IEEE 802.3by 25G Ethernet TF
A BASELINE PROPOSAL FOR RS, PCS, AND FEC

Eric Baden (ericb at broadcom com), presenting
Ankit Bansal (ankitb at broadcom com),
Cedric Begin (cbegin at cisco com),
Gary Nicholl (gnicholl at cisco com)

Supporters

Adil Haque, Altera

Ali Ghiasi, Ghiasi Quantum LLC

Brad Booth, Microsoft

Dan Dove, Dove Networking Solutions / Emulex

Daniel Koehler, MorethanIP

Don Cober, CoMira Solutions

Erdem Matoglu, Amphenol

Howard Frasier, Broadcom

Jeff Maki, Juniper

Joel Goergen, Cisco

John D’Ambrosia, Dell

Jon Lewis, Dell

Jonathan King, Finisar

Kapil Shrikhande, Dell

Kent Lusted, Intel

Mike Andrewartha, Microsoft

Mike Dudek, Qlogic

Mike Peng Li, Altera

Oded Wertheim, Mellanox

Rich Mellitz, Intel

Rob Stone, Broadcom

Ron Muir, JAE

Vineet Salunke, Cisco

Vittal Balasubramanian, Dell

Yong Kim, Broadcom

MOTIVATION & GOALS

- Goal is to develop a simple & efficient PCS architecture with maximal re-use of current 802.3 specifications
- Two approaches available: scale up from 10G or scale down from 40G/100G
- Interest in supporting 3 FEC modes which cross the two available approaches
- Implementation complexity was examined and reviewed

- **CONCLUSION: It is simplest to leverage the existing single-lane PCS architecture to develop this next-gen single-lane PCS architecture**

HISTORY

This baseline proposal is the culmination of many joint presentations that have been presented and reviewed by the study group at regular and ad-hoc meetings.

- "25GbE PCS Technical Feasibility" - gustlin_081214_25GE_adhoc.pdf (initial overview of options)
- "PCS Thoughts and Considerations" - kim_100114_25GE_adhoc.pdf
- "25G RS/PCS Considerations – A follow up" - kim_100814_25GE_adhoc.pdf
- "Layering and Gaps" - baden_102214_25GE_adhoc.pdf
- "Architectural Thoughts – 25G Interconnect" - booth_102914_25GE_adhoc.pdf
- "25G Ethernet Layering and Gaps" - baden_25GE_01a_1114.pdf
- “Architectural Thoughts – 25G Interconnect”- booth_25GE_01a_1114.pdf

This baseline proposal to follow is consistent with the work built up during these contributions.

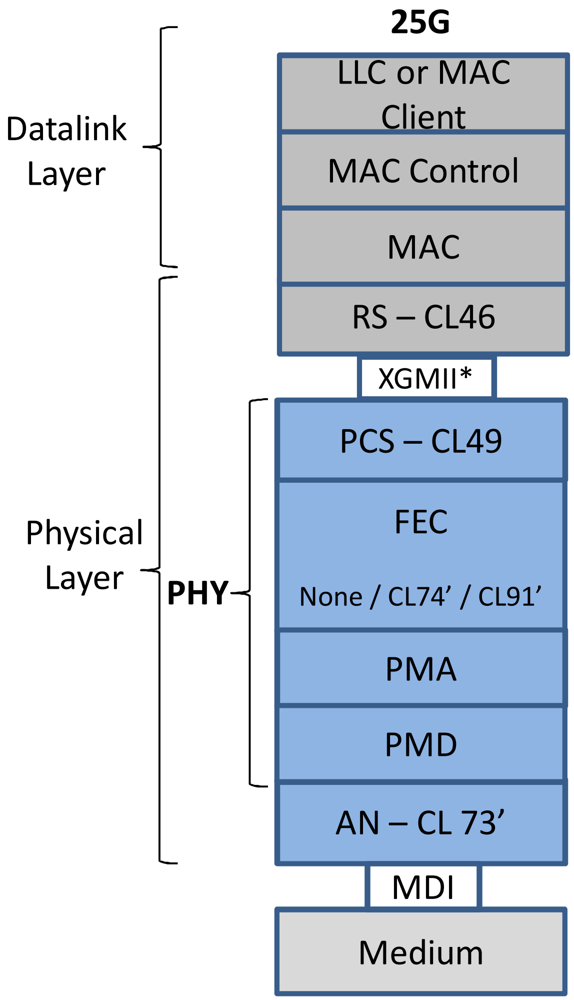
Source:
Plenary presentations: http://www.ieee802.org/3/25GSG/public/Nov14/index.html
Ad hoc presentations: http://www.ieee802.org/3/25GSG/public/adhoc/architecture/index.html

