802.3bj Scrambling Options

IEEE P802.3bj
July 2012 San Diego

Roy Cideciyan – IBM

Mark Gustlin – Xilinx

Jeff Slavick – Avago

Contributors and supporters

Pete Anslow – Ciena

Andre Szczepanek – Inphi

Stephen Bates – PMC-Sierra

Pravin Patel – IBM

Gary Nicholl – Cisco

Ran Adee – Intel

Kent Lusted – Intel

John Ewen - IBM

Scrambling Summary

- Draft 1.0 transcoding requires that you unscramble the data before transcoding, then you rescramble the data after transcoding (all with a self sync scrambler)
- Roy presented (in cideciyan_02_0512) a methodology that allows you to not descramble the data before transcoding, he proposed XORing the control bits added by transcoding with the payload data and implementing a synchronous scrambler of the FEC frame (PN-5280)
- His proposal runs a shadow scrambler to help recover the compressed data when transcoding back to 64B/66B
- A variation of Roy's proposal is to do everything that he says in the slides except to not use the synchronous scrambler or descrambler (PN-5280 for NRZ). This is called Roy's optimized proposal in the following slides.

Pros/Cons

Draft 1.0

- No known issues
- Complexity on par with Roy’s proposal (depends on collocation or not)
- Pete Anslow will show that the data has well behaved properties
- BIP must be recalculated on FEC RX side (no end to end BIP coverage)

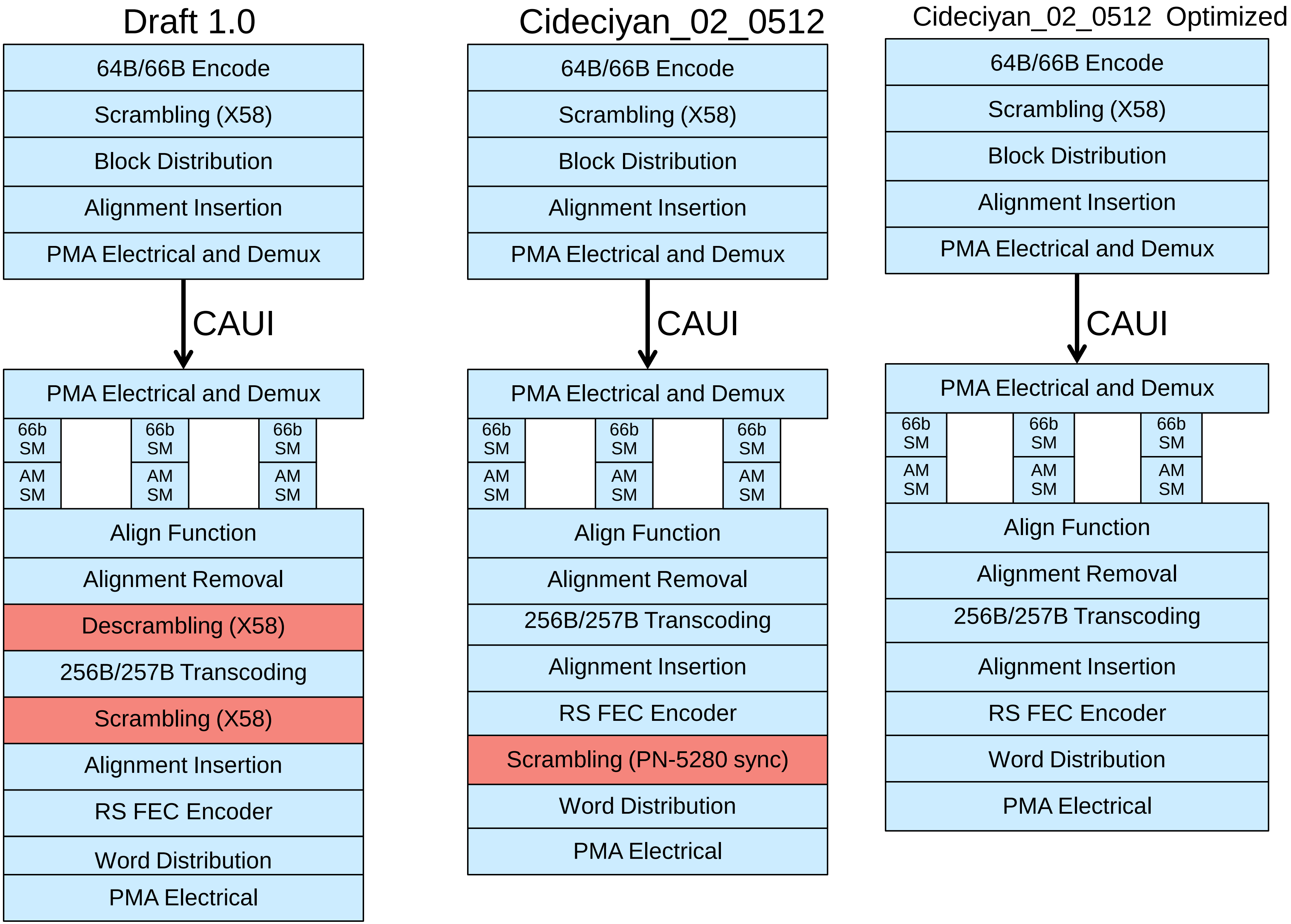
Roy’s proposal (cideciyan_02_0512)

