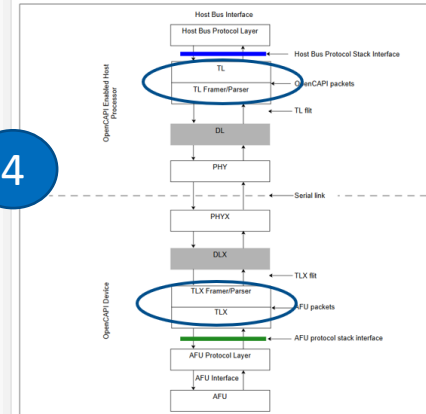
Note: The term "X µs" indicates a configurable number of microseconds as defined Section 2.1 PHY training on page 20.

9 | ©2022 Flash Memory Summit. All Rights Reserved.

TL 3.1 Architecture specification



Figure 1-1. OpenCAPI stack



Same for host and device

Up to 15 Templates for each direction for mixing data and control information together.

Transaction Layer (TLx) Version	Physical Form Factor	Protocols	Protocol Description
3.0	PCIe Add-In Card	C1, M0, C1, M1, C0, M1	Non-Caching DMA, Non-Caching DMA + LPC
3.1 (OMI)	DDIMM	Memory Controller	

11 | ©2022 Flash Memory Summit. All Rights Reserved.



OpenCAPI in POWER9



Flash Memory Summit

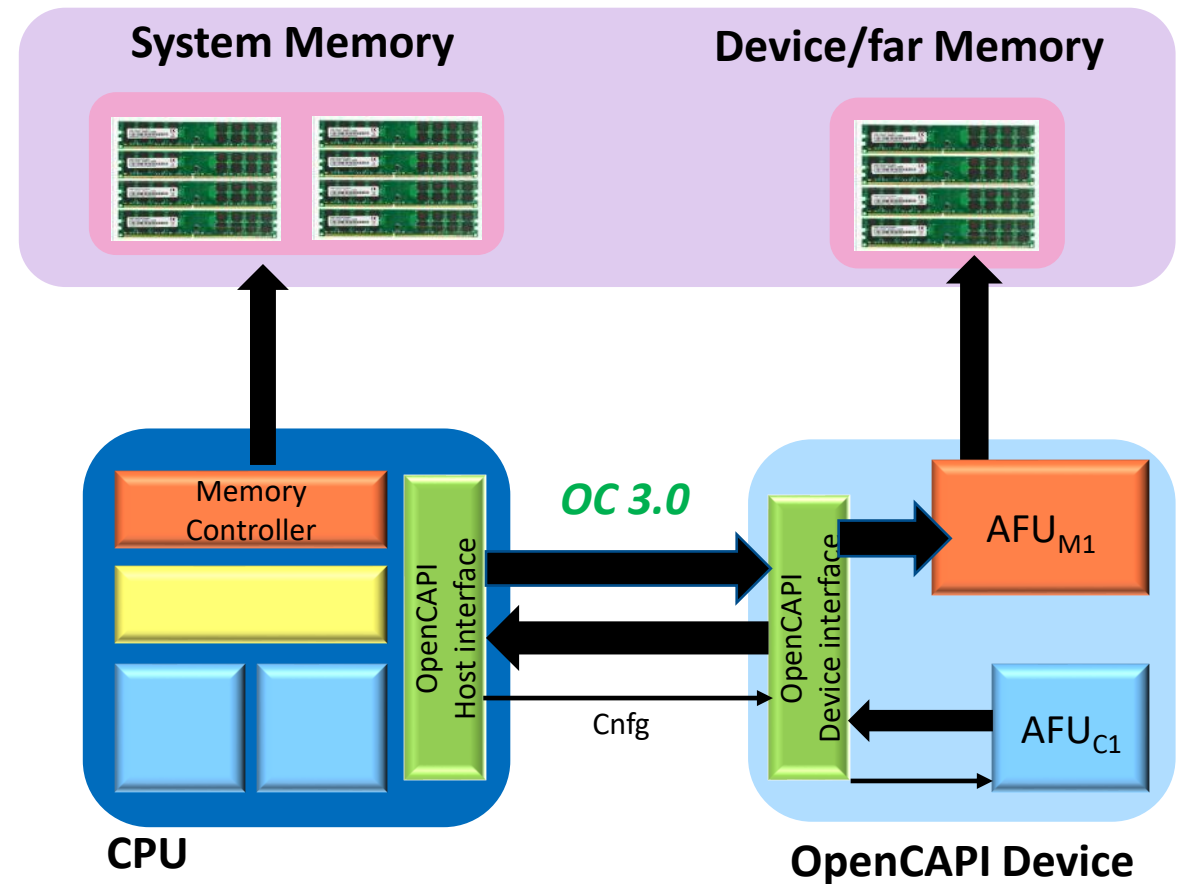
OpenCAPI attach capabilities are broken into two subclasses

- Compute (AFU_c) for function acceleration using a more traditional IO model with DMAs mastered by the device
- Memory (AFU_m) for attaching various memory technologies using Loads / Stores mastered by the host

OpenCAPI 3.0 – 25 Gbps (P9)

- AFU_{C1} , AFU_{M1}
 - AFU_{C1} can be perform non-cachable DMA Reads and Writes to system memory
 - AFU_{M1} is accelerator device memory that is part of the overall coherent system address map.