NOMENCLATURE

- **Throughout this proposal, there are references to existing clauses, as well as modified versions of the existing clauses.**
- **To minimize confusion, the following nomenclature will be adopted:**
  - **Existing clauses will be referred to in the usual manner (i.e. CL*nn*)**
  - **Modified clauses based on an existing clause will be referred to as CL*nn’***
- **It will be the subject of future work to determine how to codify the proposed specification into modified or creation of new clauses.**

OVERVIEW

- **25G IS SINGLE LANE PCS**
- **LEVERAGE EXISTING SINGLE LANE PCS (CL49)**
- **START WITH 10G KR AND SPEED UP**
  - USE CL46 RS LAYER (XGMII)
  - USE CL49 PCS
- **FEC OPTIONS:**
  - NO FEC
  - USE CL74 FEC (speed up)
  - USE CL91 (equivalent) FEC



*Will likely be renamed, left as-is for consistency with .bj

CHANGES SUMMARY

CLAUSE	Changes from existing Clause
CL45’ MDIO	• Add CL49’ PCS control for CWM insertion and IDLE deletion
	• Modify CL91 related registers for single lane support only • Add 25G CR4 / KR4 to port type and associated status registers
CL46’ RS	• Increase rate 10G → 25G
CL49’ PCS	• Increase rate 10G → 25G
CL74’ FEC	• Increase rate 10G → 25G
CL91’ RS-FEC	• Add provision for insertion of CodeWord Markers (CWM) and deletion of IDLE / ordered sets • Add provision for removal of CWMs and replacement with IDLEs. • Modify AM removal, mapping and insertion → CWM • Modify transcoding to extend control blocks for CL49’ and CL82 types • Delete multi-lane support (lane block sync → block sync, alignment lock & deskew → CWM lock, delete lane reorder)

OVERVIEW with optional FEC(s)

PCS/FEC	25G without any FEC	25G with CL74 FEC	25G with RS FEC
Block Coding	64/66B		
Lanes	1	1	1
RS	CL46 (4B)	CL46 (4B)	CL46 (4B)
PCS	CL49	CL49	CL49
Codeword Markers	N	N	Y
Transcode	N/A	N/A	256/257B
Reach	TBD	3m	5m
Latency	Low	Medium	High

SOME POINTS ABOUT AN RS FEC

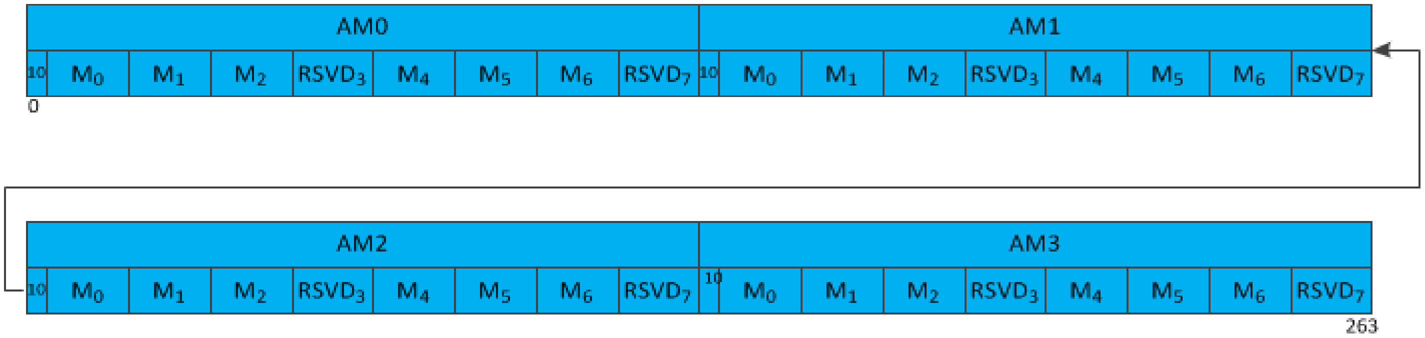
- AN RS FEC REQUIRES CODEWORD (CW) BOUNDARY IDENTIFICATION:

