OpenCAPI 3.1 Data Link (DL) Workbook

Version 1.0

29 August 2021

DL Reference Design for the OpenCAPI 3.1 Data Link Specification from the OpenCAPI Consortium:

https://opencapi.org/technical/specifications

Used in Open Memory Interface (OMI) ASIC Devices

Table of Contents

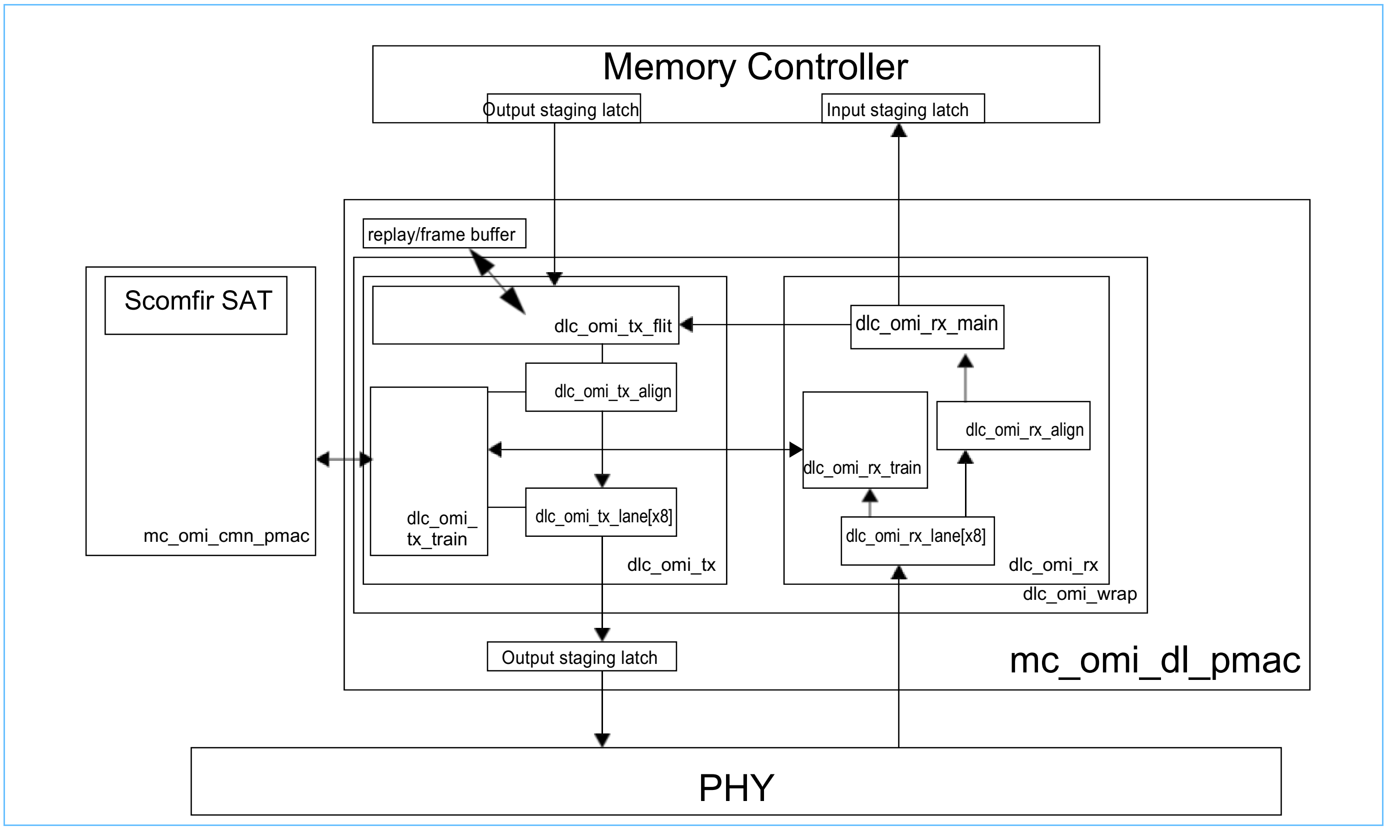
List of Tables

List of Figures

1. OpenCAPI 3.1 Data Link (DL) Workbook

1.1 Block Diagrams for OMI-DL

Figure 1-1. Block Diagram for OMI DL



1.2 RLMs

1.2.1 mc_omi_cmn_pmac

This rlm contains a scomfir satellite, a performance monitor, and registers for up to three mc_omi_dl_pmacs.

1.2.2 mc_omi_dl_pmac

The RLM pulls in the common verilog wrapper dlc_omi_wrap, plus the arrays for the frame/replay buffer.

1.2.3 dlc_omi_wrap

This wrapper supports one x8 OpenCAPI link and is divided into two sections, one for transmitting flits(dlc_omi_tx), and one for receiving flits across the link(dlc_omi_rx). All modules in the RLMs are named with a prefix of dlc_tx_ or dlc_rx_ respectively. The replay buffer is connected externally to allow the verilog code to be technology independent. An OMI Memory Buffer chip can reuse the dlc_omi_wrap and add its own frame buffer and configuration registers.

1.2.3.1 dlc_omi_tx_train (trn)

The dlc_omi_tx_train is responsible for training the individual lanes. It controls what each lane is transmitting, counts the number of patterns received, and maintains the training state machine.

1.2.3.2 dlc_omi_tx_flit (flt)

The dlc_omi_tx_flit manages the flits to be transmitted on the link. For the TL data or Replay data, it checks the ECC. For all types of flits, the dlc_omi_tx_flt appends on the link credits and calculates the CRC for the flit. It sends out 16 bytes every cycle.

1.2.3.3 dlc_omi_tx_align (agn)

The dlc_omi_tx_align pipelines the data to be transmitted on a per lane basis. Depending on the width of the link, different amounts of the 16 bytes of data from the dlc_omi_tx_flit are destined for each lane. The dlc_omi_tx_align captures the correct data for each lane and then transmits 16 bits per cycle to the dlc_omi_tx_lane. During training, the dlc_omi_tx_align sends the different DL training sets as determined by the dlc_omi_tx_train macro.

1.2.3.4 dlc_omi_tx_lane (ln[x])

The dlc_omi_tx_lane scrambles the data, inserts the 64/66 sync headers, and ensures the transmission order of the bits.

1.2.3.5 dlc_omi_rx_lane (ln[x])

This macro captures 16 bits every cycle, deskews the data and passes it to the dlc_omi_rx_main. During training, it detects the different training patterns and reports them to the dlc_omi_tx_train.

1.2.3.6 dlc_omi_rx_align (agn)

This macro aligns the data in 16Byte blocks to be passed up to dlc_omi_rx_main.

1.2.3.7 dlc_omi_rx_main (main)

The dlc_omi_rx_main receives 16 bytes of data, groups the data into flits, and checks the CRC. The data is passed to the memory buffer and the CRC indication is sent one cycle after the last beat of the control flit. Parity is calculated and passed with each beat of data for the TL to check.

When a control flit arrives, the credits are parsed and sent back to the TX data flow. The run length is also parsed to indicate when the next control flit with CRC will arrive.

Table 1-1. Macro Instances

Macro	instance name	instances
mc_omi_cmn_pmac	reg	3
mc_omi_dl_pmac	dl	8
dlc_omi_wrap	dlc_omi	1
dlc_omi_rx_pmac	dlrx	1
dlc_omi_rx_train	trn	1
dlc_omi_rx_lane	ln<i>	8
dlc_omi_rx_align	agn	1
dlc_omi_rx_main	main	1
dlc_omi_tx_pmac	dltx	1
dlc_omi_tx_train	trn	1
dlc_omi_tx_flit	flt	1
dlc_omi_tx_align	alg	1
dlc_omi_tx_lane	ln<i>	8

1.3 Interfaces

1.3.1 DL-to-TL Interface

Table 1-2. DL-to-TL Interface

DL2TL Signal Source 2 Destination	Qualifier	Comment
dl2tl_flit_vld	N/A	Indicates this cycle has valid data and parity
dl2tl_flit_error	N/A	Indicates that all previous flits since the last good control flit must be discarded. This can be asserted when flit_vld at any time.
dl2tl_flit_badcrc	N/A	Indicates that the dl2tl_flit_error was due to incorrect CRC being received.
dl2tl_flit_act	N/A	Active to be used for the clock gating the staging latches to the TL
dl2tl_flit_data(127:0)	dl2tl_flit_vld	Partial flit data. (16 bytes of the 64 bytes flit)
dl2tl_flit_pty(15:0)	dl2tl_flit_vld	Parity protecting the flit_data 1 bit per 16 bits of flit_data.
dl2tl_flit_credit	N/A	Credit returned to the TL. Each credit is for 1 cycle of flit_data. After the initial IPL ofthe link, the DL will send a programmable number of credits to the TL to indicate theframe buffer depth.
dl2tl_link_up	N/A	Indicates that the OpenCAPI link is trained and ready to accept flits.
dl2tl_dead_cycle	N/A	Indicates that the OpenCAPI link didn’t receive enough data this cycle due to 64/66 decoding overhead.
dl2tl_idle_transition	N/A	Indicates that the previous cycle was an idle flit.
dl2tl_fast_act_info(34:0)	dl2tl_idle_ transition	Fast active encoding. Same information as dl2tl_flit_data but delivered earlier in the cycle.

1.3.2 TL-to-DL Interface

Table 1-3. TL-to-DL Interface

TL2DL signal Source 2 Destination	Qualifier	Comment
tl2dl_flit_early_vld	N/A	Indicates the next cycle has valid data and ecc
tl2dl_flit_vld	N/A	Indicates this cycle has valid data and ecc
tl2dl_flit_data(127:0)	tl2dl_flit_vld	Partial flit data. (16 bytes of the 64 bytes flit)
tl2dl_flit_ecc(15:0)	tl2dl_flit_vld	ECC protecting the flit data (8 bits per 64 bits of flit data).
tl2dl_flit_lbip_vld	N/A	Indicates this cycle has valid last beat in parallel data and ecc
tl2dl_flit_lbip_data(81:0)	tl2dl_flit_vld	Last beat in parallel data. (16 bytes of the 64 bytes flit) -- last beat of a control flit Correlates to tl2dl_flit_data(81:0) – just one cycle earlier.
tl2dl_tl_error	N/A	Indicates the TL detected an error and the DL will take the configured action
tl2dl_tl_event	N/A	Indicates the TL detected an event and the DL will take the configured action (typically less severe than an error)

1.3.3 DL/PHY Interface

Table 1-4. DL/PHY Interface

DL <-> PHY Signal Source_Destination	Qualifier	Comment
phy_dl_clock_<7:0>	N/A	Recovered capture clock for this lane
phy_dl_lane_<7:0>(15:0)	N/A	16 bits of Rx data for this lane
phy_dl_init_done_<7:0>	N/A	Indication from the PHY that it is trained and has good eyes.
phy_dl_recal_done_<7:0>	N/A	Indication from the PHY that calibration is complete
phy_dl_iobist_reset	N/A	Reset to the DL driven from the PHY. Used to reset the checker in the DL before PRBS15 pattern generation/checking. This Link Layer test is typically run at MFG test (wafer and/or module) but could be used for functional lab diagnostics as well. The link layer test will use a PRBS 15 generator in the PHY that will send this pattern to a PRBS 15 checker in the DL. Suggested sequence 1.) Assert phy_dl_iobist_reset 2.) Drive PRBS 15 pattern into the DL. Could be sourced from a local PHY Rx or a remote PHY Tx 3.) Once pattern is guaranteed to be received into the DL de-assert phy_dl_iobist_reset 4.) Run for chosen time period 5.) Check dl_phy_prbs_error(7:0). If asserted at least 1 bit flip was detected on a lane.
phy_dl_rx_psave_sts_<7:0>	N/A	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>	N/A	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)	N/A	DL to PHY to indicate a PRBS15 error on corresponding lane. Checker is in the Data Link Layer logic
dl_phy_lane_<7:0>(15:0)	N/A	16 bits of Tx data for this lane
dl_phy_run_lane_<7:0>	N/A	Indication to the PHY to run in high-speed mode
dl_phy_tx_psave_req_<7:0>	N/A	Indication to the PHY to turn off the driver logic to save power
dl_phy_rx_psave_req_<7:0>	N/A	Indication to the PHY to turn off the receiver logic to save power
dl_phy_recal_req_<7:0>	N/A	Indication to the PHY to run calibration on this lane

1.3.4 DL Register Interface

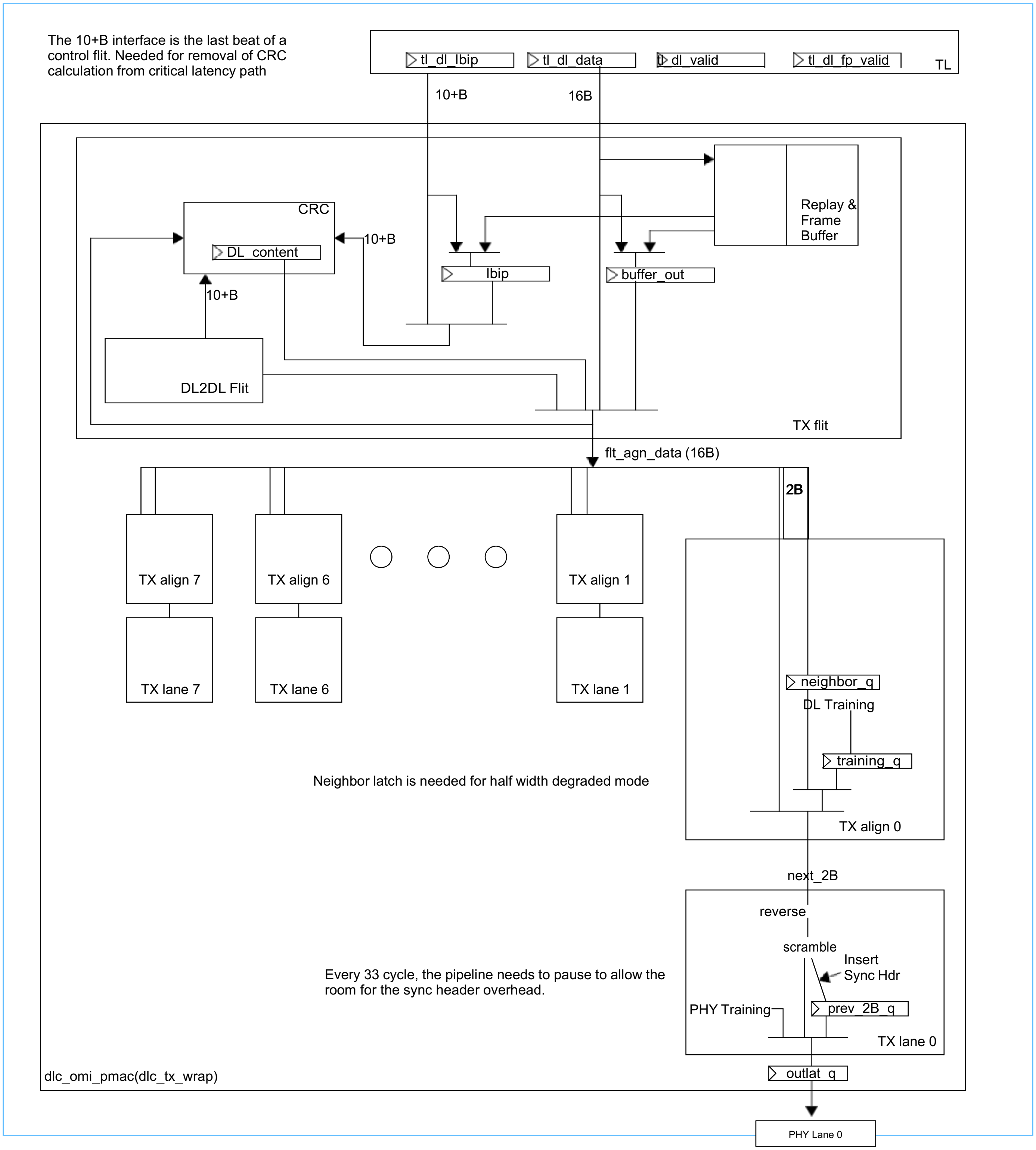
Table 1-5. DL Register Interface

DL <-> REG Signal Source_Destination	Qualifier	Comment
reg_dl_config0<63:0>	N/A	Configuration register. See Chapter 4 for bit definitions
reg_dl_config1<63:0>	N/A	Configuration register. See Chapter 4 for bit definitions
reg_rmt_config<31:0>	N/A	Configuration register. See Chapter 4 for bit definitions

reg_dl_cya_bits<31:0>	N/A	Configuration register. See Chapter 4 for bit definitions
reg_dl_1us_tick	N/A	One cycle pulse every 1 us
reg_dl_100ms_tick	N/A	One cycle pulse every 100ms
reg_dl_recal_start	N/A	One cycle pulse to indicate a lane should start calibration. Each lane needs tobe calibrated once every 100ms.
dl_reg_errors<47:0>	N/A	Status register. See Chapter 4 for bit definitions
dl_reg_rmt_message<63:0>	N/A	Status register. See Chapter 4 for bit definitions
dl_reg_status<63:0>	N/A	Status register. See Chapter 4 for bit definitions
dl_reg_training_status<63:0>	N/A	Status register. See Chapter 4 for bit definitions
dl_reg_error_capture<62:0>	N/A	Status register. See Chapter 4 for bit definitions
dl_reg_edpl_max_count<63:0>	N/A	Status register. See Chapter 4 for bit definitions
dl_reg_data<87:0>	N/A	Status register. See trace section definitions
dl_reg_trig<1:0>	N/A	Status register. See trace section definitions
dl_reg_perf_mon<11:0>	N/A	Status register. See performance monitor section definitions
reg_dl_err_cap_reset	N/A	Reset the error capture information
reg_dl_edpl_max_count_reset	N/A	Reset the EDPL max count information

1.4 TX Flow Diagram

Figure 1-2. TX Flow Diagram



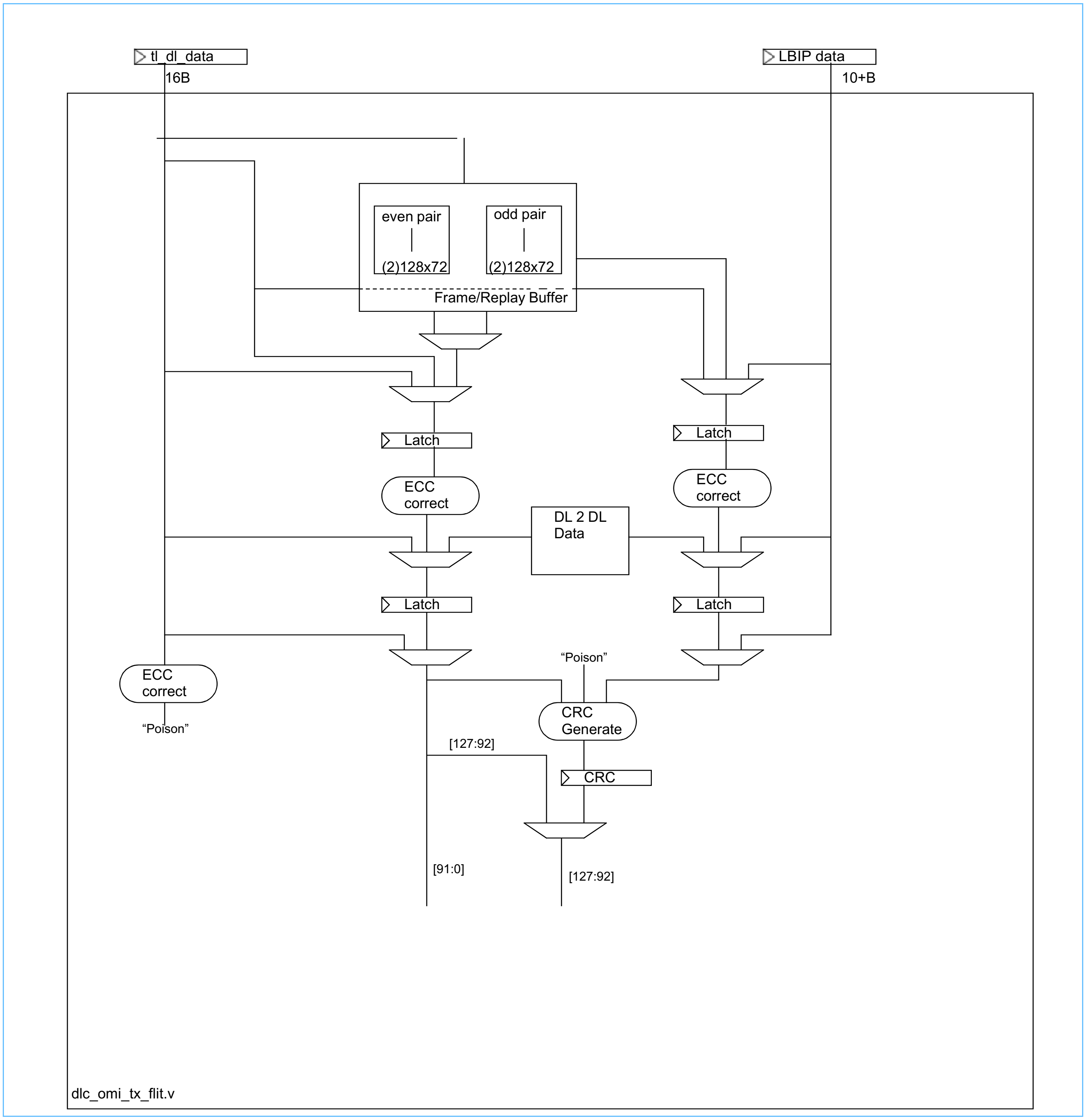
The 10+B interface is the last beat of a control flit. Needed for removal of CRC calculation from critical latency path

Neighbor latch is needed for half width degraded mode

Every 33 cycle, the pipeline needs to pause to allow the room for the sync header overhead.