- Avoids multiple self synchronous scramblers
- Complexity on par with Draft 1.0
- BIP can pass straight through (end to end BIP coverage)

Roy’s optimized proposal

- Avoids multiple self synchronous scramblers
- Least complicated implementation
- Pete Anslow will show that the data has well behaved properties
- BIP can pass straight through (end to end BIP coverage)

Side by side TX, Separated Protocol Stacks



Draft 1.0

64B/66B Encode

Scrambling (X58)

Block Distribution

Alignment Insertion

PMA Electrical and Demux

CAUI

PMA Electrical and Demux

66b SM | 66b SM | 66b SM

AM SM | AM SM | AM SM

Align Function

Alignment Removal

Descrambling (X58)

256B/257B Transcoding

Scrambling (X58)

Alignment Insertion

RS FEC Encoder

Word Distribution

PMA Electrical

Cideciyan_02_0512

64B/66B Encode

Scrambling (X58)

Block Distribution

Alignment Insertion

PMA Electrical and Demux

CAUI

PMA Electrical and Demux

66b SM | 66b SM | 66b SM

AM SM | AM SM | AM SM

Align Function

Alignment Removal

256B/257B Transcoding

Alignment Insertion

RS FEC Encoder

Scrambling (PN-5280 sync)

Word Distribution

PMA Electrical

Cideciyan_02_0512 Optimized

64B/66B Encode

Scrambling (X58)

