Symbol-muxing PMA architecture proposal

Adee Ran, Cisco

Xiang He, Huawei

Support

- Phil Sun, Credo
- Dave Cassan, Alphawave
- Henry Wong, Alphawave
- Zvi Rechtman, NVIDIA

Outline

- Motivation
- Goals
- Previous work
- Proposed PMA architecture
- Backup material
  - FEC performance comparisons
  - Detailed diagrams and descriptions
  - Other methods considered
  - PMA partitioning examples
  - Proposed new PMA clause structure

Motivation

- In P802.3dj we are defining new PMDs and AUIs for old PCSs
  - The result is an **8:1** PMA muxing ratio (e.g., 32:4)
- We also assume complex receivers that can generate strongly correlated errors
- The bit-muxing PMA which we have been using for multiple generations of Ethernet **is breaking down under these conditions**
  - In the worst case (200G/400G), with precoding, the SNR penalty is **1.1 dB** and the BER penalty is x5 (a reduction to 20% is required compared to what we assumed) (see backup)
  - Without precoding, the penalty goes through the roof
  - This would narrow down our channel range and/or increase SerDes complexity
- **This project is challenging in many ways – we must improve our tools!**

Goals

- A PMA architecture for all 200 Gb/s per lane* interfaces
- **Compatible with existing PCSs**: 200GBASE-R, 400GBASE-R, and 800GBASE-R (1, 2, and 4 lanes, respectively)
  - Evolving 1.6TBASE-R PCS also in mind, but not addressed in this presentation
- Enabling **strong FEC performance with significantly correlated errors**
- Supporting **rate conversion** between 200 Gb/s per lane (new devices) and 100 Gb/s per lane (existing devices)
- Enabling **efficient design**, including modules and retimers
  - As protocol agnostic as possible
  - Avoid full deskew and alignment of all PCSLs

** In this presentation, the unqualified term "lane" means PMA lane, as opposed to PCSL lane (PCSL)*

Previous work

- Bit-muxing PMAs have been used since 40G Ethernet (Clause 83), and recently in 200G and 400G (Clause 120) and 800G (Clause 173)
- ran_3df_01a_220518 highlighted issues with bit-muxing for 200 Gb/s per lane interfaces (8:1 bit muxing)
- wang_3df_01b_220928 studied 2:1 muxing for 2-way and 4-way codeword interleaving
- ran_3df_01a_2211 compared the effect of 8:1 bit-muxing and symbol-muxing on FEC performance with correlated errors
  - See backup section for summary of results
- ran_3df_02a_2211 presented a symbol-muxing PMA concept
- he_3df_01_2211 analyzed the effect of bit/symbol muxing with multi-part link and concatenated FEC

PMA architecture – scope

- In 200GbE, 400GbE, and 800GbE, the PCS distributes symbols* to PCSLs; each PCSL carries 25 Gb/s
- There are existing physical interfaces (AUIs) with up to 100 Gb/s per lane, bitwise muxed with a ratio of 4:1
  - For example, 32:8 PMA for 800GAUI-8
- The PMAs for new 200 Gb/s per lane AUIs should include:
  - PMAs with **8:1** ratio mux/demux (below the PCS/XS)
  - PMAs with **2:1** ratio mux/demux (conversion from 4:1 muxing to 8:1 muxing and vice versa, in modules and gearboxes)
  - PMAs with **1:1** ratio (retimers)