1.5 TX Flit Flow Diagram

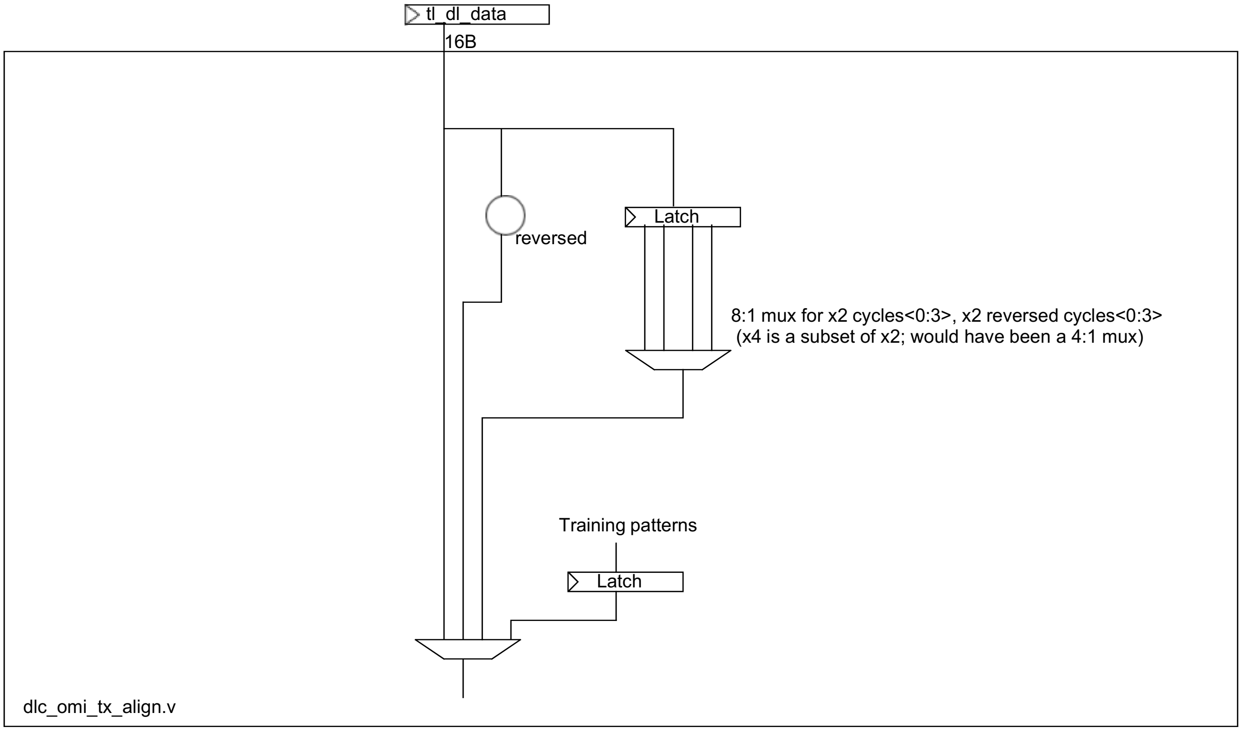
Figure 1-3. TX Flit Flow Diagram



1.6 TX Align

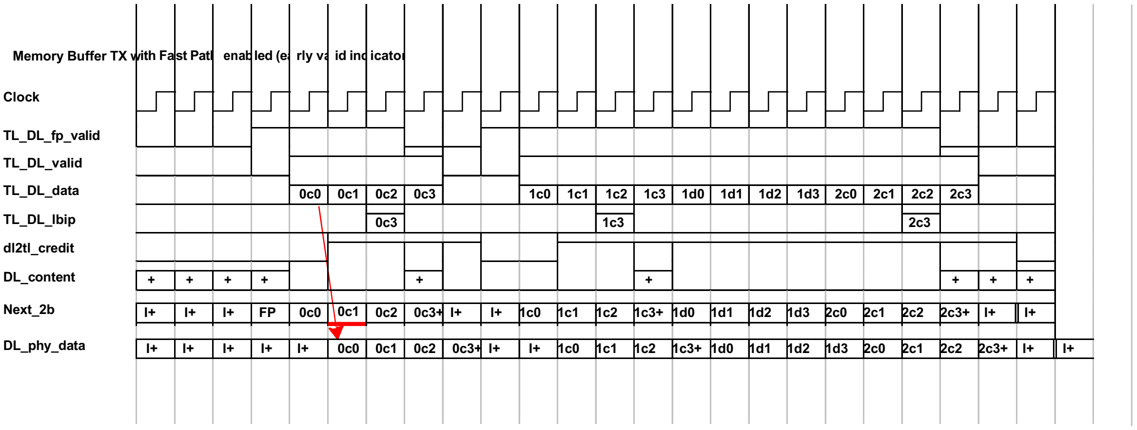
To allow the x8 data to have the minimum delay, all degraded lane widths are latched one cycle.

Figure 1-4. TX Align


tl_dl_data
16B
reversed
Latch
8:1 mux for x2 cycles<0:3>, x2 reversed cycles<0:3>
(x4 is a subset of x2; would have been a 4:1 mux)
Training patterns
Latch
dlc_omi_tx_align.v

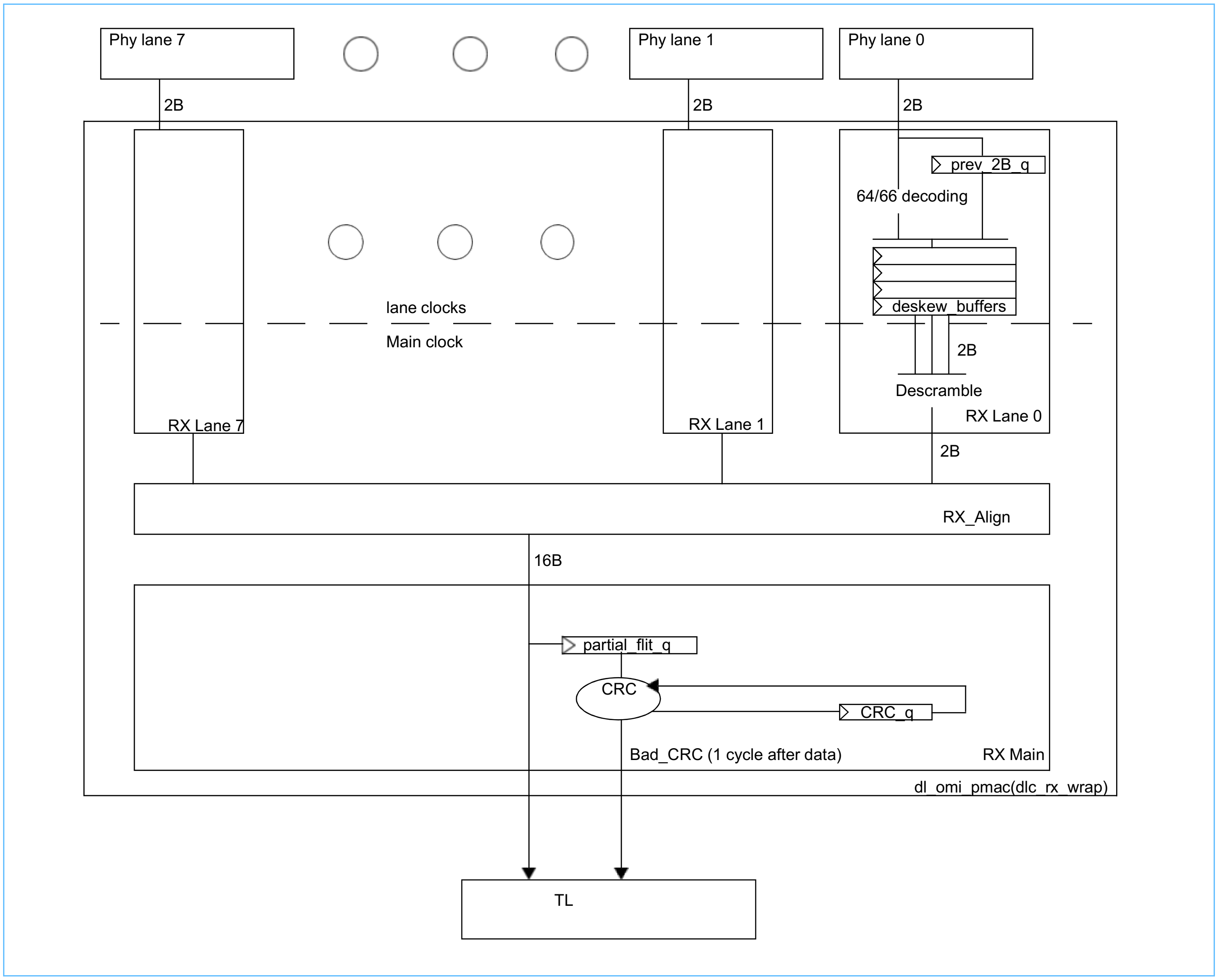
1.7 TX Timing Diagram

Figure 1-5. TX Timing Diagram


Memory Buffer TX with Fast Path enabled (early valid indicator
Clock
TL_DL_fp_valid
TL_DL_valid
TL_DL_data
0c0 0c1 0c2 0c3 1c0 1c1 1c2 1c3 1d0 1d1 1d2 1d3 2c0 2c1 2c2 2c3
TL_DL_lbip
0c3 1c3 2c3
dl2tl_credit
DL_content
+ + + + + + + + +
Next_2b
I+ I+ I+ FP 0c0 0c1 0c2 0c3+ I+ I+ 1c0 1c1 1c2 1c3+ 1d0 1d1 1d2 1d3 2c0 2c1 2c2 2c3+ I+ I+
DL_phy_data
I+ I+ I+ I+ I+ 0c0 0c1 0c2 0c3+ I+ I+ 1c0 1c1 1c2 1c3+ 1d0 1d1 1d2 1d3 2c0 2c1 2c2 2c3+ I+ I+

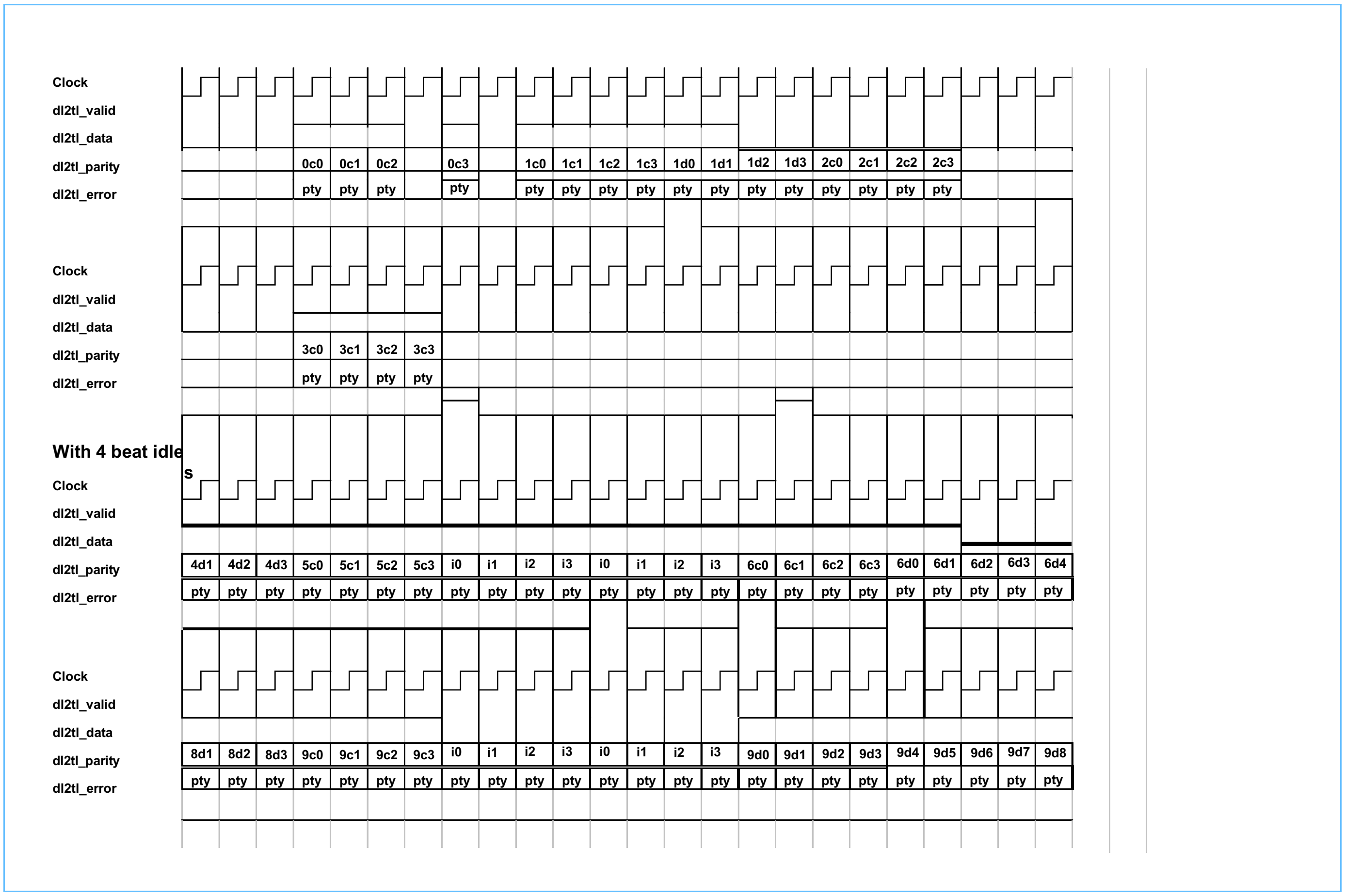
1.8 Rx Flow Diagram

Figure 1-6. RX Flow Diagram



1.9 RX Timing Diagram

Figure 1-7. RX Timing Diagram

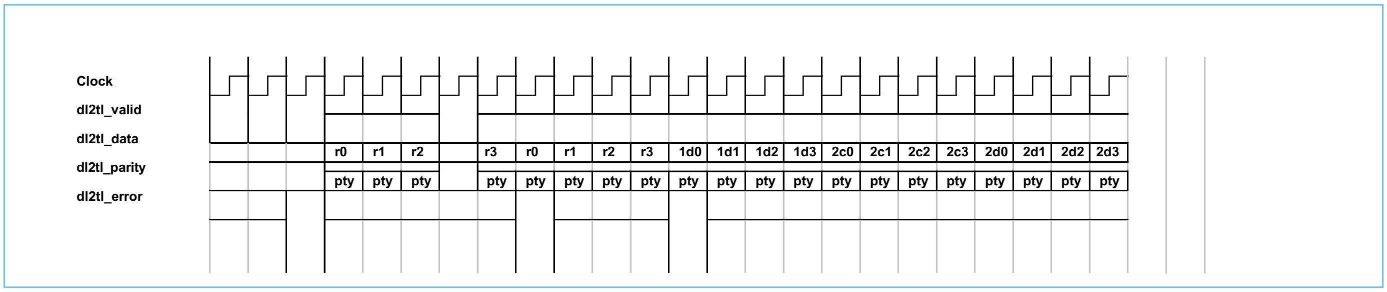


The dl2tl_error is one cycle delayed from the 4th beat of the control flit. In the diagram above, frame 0cx is received successfully, whereas frame 1c had a CRC error on it and should be ignored. The error signal can also be asserted at anytime, and the TL will need to discard all data since the last successful control flit.

When four beat idles are received, the DL doesn't decode it is an idle until after the data is passed to the TL, so the DL raise the dl2tl_error signal to notify the TL that this is not a TL flit. Note: The 6c0 control beat is valid even though the dl2tl_error signal is raised the same cycle. The bottom diagram illustrates the replay buffer being full, forcing idles to be inserted before the data flits start. In this case, the TL will have to accept 9d0 even though the error signal is asserted. When a CRC error is detected on the bus it takes a cycle or two to drop the valid after the error is detected. Before the replay flits are received, the dl2tl_error signal will be asserted to clear out any partial flits that might have been accepted. The DL guarantees that the next valid flit sent to the TL will be the replayed flit that follows the last flit that was delivered to the TL with good crc.