Processor Family	Transaction Layer (TL) Version	DataLink Layer (DL) Version	Physical Form Factor	Rate Gbps	Ref Clock MHz
Power9	3.0	3.0	PCIe Add-In Card	25.78125	156.25



OpenCAPI in POWER10



Flash Memory Summit

OpenCAPI attach capabilities are broken into two subclasses

- Compute (AFU_C) for function acceleration using a more traditional IO model with DMAs mastered by the device
- Memory (AFU_M) for attaching various memory technologies using Loads / Stores mastered by the host

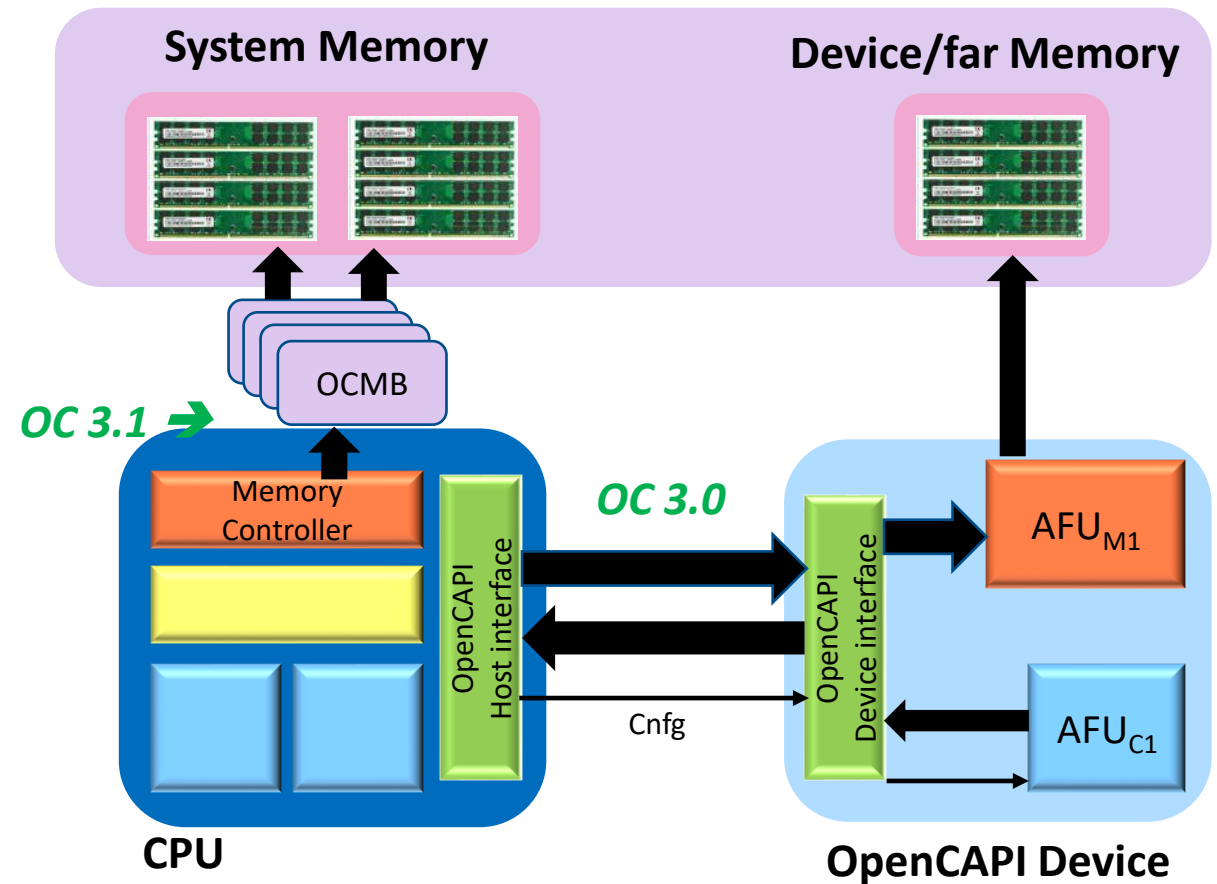
OpenCAPI 3.0 – 25 Gbps (P9+P10)

- AFU_{C1} , AFU_{M1}
 - AFU_{C1} can be perform non-cachable DMA Reads and Writes to system memory
 - AFU_{M1} is accelerator device memory that is part of the overall coherent system address map.

Power 10: OpenCAPI 3.1 @ 25.6 Gbps

- OpenCAPI Memory Interface (OMI)

Processor Family	Transaction Layer (TL) Version	DataLink Layer (DL) Version	Physical Form Factor	Rate Gbps	Ref Clock MHz
Power10	3.0 3.1 (OMI)	3.0 3.1 (OMI)	PCIe Add-In Card DDIMM	25.78125 25.6/21.33	156.25 133.33