Block Distribution

Alignment Insertion

PMA Electrical and Demux

CAUI

PMA Electrical and Demux

66b SM | 66b SM | 66b SM

AM SM | AM SM | AM SM

Align Function

Alignment Removal

256B/257B Transcoding

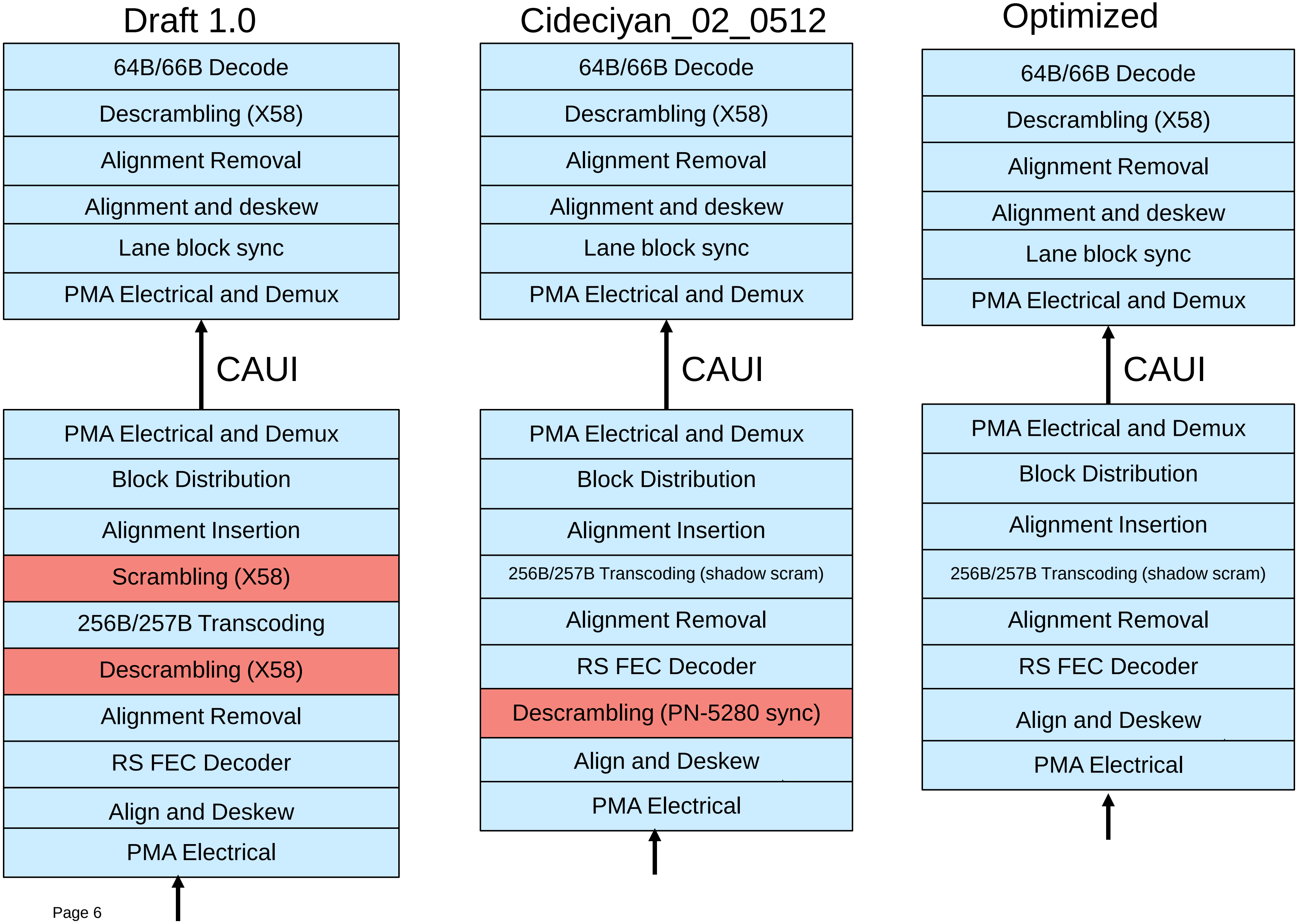
Alignment Insertion

RS FEC Encoder

Word Distribution

PMA Electrical

Side by side RX, Separated



Draft 1.0

- 64B/66B Decode
- Descrambling (X58)
- Alignment Removal
- Alignment and deskew
- Lane block sync
- PMA Electrical and Demux

CAUI

- PMA Electrical and Demux
- Block Distribution
- Alignment Insertion
- Scrambling (X58)
- 256B/257B Transcoding
- Descrambling (X58)
- Alignment Removal
- RS FEC Decoder
- Align and Deskew
- PMA Electrical

Cideciyan_02_0512

- 64B/66B Decode
- Descrambling (X58)
- Alignment Removal
- Alignment and deskew
- Lane block sync
- PMA Electrical and Demux

CAUI

- PMA Electrical and Demux
- Block Distribution
- Alignment Insertion
- 256B/257B Transcoding (shadow scram)
- Alignment Removal
- RS FEC Decoder
- Descrambling (PN-5280 sync)
- Align and Deskew
- PMA Electrical