- THE **RS FEC** IDENTIFIES CODE WORD BOUNDARIES THROUGH THE USE OF CODEWORD MARKERS (CWMs):
  - HOW?
    - THE **RS FEC** SHALL PERIODICALLY INSERT A CWMs INTO THE PCS STREAM AT A FEC CODEWORD BOUNDARY
    - THE **RS FEC** SHALL DELETE EQUIVALENT IDLES/ORDERED SETS AS NECESSARY TO COMPENSATE FOR THE INSERTION OF THE CWMs.

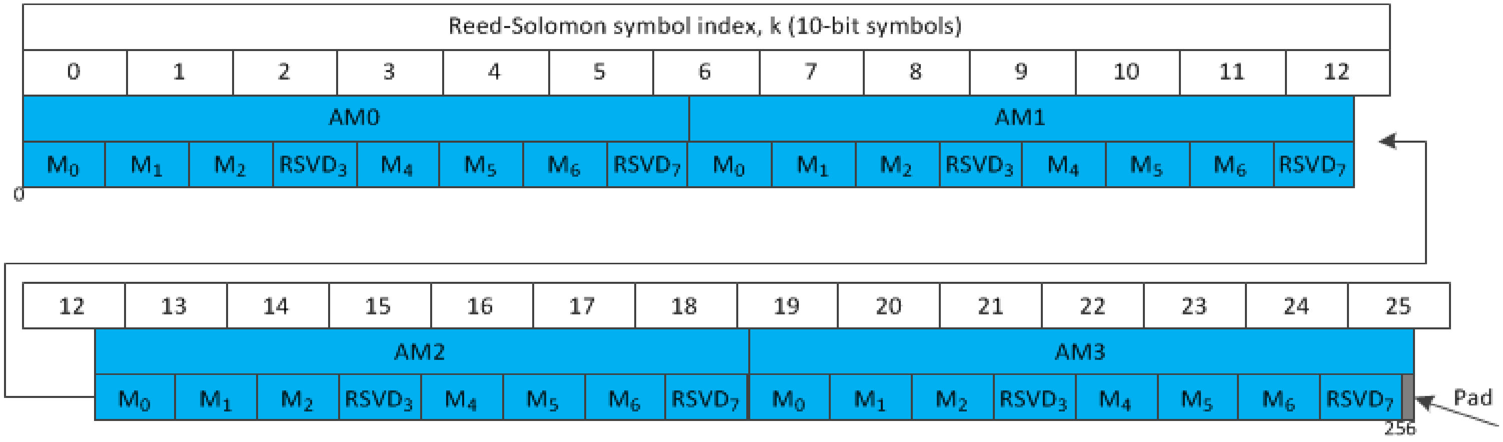
RS FEC Codeword Markers

- Codeword Markers (CWM):
  - Constructed from the concatenation of CL82 MLD 4 Alignment Markers 0 thru 3

25G W/CL91 FEC Code Word Marker (CWM)



CWM Mapping to RS Symbols



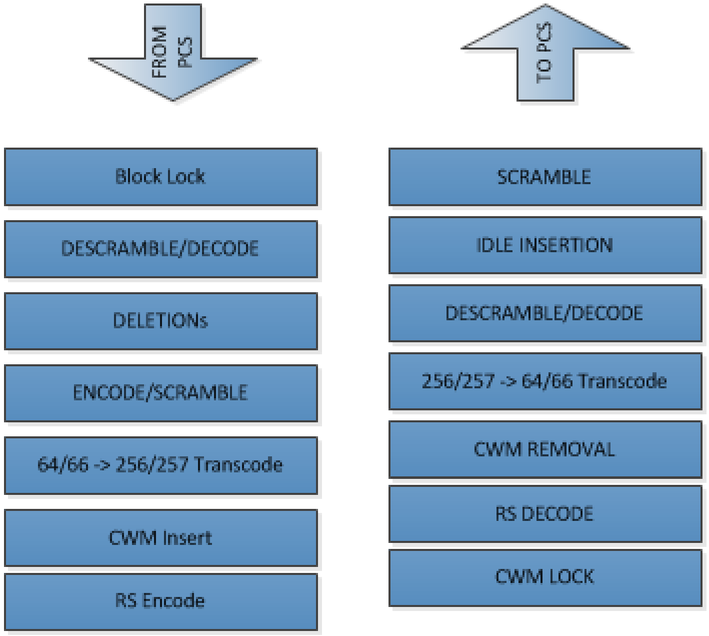
RS FEC CWM (cont.)