1.9.1 RX Replay Sequence without Duplicate Data

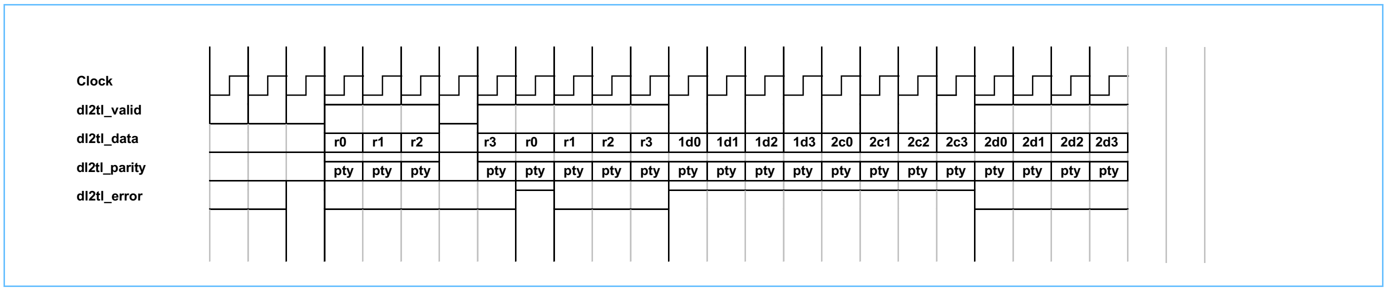
Figure 1-8. RX Replay Sequence (without duplicate data)



This timing diagram represents the replay sequence followed by new data. The error signal will be on the cycle after the replay flit to indicate to the TL that the data is to be ignored. Note that the data 1d0 was the first new flit of data and that the error signal for the replay flit is also active. Also note that the dl2tl error signal proceeds the replay flits to clean up any partial flits that might have been sent to the TL as the CRC error was being detected. This diagram only shows two replay flits in the sequence whereas the DLs always tries to transmit 9 replay flits.

1.9.2 RX Replay Sequence with Duplicate Data

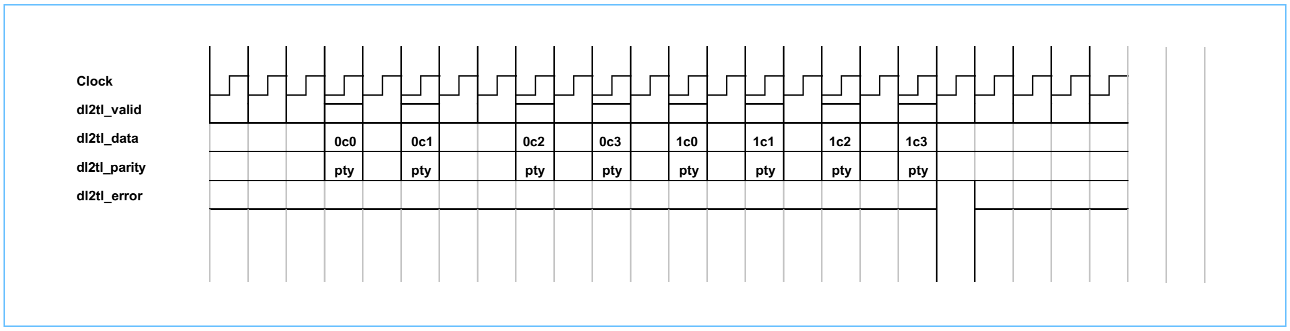
Figure 1-9. TX Replay Sequence (with duplicate data)



This timing diagram represents the replay sequence followed by duplicate data that was already trans- mitted to the TL. The error signal will be on to indicate to the TL that the data is to be ignored. Note that the data 2d0 was the first new flit of data and that the error signal dropped the cycle before this data arrived.

1.9.3 Half-Width Timing Diagram

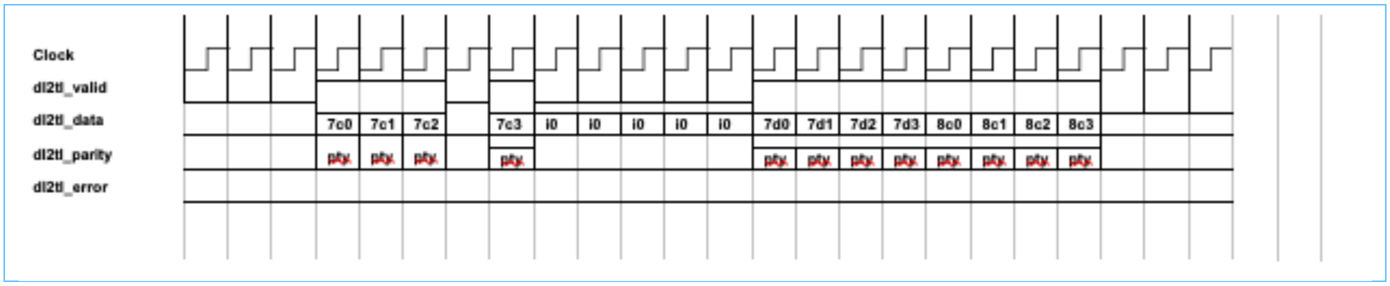
Figure 1-10. Half-Width Timing Diagram



When in half-width mode, the valid will be on every other cycle with an occasional two cycle gap where the 64/66 overhead is removed. Also note that the error signal will still be one cycle delayed.

1.9.4 Data Stalled Timing Diagram

Figure 1-11. Data Stalled Timing Diagram



When the remote replay buffer gets full, the remote DL will inject idle flits between the control flit and the data flits of a frame. The last idle flit will contain the run length field to allow the DL to keep track of how many data flits are coming.

1.10 Latches/Arrays and Floorplan

Table 1-6. Latches/Arrays and Floorplan

New Macros	Instances	Actual Latches (approximate)
mc_omi_dl_pmac	16	9415
● dlc_omi_tx		3434
− dlc_omi_tx_train		798
− dlc_omi_tx_flit		1866
− dlc_omi_tx_align		274
− dlc_omi_tx_lane		496
● spare		
● dlc_omi_rx		5981
− dlc_omi_rx_main		473
− dlc_omi_rx_align		154
− dlc_omi_rx_lane		506*8
− dlc_omi_rx_train		306
● spare		
arrays		
● 128x72		
dlc_common_pmac	6	2764
● registers		1865
● scomfir		700
● performance monitor		199
● spare		

1.11 RAS Overview

Table 1-7. RAS Overview

	Control Signals	Control Flit	Data Flit
TL 2 DL	none	ECC	ECC
DL 2 DLX	CRC	CRC	CRC
DLx 2 TLx	none	parity	parity
TLx to DLx	none	ECC	ECC
DLx to DL	CRC	CRC	CRC
DL to TL	none	parity	parity

1.11.1 ECC Stomping of Fast Path Data

When the flit macro is bypassing the frame buffer, the first two cycles of the control flit are checked for ECC and if a CE, UE or SUE are detected, the flit macro will stomp the CRC forcing the remote side to detect a CRC error and replay the flit. As the data is read out of the frame buffer, the CE ECC error will be corrected and transmitted. The last two beats of the control flit are sourced by latches internal to the TL and passed directly to the CRC algorithm without being ECC checked.

1.12 Retraining

1.12.1 First Replay FLIT after a Training or Retraining

The first replay flit after a retrain is not checked for CRC errors by the Rx side because the sync headers don’t get decoded quick enough to start the CRC logic. Therefore, only eight replay flits will be received even though nine were transmitted.

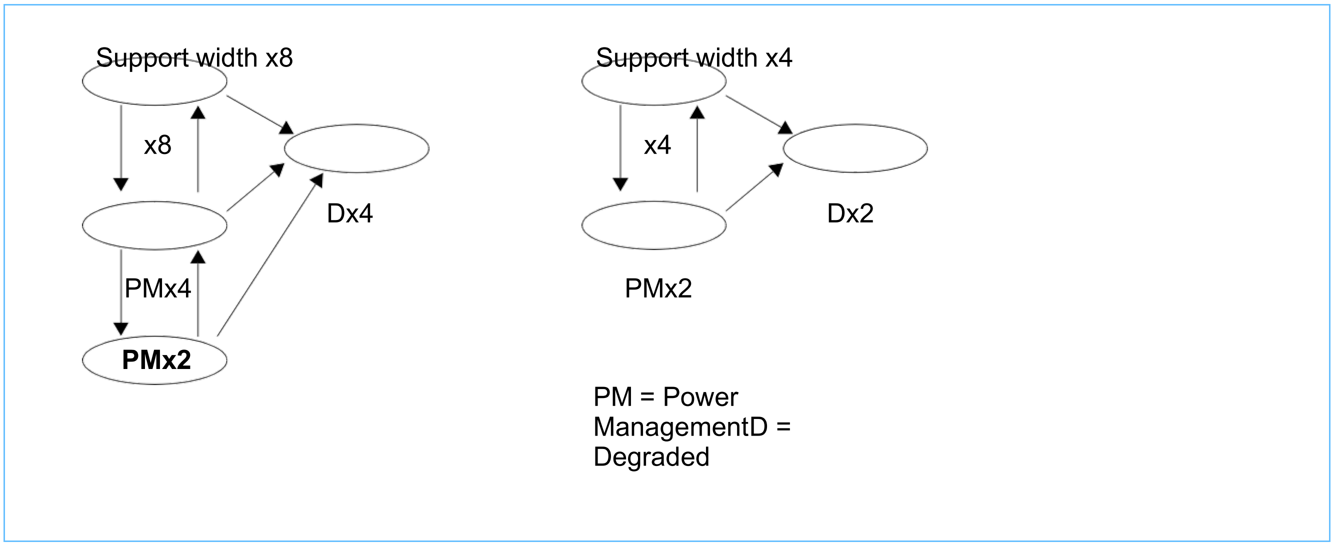
1.12.2 Fast Retraining

The information exchanged in a deskew block is not protected by parity or ECC, therefore during the initial training, eight matching deskew blocks are required to ensure the data is accurate. During a retraining, stable data has already been observed, so if the first deskew block matches the initial deskew blocks, then only one deskew block is required.

1.13 Dynamic Powering Down Lanes

The DL will implement a procedure to change the number of lanes being used to connect a device so that the extra lanes can be powered off to save power. When the device falls behind on its bandwidth, more lanes can be enabled to increase the bandwidth.

Figure 1-12. Dynamic Powering Down Lanes



1.13.1 Requirement and Restrictions

Power management will only be enabled if all lanes are functional. While running in half or quarter width mode for power management, and a lane goes bad, the link will retrain in a degraded half width mode.

A lane width request must be completed before starting a new request. Change lane options

- full width -> half width
- half width -> quarter width; half width -> full width
- quarter width -> half width

Each lane is calibrated once every 100ms, and it takes about 400 us to complete the recalibration. Please refer to the PHY specification for exact details of the recalibration.

1.13.2 Power Management Ports

1.13.2.1 TL Interface

TL2DL_lane_width_request (1:0) --> indicates the desired lane width

- “00” = power management not invoked
- “01” = quarter width
- “10” = half width
- “11” = full width

DL2TL_lane_width_status (1:0)

- “00” = not trained or retraining
- “01” = quarter width
- “10” = half width
- “11” = full width

1.13.2.2 PHY Interface per Lane

Table 1-8. PHY Interface per Lane

DL2PHY_recal_req	Indicates the PHY should recalibrate this lane.
PHY2DL_recal_done	Indicates the recalibration is complete.
DL2PHY_RX_psave_req	This lane is not being used and can be turned off to save power.
PHY2DL_RX_psave_sts	Indicates this lane is currently powered off.
DL2PHY_TX_psave_req	This lane is not being used and can be turned off to save power.
PHY2DL_TX_psave_sts	Indicates this lane is currently powered off.

1.13.2.3 Common Macro Interface

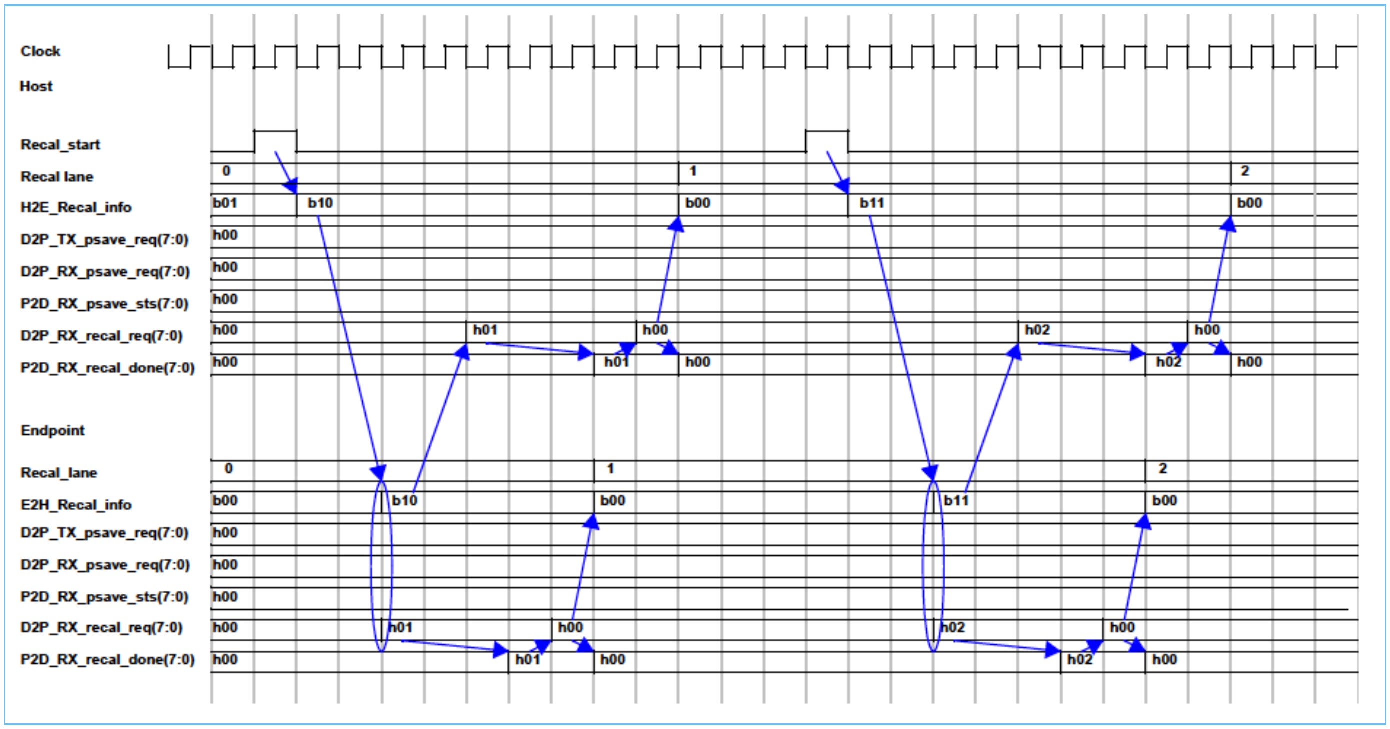
Table 1-9. Common Macro Interface

REG2DL_recal_start	Indicates this DL should recalibrate the next lane in its sequence.
DL2REG_pwrchg_req	This DL wants to change the power of one or more of its lanes. Needs to arbitrate with other DLs using this PHY to ensure only one lane is changed at a time.
REG2DL_pwrchg_gnt	Indicates this DL can change power states of its lanes. Once a DL is granted, this line will stay active as long as the DL2REEG_pwrchg_req for this DL is active.
REG2DL_pm_cdr_timer	Indicates the amount of time that is needed for the PHY to CDR lock a new lane.
REG2DL_pm_didt_timer	Indicates the amount of time required between two lanes changing their power states.

1.13.3 Handshake for Recalibration

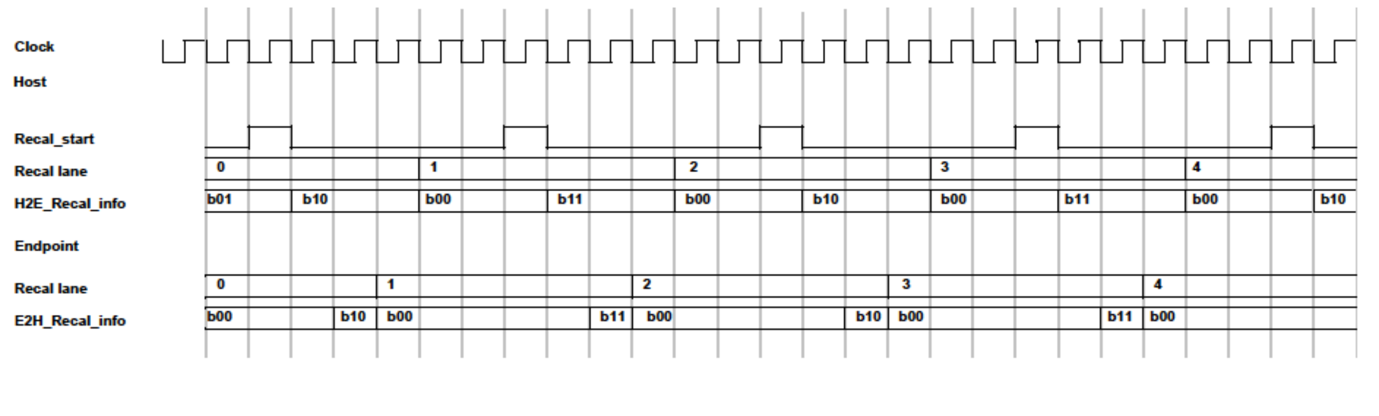
A periodic timer will initiate the recalibration sequence. Each lane needs to be powered on and when the PHY indicates that the lane is CDR locked, the recalibration can begin. After calibration is completed, the Rx can be powered down, but the TX needs to receive a message that the remote side of the link is complete before powering down.

Figure 1-13. Recalibration when all Eight Lanes are Powered On



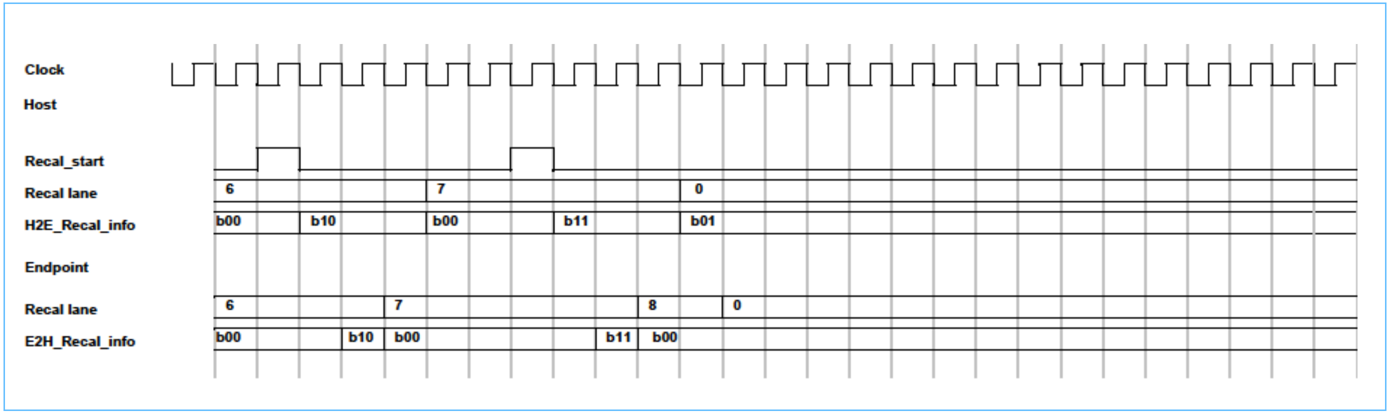
The diagram above shows the recalibration when all 8 lanes are powered on. The diagram below shows the sequencing of the 8 lanes. When the recalibration is finished, the Recal_lane advance by one.