** In this presentation, the unqualified term "symbol" means RS symbol (10 bits), as opposed to PAM4 symbol*

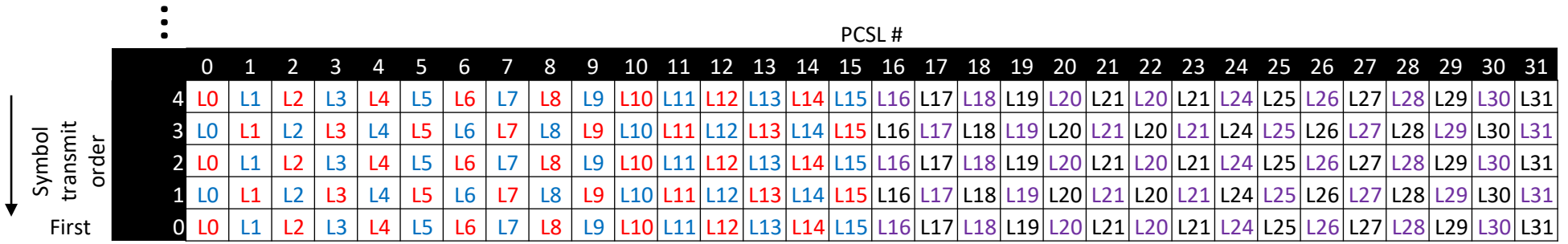
Bit or symbol muxing for 200 Gb/s signaling?

- **Bit muxing**
  - Used in existing 200G, 400G, and 800G PMAs (for 200G and 400G, 50 and 100 Gb/s per lane AUIs)
  - Simple and protocol agnostic
  - High muxing ratio degrades the FEC performance with correlated errors; with 8:1 the penalty is unacceptable (see ran_3df_01a_2211)
  - With highly correlated errors on high-loss AUI links, even 2:1 bit muxing causes a degradation (see wang_3df_01b_220928).
- **Symbol muxing**
  - Enables good FEC performance even with strong error correlation
  - To maximize FEC performance with bursts, we should **maximize the temporal separation between symbols of the same FEC codeword on each lane**
  - Round-robin distribution from FEC codeword to symbols on each lane is optimal
  - Compatibility with existing AUIs requires conversion from/to bit muxing
  - Requires knowledge of symbol boundaries – somewhat protocol aware
  - Current and future high-speed SerDes have large logic content (DSP, FEC) – symbol muxing logic is small in comparison

New PMA concept: symbol-pair muxing

- The PCS checkerboard pattern ensures each pair of successive symbols contains symbols from two codewords
  - In 800GBASE-R, taking one symbol-pair from each flow gives us symbols from all four codewords
  - In 200/400GBASE-R, taking one symbol-pair from each lane gives us symbols from both codewords
- In order to get a consistent order on each output lane, we need to convert the **checkerboard pattern** into a **striped pattern** (see next slide)
  - This can be done in a couple of ways
- From the striped pattern, symbols are muxed in pairs to an output lane
- In the receive direction, alignment on any AM gives us the symbol-pair boundary
- A more detailed description can be found in the backup section
  - This method was also referred to as “half-blind symbol muxing”

