TX EQUALIZATION FOR C2C LINKS

Adee Ran, Intel

June 2020

Supporters

- Rick Rabinovich, Keysight Technologies

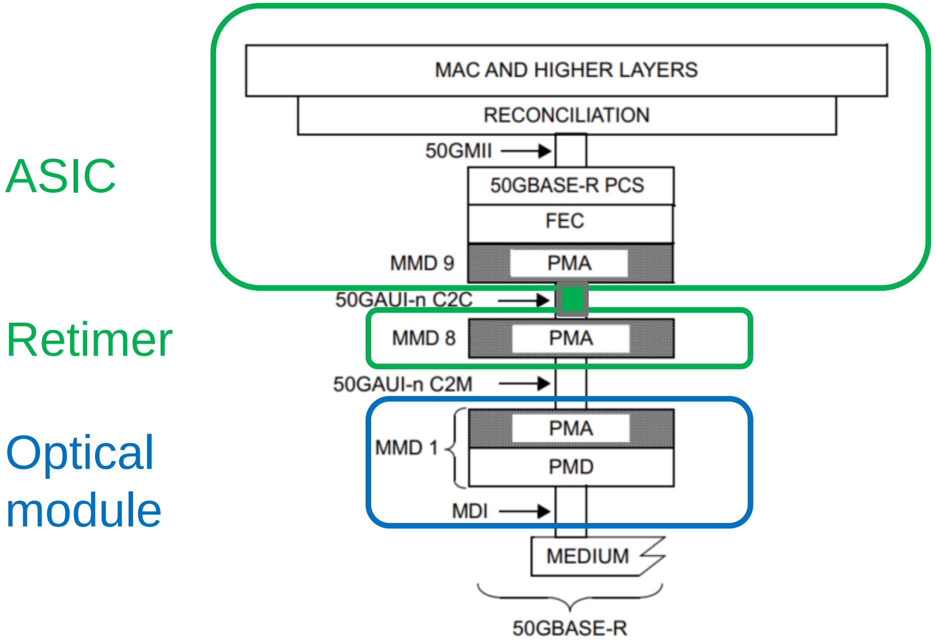
- (your name could be here)

Agenda

- Look at possible C2C usage models and applications
- Explore methods of Tx equalization control
- A possible solution?

- Note: This presentation is focused on C2C transmitter control. C2M is a related but quite different discussion, and is out of scope of this presentation.

C2C usage models (from 802.3cd)

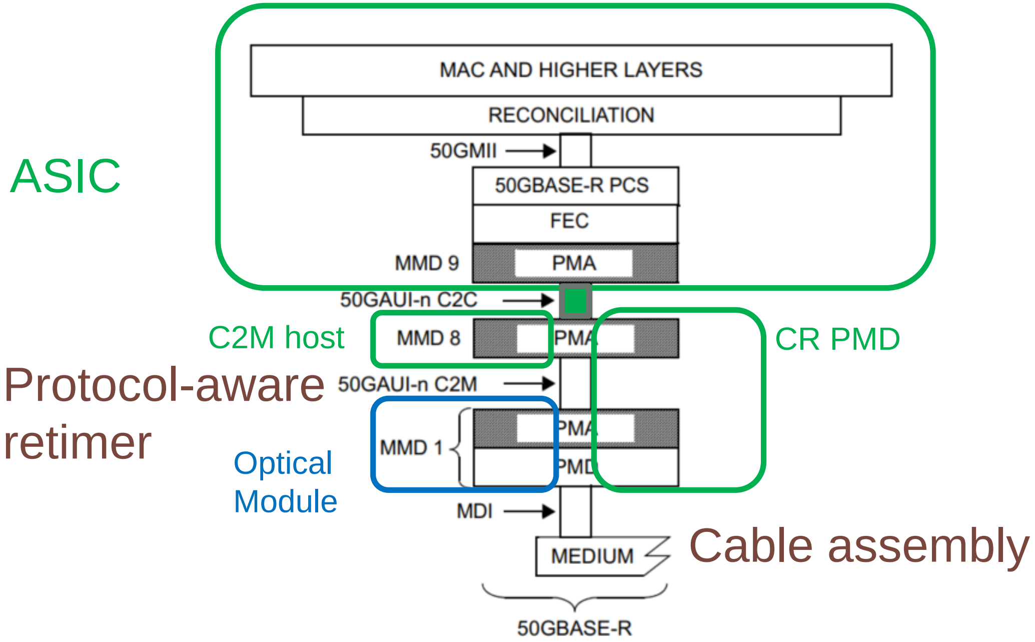


50GAUI-n = 50 Gb/s ATTACHMENT UNIT INTERFACE
50GMII = 50 Gb/s MEDIA INDEPENDENT INTERFACE
FEC = FORWARD ERROR CORRECTION
MAC = MEDIA ACCESS CONTROL

MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT

Figure 135A–4—Intermediate PMA device for module interface, FEC implemented with PCS

Retimer for extending host channel, supports only C2M
Minimum power and complexity (but PAM4, likely multi-rate)



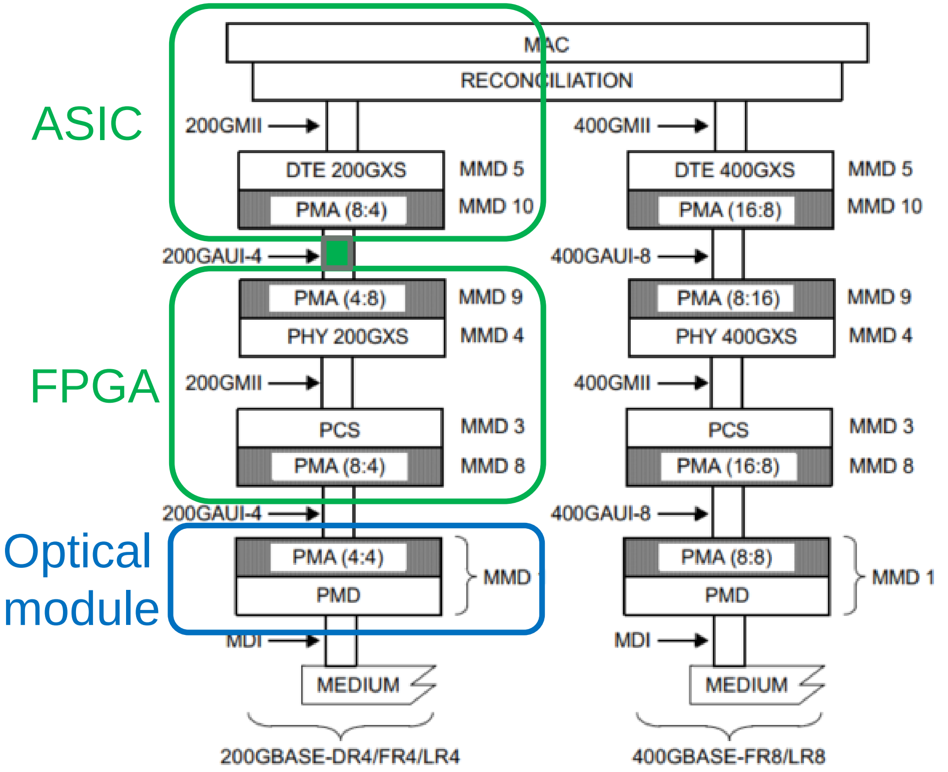
50GAUI-n = 50 Gb/s ATTACHMENT UNIT INTERFACE
50GMII = 50 Gb/s MEDIA INDEPENDENT INTERFACE
FEC = FORWARD ERROR CORRECTION
MAC = MEDIA ACCESS CONTROL

MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT

Figure 135A–4—Intermediate PMA device for module interface, FEC implemented with PCS

Protocol-aware retimer for extending host channel supporting both optics and cable
Needs full CR PMD capability – training, strong Rx

C2C usage models (from 802.3bs)



200GAUI = 200 Gb/s ATTACHMENT UNIT INTERFACE
200GMII = 200 Gb/s MEDIA INDEPENDENT INTERFACE
200GXS = 200 Gb/s EXTENDER SUBLAYER
400GAUI = 400 Gb/s ATTACHMENT UNIT INTERFACE
400GMII = 400 Gb/s MEDIA INDEPENDENT INTERFACE
400GXS = 400 Gb/s EXTENDER SUBLAYER
DTE = DATA TERMINAL EQUIPMENT