Figure 1-14. Sequencing of the Eight Lanes



Showing the case where wrapping the lanes back to zero. Host side will send the "01" pattern

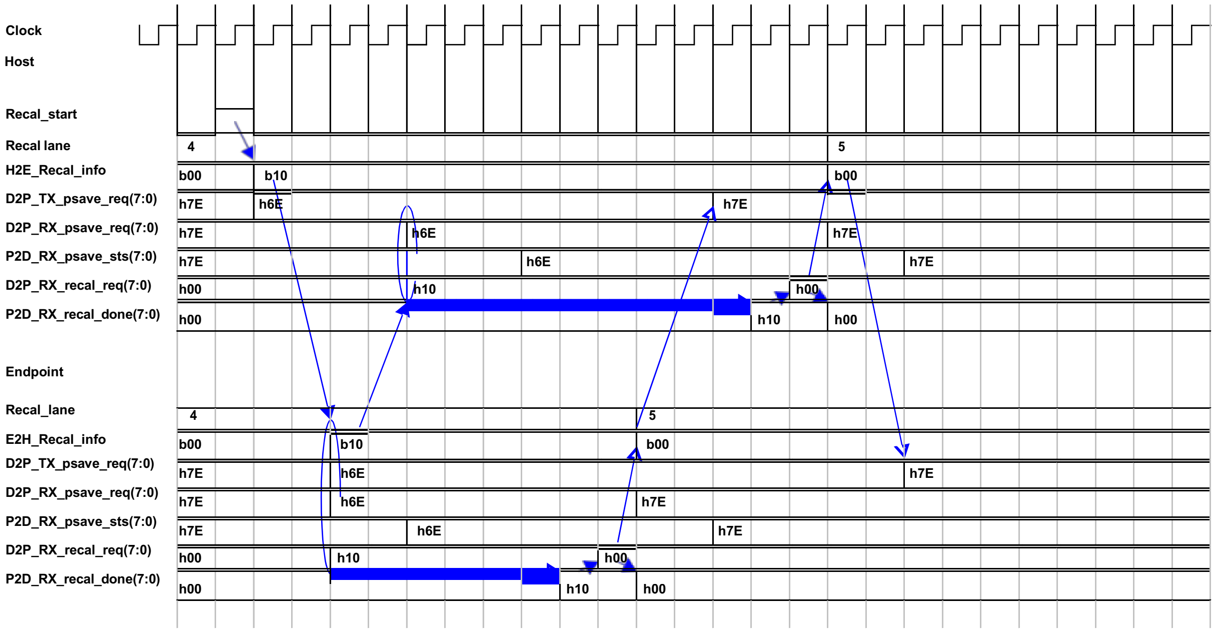
Figure 1-15. Example: Wrapping the Lanes Back to Zero



1.13.4 Handshake for Recalibration when Lanes are Powered Down

Each lane needs to be powered on. The recalibration request and Rx power save request can be changed at the same time. The PHY ensures that the lane is receiving good data before the recalibration starts. To the DL, it just looks like the recalibration done signal is delayed. After calibration is completed, the Rx can be powered down, but the TX needs to receive a message that the remote side of the link is complete before powering down. The power save status signal is not needed for recalibration, only for when switching the lane width.

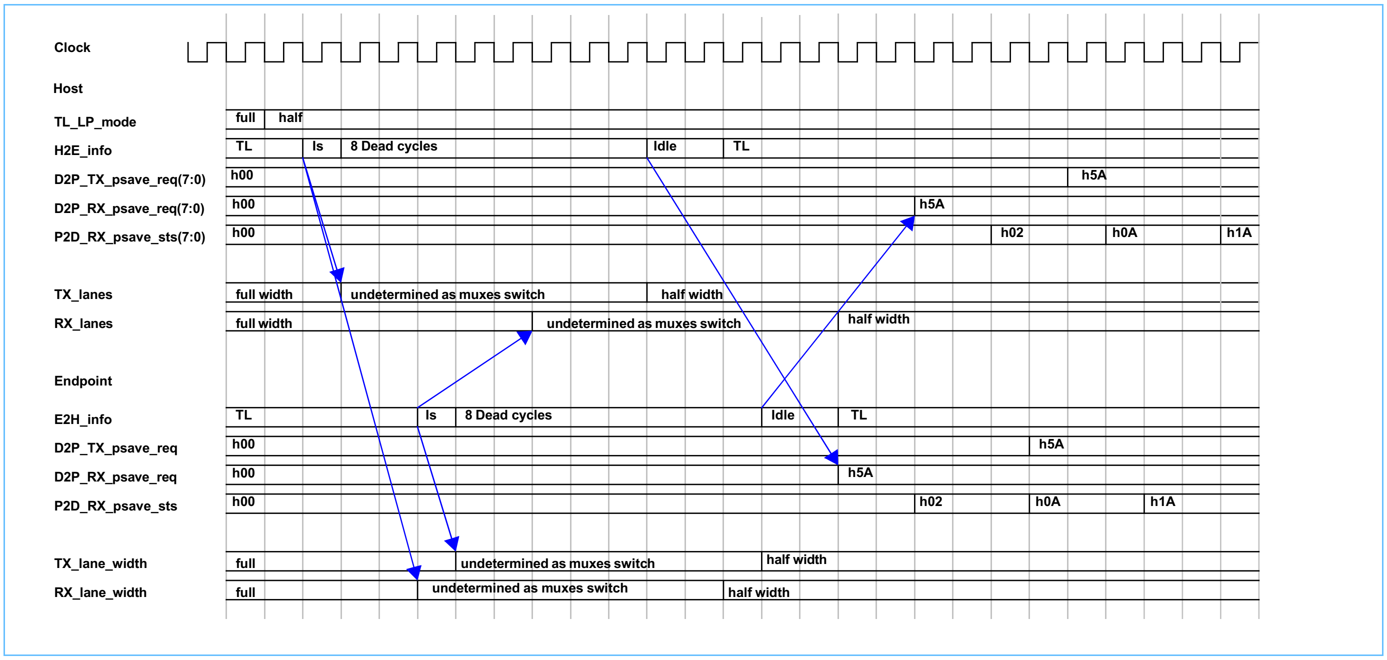
Figure 1-16. Handshake for Recalibration when Lanes are Powered Down



1.13.5 Transition into Low-Power Mode

When the TL indicates that it can drop into a low power mode, the DL will break into the data stream and send an idle flit with the switch to desired lane width to the remote side. After the idle flit, 8 cycles of idle (dead cycles) will be transmitted to allow the mux'ing to switch over. After the eight cycles, an idle flit using the new lane width will be transmitted next in which the previous run length will be valid. If a replay was in progress, the DL will wait until the replay is completed before transitioning into low power mode. Total amount of non-TL data required is 1 idle flit in full-width mode, 8 dead cycles, plus 1 idle flit in half-width mode; or 1 idle flit in half-width mode, 8 dead cycles, plus 1 idle flit in quarter-width mode;(2+8+4 = 14 cycles [8.75 ns]). After the eight dead cycles, the TX will transmit TS1 blocks for 16 cycles on the lanes that are no longer being used. After the 16 cycles of TS1 blocks, the TX lanes can be powered off. If the Rx missed the idle flit with switch, the TS1 will cause the remote side to start a retrain of the link to the new desired width. If the Rx received the idle flit with switch, it can power down the receive lanes. If a lane is being calibrated, that lane will not be powered down until after the calibration is finished. The other lanes will be powered down one lane at a time to avoid any di/dt spikes.

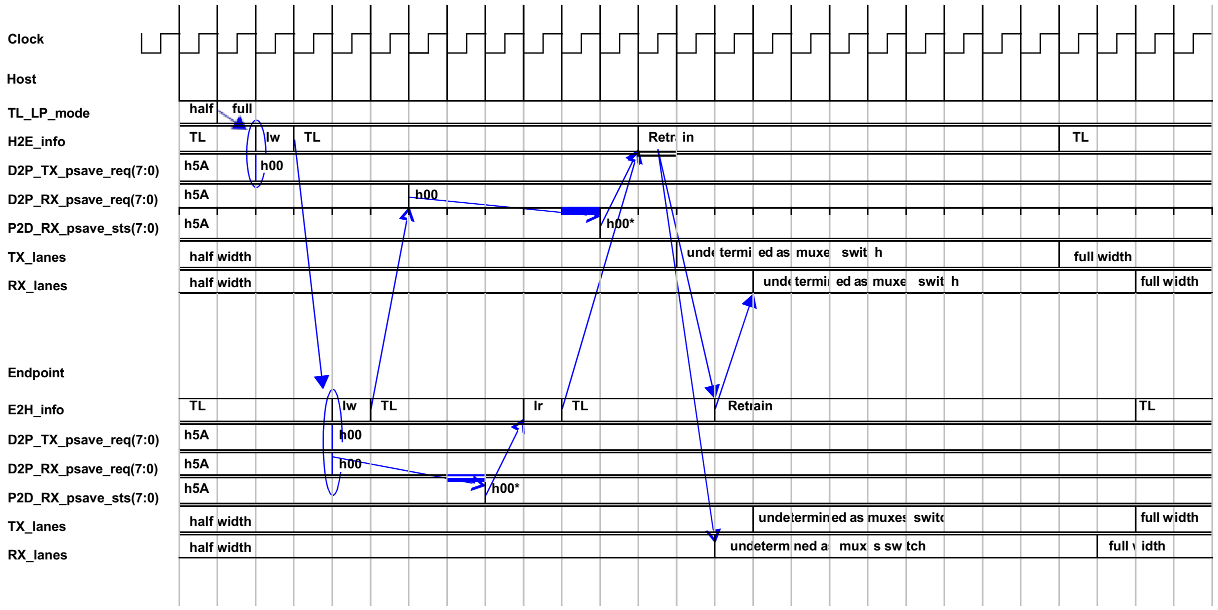
Figure 1-17. Transition into Low-Power Mode



1.13.6 Transition into Full-Width Mode

When the TL indicates that the DL should run in full width mode, the DL will turn on its TX lanes and sends an idle flit with wake-up message to indicate that the endpoint should do the same. The lanes will power on one at a time to avoid di/dt spikes. The newly turned on lanes will transmit TS1 training sets. After the lanes indi- cate that they are ready for TL traffic, the DL will launch a retrain to deskew the lanes. After the retrain, the link runs at the wider width. If a recalibration was in progress, that lane was already powered on, so only the other lanes will be powered up.

Figure 1-18. Transition into Full-Width Mode



* the psave_sts will be returned one at a time as the lanes power up.

Iw = Idle with wake up message

Ir = Idle with ready message

Is = Idle with switch lane message

1.13.6.1 Timer for di/dt

programmable values of 16ns, 32ns, 48ns, 64ns, 128ns, 256ns, 512ns, 1us (default to 32ns)

1.13.6.2 Timer for CDR Lock

programmable values of 64ns, 128ns, 256ns, 512ns, 1us, 2us, 4us, 8us, 16us, 32us, 64us, 128us (default to 512ns)

1.13.6.3 Timer for Recalibration

Waterfall timer across 24 lanes. programmable values of OFF, 24 ms, 48 ms, 100 ms, 200 ms, 500 ms, 1 sec. (default to 100 ms). If a lane is powered down when the recalibration timer fires, the lane will need to be powered up first.

1.13.7 TL Controls to Determine Bandwidth

The TL keeps track of the bandwidth of the link and sets the desired lane width based on the bandwidth. The goal is to ensure the link is at a reduce bandwidth before narrowing the width, and being able to quickly increase the width when more bandwidth is required. The TL counts the number of cycles that it sent data to the DL per each window. The number of consecutive windows are counted to determine if the lane width change is desired. Increasing and decreasing the lane width use different configuration registers to allow the system to be fine-tuned.

1.13.7.1 Configuration Registers

Table 1-10. Configuration Registers

Lane width size enabled	1	Enable the TL to change lane width
Window width	4	The number of cycles that define a window of time. 256 to 65K in powers of 2.
Low usage count	8	Number of consecutive windows that need to meet the threshold before reducing bandwidth
High usage count	8	Number of consecutive windows that need to meet the threshold before increasing bandwidth
Low threshold	5	Utilization of the link in 3.1% granularity
High threshold	5	Utilization of the link in 3.1% granularity
Low to middle threshold	5	Utilization of the link in 3.1% granularity
High to middle threshold	5	Utilization of the link in 3.1% granularity

1.13.7.2 Reducing the Lane Width

When running at full width, and the link utilization is running below the high to middle threshold for the required number of consecutive windows as defined by the low usage count, the TL indicate to the DL that half width is required, and all the counters are reset. When running at half width, and the link utilization is running below the low threshold for required number of consecutive windows as defined by the low usage count, the TL indicate to the DL that quarter width is required, and all the counters are reset.

1.13.7.3 Increasing the Lane Width

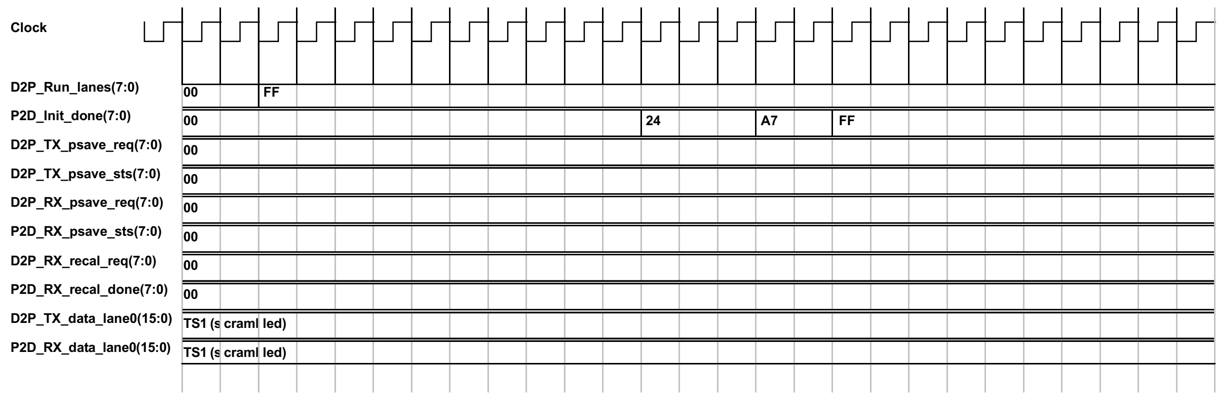
When running at quarter width, and the link utilization is running above the low to middle threshold for the required number of consecutive windows as defined by the high usage count, the TL indicate to the DL that half width is required, and all the counters are reset. When running at half width, and the link utilization is running above the high threshold for required number of consecutive windows as defined by the low usage count, the TL indicate to the DL that full width is required, and all the counters are reset.

1.13.8 PHY Interface Details

The following timing diagrams represent a link with 8 lanes. Each lane has its own control interface, and the response times for individual lanes can be varied. The signals prefixed with D2P are from the DL to the PHY; and the signals prefixed with P2D are from the PHY to the DL. All values are in hexadecimal.

1.13.8.1 Run_lanes and Init_done

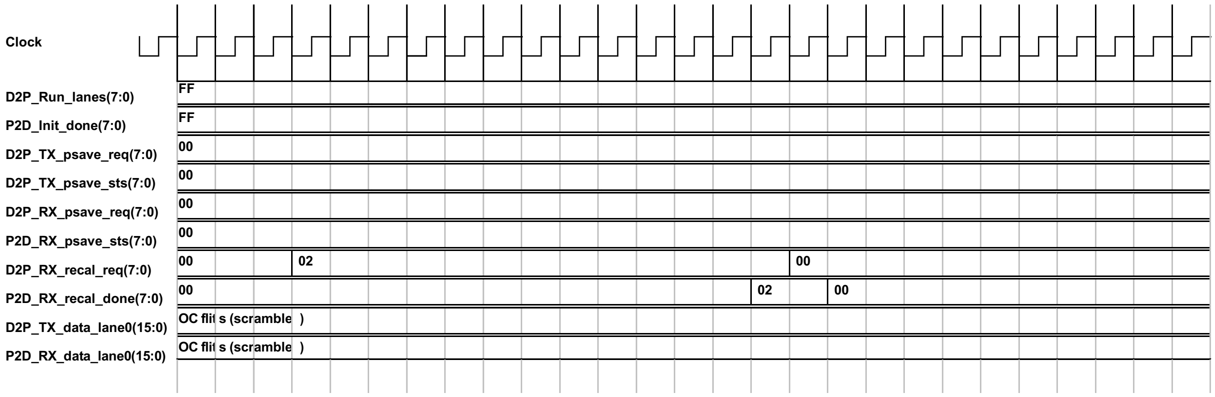
Figure 1-19. Run_lanes and Init_done



The DL will indicate that the PHY is receiving scrambled data via the Run_lanes signal for all lanes at the same time. Each lane from the PHY will respond via the Init_done signal when that lane has CDR locked.

1.13.8.2 Recalibration of an Active Lane

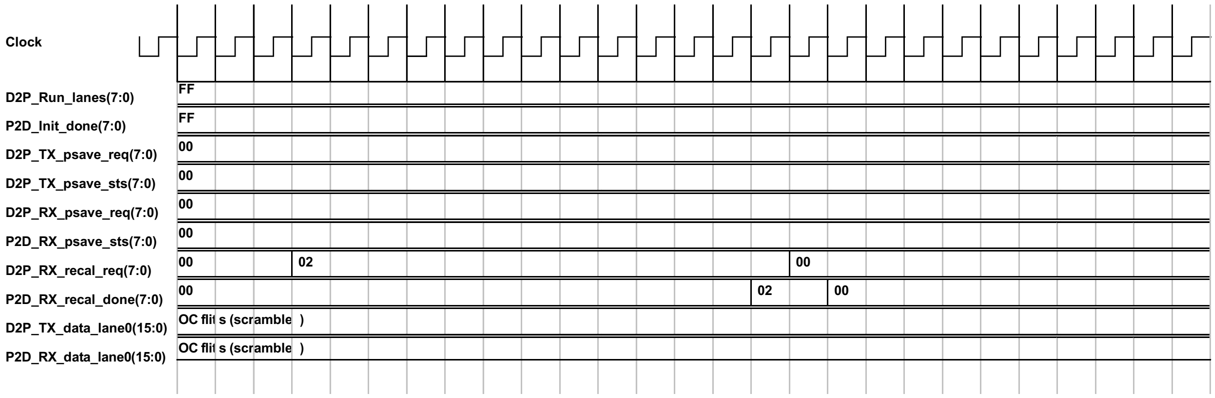
Figure 1-20. Recalibration of an Active Lane



The above diagram shows a request for lane 1 to be calibrated while OpenCAPI Flits are being transmitted. This is a four-point handshake. Only 1 lane will ever be re-calibrated at a time. The PHY is required to always complete the handshake otherwise the DL would be in a hung state.