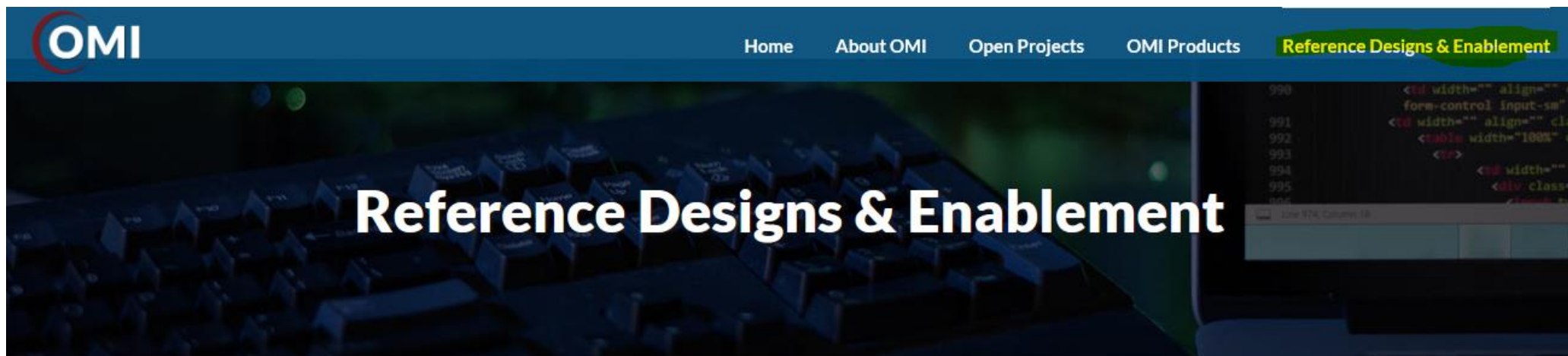


Let's start...

<https://openmemoryinterface.org/>



Flash Memory Summit



A critical component of growing any ecosystem is to have enablement available for respective developers. This includes specifications, tools, and reference designs. Below is a list of available enablement today. It is suggested to periodically visit as more enablement will be added over time.

OMI ARCHITECTURE SPECIFICATIONS

Design Overview

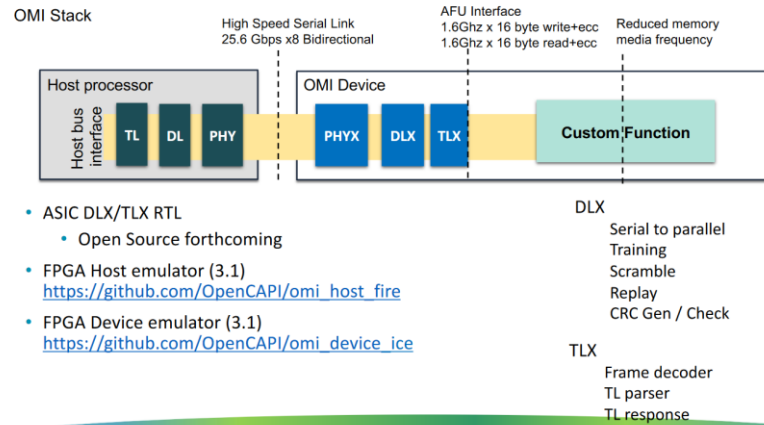
A general understanding of OMI is helpful before consuming architecture and design details. The following [OMI reference material \(PDF\)](#) shows block diagrams of the OMI stack, PCIe interoperability, and the relationship of OMI to the Data Link layer and Transaction Link layer.

OMI reference material (pdf)



Flash Memory Summit

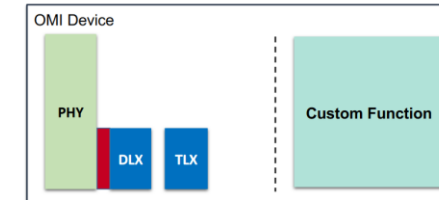
Current Open Memory Interface (OMI) Reference



- OMI and PCIe PHY are compatible/interoperable
 - OMI PHY layer is Based on the OIF CEI 28G SR specification
 - PCIe 5 SerDes PHY x16
 - OMI 32 Gbps PHY x8
 - OMI reference clock is 133.33 MHz
 - PCIe reference clock is 100 MHz

- OMI 3.1 available with P10

The DLX of OMI
Is directly connected to the serdes pins of the PHY
No connection to PCS/PIPE as in the PCI-e stack



DL <-> PHY Interface

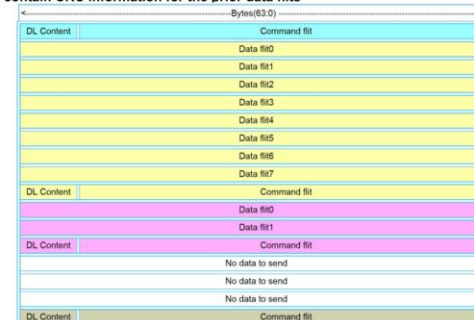


DL <-> PHY Signal	Comment
PHY_DL_CLOCK_<7:0>	Recovered captured clock for this lane
PHY_DL_LANE_<7:0>(15:0)	16 bits of Rx Data for this lane
PHY_DL_INIT_DONE_<7:0>	Indication from the PHY that it is trained and has good eyes
PHY_DL_RECAL_DONE_<7:0>	Indication from the PHY that calibration is complete
PHY_DL_IOBIST_RESET	Reset to the DL driven from the PHY to kick off IOBIST
PHY_DL_RX_PSAVE_STS_<7:0>	Indicates if the Rx Lane has responded to a Power Saving Request and is in Low Power State
PHY_DL_TX_PSAVE_STS_<7:0>	Indicates if the Tx Lane has responded to a Power Saving Request and is in Low Power State
DL_PHY_IOBIST_PRBS_ERROR(7:0)	DL to PHY to indicate a PRBS error
DL_PHY_LANE_<7:0>(15:0)	16 bits of Tx Data for this lane
DL_PHY_RUN_LANE_<7:0>	Indication to the PHY to run in high speed mode
DL_PHY_TX_PSAVE_REQ_<7:0>	Indication to the PHY to turn off the driver logic to save power
DL_PHY_RX_PSAVE_REQ_<7:0>	Indication to the PHY to turn off the receiver logic to save power
DL_PHY_RECAL_REQ_<7:0>	Indication to the PHY to run calibration on this lane