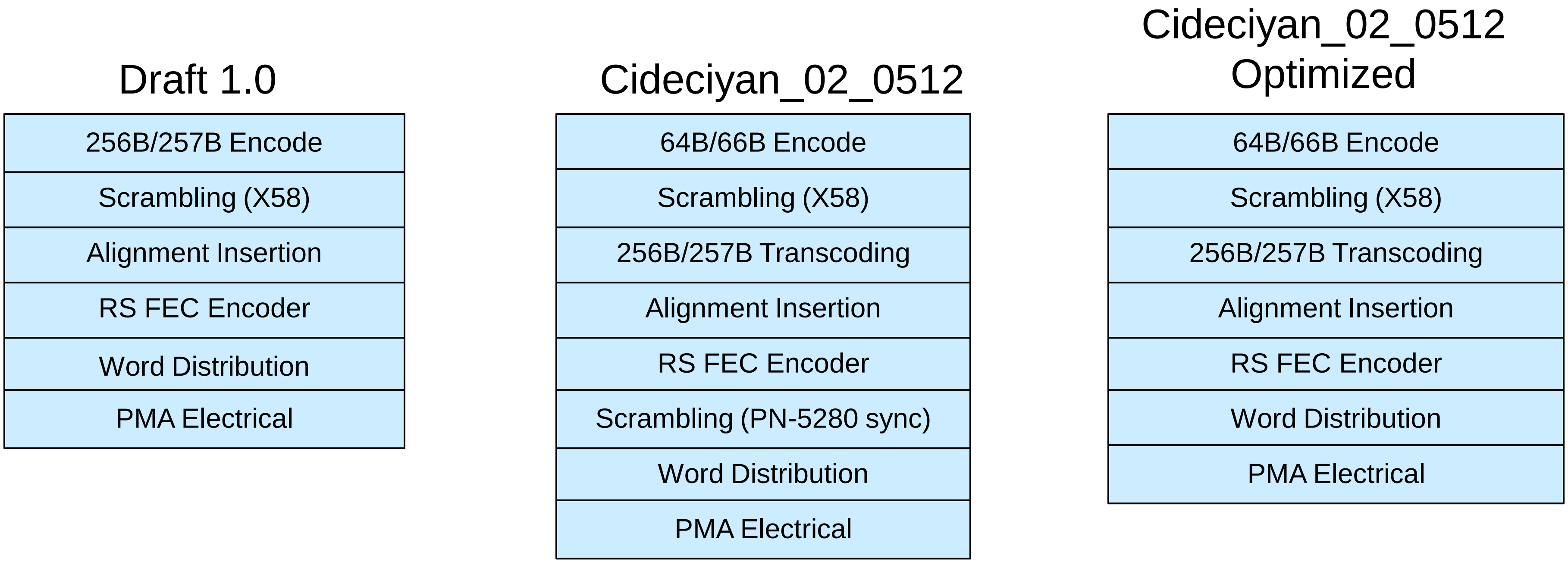
Cideciyan_02_0512 Optimized

- 64B/66B Decode
- Descrambling (X58)
- Alignment Removal
- Alignment and deskew
- Lane block sync
- PMA Electrical and Demux

CAUI

- PMA Electrical and Demux
- Block Distribution
- Alignment Insertion
- 256B/257B Transcoding (shadow scram)
- Alignment Removal
- RS FEC Decoder
- Align and Deskew
- PMA Electrical

Side by side TX, Collocated and Optimized



Draft 1.0

- 256B/257B Encode
- Scrambling (X58)
- Alignment Insertion
- RS FEC Encoder
- Word Distribution
- PMA Electrical

Cideciyan_02_0512

- 64B/66B Encode
- Scrambling (X58)
- 256B/257B Transcoding
- Alignment Insertion
- RS FEC Encoder
- Scrambling (PN-5280 sync)
- Word Distribution
- PMA Electrical