1.13.8.3 Recalibration of an Inactive Lane

Figure 1-21. Recalibration of an Inactive Lane

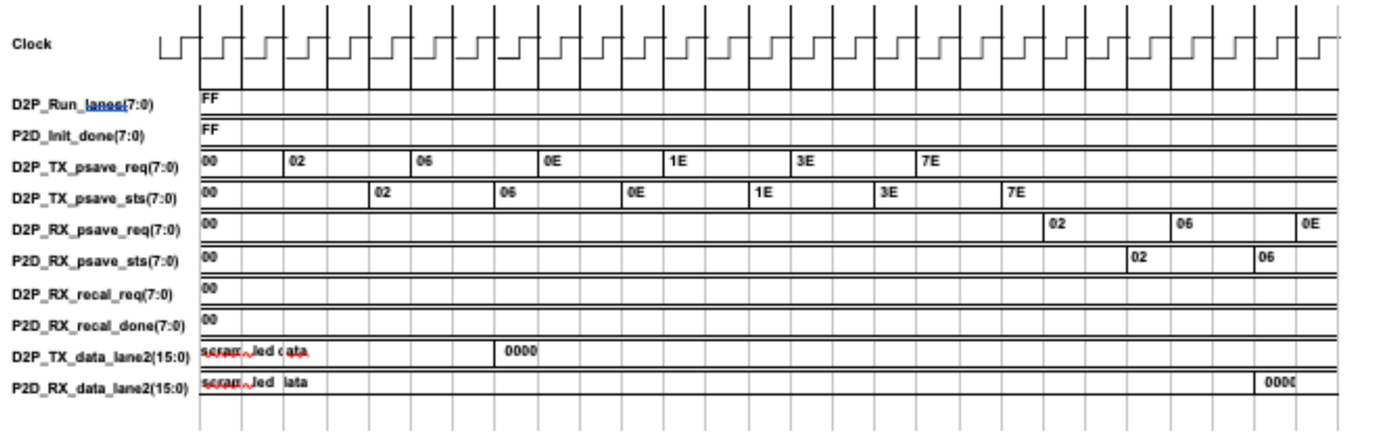


The above diagram shows a calibration request to lane 4 that is currently in a powered down state. Only the RX side is required to be powered on to calibrate, however the OpenCAPI protocol calibrates both ends of the link at the same time, so both the RX and TX are powered on. To avoid a di/dt spike, the TX is powered on before the RX. There is a programmable register (defaulted to 32ns) that delays when the RX can be powered on. The DL will send scrambled data to the PHY when the psave_req lowered. The RX psave_sts indicates that the lane is powered on and CDR locked. This delay should be around 500ns. If the PHY doesn't output a status of when the lane is ready, the DL has a programmable timer that can be used to guarantee the lane is locked. Values range between 64ns and 128 us. Since both sides are the link are independent, the time when the TX is allowed to power off the lane could be before or after the RX side. The DL guarantees that both sides don't transition within the programmable value of each other.

In degraded mode, either one direction or both directions, the calibration logic sends calibration requests to all lanes in the same sequential order. The only difference is that the lane the psave requests are not modified when the lane is degraded.

1.13.8.4 Powering Down of Six Lanes

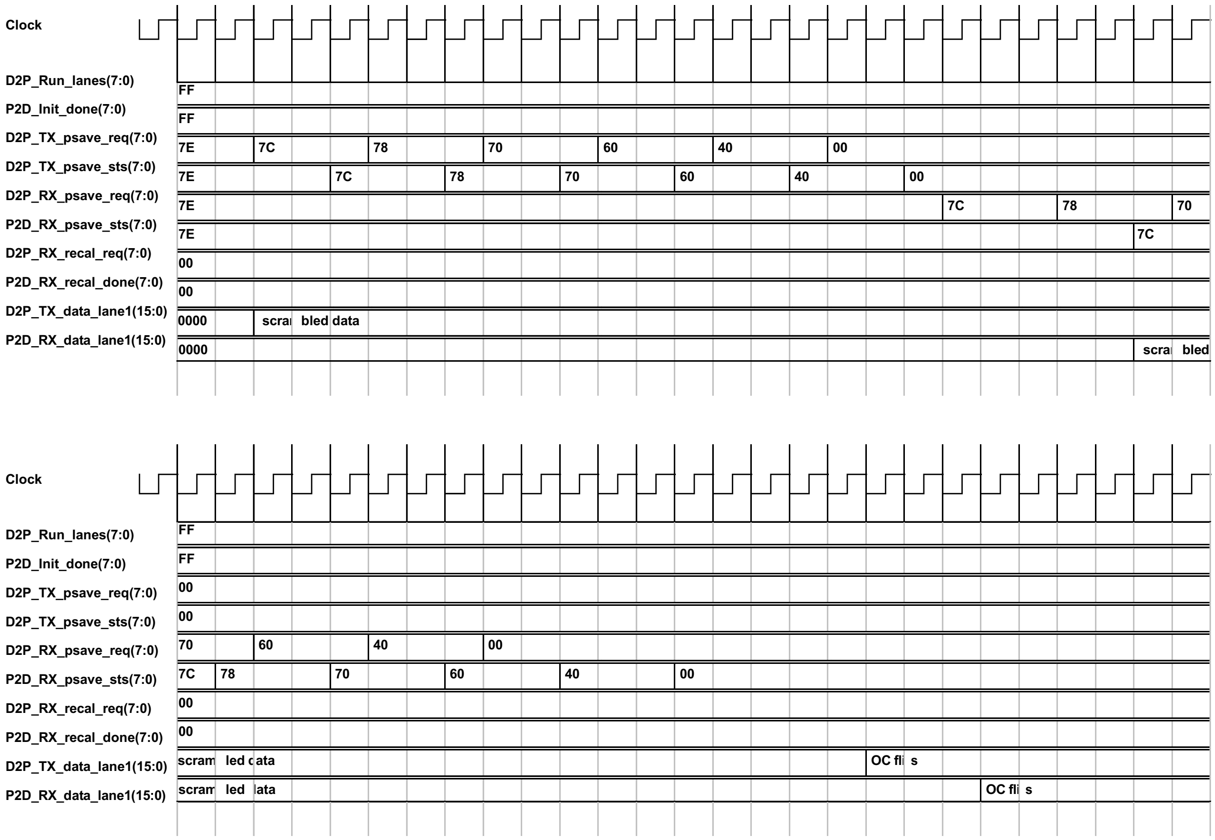
Figure 1-22. Powering Down of Six Lanes



The above diagram shows the powering down of 6 lanes. All of the TX lanes are powered off before the RX lanes are started. The DL uses a programmable register to control the delay between when two lanes can change power states. When the TX status indicates the PHY is powered down, the DL stops sending data to that lane.

1.13.8.5 Powering up of Six Lanes

Figure 1-23. Powering Up of 6 Lanes



The above diagram shows the powering up of 6 lanes. All of the TX lanes are powered on before the RX lanes are started. The DL uses a programmable register to control the delay between when two lanes can change power states, it does not wait for the psave status. The DL starts to send scrambled data when the psave_req is lowered. The RX_psave_sts indicates that the lane is powered on and receiving reliable data. The DL uses the psave_sts to indicate when it can change the type of data being transmitted on the link.

If a recalibration is in progress, the powering up or down of that lane will be skipped until the recalibration is completed.

1.13.8.6 Calibration of lanes in degraded mode.

When the link is in degraded mode, the calibration requests can skip lanes that are not being used. This may cause one side of the link to do a calibration when the other side is skipping a lane. The overall periodicity of calibration requests to an individual lane should not be extended.

1.14 DL-to-TL Flit Activate

In order to clock gate the staging latches between the DL and the TL, the DL can send out a dl2tl_flit_act signal that will clock gate the dl2tl_flit_data only. The dl2tl control signals (i.e. valids and errors) will not be clock gated in the staging latches.

1.15 Unexpected Retrains

When the DL receives an event that causes an unexpected retrain, it will power up all the lanes and then attempt to train at full width. During a power management width change from quarter width to half width, if the link does not retrain within 16 us the DL treats this as an unexpected retrain and will power up all the lanes.

Assuming all the lanes train, the DL will honor the pm width change to half width and then to quarter width or stay at half width depending on the request. If the link retrains as a degraded half width, it ignores the power management requests.

If a retrain does not complete within 24 us for the memory device or 31 us for the host, the link training will revert back to sending ts1 and attempt to train again.

During the initial training, if a lane reports as being usable but then is removed from operation, the link will be hung waiting for valid information from the flaky lane. Software will need to reset the link and try again. During retrains, the link will attempt to automatically retrain.

1.16 Performance Degradation

Memory links have a performance trigger that allows software to know if a memory link is degraded due to crc errors on the link. The percentage of degradation is programmable, 1, 2, 3, or 4 percent over a 100 ms window. Each DL counts the number of cycles that it's receive or transmit links are not being utilized due to errors on the link and sets a FIR if the threshold is breached. The common macro sends a 100 ms tick to the DLs to reset these counts.

For the MCA main running at 1.6 Ghz, (.625ns/cycle), count 159.9 million cycles to reach 99.94 ms. This is accomplished by taking bits 27, 24, 23, and 19 of a 28-bit counter. For the 1ms counter, this requires bits 20,19,14, and 13 to be 0.998 ms.

1.17 Performance Monitor Events

Each OMI-DL sends 12 performance events to the common macro to be captured. These events are:

- 11: TX Fast Path Start
- 10: TX from replay buffer
- 9: TX data flit
- 8: TX control flit
- 7: TX replay flit
- 6: TX idle flit
- 5: RX nack received
- 4: RX crc error detected
- 3: RX data flit
- 2: RX control flit

- 1: RX replay flit
- 0: RX idle flit

The PMU component contains 4 counters. All the counters have to log events from the same group of 8 input signals. The common macro uses the following grouping as inputs into the PMU component.

Table 1-11. Performance Monitor Events

Group	Events							
	0	1	2	3	4	5	6	7
0	‘1’	DL0(6)	DL0(0)	DL0(2)	DL0(3)	DL0(4)	DL0(5)	DL0(1)
1	‘1’	DL0(0)	DL0(6)	DL0(8)	DL0(9)	DL0(10)	DL0(7)	DL0(11)
2	‘1’	DL1(6)	DL1(0)	DL1(2)	DL1(3)	DL1(4)	DL1(5)	DL1(1)
3	‘1’	DL1(0)	DL1(6)	DL1(8)	DL1(9)	DL1(10)	DL1(7)	DL1(11)
4	‘1’	DL2(6)	DL2(0)	DL2(2)	DL2(3)	DL2(4)	DL2(5)	DL2(1)
5	‘1’	DL2(0)	DL2(6)	DL2(8)	DL2(9)	DL2(10)	DL2(7)	DL2(11)
6	‘1’	‘0’	DL2(6)	DL2(0)	DL1(6)	DL1(0)	DL0(6)	DL0(0)
7	‘1’	‘0’	DL2(9)	DL2(3)	DL1(9)	DL1(3)	DL0(9)	DL0(3)

1.18 Replay/Frame Buffer Sizing

The replay buffer and frame buffer share the same physical array. To ensure the replay buffer is large enough, take the worst-case scenario for each section.

1.18.1 Replay Buffer Sizing

Latency for round trip credit return = 168 cycles

- 9 flit frame out = 36 cycles
- Tx scrambling = 1 cycle
- PHY Tx = 2 cycles
- Latency of card wiring = 32 cycles
  - – (1ns / inch for card wiring) x (.5 meter) = 19 ns * 1.6 GHz = 32 cycles.
- PHY Rx = 3 cycles
- Max deskew = 3 cycles
- Descramble = 1 cycle
- Frame in = 4 cycles
- Staging from Rx to Tx = 2 cycles
- 9 flit frame out = 36 cycles
- Tx scrambling = 1 cycle
- PHY Tx = 2 cycles
- Latency of card wiring = 32 cycles
- Phy Rx = 3 cycles

- Max deskew = 3 cycles
- Descramble = 1 cycle
- Frame in = 4 cycles
- Staging from Rx to Tx = 2 cycles

Total latency

- 168 cycles (@ 1.6GHz) = 105 ns

1.18.2 Frame Buffer Sizing

The TL is within 3 cycles of the DL, so the round trip credit delay is approximately 8 cycles. The DL needs to ensure it never runs empty, so it must be sized with some cushion. If it is sized too large, the latency is increased because the TL won't be able to return TL to TL credits as efficiently. The OMI DL has 32 entries and it has a configuration register to indicate the number of credits that should be used.

1.18.3 Replay/Frame Buffer Minimum Size

Replay buffer (168 entries) plus frame buffer (16 entries) = 186 entries. Round up to the next power of 2 yields 256 entries with each entry being 18 Bytes (16B of data + 2B of ecc/parity). In order to read 2 logical array entries per cycle for early CRC computation, this design requires two logical arrays of 128 entries to read 32 bytes in one cycle. In addition, the physical array width is 9 bytes, so two physical arrays are used in parallel to get the required data width of each logical array.

End result is four 128x72 arrays are required.

1.19 Selectable Idle Flit Size

Since the first beat of flit data can arrive from the TL on any cycle, the OMI DL offers a 16 byte short idle flit size. This allows the DL to switch over to TL flits on the next cycle. This can only be used if both sides are running with a 16:1 serdes ratio. The version number of both ends of the link must be between 8 and F to support this mode.

Assuming both sides are 16:1, and the OMI DL is configured to use 16Byte idle flits, as soon as the TL indi- cates that it has a valid flit, the OMI DL will indicate that the next flit will be a 64 byte control flit and allow the TL flit to be sent without delays.

When short idle flits are enabled, before transmitting a replay sequence of flits (excluding the initial replay sequence), the TX must transmit 4 cycles of idle flit. The first 3 cycles have the short flit next asserted, and the last short idle flit has the short flit next de-asserted. This ensures that the receiving side can find the starting beat of the replay sequence.

1.20 Early CRC Generation

To remove the CRC generation from the critical latency path, the OMI DL implements a last beat in parallel (LBiP) CRC algorithm. The data flow in the OMI is 16 Bytes. When the TL is sending a control flit, it will use a parallel side channel to pass the 10.5 Bytes of the final beat of data to the DL a cycle early. By receiving the last beat early from the TL, the CRC can be generated and flopped that cycle by using the LBiP algorithm.

1.20.1 LBiP CRC

There are two CRC algorithm running in parallel. The first is a normal serial CRC algorithm that feeds back the output from one cycle to be used as the seed for the next cycle. The parallel CRC algorithm has an advance feature that will advance the CRC for 128 bits (CRC plus Advance). Each control flit takes 4 cycles to be received from the TL as indicated by C0, C1, C2, C3 with the number indicating the cycle the control flit beat arrives.

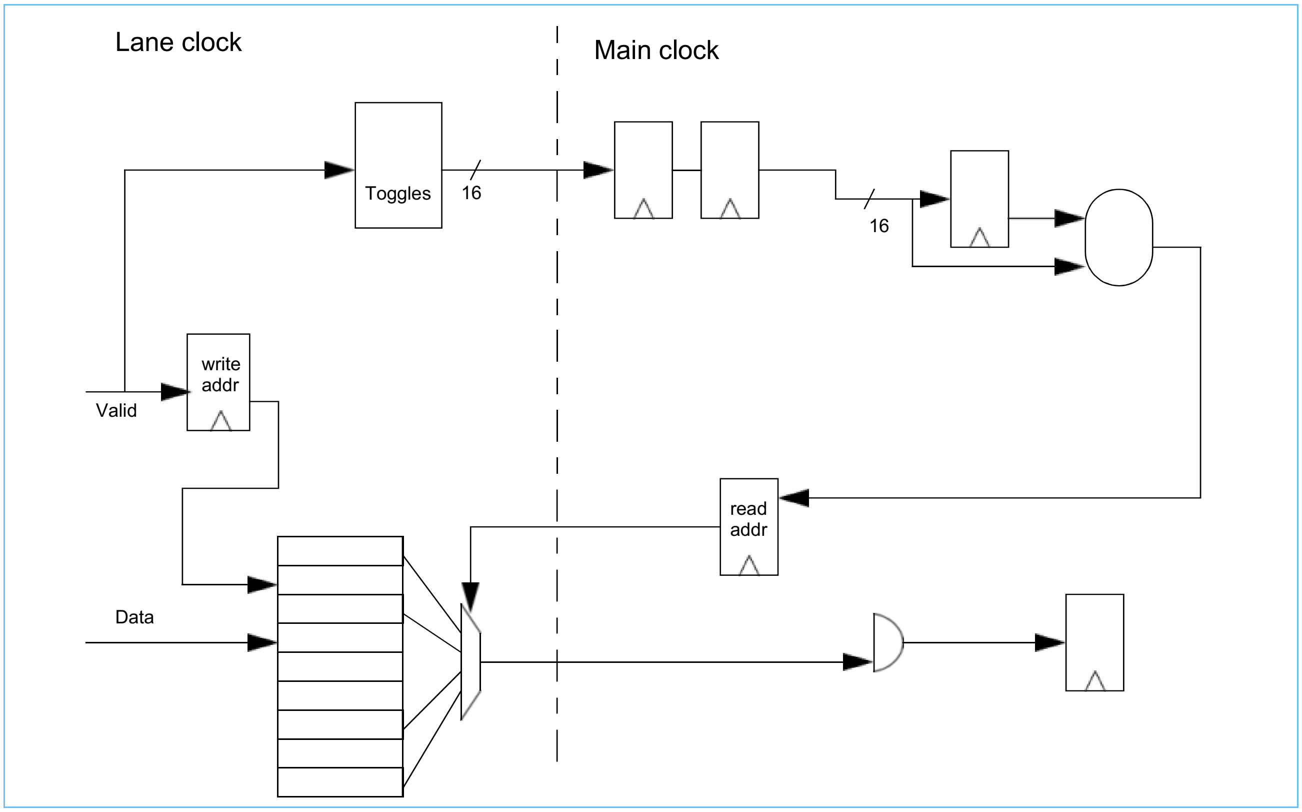
Table 1-12. LBiP for a TL Flit

	CRC			CRC plus Advance1			CRC OUT
Cycle	Data In	Feedback	Output	Data In	Feedback	Output	
0	0C0	zeros	CRC0_0	zeros	zeros	zeros	
1	0C1	CRC0_0	CRC0_1	zeros	zeros	zeros	
2	0C3	zeros	CRC0_3	0C2	CRC0_1	CRC0_2	
3	zeros	zeros	zeros	zeros	zeros	zeros	flopped version of CRC0_2 xor CRC0_3
4	0D0	zeros	CRC1_0	zeros	zeros	zeros	
5	0D1	CRC1_0	CRC1_1	zeros	zeros	zeros	
6	0D2	CRC1_1	CRC1_2	zeros	zeros	zeros	
7	0D3	CRC1_2	CRC1_3	zeros	zeros	zeros	
8	0D4	CRC1_3	CRC1_4	zeros	zeros	zeros	
9	0D5	CRC1_4	CRC1_5	zeros	zeros	zeros	
10	0D6	CRC1_5	CRC1_6	zeros	zeros	zeros	
11	0D7	CRC1_6	CRC1_7	zeros	zeros	zeros	
12	1C0	CRC1_7	CRC1_8	zeros	zeros	zeros	
13	1C1	CRC1_8	CRC1_9	zeros	zeros	zeros	
14	1C3	zeros	CRC1_11	1C2	CRC1_9	CRC1_10	
15	zeros	zeros	zeros	zeros	zeros	zeros	flopped version of CRC1_10 xor CRC1_11

1.21 RX Lane Asynchronous Crossings

Since the recovered capture clocks from the RX lanes have no phase relationship with the main clock, the OMI DL treats them as asynchronous crossing to ensure that the clock edges are not causing metastable latch glitches.