- CW MARKERS (CWM):
  - FORMAT
    - Almost identical to CL82 40G AMs
    - BIP is not used and forced to fixed (tbd) values.
  - INSERTION RATE
    - Every 16384***5***66 bit blocks= 1024 Code Words.
  - TX: DELETE EQUIVALENT IDLES
    - Re-use CL46/CL49 Clock Compensation Rules
    - Four (4) Bytes of IDLE deleted per frame. Requires eight (8) frames.
    - 1024 Code Words = 655360 Bytes =~ 400 frames (of about 1.5KBytes/frame).
  - RX: DELETE CWMs
    - Replace with IDLEs or Ordered Sets
  - MAINTAINs BASICS of:
    - CL91 FEC SYNCHRONIZATION STATE FSM (**Figure 91-8)**
    - CL91 FEC ALIGNMENT STATE FSM (**Figure 91-9)**

RS FEC CWM (cont.)

- TX: IDLE or ORDERED SET DELETION:
  - Perform CL49 equivalent Block Lock and Descramble/Decode.
  - Delete IDLEs or ORDERED SETs
    - Re-use CL46/CL49 Clock Compensation Rules
  - Perform CL49 equivalent Encode/Scramble
- TX: CWM INSERTION
  - After CL49 equivalent Scrambler (TX)
  - Before CL91 equivalent RS Encoder
- RX: CWM DELETION
- RX: IDLE or ORDERED SET INSERTION
  - Perform CL49 equivalent Descramble/Decode.
  - Insert IDLEs or ORDERED SETs
  - Perform CL49 equivalent Scramble

RS FEC FUNCTIONAL DIAGRAM



FROM PCS

Block Lock

DESCRAMBLE/DECODE

DELETIONs

ENCODE/SCRAMBLE

64/66 -> 256/257 Transcode

CWM Insert

RS Encode

TO PCS

SCRAMBLE

IDLE INSERTION

DESCRAMBLE/DECODE

256/257 -> 64/66 Transcode

CWM REMOVAL

RS DECODE

CWM LOCK

AFFECTED CLAUSES

- CL49' – Identical to CL49 but operates at 25G.
- CL91' – RS FEC for 25G
  - Single lane RS FEC based on CL91
  - Contains CWM related functions
- CL45’ – MDIO (addressable register set)
- CL46’ – XGMII Reconciliation Layer – Speed up only
- CL74’ – KR FEC – Speed up only

CL45 (MDIO) CHANGES

- **CL91'** Related Register Changes:
  - Maintain 100GBASE-R FEC Control/Status bits
    - Still per ‘PHY’
    - PCSLane alignment status register applicable, but for one lane only.
    - FECLane alignment register applicable, but for one lane only.
    - RS FEC lane mapping register is not applicable.
    - Maintain RS FEC symbol errors counter for lane 0 only.
    - RS FEC BIP Error Counters are not applicable.
    - RS FEC PCS Alignment status register is not applicable.
  - PCS/CWM register changes?
- **GENERAL** Changes:
  - Add 25G CR4/KR4 to Negotiated Port Type
  - Add 25G CR4/KR4 to Backplane Ethernet, BARE-R copper status register
  - OTHER?

CL46' (PCS)

- **CL46' is based on CL46.**
- **All CL46 functions are maintained.**
- **CL46' operates at 25G instead of 10G.**

CL49' (PCS)

- **CL49' is based on CL49.**
- **All CL49 functions are maintained.**
- **CL49' operates at 25G instead of 10G.**

CL74' (PCS)

- **CL74' is based on CL74.**
- **All CL74 functions are maintained.**
- **CL74' operates at 25G instead of 10G.**