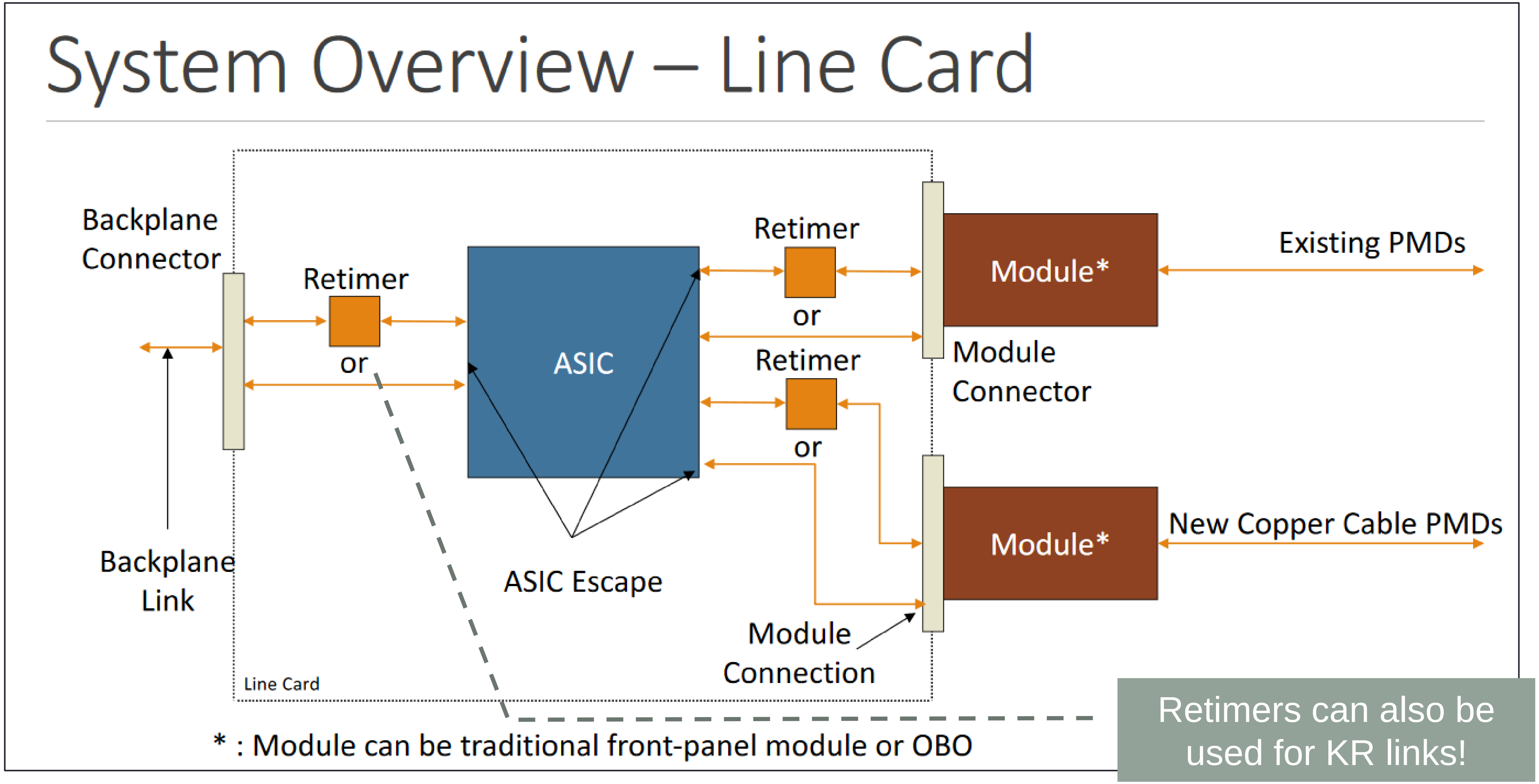
MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
MMD = MDIO MANAGEABLE DEVICE
PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT

Figure 120A–7—Example 200GBASE-DR4/FR4/LR4 and 400GBASE-FR8/LR8 PMA layering with 200GXS, 400GXS, and two 200GAUI-4, 400GAUI-8 interfaces

"Bump-in-the-wire" device
Likely also capable of being CR/KR PMD
Logic area is not a problem