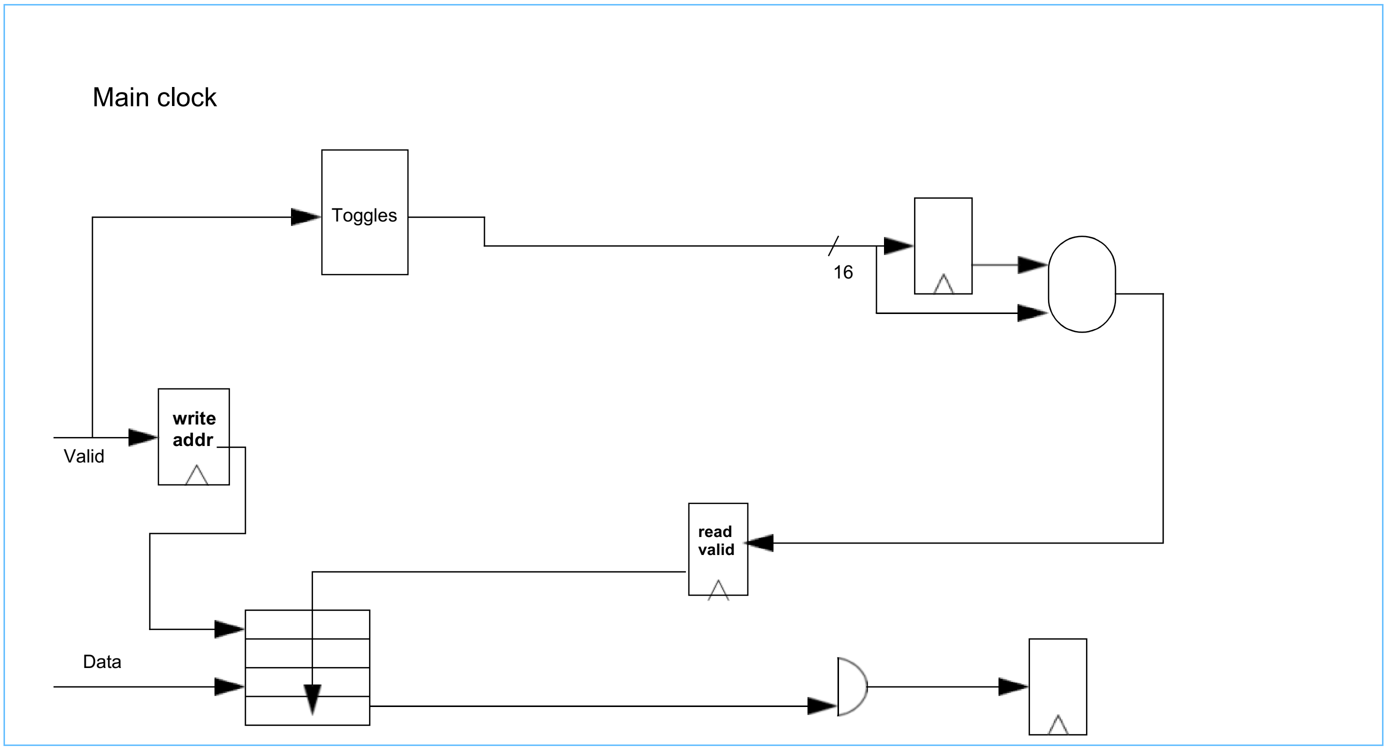
Figure 1-24. RX Lane Asynchronous Crossings



1.22 Synchronous RX Lane to Main Crossing

For an OMI ASIC, the entire DL runs on the main clock and the timing of the signals from the foxhound to the DL will be short enough to only use 12/16 of the cycle time. This guarantees that it can capture the signals using the main clock. The deskew buffer will shift the data to the first entry every valid cycle to reduce the timing from the buffer to the RX logic.

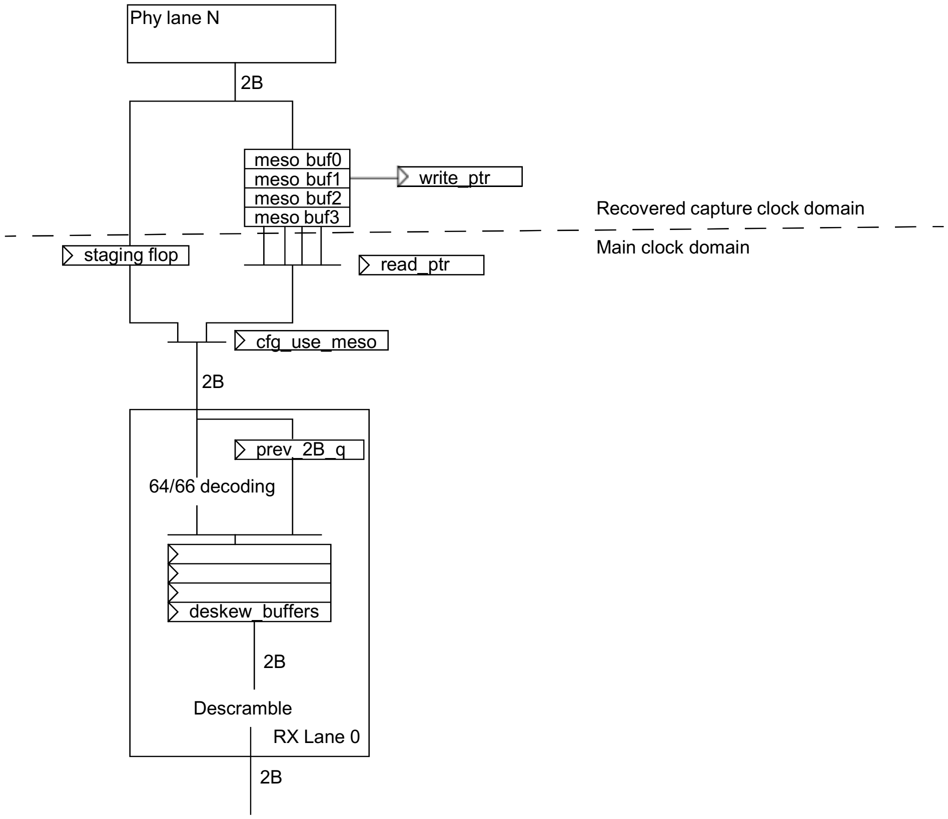
Figure 1-25. Synchronous RX Lane to Main Crossing



1.22.1 RX Meso-Synchronous Interface

As an insurance policy that for a memory buffer design can maintain a minimum of a 4UI difference between the recovered capture clock and the main clock as the temperature changes, the DL implemented a meso synchronous buffer that can be used instead of the staging flop. After the chip comes out of reset and the recovered capture clock is running, the read pointer will be updated to a configured value to allow the depth of the buffer to be programmable. Every other cycle the read pointer will increment to the next buffer entry.

Figure 1-26. RX Meso-Synchronous Interface



1.23 Error Handling

Error handling is done by FIR architecture on the host. There are no interrupts for DL reported errors. If a fatal DLx error is detected during transmission of a flit, it reports the error across the link in the replay frame.

If the TLx has a fatal error, it notifies the DLx and the DLx inserts a replay flit with an error signal asserted. The host DL will report the error in the FIR register. A register access can clear this bit on the host.

1.23.1 DL Error Detection

The DL errors can be broken into three 4 types of errors. Bits 0 - 11 indicate fatal internal errors. Bits 12 - 27 indicate notifications to software. Bits 28 – 39 indicate reasons for a retrain of the link, and bits 40 - 47 are messages or errors from the remote device.

Table 1-13. DL Errors

Bit	Name	Description	FIR Bit	Action
0	UE_FRB_CF	Uncorrectable error on a control flit from the frame buffer	0	Local check stop
1	UE_RPB_CF	Uncorrectable error on a control flit from the replay buffer	0	Local check stop
2	ACK_ptr_error	TX ack pointer passes Transmit pointer (Remote device sent too many acknowledges)	0	Local check stop
3	TL_RL_error	TL sends a reserved run length	0	Local check stop
4	TL_flit_truncated	TL truncated a frame/flit coming to the DL	0	Local check stop
5	Data_parity_error	Parity error on the data before the CRC covers it	0	Local check stop
6	CTL_parity_error	Parity error on the control state machine.	0	Local check stop
7	Rx_illegal_run_length	Received an illegal run length from the endpoint	0	Local check stop
8	Rx_slow_clock	Remote Tx is driving on a slower oscillator	0	Local check stop
9	Tx_lane_reversal	Tx lane reversal requested illegally	0	Local check stop
10	Flit_hammer	Flit detected a possible Data integrity issue and brought the link down to prevent the potential data corruption	0	Local check stop
11	Spare	Spare	0	Local check stop
12	UE_FRB_DF	Uncorrectable error on a data flit from the frame buffer	1	Recoverable
13	UE_RPB_DF	Uncorrectable error on a data flit from the replay buffer	1	Recoverable
14	CE_FRB	Correctable error on a flit from the frame buffer	2	Recoverable
15	CE_RPB	Correctable error on a flit from the replay buffer	2	Recoverable
16	Rx_CRC_Error	Detected a CRC on the data received from the link	3	Recoverable
17	Rx_NACK	Received a NACK from the remote chip	4	Recoverable
18	degraded Tx	Tx is running in degraded mode	5	Recoverable
19	degraded Rx	Rx is running in degraded mode	5	Recoverable
20	EDPL lane0	64/66 block parity error detected on lane0	6	Recoverable
21	EDPL lane1	64/66 block parity error detected on lane1	6	Recoverable
22	EDPL lane2	64/66 block parity error detected on lane2	6	Recoverable
23	EDPL lane3	64/66 block parity error detected on lane3	6	Recoverable
24	EDPL lane4	64/66 block parity error detected on lane4	6	Recoverable

Bit	Name	Description	FIR Bit	Action
25	EDPL lane5	64/66 block parity error detected on lane5	6	Recoverable
26	EDPL lane6	64/66 block parity error detected on lane6	6	Recoverable
27	EDPL lane7	64/66 block parity error detected on lane7	6	Recoverable
28	Spare	Spare	7	Recoverable
29	flit_retrain	TX_Flit macro detected a reason to retrain the link. Look at Chicken Switch register bits for details.	7	Recoverable
30	No_fwd_progress	timeout due to no forward progress	7	Recoverable
31	Remote_retrain	Remote side of the link started a retrain	8	Recoverable
32	Software_retrain	Software initiated retrain	9	Recoverable
33	Lost_Block_lock	RX lost block lock	9	Recoverable
34	Deskew_overflow	Deskew buffer overflowed	9	Recoverable
35	Lost_data_SH	Received illegal (non-data) sync headers	9	Recoverable
36	EDPL	Error detected per lane threshold reached	10	Recoverable
37	Rx degraded threshold	Receive hit a threshold of 1-4% of replays over 100ms window	10	Recoverable
38	Tx degraded threshold	Transmit hit a threshold of 1-4% of replays over 100ms window	10	Recoverable
39	train_done	Training completed successfully	11	Recoverable
40	TLx_msg_errrors	TLX initiated messages and errors (TLx to define at a later time)	12	Local check stop
41	TLx_msg_errrors	TLX initiated messages and errors (TLx to define at a later time)	13	Recoverable
42	TLx_msg_errrors	TLX initiated messages and errors (TLx to define at a later time)	14	Recoverable
43	TLx_msg_errrors	TLX initiated messages and errors (TLx to define at a later time)	15	Recoverable
44	DLx_msg_errors	DLX initiated messages and errors (DLx to define at a later time)	16	Local check stop
45	DLx_msg_errors	DLX initiated messages and errors (DLx to define at a later time)	17	Recoverable
46	DLx_msg_errors	DLX initiated messages and errors (DLx to define at a later time)	18	Recoverable
47	DLx_msg_errors	DLX initiated messages and errors (DLx to define at a later time)	19	Recoverable

Each of the 48 errors per DL can be masked prior to setting the c_err_report. The individual errors are or'd together to set 1 of the 20 FIR bits as indicated in the above table. The individual Fir bits can also be masked or configured to report as a recoverable err, local checkstop, or checkstop. Each common macro supports 3 DL macros, and Fir bits 0:19 are associated with DL0, bits 20:39 are associated with DL1, and bits 40:59 are associated with DL2.

Error bit 4, TL_flit_truncated, is a violation of the architecture specification. The TL must supply all required data thru the book-end control flit without gaps (unless the TL doesn't have credits). Due to timing constraints internal to the DL, the DL doesn't attempt to stomp the data if the TL truncated the data steam. The data sent to the remote side will be unpredictable.

During a retrain of the link, the data flow is abruptly interrupted therefore the remote side may see crc errors, EDPL errors, lost block lock errors, and/or lost data sync header errors.

1.24 Endpoint Control from the TL

The TL_error and TL_event signals from the TL that are received by the DL that control the remote endpoint. Up to four actions can be taken on the remote endpoint when either of these signals is asserted. The actions are control by the OMI DL configuration register. When the TL raises TL_error or TL_event signals, it must leave them active until a reset of the chip occurs.

- Freeze the entire endpoint
- Freeze the AFU
- Trigger the ILA
- Stop the link

1.25 OpenCAPI Error Detection Per Lane (EDPL)

To identify a lane that is receiving bit errors at rates greater than the specified rates, each lane will calculate odd parity on every block of 64 bits.

1.25.1 Transmission of Lane Parity

Each lane currently sends a 2-bit sync header at the beginning of every block and using the following encoding for these 2 bits, it can transmit parity on a block-by-block basis. The parity encoding on the sync header covers the previous block of data and is calculated before the data is scrambled. Parity is only generated when data sync headers are being used.

Note: This method maintains DC balance on each lane.

Table 1-14. 64/66 Encoding Description

64/66 Header	Encoding
“10”	Control Sync header
“01”	Data Sync header when even number of 1’s were sent in previous block of 64 bits
“00” or “11”	Data Sync header when odd number of 1’s were sent in previous block of 64 bits TX will alternate sending these two encodings to ensure DC balance across the link

1.25.2 Reception of Lane Parity

Each Rx lane checks for odd parity across the previous block (64 bits) and current sync header. When a parity error is detected, the Rx logic will increment the error bit rate count. If this count exceeds a program- mable error threshold within a programmable time window, a lane error will be reported. When the time window expires or the link is retrained, the current error count is reset to zero. This lane error will set a flag to disable this lane from being used and starts a retrain of the link. Since a lane has been removed, the training logic will attempt to train to a degraded mode when degraded modes are enabled. The EDPL max register records the maximum number of errors that each lane has seen in the given time window. If the current count is greater than the previous max, this register will be updated with the new maximum. This allows software to determine the state of the individual lanes.

1.25.3 Timing Diagram

Note: Show that the transition of control sync headers to data sync headers will have a “01” for the first data sync header.

1.26 TX Lane Ordering for Degraded Mode

While in x8 half width degraded mode, the OMI DL will always send the inside lanes (1,3,4,6) before the outside lanes (0,2,5,7) regardless of which lanes are disabled. While in x4 half width mode, the inside lanes (2,5) will be transmitted before the outside lanes (0,7).

1.27 Initial Program Load IPL Procedures

For the OMI device interface, the following procedure needs to be performed to train the link.

1.27.1 OMI Memory

1. Assert reset on OMI-DL (useful on a warm reset scenario)
2. Assert reset on adapter
3. De-assert reset on ODL
4. Start autonomous link training on ODL
5. De-assert reset on adapter
6. Memory Buffer firmware configures OMI ASIC
    - Write to configurations register for any overrides that are needed
    - Write to config0 register to start autonomous link training
7. Poll on ODL status for link training state of “111”

1.28 Spare Latches

Table 1-15. Spare Latches

Macro	Spare Latches per instance	Total
rx_align	8	8
rx_lane	12	96
rx_main	20	20
rx_train	16	16
tx_align	8	8
tx_flit	64	64
tx_lane	20	160
tx_train	32	32
Total		404

1.29 Debug Bus

There is one 88-bit debug bus coming out of the omi-dl. The default bits will be route out these signals:

- Tx side (17)
  - Frame buffer credit depth (5:0)
  - Flit type (1:0)
  - Run_length field (3:0)
  - Short flit next
  - stall_d3_q
  - Fast_path_state (2:0)
- Rx side (10)
  - Run_length field (3:0)
  - short flit next
  - Is_control_flit
  - Is_idle_flit
  - Is_replay_flit
  - Is_data_flit
  - Is_duplicate_dat

1.30 Endianness

Table 1-16. Endianness

Interface	Cycle	Data
TL to DL Flit (127:0)	0	C0B7_C0B6_C0B5_C0B4_C0B3_C0B2_C0B1_C0B0
	1	C1B7_C1B6_C1B5_C1B4_C1B3_C1B2_C1B1_C1B0
	2	C2B7_C2B6_C2B5_C2B4_C2B3_C2B2_C2B1_C2B0
	3	C3B7_C3B6_C3B5_C3B4_C3B3_C3B2_C3B1_C3B0
AGN_LANE 0 (15:0)	0	C0B0
	1	C1B0
	2	C2B0
	3	C3B0
AGN_LANE 7(15:0)	0	C0B7
	1	C1B7
	2	C2B7
	3	C3B7
DL 2 PHY lane 0	0	0D03
-- w/o scrambling and encoding	1	0D83
-- bits reversed	2	0D43
	3	0DC3
Lane 0	15:0	(transmitted last) 1100_0000_1011_0000 (transmitted first)
-- 16:1 serdes	31:16	(transmitted last) 1100_0001_1011_0000 (transmitted first)
	47:32	(transmitted last) 1100_0010_1011_0000 (transmitted first)
	63:48	(transmitted last) 1100_0011_1011_0000 (transmitted first)
PHY 2 DL lane0	0	0D03
	1	0D83
	2	0D43
	3	0DC3
LANE_AGN 0(15:0)	0	C0B0
-- bits reversed	1	C1B0
	2	C2B0
	3	C3B0
DL to TL Flit (127:0)	0	C0B7_C0B6_C0B5_C0B4_C0B3_C0B2_C0B1_C0B0
	1	C1B7_C1B6_C1B5_C1B4_C1B3_C1B2_C1B1_C1B0
	2	C2B7_C2B6_C2B5_C2B4_C2B3_C2B2_C2B1_C2B0
	3	C3B7_C3B6_C3B5_C3B4_C3B3_C3B2_C3B1_C3B0

2. Version Features

Any version and feature information is detailed in the OpenCAPI 3.0/3.1/4.0 DL Architecture Specification that can be obtain at www.opencapi.org.

3. Software Procedures

3.1 Lab Debug

3.1.1 OMI-DL CRC Errors and NACKs Received

To enable the performance monitor to count the number of CRC errors received on a link, common configuration register bits 31:0 at address offset x0E must be configured. See section 1.17 for the detailed PMU information. The same PMU counts up to three links, but only one link at a time.

A value of x"F8001403" will count DL0, crc errors in PMU counter 0 and nacks received in PMU counter 1.
A value of x"FA001403" will count DL1, and a value of x"FC001403" will count DL2.

Reading of the PMU counters at address offset x0F will give you crc and nacks counts since the last scom read.

Table 3-1. OMI-DL CRC Errors and NACK (Received)

Common Config SCOM Address	Value (31:0)	DL Being Monitored	PMU Counter SCOM Address	Counter 1 (31:16)	Counter 0 (15:0)
x"701334E"	x"F8001403"	MC01_DL0	x"701334F"	CRC errors	NACK received
x"701334E"	x"FA001403"	MC01_DL1	x"701334F"	CRC errors	NACK received
x"701334E"	x"FC001403"	MC01_DL2	x"701334F"	CRC errors	NACK received
x"701338E"	x"F8001403"	MC01_DL3	x"701338F"	CRC errors	NACK received
x"701338E"	x"FA001403"	MC01_DL4	x"701338F"	CRC errors	NACK received
x"701338E"	x"FC001403"	MC01_DL5	x"701338F"	CRC errors	NACK received
x"70133CE"	x"F8001403"	MC01_DL6	x"70133CF"	CRC errors	NACK received
x"70133CE"	x"FA001403"	MC01_DL7	x"70133CF"	CRC errors	NACK received
x"801334E"	x"F8001403"	MC23_DL0	x"801334F"	CRC errors	NACK received
x"801334E"	x"FA001403"	MC23_DL1	x"801334F"	CRC errors	NACK received
x"801334E"	x"FC001403"	MC23_DL2	x"801334F"	CRC errors	NACK received
x"801338E"	x"F8001403"	MC23_DL3	x"801338F"	CRC errors	NACK received
x"801338E"	x"FA001403"	MC23_DL4	x"801338F"	CRC errors	NACK received
x"801338E"	x"FC001403"	MC23_DL5	x"801338F"	CRC errors	NACK received
x"80133CE"	x"F8001403"	MC23_DL6	x"80133CF"	CRC errors	NACK received
x"80133CE"	x"FA001403"	MC23_DL7	x"80133CF"	CRC errors	NACK received

3.1.2 ODL CRC Errors and NACKs Received

Setting up the ODL crc and nack counters, all ODL can be monitored at the same time.