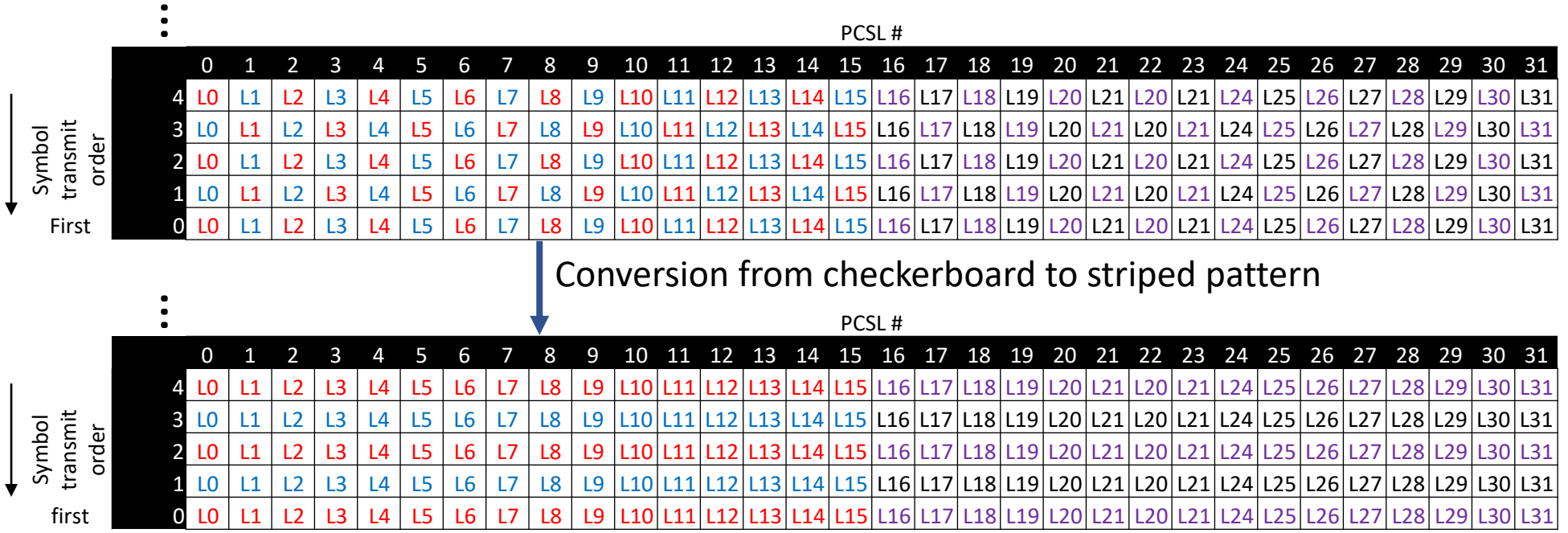
Symbol-pair muxing illustration (800GBASE-R)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
4	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L20	L21	L24	L25	L26	L27	L28	L29	L30	L31
3	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L20	L21	L24	L25	L26	L27	L28	L29	L30	L31
2	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L20	L21	L24	L25	L26	L27	L28	L29	L30	L31