Data Link Layer Frames



- 64 bytes in size
- Command flits include 'DL Content'
- Up to 8 data flits for each command flit
- Control flits contain CRC information for the prior data flits



The OMI architecture specification is a subset of the OpenCAPI bus and includes:

- *OpenCAPI Transaction Link (TL) Architecture Specification 3.1*
- *OpenCAPI Data Link (DL) Architecture Specification.*

The TL Architecture Specification 3.1 is highly tuned based on decades of experience in making a 'clean sheet architecture' by making tradeoff decisions to focus the architecture for a direct link to memory. The specifications can be downloaded here [OpenCAPI Consortium: Official Site.](#)

PHY SIGNALING SPECIFICATIONS

The OMI bus protocol will be running over a 32Gbps PHY. Equally important to the OM

- *OpenCAPI 32Gbps PHY Signaling Specification*

This PHY Signaling Specification will provide you further guidance in how the TL will int here [OpenCAPI Consortium: Official Site.](#)

Please download the standards below:

[OpenCAPI 3.0 Transaction Layer Specification](#)

[OpenCAPI 3.1 Transaction Layer Specification](#)

[OpenCAPI 4.0 Transaction Layer Specification](#)

[OpenCAPI Data Link Layer Specification**](#)

[OpenCAPI 32Gbps PHY Signaling Specification](#)

[OpenCAPI 32Gbps PHY Mechanical Specification](#)

Available documentation

- Specifications

- <https://openmemoryinterface.org/>

- Workbooks

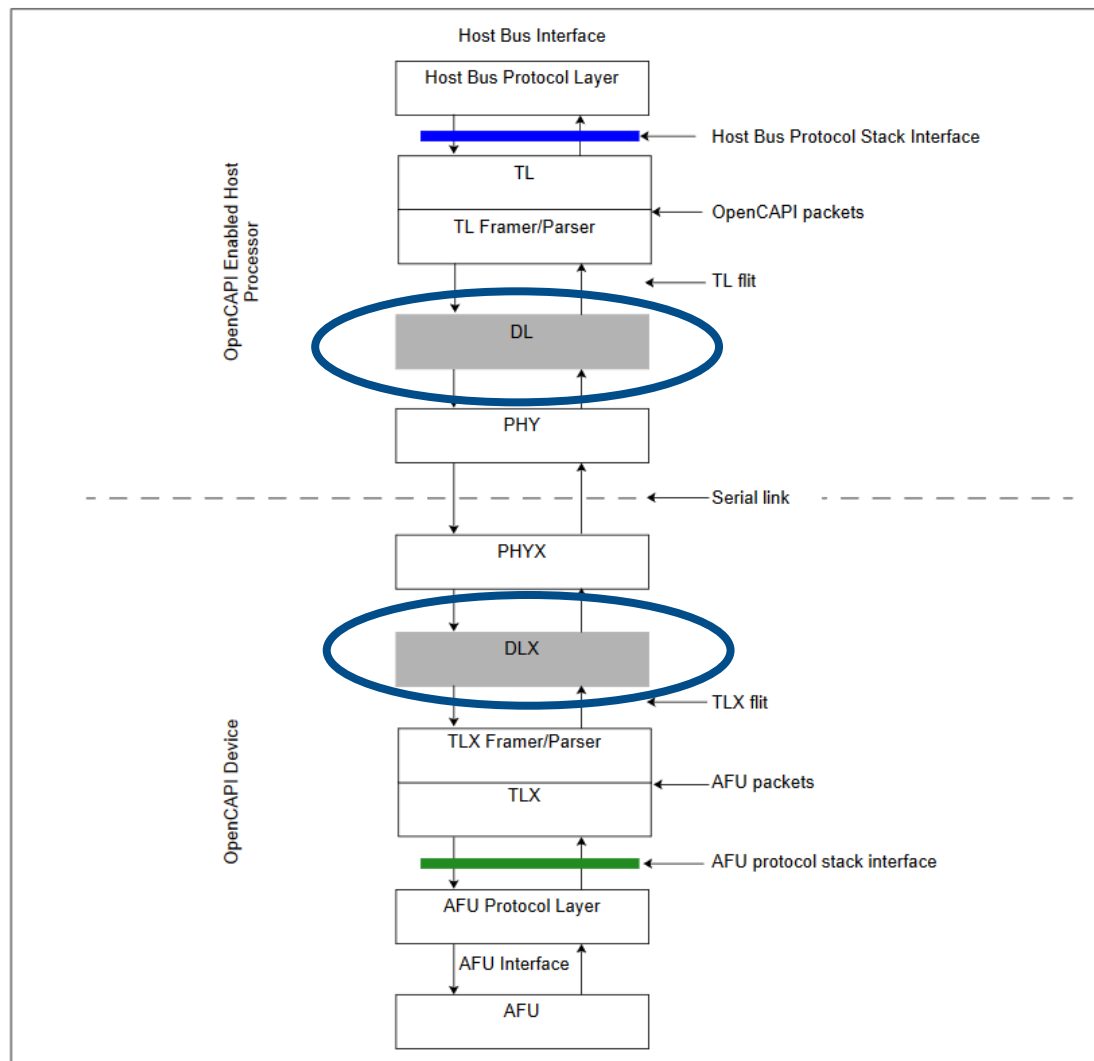
- https://github.com/OpenCAPI/omi_asic_device_reference_design/tree/master/dlx/docs
 - DL_workbook_OpenCAPI3.1_30AUG2021.pdf
- https://github.com/OpenCAPI/omi_asic_device_reference_design/tree/master/tlx/docs
 - OpenCAPI 3.1 Transaction Layer RX Workbook for OMI ASIC v 1.0.pdf
 - OpenCAPI 3.1 Transaction Layer TX Workbook for OMI ASIC v 1.0.pdf