Source: ofelt_100GEL_01_0118

Practical application example



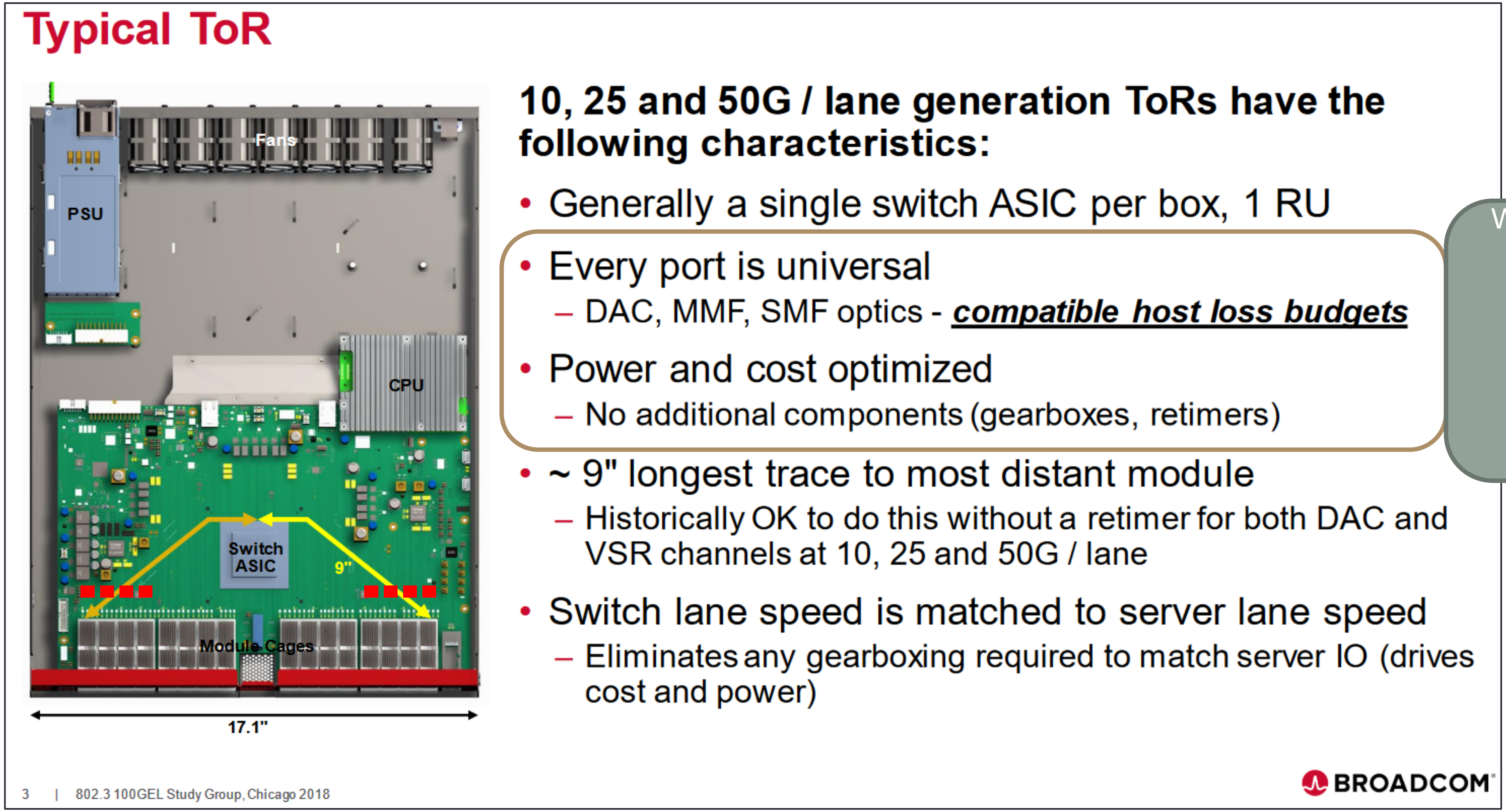
System Overview – Line Card

* : Module can be traditional front-panel module or OBO

Retimers can also be used for KR links!

Source: stone_100GEL_01_0318

Universal ports?