1	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L20	L21	L24	L25	L26	L27	L28	L29	L30	L31
0	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20	L21	L20	L21	L24	L25	L26	L27	L28	L29	L30	L31

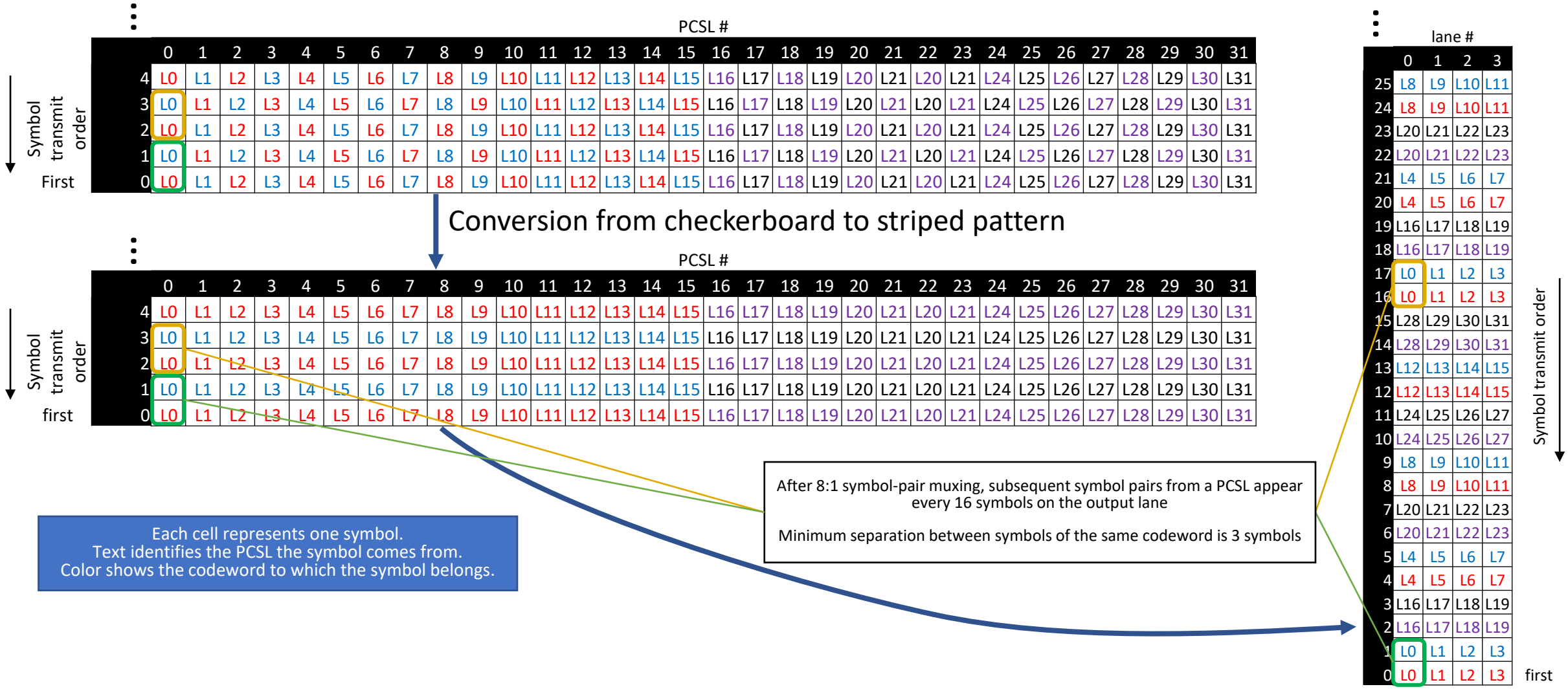
Each cell represents one symbol.
Text identifies the PCSL the symbol comes from.
Color shows the codeword to which the symbol belongs.