Table 3-2. ODL CRC Errors and NACKs (Received)

Optical Region	Common Config SCOM Address	Value (0:31)	DL being Monitored	PMU Counter SCOM Address	Counter 0 (0:15)	Counter 1 (0:15)
0	x"901081C"	x"A500FF00"				
0	x"901081D"	x"44454445"				
0			DL0	x"901081E"	CRC errors	NACK received
0			DL1	x"901081F"	CRC errors	NACK received
1	x"A01081C"	x"A000F000"				
1	x"A01081D"	x"44450000"				
1			DL2	x"A01081E"	CRC errors	NACK received
3	x"C01081C"	x"A500FF00"				
3	x"C01081D"	x"44454445"				
3			DL3	x"C01081E"	CRC errors	NACK received
3			DL4	x"C01081F"	CRC errors	NACK received
2	x"B01081C"	x"A000F000"				
2	x"B01081D"	x"44450000"				
2			DL5	x"B01081E"	CRC errors	NACK received

3.2 OMI DL Dynamic Lane Reduction

To test this function you would use the SCOM address in the DL to change the desired widths. Bits 61:60 of the config1 register control the desired width.

Table 3-3. Config1 Register (Bits 61:60)

Value of bits 61:60	Lane width
"00"	full width
"01"	quarter
"10"	half
"11"	full

Table 3-4. DL Controlled by Config1 SCOM Address

Config1 SCOM Address	DL being Controlled
x"7013351"	MC01_DL0
x"7013361"	MC01_DL1
x"7013371"	MC01_DL2
x"7013391"	MC01_DL3
x"70133A1"	MC01_DL4

Config1 SCOM Address	DL being Controlled
x”70133B1”	MC01_DL5
x”70133D1”	MC01_DL6
x”70133E1”	MC01_DL7
x”8013351”	MC23_DL0
x”8013361”	MC23_DL1
x”8013371	MC23_DL2
x”8013391”	MC23_DL3
x”80133A1”	MC23_DL4
x”80133B1”	MC23_DL5
x”80133D1”	MC23_DL6
x”80133E1”	MC23_DL7

4. Errata

4.1 OMI ASIC Errata

4.1.1 Replay Flits have good crc as a single beat

ERROR: Data integrity error when a replay flit is mis-interpreted.

If the first 3 beats of a replay flit are all zeroes, the fourth beat can pass CRC checking as a single beat with good CRC. The Host side will transmit a non-zero third beat to ensure the crc on the fourth beat is not the same as a single beat transfer to work around this issue. The host receiving side will also ensure single beat transfers with good CRC have a run length of “F” to validate it as a short idle flit.

4.1.2 Retraining while calibrating a lane that was powered down

ERROR: Retrain hang when calibrating lane doesn’t send TL data.

After a retrain, the host side needs to send all nine replay flits before retraining again to ensure that the calibration complete messages can be exchanged.

4.1.3 Calibration reset to lane zero

ERROR: A recal message of “01” was supposed to reset the next lane to recalibrate to be lane 0 (or 7 if reversed) but it does not.

Workaround is to disable power management if the host ever wants to reset the lanes to zero because of an error.

4.1.4 Pre-IPL PRBS configuration bit needs to be disabled

ERROR: The Pre-IPL PRBS mode corrupts the attached PHY because it turns on the run_lanes signal to PHY before random data is being received. Therefore the PHY can’t train properly Configuration1 register bit59 needs to be a zero.

Workaround is to manually drive the PRBS pattern before training by writing the training mode to 0x6 to drive random data, and write it to a 0x1 to drive pattern A. and then write it to an 0x8 to start auto link training. the IBM phy doesn’t need to receive a PRBS first, so the attached device can start in training mode 0x8.

4.1.5 More than 9 replay flits can cause a data integrity issue

ERROR: When 10 or more replay flits are received followed by a stream of flits which contain exactly 1 duplicate flit, the receive logic does not assert the valid to the TL on the right beat causing a data integrity issue.

Workaround is to ensure that the device connected to the OCMB sends exactly 9 replay flits when doing a replay. The IBM host has been updated to always send 9 replay flits. Any other hosts would also have to ensure this restriction.

4.2 DLR Errata with OMI ASIC

4.2.1 Quarter width mode can force degraded mode

ERROR: When running in quarter width mode and there is a request to switch to half width mode, but one side of the link detects an unexpected retrain, it will attempt to train to full width. So one side is attempting togo to x4 and the other side to x8. This causes the link the train in a degraded half width mode.

Workaround is to prohibit quarter width mode from every being requested.

4.2.2 PM message can cause the link to hang

ERROR: When a PM message hits a stream of data from the TL that the following pattern, the link will reportan error and turn off.

- Link is idle and array bypass (fastpath) is enabled
- 28-38 flits in a row
- 1 cycle gap
- 1 flit
- 1 cycle gap
- 1 flit -> new pm message request
- 1 cycle gap
- 1+ flits

Workaround is to disable array bypass. This has only been seen in simulation. We are not sure the TLX cansend this data stream.

4.2.3 1 us dead time when up-shifting in width

ERROR: When the link is up-shifting, it is doing a full retrain of the link with takes 925ps, the goal was around200ps.

Workaround is to wait for it to finish.

5. ODL Registers

The registers in the ODL are accessible via SCOM (dynamic scan). There is no TL to DL interface to send an MMIO (Memory Mapped I/O) to the ODL, therefore a device driver will need to access the XSCOM to access the ODL registers.

Table 5-1. ODL Registers

SCOM Address	Name	Width	Comment
x0D	DL TX Sync	64	Synchronize DL TX counters
x0E	Common Configuration	32	Configuration settings that are common to multiple DLs
x0F	Performance Counters	64	(4) 16-bit performance counters
x10	DL0 DL Configuration0	64	Configuration setting for this DL
x11	DL0 DL Configuration1	64	Extra configuration settings for this DL
x12	DL0 Error Mask Register	48	Error mask for each error detectable by the DL
x13	DL0 Error Hold Register	48	Sticky register to show if the error was hit since the last reset of errors
x14	DL0 Error Capture Register	63	Error information from the first error seen by this DL
x15	DL0 EDPL Max Count register	64	(8) 8-bit counters reflect the maximum parity errors hit in the defined window
x16	DL0 Status	64	Status of this DL
x17	DL0 Training Status	64	Status of the training states
x18	DL0 Remote Configuration	32	Configuration setting for the endpoint of this link
x19	DL0 Remote Message	64	Remote message from the remote TL
x1D	DL0 Error Actions	48	Indicates which DL link error bits are sent across the link
x1E	DL0 Debug Aid register	64	Indicates the state of the prbs7 checker
x1F	DL0 Chicken Switch bits	32	Chicken switches for future enhancements
x20	DL1 DL Configuration	64	Configuration setting for this DL
x21	DL1 DL Configuration1	64	Extra configuration settings for this DL
x22	DL1 Error Mask Register	48	Error mask for each error detectable by the DL
x23	DL1 Error Hold Register	48	Sticky register to show if the error was hit since the last reset of errors
x24	DL1 Error Capture Register	63	Error information from the first error seen by this DL
x25	DL1 EDPL Max Count register	64	(8) 8-bit counters reflect the maximum parity errors hit in the defined window
x26	DL1 Status	64	Status of this DL
x27	DL1 Training Status	64	Status of the training states
x28	DL1 Remote Configuration	32	Configuration setting for the endpoint of this link
x29	DL1 Remote Message	64	Remote message from the remote TL
x2D	DL1 Error Actions	48	Indicates which DL link error bits are sent across the link
x2E	DL1 Debug Aid register	64	Indicates the state of the prbs7 checker
x2F	DL1 Chicken Switch bits	32	Chicken switches for future enhancements
x30	DL0 DLR_APCR	64	DLR Activity Proxy Control Register (see DLR spec for details)
x31	DL0 DLR_APSR	64	DLR Activity Proxy Sample Register (see DLR spec for details)

SCOM Address	Name	Width	Comment
x32	DL0 DLR_APOR	64	DLR Activity Proxy Override Register (see DLR spec for details)
x33	DL0 DLR_APST	64	DLR Activity Proxy Status Register (see DLR spec for details)
x38	DL1 DLR_APCR	64	DLR Activity Proxy Control Register (see DLR spec for details)
x39	DL1 DLR_APSR	64	DLR Activity Proxy Sample Register (see DLR spec for details)
x3A	DL1 DLR_APOR	64	DLR Activity Proxy Override Register (see DLR spec for details)
x3B	DL1 DLR_APST	64	DLR Activity Proxy Status Register (see DLR spec for details)

5.1 DL TX Sync Register [SCOM addresses of x‘0D’]

Table 5-2. DL TX Sync Register

Bits	Access	Init Value	Name	Description
63:33	RO	0	Reserved	Reserved
32	RW	0	TX sync enable	When set to ‘1’ the sync logic will be enabled. The delay value added to the number of physical staging latches to that DL must be equal across all 8 DLs. When the 33-cycle counter matches the delay value of DL, a pulse willbe launched toward that DL to synchronize their TX sync header pointers. One TX sync register will be the master for the group of 8 DLs. The other registers will be unused and their contents will not be used.
31:28	RW	0001	DL7 Delay value	Delay value for this DL
27:24	RW	0001	DL6 Delay value	Delay value for this DL
23:20	RW	0001	DL5 Delay value	Delay value for this DL
19:16	RW	0001	DL4 Delay value	Delay value for this DL
15:12	RW	0001	DL3 Delay value	Delay value for this DL
11:8	RW	0001	DL2 Delay value	Delay value for this DL
7:4	RW	0001	DL1 Delay value	Delay value for this DL
3:0	RW	0001	DL0 Delay value	Delay value for this DL

5.2 Common Configuration Register [SCOM addresses of x‘0E’]

Table 5-3. Common Configuration Register (SCOM address x‘0E')

Bits	Access	Init Value	Name	Description
63:60	RW	0	Spare	Spare
59:56	RW	0100	PM CDR timer	Time needed after a lane is turned on until the PHY guarantees that the CDR is locked assuming scrambled data is being received. These valuesare the number of clock cycles and the times specified assume a 625ps period. This timer value must be greater than the di/dt timer in bits 55:52. “0000” - disabled “0001” - 96 cycles = ~60 ns “0010” - 200 cycles = ~125 ns “0011” - 296 cycles = ~185 ns “0100” - 400 cycles = ~250 ns “0101” - 600 cycles = ~375 ns “0110” - 800 cycles = ~500 ns “0111” - 1200 cycles = ~750 ns “1000” - 1600 cycles = ~1 us “1001” - 3200 cycles = ~2 us “1010” - 6400 cycles = ~4 us “1011” - 12,800 cycles = ~8 us “1100” - 25,600 cycles = ~16 us “1101” - 51,200 cycles = ~32 us “1110” - 102,400 cycles = ~64 us “1111” - 204,800 cycles = ~128 us
55:52	RW	0010	PM di/dt timer	Time needed after a lane is turned on or off, and the next lane can be turnedon or off. This is needed to prevent a current spike. These values are the number of clock cycles and the times specified assume a 625ps period. “0000” -disabled “0001” - 8 cycle = ~5 ns “0010” - 16 cycles = ~10 ns “0011” - 24 cycles= ~15 ns “0100” - 32 cycles = ~20 ns “0101” - 48 cycles = ~30 ns “0110” - 72 cycles = ~45 ns “0111” - 96 cycles = ~60 ns “1000” - 144 cycles = ~90 ns “1001” - 200 cycles = ~125 ns “1010” - 296 cycles = ~185 ns “1011” - 400 cycles = ~250 ns “1100” - 600 cycles = ~375 ns “1101” - 800 cycles = ~500 ns “1110” - 1200 cycles = ~750 ns “1111” - 1600 cycles = ~1 us
63:58	RW	0	Spare	Spare
57	RW	0	Rx Edge Align	When set, each Rx lane will use the bump UI to the Phy to align the recovered lane clock to the main clock
56:52	RW	00000	Rx Clock Margin	Number of UI away from aligned clocks each Rx lane is bumped

Bits	Access	Init Value	Name	Description
51	RW	1	Psave_statusenable	When ‘1’, the DL will wait for the psave_status inputs from the PHY to determine when the link is up ready to be retrained.
50:48	RW	011	Calibration Timer	Time between calibration request to a given lane. This timer uses the 1ustimer as defined in bits 47:36. “000” -disabled “001” - 25 ms “010” - 50 ms “011” - 100 ms “100” - 200 ms “101” - 400 ms “110” - 800 ms “111” - 1.6 seconds
47:36	RW	x“640”	1us Timer	Number of clock cycles needed to equal 1us. The timers use this value or a multiple of it.
35	RW	1	Debug enable	Clock gating for the debug/trace logic
34:32	RW	001	Debug Select	Each common configuration register supports upto 3 DLs, and the registers work together to select with DL information traced. “000” - zeros “001” - DL0 trace information “010” - DL1 trace information “011” - DL2 trace information “100” - trace information common macro 0 “101” - trace information common macro 2“110” - 11 bits from all 8 DLs . . . . . .(22 bits from common 0 plus . . . . . .11 bits from all 3 DLs plus . . . . . .33 bits from common macro) “111” - zeros
31	RW	1	reset on read	Reset the performance counters when the counters are read
30:28	RW	0	pre-scaler	“000” -16-bit pre-scaler “001” – 8-bit pre-scaler “100” – 20-bit pre-scaler“111” - no pre-scaler
27	RW	1	Freeze Mode	“1” - Freeze when any of the four counters wrap
26:24	RW	0	Port Select	Select which of the 8 input ports will be counted
23:22	RW	01	Cntr3 Bit pair select	“00” - select odd bit “01” - select even bit “10” - select and of both bits“11” - select xor of both bits
21:20	RW	11	Cntr3 event select	“00” select bits 7:6 “01” select bits 5:4 “10” select bits 3:2 “11” select bits 1:0
19:18	RW	01	Cntr2 Bit pair select	“00” - select odd bit “01” - select even bit “10” - select and of both bits“11” - select xor of both bits
17:16	RW	00	Cntr2 event select	“00” select bits 7:6 “01” select bits 5:4 “10” select bits 3:2 “11” select bits 1:0

Bits	Access	Init Value	Name	Description
15:14	RW	01	Cntr1 Bit pair select	“00” - select odd bit “01” - select even bit “10” - select and of both bits“11” - select xor of both bits
13:12	RW	10	Cntr1 event select	“00” select bits 7:6 “01” select bits 5:4 “10” select bits 3:2 “11” select bits 1:0
11:10	RW	00	Cntr0 Bit pair select	“00” - select odd bit “01” - select even bit “10” - select and of both bits“11” - select xor of both bits
9:8	RW	11	Cntr0 event select	“00” select bits 7:6 “01” select bits 5:4 “10” select bits 3:2 “11” select bits 1:0
7	RW	0	Positive edge count	Count positive edges for counter 3
6	RW	0	Positive edge count	Count positive edges for counter 2
5	RW	0	Positive edge count	Count positive edges for counter 1
4	RW	0	Positive edge count	Count positive edges for counter 0
3	RW	1	Counter enable	Enable counter 3
2	RW	1	Counter enable	Enable counter 2
1	RW	1	Counter enable	Enable counter 1
0	RW	1	Counter enable	Enable counter 0

6. Performance Monitor Counters [SCOM addresses of x‘0F’]

There are three ways that this register can be reset:

- Write to this register
- Write to the Common Configuration register
- Read of this register while Common Configuration register bit 31 is set

Table 6-1. Performance Monitor Counters

Bits	Access	Initial Value	Name	Description
63:48	RW	0	Counter3	16-bit counter
47:32	RW	0	Counter2	16-bit counter
31:16	RW	0	Counter1	16-bit counter
15:0	RW	0	Counter0	16-bit counter

6.1 Configuration0 Register0 (1 per ODL) [SCOM addresses of x‘10’, x‘20’, x‘30’]

Table 6-2. Configuration0 Register

Bits	Access	Init Value	Name	Description
63	RW	0	Enable	Enable the DL (used for clock gating)
62:58	RW	0	Spare	Spare
57:52	RW	0	TL credits	Maximum number of frame buffer credits that can be sent to the TL. Maximum: 32 Minimum: 12
51:48	RW	0	TL event Actions	When a TL event arrives, what action should be sent to the endpoint. Multi-ple actions can be taken at the same time. 51: Freeze entire endpoint: AFU, TL, and DL 50: Freeze the AFU only 49: Trigger the Internal Logic Analyzers 48: Bring down the link
47:44	RW	0	TL error Actions	When a TL error arrives, what action should be sent to the endpoint. Multiple actions can be taken at the same time. 47: Freeze entire endpoint: AFU, TL, and DL 46: Freeze the AFU only 45: Trigger the Internal Logic Analyzers 44: Bring down the link

Bits	Access	Init Value	Name	Description
43:40	RW	0100	No Forward Prog-ress Timer	Timer for the “no forward progress” error. If a head flit is stuck for thisamount of time, a retrain of the link is preformed. “0000” - 1 us “0001” - 2 us “0010” - 4 us “0011” - 8 us “0100” - 16 us “0101” - 32 us “0110” - 64 us “0111” - 128 us “1000” - 256 us “1001” - 512 us “1010” - 1 ms “1011” - 2 ms “1100” - 4 ms “1101” - 8 ms “1110” - 16 ms “1111” - disabled
39:36	RW	0	Replay Buffers reserved	There are 256 - 16B entries in the replay buffer. This value multiplied by 32is the number of entries that can be reserved for testing buffer full conditionsA value of “F” is reserved.
35:33	RW	111	Debug Select	“000” - zeros “001” - Rx information “010” - Tx FLT information “011” - Tx Train information “100” - bits 10:0 only of Rx information “101” - bits 10:0 only of Tx flit information “110” - bits 10:0 only of Tx train information “111” - default trace settings
32	RW	1	Debug enable	Clock gating for the debug/trace logic. Also allows the status registers to update every cycle instead of every 1us. (Status, Training Status, Remote Message, EDPL Max Count, and Error Capture Register)
31	RW	0	Inject DL2TL parity error	Inject a single parity error into the DL2TL flit. Injection occurs on the risingedge of this bit. This injects a parity error on the data coming into the DL, so if the flit was an idle or replay, it would not be sent to the TL. This will need to be pulsed many times to ensure a parity error is sent to the TL.
30	RW	0	Reserved	Reserved
29	RW	0	Inject ECC UE	Inject a single ECC UE into the Frame buffer/Replay buffer data
28	RW	0	Inject ECC CE	Inject a single ECC CE into the Frame buffer/Replay buffer data
27	RW	0	Disable Fast path	Disable frame buffer bypass “fast path” in TX flit
26	RW	0	Spare	Spare
25	RW	0	Enable TX lane reversal	When set, the TX is allowed to reverse the lanes if the remote side requested it in the deskew marker. When not set, the DL will set error bit 9when the remote side requests it.
24	RW	0	128/130 Encoding Enable	Allow the link to attempt to train using 128/130 encoding