Fans
PSU
CPU
Switch ASIC
9"
Module Cages
17.1"

Typical ToR

10, 25 and 50G / lane generation ToRs have the following characteristics:

- Generally a single switch ASIC per box, 1 RU
- Every port is universal
  - DAC, MMF, SMF optics - ***compatible host loss budgets***
- Power and cost optimized
  - No additional components (gearboxes, retimers)
- ~ 9" longest trace to most distant module
  - Historically OK to do this without a retimer for both DAC and VSR channels at 10, 25 and 50G / lane
- Switch lane speed is matched to server lane speed
  - Eliminates any gearboxing required to match server IO (drives cost and power)

3 | 802.3 100GEL Study Group, Chicago 2018

BROADCOM

We now know that CR can't work with the same loss as C2M… Universal port may require a protocol-aware retimer (at least at some ports)

Marked by 🟥(Note: variable distance from the ASIC)

Other applications?

- There are many more use cases for “Ethernet” ports than we consider here.
- Some devices are not fully compliant…
  - For example, may not include training functionality, even when used as a port that has mandatory requirements.
  - Many applications have engineered links and don’t need plug-and-play.
  - Some applications use a custom preset setting, and then run “training” without allowing any equalization change.
- Some devices have link training capability built in!
  - And possibly AN too
  - State machine implementation is not an issue.
- We should acknowledge and provision for many possible use cases.

Summary of likely C2C use cases

Types of “downstream” devices

- Optical-only minimal retimer
  - Training logic increases complexity/cost, not desired
- Universal-port, Protocol-aware retimer
  - Training logic exist on MDI side anyway (for CR)
  - Having another copy has insignificant complexity/cost impact
- Bump-in-the-wire
  - Large device, more package loss
  - Additional logic is not a problem
- Custom applications

What about the “big ASIC” port?

- If it supports universal ports with no-retimer, then it may have CR training anyway.

There are also custom applications which may or may not use training, or have unexpected behavior.

Does equalization need to be configurable at all?

- It was configurable at previous speeds 50G, 25G
  - Life isn’t getting simpler
- Different lanes from ASIC to retimers may be quite different channels
- We are assuming it in the COM reference receiver for C2C, and have corresponding electrical specs.

Also…

- Even if optimal Tx equalization is known, the “training” phase is sometimes desired for bringing up the link.

But…

- We should acknowledge that in some cases training logic will not be available.

Possible way forward

- Specify configurable Tx equalization as mandatory
  - Define MDIO or equivalent control registers
  - (more details later)
- Specify the “training protocol” PMD control function (CR/KR PMDs) as optional
  - Use the same MDIO/equivalent control and status register definitions
- Management:
  - If the optional PMD training protocol is implemented on both sides of the C2C – may use it and/or MDIO control/status registers on both devices
  - If not implemented on either side – may only use MDIO registers

Register interface?

- Reuse the training registers (Table 162–7) for simplicity
  - But they provide only relative changes and presets…
  - What if I want to program a known value?
- Add another R/W register for direct access to c(coef_sel)
  - On read, returns a value that corresponds to the current setting of c(coef_sel)
    - Encoding is implementation dependent; 16 bits per coefficient are sufficient
    - Default values for c(coef_sel) correspond to the OUT_OF_SYNC settings listed in Table 136–12
    - The setting in OUT_OF_SYNC state is taken from the programmed registers
  - On write, configures c(coef_sel) to the appropriate setting as though it was reached using relative changes
    - Enables programming all coefficients to known values quickly
    - Writing order requirements (for multiple coef_sel) are implementation dependent

Possible use cases

- Pre-programming
  - Optimize BER in “factory tuning” by using relative changes to the registers
  - Record the resulting values on both sides in NVM
  - After deployment, at system startup, values are read from NVM and re-programmed
  - Can work without the training protocol!
- Training for link startup with known coefficients, when both devices support the training protocol
  - Optimize and store values for both devices as above
  - Program values at startup – override the OUT_OF_SYNC values
  - Enable (restart) training on both sides
  - Both sides start at the programmed values and go through the state diagram flow
    - Management intervention is not required for this state diagram
    - Coefficient change requests may be disabled if so desired
- Blind training, when both devices support the training protocol
  - Start from a standard preset and perform training as in 136.8.11
- Other use cases?

Summary

- Configurable equalization is assumed necessary for C2C devices
- Training protocol or register programming are both possible for a C2C link
- Training protocol implementation may be available or not depending on device
- Using either training protocol or direct register programming may be preferable for a given system.

- In this presentation we showed guidelines for a solution that enable both ways, and supports multiple use cases.

THANK YOU

Questions/feedback?