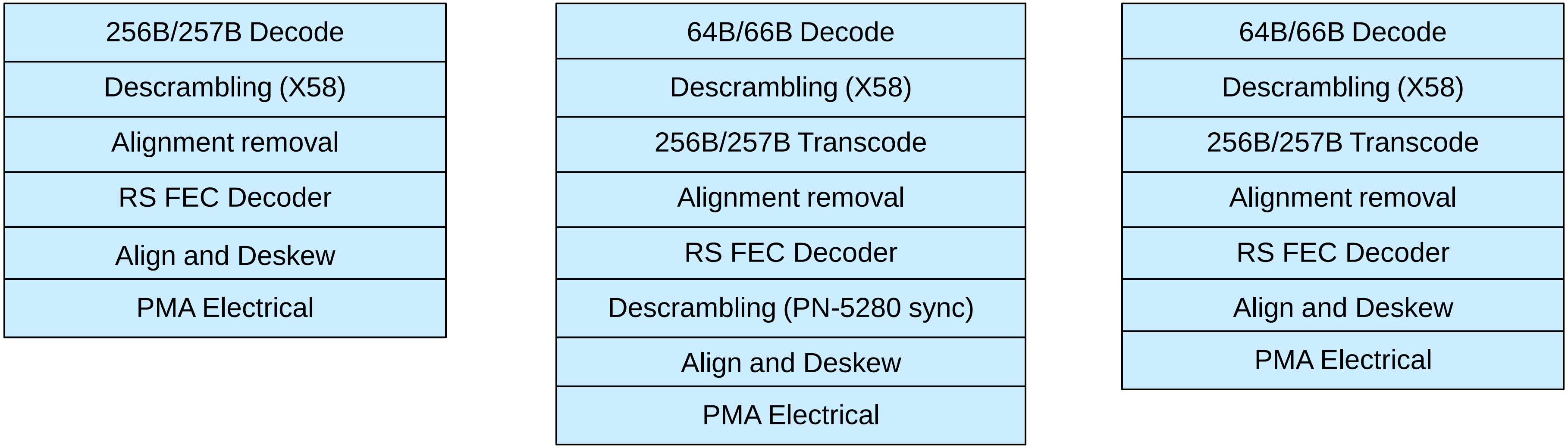
Cideciyan_02_0512
Optimized

- 64B/66B Encode
- Scrambling (X58)
- 256B/257B Transcoding
- Alignment Insertion
- RS FEC Encoder
- Word Distribution
- PMA Electrical

The collocated and optimized stack is only for a device that only sends 100GBASE-KR4/CR4/KP4 FEC encoded data!

Note the 64B/66B protocol stack and scrambler maybe shared logic in devices which support multiple PMD types, i.e.. in a device which supports both 100GBASE-LR4 and 100GBASE-KR4/CR4 PMDs

Side by side RX, Collocated and Optimized



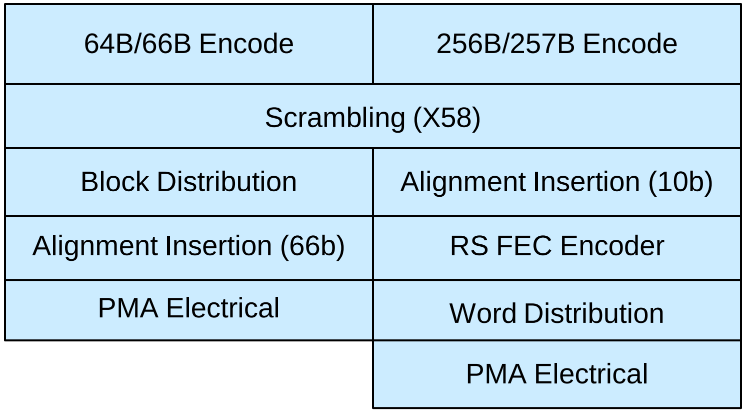
Draft 1.0	Cideciyan_02_0512	Cideciyan_02_0512 Optimized
256B/257B Decode	64B/66B Decode	64B/66B Decode
Descrambling (X58)	Descrambling (X58)	Descrambling (X58)
Alignment removal	256B/257B Transcode	256B/257B Transcode
RS FEC Decoder	Alignment removal	Alignment removal
Align and Deskew	RS FEC Decoder	RS FEC Decoder
PMA Electrical	Descrambling (PN-5280 sync)	Align and Deskew
	Align and Deskew	PMA Electrical
	PMA Electrical	

The collocated and optimized stack is only for a device that only sends 100GBASE-KR4/CR4/KP4 FEC encoded data!

Note that the 64B/66B protocol stack and scrambler can be shared logic in devices which support multiple PMD types, i.e. in a device which supports both 100GBASE-LR4 and 100GBASE-KR4/CR4 PMDs

Side by side TX, Collocated and Optimized

Draft 1.0



64B/66B Encode

256B/257B Encode

Scrambling (X58)

Block Distribution

Alignment Insertion (10b)

Alignment Insertion (66b)