CL91' (RS FEC)- OVERVIEW

- **CL91' is based on CL91 RS FEC.**
  - **Changes AND additions are required.**
  - **All 'changes' are relative to the CL91 specification.**
- **Add** CWM functions:
  - PCS descramble and decode
  - IDLE/Ordered Set deletion
  - PCS encode and scramble
  - CWM insertion
- **Maintain** the same register set as the 100G RS-FEC
- **Change** sections to show the following:
  - Remove **MLD** functions:
  - Change AM removal, mapping, and insertion for both RX and TX
    - Different number of blocks representing the CWM are inserted and deleted.
    - Single FEC lane (0) and no AM to FEC lane mapping.
      - For TX and for RX (removal)
  - Transcoding changes
    - Extending the control blocks to support CL49 as well as CL82 types

CL91' DETAILS – CWM FUNCTIONS

- **These changes are based on CL49 as the starting point.**

- **TX:**
  - Add CL49 Block Lock (**Figure 49-12**, **Lock State Diagram**)
  - Add CL49 equivalent Descramble (**Section 49.2.10 Descrambler)**
  - Add CL49 equivalent Decode (**Figure 49-15, Receive State Diagram)**
  - Add Function to Delete IDLEs or ORDERED SETs
  - Add CL49 equivalent Encode (**Figure 49-14, Transmit State Diagram)**
  - Add CL49 equivalent Scrambler (**Section 49.2.6, Scrambler)**
  - Add CWM Insertion Function
- **RX:**
  - Add CWM Deletion Function
  - Add CL49 equivalent Descramble (**Section 49.2.10 Descrambler)**
  - Add CL49 equivalent Decode (**Figure 49-15, Receive State Diagram)**
  - Add IDLE/Ordered Set Insertion Function
  - Add CL49 equivalent Scrambler (**Section 49.2.6, Scrambler)**

CL91' CHANGES - DETAILS

- **THE REMAINING CHANGES ARE BASED ON CL91 AS THE STARTING POINT.**

- **FEC Service Interface** (91.2)
  - Change text to indicate the interface operates over a single PCS stream running at 25.78125GBd.
- **Lane block synchronization** (91.5.2.1)
  - Change to Block Synchronization
  - Change text to indicate a single lane (FEC:IS_UNITDATA_i.request) provides a stream of data to this function. It obtains lock to the 66-bit blocks in the bit stream using the sync headers, and outputs 66-bit blocks. Block lock is obtained as specified in the block lock state diagram in Figure 82-10.
- **Alignment lock and deskew** (91.5.2.2)
  - Change to Codeword Marker Lock
  - Change text to indicate Codework Marker lock is obtained by using the alignment marker lock state diagram in Figure 82-11, with the following changes:
    - X has a value of 0 only, representing the 40G MLD AM0 encoding (only lane 0 is considered)
    - Change am_counter variable in 82.2.18.2.4 Counters to cwm_counter, and to indicate the CWMs are separated by 16383*5 66 bit blocks
- **Lane Reorder** (91.5.2.3)
  - Remove this section.

CL91 CHANGES – MORE DETAILS

- **Alignment marker removal** (91.5.2.4)
  - Change to Codeword Marker removal
  - Change text to indicate that once Codeword Marker lock is obtained (as indicated by am_lock), the Codework Marker is removed from the data stream.
- **Figure** 91-2 (Functional block diagram):
  - Change TX interface to include FEC:IS_UNIDATA_0.request only
  - Change “Lane block synchronization” to “Block Synchronization”
  - Change TX direction “Alignment lock and deskew” to “CWM lock”
  - Remove TX direction “Lane reorder”
  - Remove “Symbol distribution”
  - Change RX direction “Alignment lock and deskew” to “CWM lock”
  - Remove RX direction “Lane reorder”
  - Remove “Block distribution”

CL91 CHANGES - MORE DETAILS

- **Alignment marker mapping and insertion** (91.5.2.4)
  - Change to CWM mapping and insertion
  - Change text to the following:

    The codeword markers that were removed per 91.5.2.4 are re-inserted after being processed by the codeword marker mapping function.

    The RS-FEC receive function uses knowledge of this mapping to identify RS-FEC codeword boundaries.

    The codeword marker mapping function operates on a group of four codeword markers. Let cwm_tx_x<65:0> be the codeword marker "x", x=0 to 3, where bit 0 is the first bit transmitted. The codeword markers shall be mapped to cwm_txmapped<256:0> in a manner that yields the same result as the process defined below.

    a) cwm_txmapped<64x+63:64x> = cwm_tx_x<65:2> for x = 0 to 3

    b) cwmp_txmapped<256> = 1

    One group of codeword markers are mapped every 5 x 16384 66-bit blocks. This corresponds to 1024 Reed-Solomon codewords. The mapped codeword markers, cwm_txmapped<256:0> shall be inserted as the first 257 message bits to be transmitted from every 1024th codeword.