Bits	Access	Init Value	Name	Description
23:20	RW	0001	PHY control limit	Length of time the Rx needs to receive a pattern A or pattern B before advancing state. x"0" = 1 us x"1" = 50 us x"2" = 100 us x"3" = 200 us x"4" = 500 us x"5" = 1 ms x"6" = 2 ms x"7" = 3 ms x"8" = 4 ms x"9" = 5 ms x"A" = 6 ms x"B" = 8 ms x"C" = 10 ms x"D" = 15 ms x"E" = 30 ms x"F" = 60 ms
19	RW	0	Run Lane override enable	When set, the DL will assert the run lane signal to the PHY for all training states causing the PHY to CDR lock on the incoming data without waiting tobe in state 4. The remote should have Pre-IPL PRBS enabled (config1 bit 59) before this DL is enabled to ensure the link has transitions.
18	RW	1	Power-Management Enable	[category DL3.0] Not supported. [category DL3.1] Allow the link to change lane width based on input from theTL and/or a DL configuration register.
17	RW	0	Quarter width degrade modeenable	[category DL3.0] Not supported [category DL3.1] Not supported [category DL4.0] Allow x8 traffic to flow as quarter width mode on lanes (7,0) or (5,2)
16	RW	1	Half width degrade mode enable	Allow the link to run in half bandwidth mode. All lanes of a group will be disabled if one lane in their group doesn’t train. If this bit is ‘0’, and a lane doesnot train, the link should not train. [category DL3.0] x8 lane groups, (6,4,2,0) & (7,5,3,1)[category DL3.1] x8 lane groups, (7,5,2,0) & (6,4,3,1) [category DL3.1] x4 lane groups, (7,0) & (5,2)
15:12	RW	0011	Supported Widths	Vector of supported widths. Both sides will negotiate to best width. Priority of widths is x32, x16, x8, x4. (15) = x32 [Not supported] (14) = x16 [Not supported] (13) = x08 (12) = x04
11:8	RW	1000	Training Mode	1xxx = enable training 0000 = send zero 0001 = send pattern A 0010 = send pattern B 0011 = send sync 0100 = send ts1 0101 = send ts2 0110 = send ts3 0111 = send ts0
7:2	RW	001000	OpenCAPI version	Matches the 1.0 OpenCAPI 3.0/3.1/4.0 DL Architecture Specification
1	RW	0	Retrain	Reset the training sequence to sending of control sync headers with TS1 pattern

Bits	Access	Init Value	Name	Description
0	RW	0	Reset	Reset ODL to Power-on Values

6.2 Configuration1 Register (1 per ODL) [SCOM Addresses of x11, x21, x31]

Table 6-3. Configuration1 Register

Bits	Access	Init Value	Name	Description
63:62	RW	0	Spare	Spare
61:60	RW	00	Lane Width Selection	Configuration override to select lane width for dynamic lane power down modes. “00” = TL controlled via input ports “01” = DL override to quarter width “10” = DL override to half width “11” = DL override to full width
59	RW	0	Pre-IPL PRBS enable	When enabled, will send scrambled data before training starts to allow the receivers to lock for the amount of time specified in the Pre-IPL timer
58:56	RW	100	Pre-IPL PRBS timer	Amount time to send scrambled data before training starts “000” = 256 us “001” = 1 ms “010” = 4 ms “011” = 16 ms “100” = 64 ms “101” = 256 ms “110” = 1 s “111” = 4 s
55:52	RW	0000	Pattern B Hysteresis	Number of consecutive pattern B seen before indicating received pattern B. “0000” = 16 “0001” = 24 “0010” = 32 “0011” = 40 “0100” = 48 “0101” = 56 “0110” = 64 “0111” = 72 “1000” = 80 “1001” = 96 “1010” = 128 “1011” = 256 “1100” = 512 “1101” = 1K “1110” = 2K “1111” = 4K
51:48	RW	0000	Pattern A Hysteresis	Number of consecutive pattern A seen before indicating received pattern A. “0000” = 16 “0001” = 24 “0010” = 32 “0011” = 48

Bits	Access	Init Value	Name	Description
				“0100” = 64 “0101” = 96 “0110” = 128 “0111” = 256 “1000” = 512 “1001” = 1024 “1010” = 2K “1011” = 4K “1100” = 8K “1101” = 16K “1110” = 32K “1111” = 64K
47:46	RW	00	Pattern B length	Number of consecutive 1’s and 0’s needed to represent training Pattern B=> “11111111111111110000000000000000” “00” = two X’s => “11111111111111XX00000000000000XX” “10” = four X’s => “111111111111XXXX000000000000XXXX” “01” = one X’s => “111111111111111X000000000000000X” “11” = same as “10”
45:44	RW	00	Pattern A length	Number of consecutive 1’s and 0’s needed to represent training Pattern A => “1111111100000000” “00” = two X’s => “111111XX000000XX” “10” = four X’s => “1111XXXX0000XXXX” “01” = one X’s => “1111111X0000000X” “11” = same as “10”
43:42	RW	01	Tx degraded threshold	Percent of Tx traffic repeated due to crc errors before setting FIR“00” = 1% “01” = 2% “10” = 3% “11” = 4%
41:40	RW	01	Rx degraded threshold	Percent of Rx traffic that due to crc errors before setting FIR“00” = 1% “01” = 2% “10” = 3% “11” = 4%
39:32	RW	0	Tx Lanes disable	Prevent Tx lanes (7:0) from training.
31:24	RW	0	Rx Lanes disable	Prevent Rx lanes (7:0) from training.
23:20	RW	0	Macro Debug select	Allow each macro to select different things to send to the debug bus
19	RW	0	Reset Error Hold register	Reset error hold register when it is read
18	RW	0	Reserved	Reserved
17	RW	0	Reset EDPL Threshold Register	Reset EDPL threshold register when it is read
16	RW	0	Reserved	Reserved
15	RW	0	Lane injection direction	‘0’ Inject error on Rx side of link‘1’ Inject error on Tx side of link
14:12	RW	0	Lane Bit Errorinjection rate	Injection rate for the lane bit errors (power of 2 injection rate)“000” = 1 us “001” = 8 us “010” = 64 us “011” = 512 us “100” = 4 ms

Bits	Access	Init Value	Name	Description
				“101” = 32 ms “110” = 256 ms “111” = 2 s
11:9	RW	b000	Lane select	Lane to inject a periodic error
8	RW	0	Lane bit error injection enable	Inject a periodic error on a lane
7:4	RW	b0101	EDPL Time Window	Power of 2 windows “0000” = no time window “0001” = 4 us = 1x10^5 bits “0010” = 32 us = 8x10^5 bits “0011” = 256 us = 6.4x10^6 bits “0100” = 2 ms = 5.0x10^7 bits “0101” = 16 ms = 4x10^8 bits “0110” = 128 ms = 3,2x10^9 bits “0111” = 1 s = 2.5x10^{10} bits “1000” = 8 s = 2x10^{11} bits “1001” = 64 s = 1,6x10^{12} bits “1010” = 512 s = 1.2x10^{13} bits “1011” = 4 ks = 1.0x10^{14} bits “1100” = 32 ks = 8x10^{14} bits “1101” = 256 ks = 6.4x10^{15} bits “1110” = 2 Ms = 5.1x10^{16} bits “1111” = 16 Ms = 4x10^{17} bits = 189 days
3:1	RW	“100”	EDPL ErrorThreshold	“000” = disabled “001” = 2 errors “010” = 4 errors “011” = 8 errors “100” = 16 errors “101” = 32 errors “110” = 64 errors “111” = 128 errors
0	RW	1	EDPL Enable	Error Detection Per Lane Override (EDPL) is based on version number. However, if this bit is a zero it will overridethe exchanged mode of operation. ‘1’ - Based on version number ‘0’ - Disabled

6.3 Error Mask Register (1 per ODL) [SCOM addresses of x‘12’, x‘22’, x‘32’]

Table 6-4. Error Mask Register

Bits	Access	Initial Value	Name	Description
63:48	RW	0	Reserved	Reserved
47:0	RW	FFE3_0003_C000	Error Mask	When set, this error will be prevented from being detected.

6.4 Error Hold Register (1 per ODL) [SCOM addresses of x‘13’, x‘23’, x‘33’]

Table 6-5. Error Hold Register

Bits	Access	Initial Value	Name	Description
63	RW	0	Reset	Resets the cerr hold registers
62:48	RO	0	Reserved	Reserved
47:0	RO	o	Cerr holds	Cerr hold registers, see implement workbook for individual descriptions

6.5 EDPL Max Count Register (1 per ODL) [SCOM addresses of x‘15’, x‘25’, x‘35’]

Any write to this register will reset the entire register.

Table 6-6. EDPL Max Count Register

Bits	Access	Initial Value	Name	Description
63:56	RW	0	Max errors lane7	Maximum parity errors detected on this lane in the given time window
55:48	RW	0	Max errors lane6	Maximum parity errors detected on this lane in the given time window
47:40	RW	0	Max errors lane5	Maximum parity errors detected on this lane in the given time window
39:32	RW	0	Max errors lane4	Maximum parity errors detected on this lane in the given time window
31:24	RW	0	Max errors lane3	Maximum parity errors detected on this lane in the given time window
23:16	RW	0	Max errors lane2	Maximum parity errors detected on this lane in the given time window
15:8	RW	0	Max errors lane1	Maximum parity errors detected on this lane in the given time window
7:0	RW	0	Max errors lane0	Maximum parity errors detected on this lane in the given time window

6.6 Status Register (1 per ODL) [SCOM addresses of x‘16’, x‘26’, x‘36’]

Table 6-7. Status Register

Bits	Access	Initial Value	Name	Description
63:60	RW	0	Trained Mode	Status of negotiated lane width. Status of degraded modes don’t getreflected this field. (63) = x 32 [not supported] (62) = x16 [not supported] (61) = x8 (60) = x4
59	RW	0	Rx lanes reversed	Rx lanes have been reversed
58	RW	0	Tx lanes reversed	Tx lanes have been reversed
57	RW	0	Idle Flit negotiatedsize	1 = short idle flits are supported0 = idle flits are 64B
56	RW	0	Replay pointersmatch	Indicates that all transmitted flits have been acknowledged
55:52	RW	0	Reserved	

Bits	Access	Initial Value	Name	Description
51:50	RW	0	Requested lanewidth	Requested lane wide for dynamic power savings. Will mirror the TL inputs when DL configuration register is set to TL, otherwise will mirror the DL configuration register setting. 00=power management not invoked01=quarter width 10=half width11=full width
49:48	RW	0	Actual lane width	Current operating lane width of this DL. Will reflect lower widths while new lanes are being powered on. 00 = training or retraining 01 = quarter width 10 = half width 11 = full width
47:40	RW	0	Tx trained lanes	Indicates which Tx lanes were trainable
39:32	RW	0	Rx trained lanes	Indicates which Rx lanes were trainable
31:28	RW	0	Endpoint supported widths	Endpoint configuration information, received from endpoint during training. Supported widths: equivalent to configuration0 register bits 15:12.
27:26	RW	0	Reserved	
25:20	RW	0	Endpoint version number	Endpoint configuration information, received from endpoint during training. Version number: equivalent to configuration0 register bits 7:2.
19	RW	0	Endpoint TX ordering	Endpoint configuration information, received from endpoint during training. Degraded mode transmission order: equivalent to configuration0 register bits 19
18	RW	0	Endpoint lane swap requested	Endpoint configuration information, received from endpoint during training. Transmit lane swap requested. Endpoint detected a Rx lane reversal but doesn’t have the logic to support lane reversal.
17	RW	0	Endpoint power management supported	Endpoint configuration information, received from endpoint during training. Power management supported: equivalent to configuration0 register bits 18.
16	RW	0	PM disabled	Power management is disabled. Either due to not being enabled or previous error (that is, degraded width).
15	RW	0	Reserved	
14:12	RW	0	Training State	Current training state 000 = send zero 001 = send pattern A 010 = send pattern B 011 = send sync 100 = send ts1 101 = send ts2 110 = send ts3 111 = link trained
11:9	RW	0	Reserved	
8	RW	0	Deskew information matched	Received eight consecutive matching deskew patterns
7:0	RW	0	Lanes Disabled	Rx lanes that have been disabled, either due to not being trained or a different lane in its lane group did not train.

6.7 Training Status Register (1 per ODL) [SCOM addresses of x‘17’, x‘27’, x‘37’]

This register is used if the link does not train to help identify the lanes that are not trainable. It is possible to train the link and have some of the lanes report that they did not see enough ts1, ts2, or ts3.

Table 6-8. Training Status Register

Bits	Access	Initial Value	Name	Description
63:56	RW	0	pattern A	Information from Rx lanes (one bit per lane)
55:48	RW	0	pattern B	Information from Rx lanes (one bit per lane)
47:40	RW	0	sync pattern	Information from Rx lanes (one bit per lane)
39:32	RW	0	Phy Init Done	Information from Rx lanes (one bit per lane)
31:24	RW	0	Block locked	Information from Rx lanes (one bit per lane)
23:16	RW	0	TS1 received	Information from Rx lanes (one bit per lane)
15:8	RW	0	TS2 received	Information from Rx lanes (one bit per lane)
7:0	RW	0	TS3 received	Information from Rx lanes (one bit per lane)

6.8 Endpoint Configuration Register (1 per ODL) [SCOM addresses of x‘18’, x‘28’, x‘38’]

Table 6-9. Endpoint Configuration Register

Bits	Access	Initial Value	Name	Description
63:32	RO	0	Reserved	Reserved
31:0	RW	0	Endpoint configuration	Configuration information to be passed to the endpoint. (exact content to be determined by the endpoint) When this register is written, the DL will send areplay sequence to get the information to the endpoint.

6.9 Endpoint Information Register (1 per ODL) [SCOM addresses of x‘19’, x‘29’, x‘39’]

Table 6-10. Endpoint Information Register

Bits	Access	Initial Value	Name	Description
63:32	RO	0	DLx Info	Dlx information that was received in a replay flit (exact content to be determined by DLx)
31:0	RO	0	TLx Info	Tlx information that was received in a replay flit (exact content to be determined by TLx)

6.10 Error Action Register (1 per ODL) [SCOM addresses of x‘1D’, x‘2D’, x‘3D’]

The 40 error bits are grouped together based on which fir bit that error sets. When an unmasked erroris detected, this register defines which of the four dlx error messages will be sent to the host. Each error can set any combination of the four error messages. The four endpoint error messages will be reflected in the host error register bits 47:44.

Table 6-11. Error Action Register

Bits	Access	Initial Value	Name	Description
63:48	RO	0	Reserved	Reserved
47:44	RW	0000	Fir 11 action	Error message to send for this fir group Bit 47: if Fir 11 is set, set link_error bit 3 Bit 46: if Fir 11 is set, set link_error bit 2 Bit 45: if Fir 11 is set, set link_error bit 1 Bit 44: if Fir 11 is set, set link_error bit 0
43:40	RW	0000	Fir 10 action	Error message to send for this fir group
39:36	RW	0000	Fir 9 action	Error message to send for this fir group
35:32	RW	0000	Fir 8 action	Error message to send for this fir group
31:28	RW	0000	Fir 7 action	Error message to send for this fir group
27:24	RW	0000	Fir 6 action	Error message to send for this fir group
23:20	RW	0000	Fir 5 action	Error message to send for this fir group
19:16	RW	0000	Fir 4 action	Error message to send for this fir group
15:12	RW	0000	Fir 3 action	Error message to send for this fir group
11:8	RW	0000	Fir 2 action	Error message to send for this fir group
7:4	RW	0000	Fir 1 action	Error message to send for this fir group
3:0	RW	0001	Fir 0 action	Error message to send for this fir group

6.11 Debug Aid Register (1 per ODL) [SCOM addresses of x‘1E’, x‘2E’, x‘3E’]

Table 6-12. Debug Aid Register

Bits	Access	Init Value	Name	Description
63:10	RO	0	Reserved	Reserved
9	RW	0	PRBS7 disable	Disable this DLs lanes from contributing to the PRBS error signal being returned to the PHY.
8	RW	1	PRBS7 reset	Reset the prbs7 error indicators. (only implemented on OMI ASIC)
7:0	RO	0	PRBS7 status	Per lane status of the prbs7 checker. The remote side TX must be transmitting a prbs7 before clearing the reset for this field to be valid

6.12 DL Chicken Switch Bits (1 per ODL)[SCOM addresses of x‘1F’, x‘2F’, x‘3F’]

Table 6-13. DL Chicken Switch Bits

Bits	Access	Init Value	Name	Description
63:17	RW	0	Chicken Switch bits	(OMI ASIC only implements bits(31:0)) Chicken switch control bits.
16	RW	0	fast_retrain_disable	Chicken switch to prevent the host from attempting to block lock thewaking lanes before the link goes into retrain.
15	RW	0	dly_lane_width_status	Chicken switch to delay the lane width status being returned to the TLto ensure proper delay between PM requests. When set to a ‘1’ the status is delayed 168 cycles.
14	RW	0	kill_crc_during_re-play	Chicken switch to bring down the link whenever a crc error is detected whilea replay is in progress. Enabled when set to a ‘1’.
13	RW	0	retrain_crc_ during_replay	Chicken switch to retrain the link whenever a crc error is detected while a replay is in progress. Enabled when set to a ‘0’.
12	RW	0	PM_disable_EDPL	Chicken switch to change function for Power management disabled due to EDPL threshold
11	RW	0	retrain_crc_retrain	Chicken switch to retrain the link whenever a crc error is detected before the tx_ack_pointer has been updated after a retrain. Enabled when set to a ‘0’.
10	RW	0	reset_crc_retrain	Chicken switch to retrain the link whenever a crc error is detected before the tx_ack_pointer has been updated after a reset. Enabled when set to a ‘0’.
9	RW	1	frbuf_full_retrain	Chicken switch to retrain the link whenever the frbuf is full and it is notdoing a replay. Enabled when set to a ‘0’.
8	RW	0	replay_frbuf_full_ retrain	Chicken switch to retrain the link whenever the frbuf is full, and it is doing a replay and the frbuf_rd_ptr is within 64 entires of the last flit sent from the replay buffer. Enabled when set to a ‘0’.
7	RW	0	pm_retrain	Chicken switch to bring the link down whenever a power management request is received before the tx_ack_pointer has been updated after aretrain. Enabled when set to a ‘1’.
6	RW	0	pm_reset	Chicken switch to bring the link down whenever a power management request is received before the tx_ack_pointer has been updated after areset. Enabled when set to a ‘1’.
5	RW	0	crc_retrain	Chicken switch to bring the link down whenever a crc error is detected before the tx_ack_pointer has been updated after a retrain. Enabled whenset to a ‘1’.
4	RW	0	crc_reset	Chicken switch to bring the link down whenever a crc error is detected before the tx_ack_pointer has been updated after a reset. Enabled when setto a ‘1’.
3	RW	0	Meso buffer enable (OMI ASIC only)	OMI ASIC: meso-synchronize buffer enable ‘1’ use meso buffer ‘0’ use tx clocked staging latch
2	RW	0	Meso buffer start(OMI ASIC only)	After the Rx clock is aligned to the tx main clock, set this bit to allow buffer depth value to be used. Must be set before training is enabled but after the remote side is sending prbs data.
1:0	RW	0	Meso buffer depth (OMI ASIC only)	1:0 Meso-synchronize starting buffer location