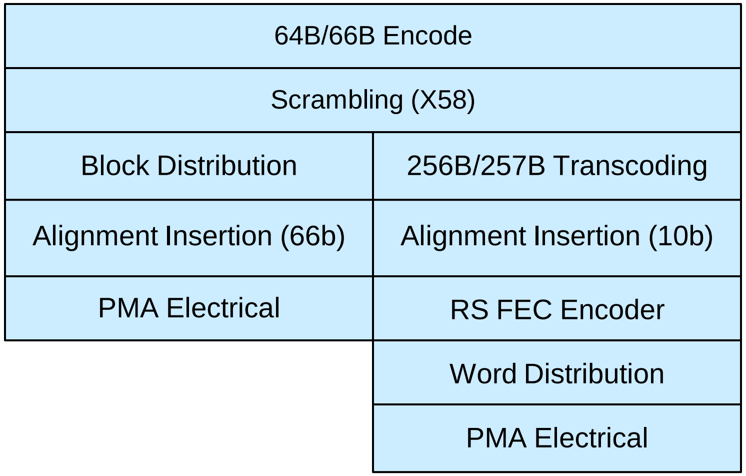
RS FEC Encoder

PMA Electrical

Word Distribution

PMA Electrical

Cideciyan_02_0512 Optimized



64B/66B Encode

Scrambling (X58)

Block Distribution

256B/257B Transcoding

Alignment Insertion (66b)

Alignment Insertion (10b)

PMA Electrical

RS FEC Encoder

Word Distribution

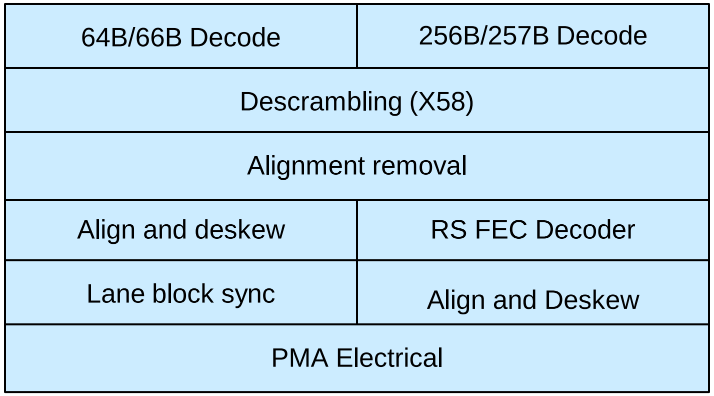
PMA Electrical

The above shows a TX protocol stack for a PHY that supports multiple PMDs (64b/66b and FEC PMDs)

- When collocated and optimized the complexity is similar for both

Side by side RX, Collocated and Optimized

Draft 1.0



64B/66B Decode | 256B/257B Decode

Descrambling (X58)

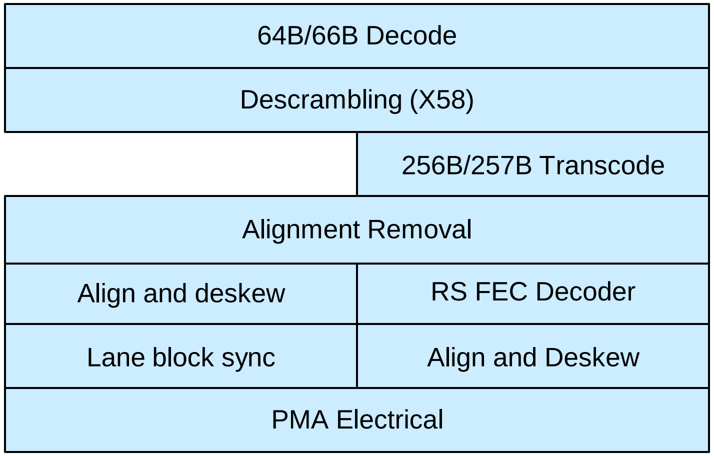
Alignment removal

Align and deskew | RS FEC Decoder

Lane block sync | Align and Deskew

PMA Electrical

Cideciyan_02_0512 Optimized



64B/66B Decode

Descrambling (X58)