CL91 CHANGES - MORE DETAILS

- **Reed Solomon Encoder** (91.5.2.7)
  - Add a paragraph at the end:
  - The output of the encoder is connected to the PMA:_IS_UNITDATA_0.request input one 10-bit symbol at a time, in a concatenated order.
- **Alignment lock and deskew** (91.5.3.1)
  - Change to Codeword Marker lock
  - Change text to indicate the RS-FEC receive function forms a bit stream from the PMA:IS_UNITDATA_0.indication primitive. It obtains codeword marker logic and performs codeword validity checks as indicated by the FEC synchronization state diagram shown in Figure 91-8 and the FEC alignment state diagram show in Figure 91-9
- **FEC synchronization state diagram** (Figure 91-8)
  - Remove FEC_lane_mapping and fec_lane variables.
  - Redefine amp_valid and rename to cwm_valid: Boolean variable that is set to true if the received 64-bit block is a valid cw marker payload. The cw marker payload consists of 48 known bits and 16 variable bits (the BIP3 and CD3 field and it's compliment BIP7 and CD7, see 82.2.7). The bits of the candidate block that are in the positions of the known bits in the cw marker payload are compared on a nibble-wise basis (12 comparisons). If no more than 3 nibbles in the candidate block fail to match the corresponding known nibbles in the cw marker payload, the candidate block is considered a valid cw marker payload. For the normal mode of operation, the lane compares the candidate block to the cw marker payload for PCS lane 0, from the 40G MLD 4 specification (See Table 82-3).
  - Change all_locked to indicate x=0 only.
  - Change amp_counter to cwmp_counter, and to count 1024 FEC codewords that separate the ends of two consecutive, normal codeword marker payload sequences.

CL91 CHANGES - MORE DETAILS

- **FEC alignment state diagram** (Figure 91-9)
  - Remove DESKEW, DESKEW_FAIL, and ALIGN_ACQUIRED states.
  - Replace test_cw from ALIGN_ACQUIRED with all_locked
- **Lane Reorder** (91.5.3.2)
  - Remove this section
- **FIGURE 91-6** (Transmit bit ordering)
  - Remove symbol distribution
  - Remove PMA_UNIDATA_[{1,2,3}].request interfaces
  - Distribute all symbols in concatenated order to PMA_UNIDATA_0.request
- **Alignment marker removal** (91.5.3.4)
  - Change text to indicate the first 257 bits in every 1024th codeword is the vector cwm_rxmapped<256:0>.
- **256B/257B to 64B/66B transcoder** (91.5.3.5)
  - Change section **f2** to refer to block types using Figure 49-7
- **Block distribution** (91.5.3.6)
  - Remove this section

CL91 CHANGES - MORE DETAILS

- **Alignment marker mapping and insertion** (91.5.3.7)
  - Change to CWM mapping and insertion
  - Change text to the following:

    The codeword marker mapping function derives the codeword markers, am_rx_x<65:0> for x = 0 to 3, from am_rxmapped<256:0>.

    The codeword markers shall be derived from am_rxmapped<256:0> in a manner that yields the same result as the following process.

    For x=0 to 3, am_rx_x<65:0> is constructed as follows:

    am_rx_x<0>=1 and am_rx_x<1>=0.

    am_rx_x<25:2> is set to M_0, M_1, M_2 as shown in Figure 82-9 using the values in Table 82-3 for PCS lane number x.

    am_rx_x<33:26> = am_rxmapped<64x+31:64x+24>

    am_rx_x<57:34> is set to M_4, M_5, M_6 as shown in Figure 82-9 using the values in Table 82-3 for PCS lane number x.

    am_rx_x<65:58> = am_rxmapped<64x+63:64x+56>
- **Receive bit ordering** (Figure 91-7)
  - Remove PMA_UNITDATA_[{1,2,3}].indication interfaces.
  - Show that all symbols are received via the PMA_UNIDATA_0.indicate interface.
  - Change “Alignment lock, deskew, and lane reorder” to “CWM lock”
  - Change arrow at top from “To Block distribution” to “To Alignment Inerstion”

THANK YOU!