- RTL code

- HOST
 - An example OMI Host FPGA for testing an OMI Device
 - https://github.com/OpenCAPI/omi_host_fire
- DEVICE
 - An example OMI Device FPGA with 2 DDR4 memory ports
 - https://github.com/OpenCAPI/omi_device_ice
 - OMI ASIC Device Reference Design
 - https://github.com/OpenCAPI/omi_asic_device_reference_design

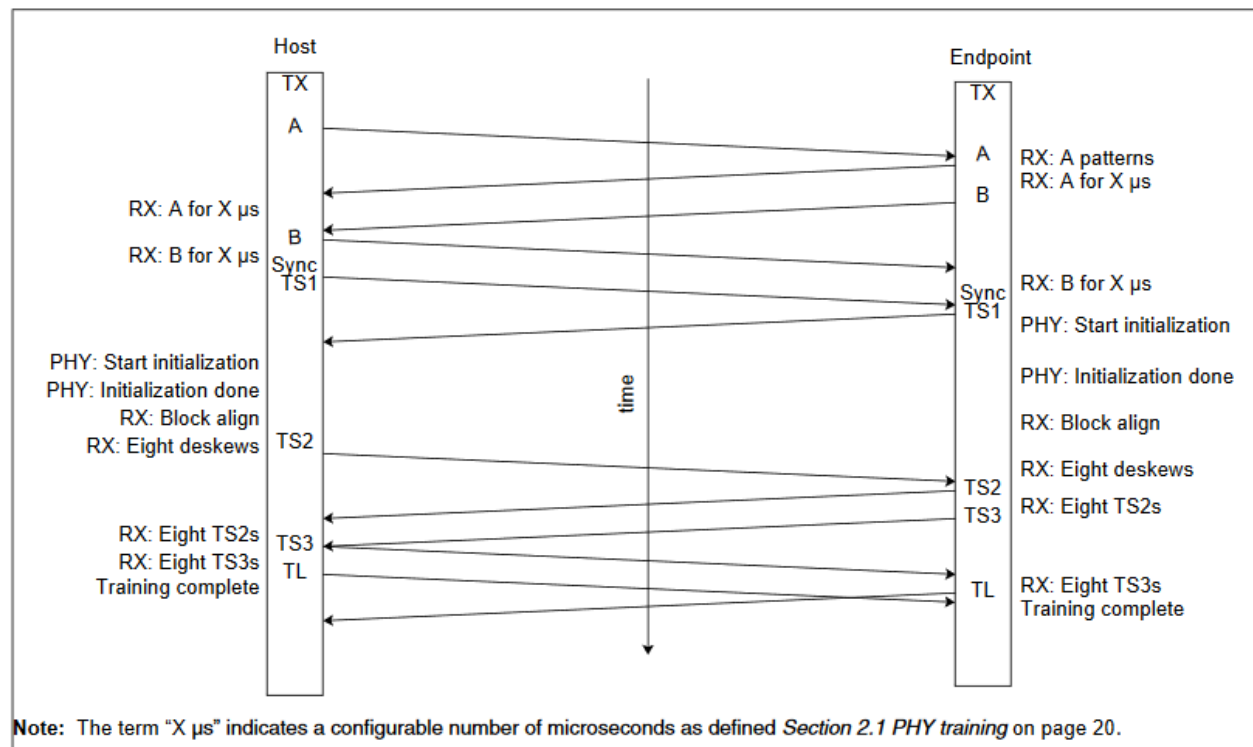
DL Architecture specification (DL3.0/3.1/4.0)