256B/257B Transcode

Alignment Removal

Align and deskew | RS FEC Decoder

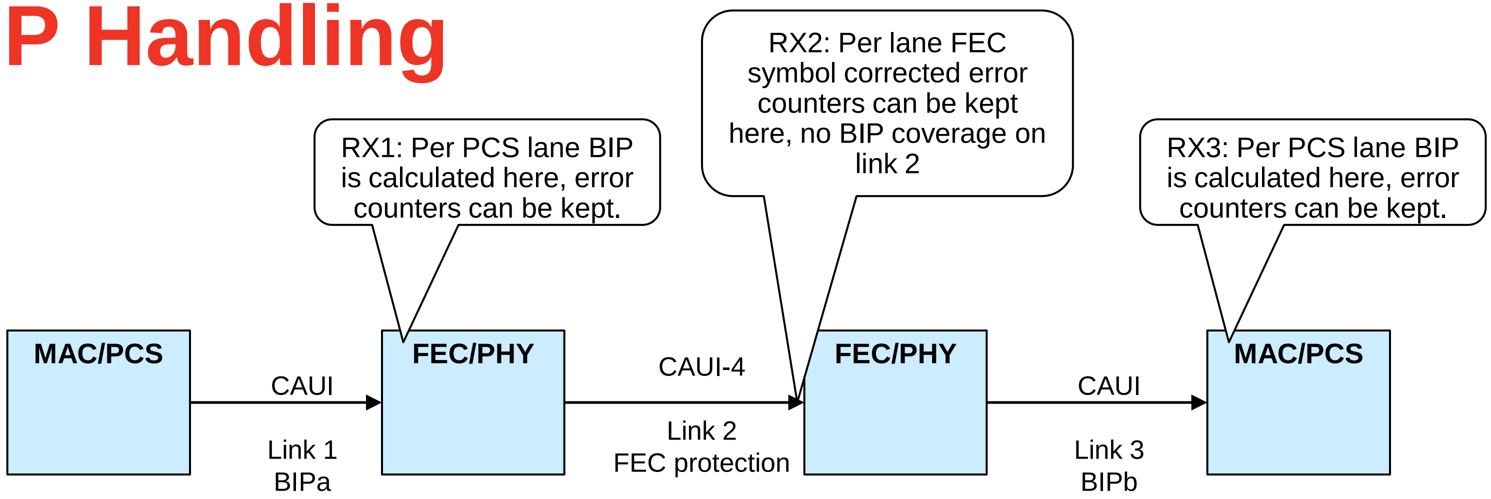
Lane block sync | Align and Deskew

PMA Electrical

The above shows a RX protocol stack for a PHY that supports multiple PMDs (64b/66b and FEC PMDs)

When collocated and optimized the complexity is similar for both

BIP Handling


RX2: Per lane FEC symbol corrected error counters can be kept here, no BIP coverage on link 2
RX1: Per PCS lane BIP is calculated here, error counters can be kept.
RX3: Per PCS lane BIP is calculated here, error counters can be kept.
MAC/PCS
CAUI
Link 1
BIPa
FEC/PHY
CAUI-4
Link 2
FEC protection
FEC/PHY
CAUI
Link 3
BIPb
MAC/PCS

- BIP values are calculated after scrambling and then inserted into the Alignment Markers as part of the 802.3ba PCS
- For Draft 1.0 when performing transcoding, we must check the BIP values before we descramble
- Draft 1.0 leaves the BIPa values as is for filler when being carried across the FEC link
- BIPb is regenerated on the far end when the 100GBASE-R PCS is recreated for Draft 1.0
- Errors can be isolated, but you do not have visibility to end to end errors at the endpoint
- If we were to adopt Cideciyan_02_0512 Optimized proposal, since there is no descrambling and re-scrambling, then the BIP can be carried transparently through the path
  - Errors can still be isolated if the BIP is checked at the FEC ingress point

Proposal

- **Change Draft 1.0 as per cideciyan_02_0512 pages 14-17, but remove the synchronous scrambler from the protocol flow.**
  - Remove boxes and references to Scramble and Descramble from figures 91-2, 91-4 and 91-5.
  - Remove sub-clauses 91.4.2.5 , 91.4.2.7, 91.4.3.5 and 91.4.3.7 which talk about scrambling and descrambling
  - Add into sub-clause 91.4.2.6 the details of the bit XORing
  - Add into sub-clause 91.4.3.6 the details of the bit XORing and shadow scrambler
  - In sub-clause 91.4.3.9 state that the BIP fields are carried transparently, not recalculated
  - In 91.4.3.3 change the method used to indicate an uncorrectable frame error to have the reverse transcoder corrupt sync headers in the FEC block. Corrupt the sync headers of 66b blocks 1, 9, 17, …, 73 and the final block 80 (11 blocks in total) by setting them to ‘11’ . This ensures all possible 64B or larger packets will be dropped.

Thanks!