Figure 1-1. OpenCAPI stack



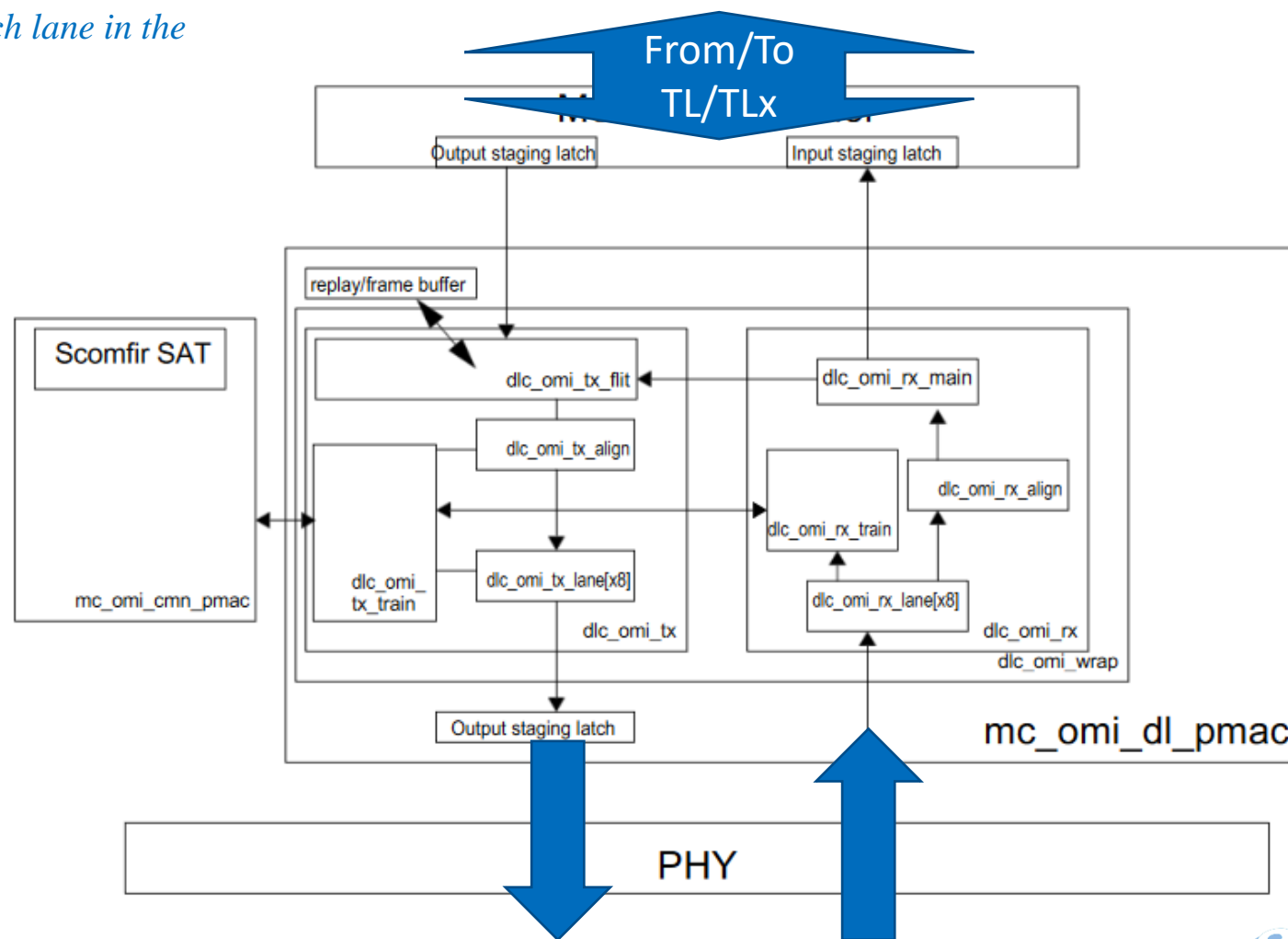
Same spec for 3.0/3.1/4.0 but with some “categories”
Same for host and device: DL=DLX

Figure 2-1. Training exchange between the host and endpoint



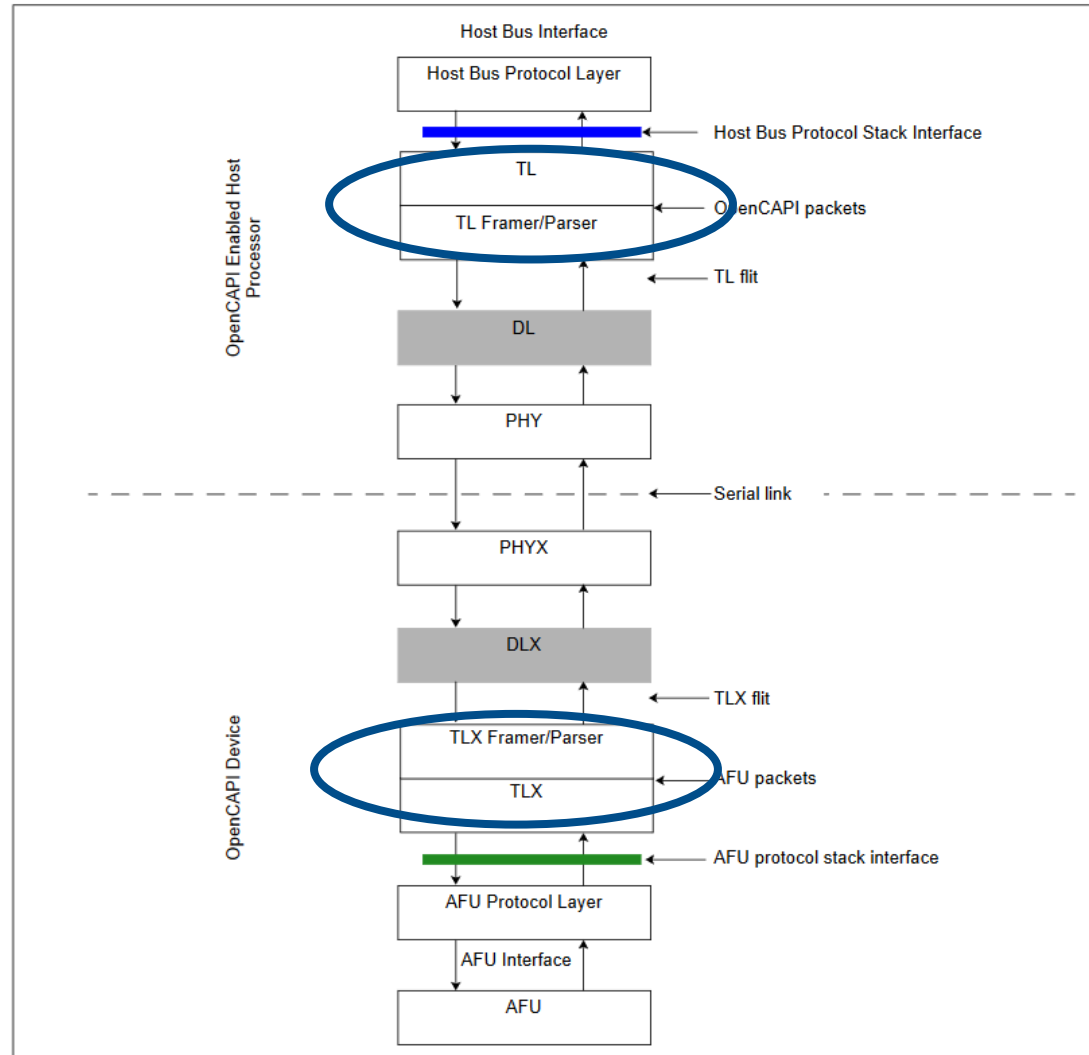
- checks the **ECC**
- calculates **CRC** for each outgoing flit.
- The partial flit (16 bytes) is **aligned** in such a way that amongst the 8 lanes.
- It also **scrambles** each 2 bytes piece,
- adds a 64/66 **sync header** and
- makes sure everything is being sent in the proper order.

- *deskews the data*
- *forms it back into a partial data flit.*
- *The **CRC** is checked and then*
- *16 bytes (parity protected) is forwarded to the TLX*



TL 3.1 Architecture specification

Figure 1-1. OpenCAPI stack



- Same specification for the asymmetric Host and Device
- Up to 15 **Templates** for each direction for mixing data and control information together.

Transaction Layer (TLx) Version	Physical Form Factor	Protocols	Protocol Description
3.0	PCIe Add-In Card	C1, M0 C1, M1 C0, M1	Non-Caching DMA Non- Caching DMA + LPC LPC
3.1 (OMI)	DDIMM	Memory Controller	

Table 5-1. DL frame format showing CRC and “bad data flit” coverage

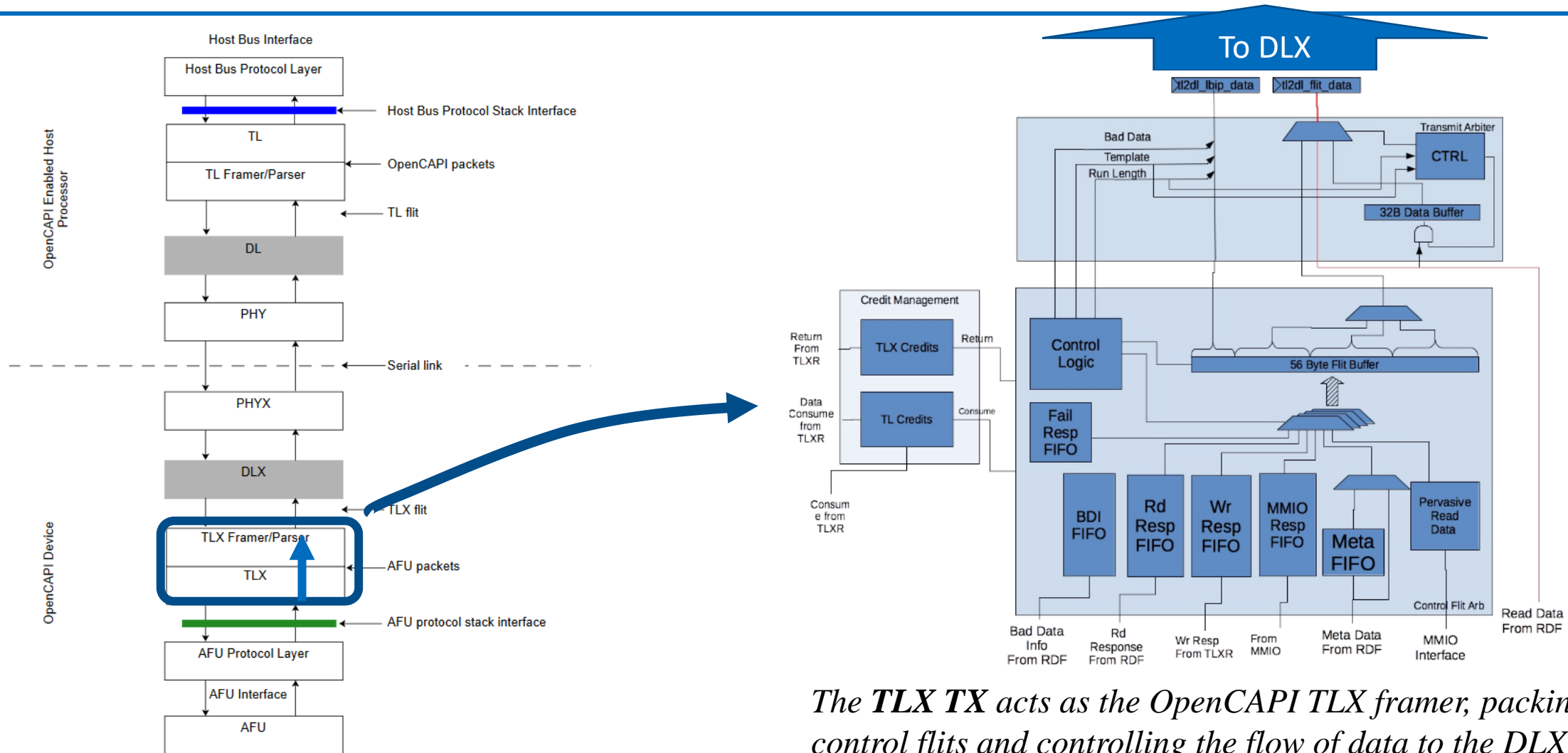
Bytes(63:0)	
DL content	TL command/response/32-, 8-byte data content
Data flit 0	
Data flit 1	
Data flit 2	
Data flit 3	
Data flit 4	
Data flit 5	
Data flit 6	
Data flit 7	
DL content	TL command/response/32-, 8-byte data content
Data flit 0	
Data flit 1	
DL content	TL command/response/32-, 8-byte data content
DL content	TL command/response/32-, 8-byte data content

- Control flits.** The control flit contains TL command/response content and DL content. The DL content contains several DL-generated subfields including the CRC that covers the control flit and any preceding data flits. There are fields in the DL content that are generated by the TL.
- Data flits.** There are 0 to 8 data flits between each control flit.

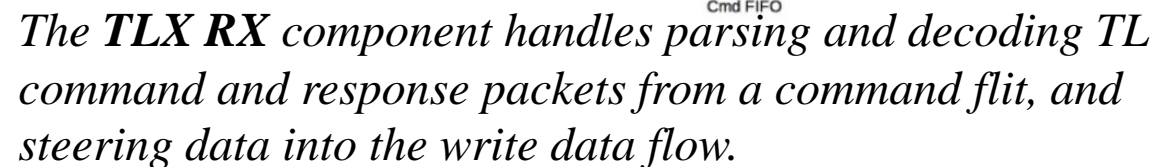
TLX Tx Workbook



Flash Memory Summit



*The **TLX TX** acts as the OpenCAPi TLX framer, packing up control flits and controlling the flow of data to the DLX from the read data flow. This component also contains the TLX credit management.*



OMI Minimum Template sets

TLx Receive (ie Memory Buffer) TL Transmit (ie Processor)	Requirement
x'00'	M
x'01	O
x'04'	O
x'07'.a	O
x'0A'.b	O

- To support metadata, an implementation must support one of the following templates: x'04', x'05' or x'06'
- x'07'.a = This template is used to support 32-byte data carriers
- x'0A'.b = This template is used to support 32-byte data carriers with extended-metadata
- See Chapters 6.1 and 6.2 of the TL 3.1 Specification for details

TL Receive (ie Processor) TLx Transmit (ie Memory Buffer)	Requirement
x'00'	M
x'01'	O
x'05'	O
x'09'.c	O
x'0B'.d	O

- To support metadata, an implementation must support one of the following templates: x'04', x'05' or x'06'
- x'09'.c = This template is used to support 32-byte data carriers
- x'0B'.d = This template is used to support 32-byte data carriers with extended-metadata

OMI Minimum Command Set

TL Command Processor Initiated	Requirement
config_read	M
config_write	M
intrap_rdy	M.ir
mem_cntl	O
nop	M
pad_mem	O
pr_rd_mem	M
pr_wr_mem	M
rd_mem	M
rd_pf	M
write_mem	M
write_mem.be	M

TLx Command Memory Buffer Initiated	Requirement
assign_actag	M
intrap_req	M
intrap_req.d	M
nop	M

M.ir = Mandatory if AFU issues any form of of intrap_req
Otherwise Unsupported

See Chapters 2.2 and 2.3 of the TL 3.1 Specification
for details

OMI Minimum Response Set

TL Response Processor Responses	Requirement
intrp_resp	M.ir
nop	M
return_tlx_credits	M

M.Ir = Mandatory if AFU issues any form of intrp_req
Otherwise Unsupported

See Chapters 2.4 and 2.5 of the TL 3.1 Specification
for details

TLx Responses Memory Buffer Responses	Requirement
mem_rd_fail	M
mem_rd_response	M
mem_rd_response.ow	M.a
mem_rd_response.xw	M.b
mem_wr_fail	M
mem_wr_response	M
nop	M
return_tl_credits	M

M.a = Mandatory if templates x'07' or x'09' are
supported

M.b = Mandatory if template x'08' is supported



Just Simple and Open

Questions



Flash Memory Summit