Symbol-pair muxing illustration (800GBASE-R)



Each cell represents one symbol.
Text identifies the PCSL the symbol comes from.
Color shows the codeword to which the symbol belongs.

Symbol-pair muxing illustration (800GBASE-R)



Functional block diagram: **PMA(32:4)** (800GBASE-R)

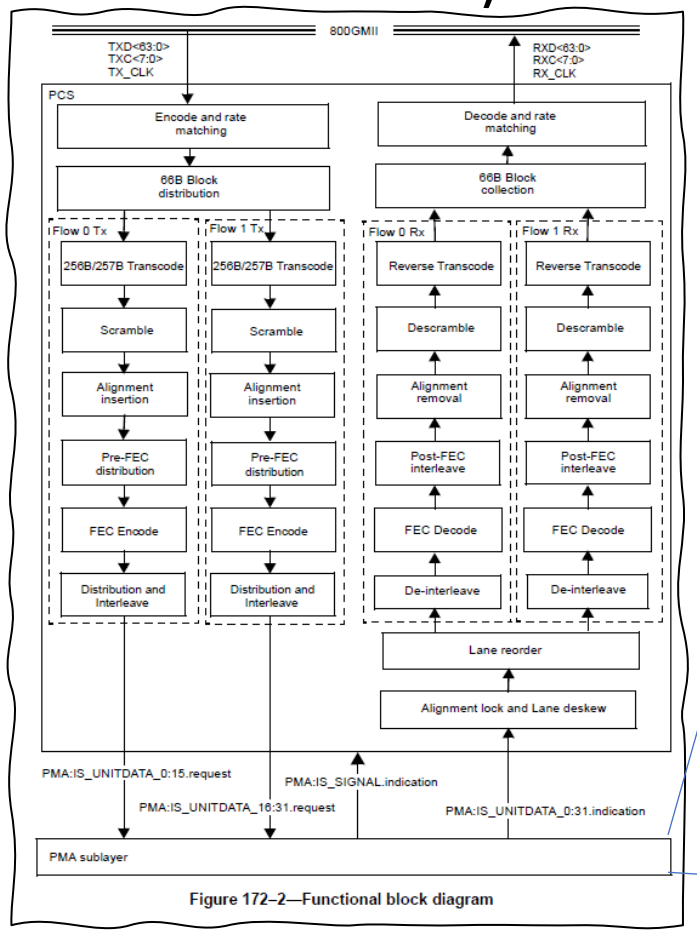
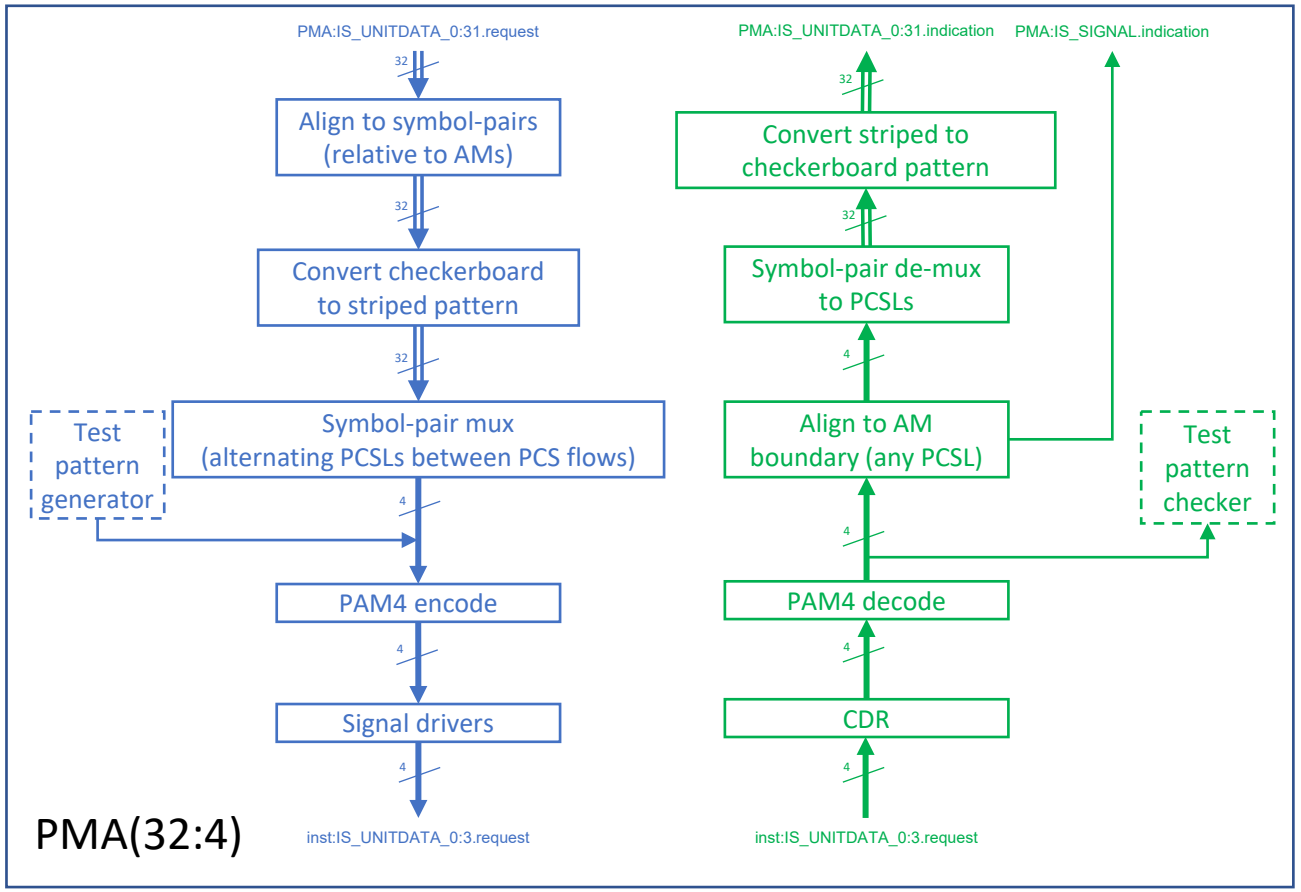
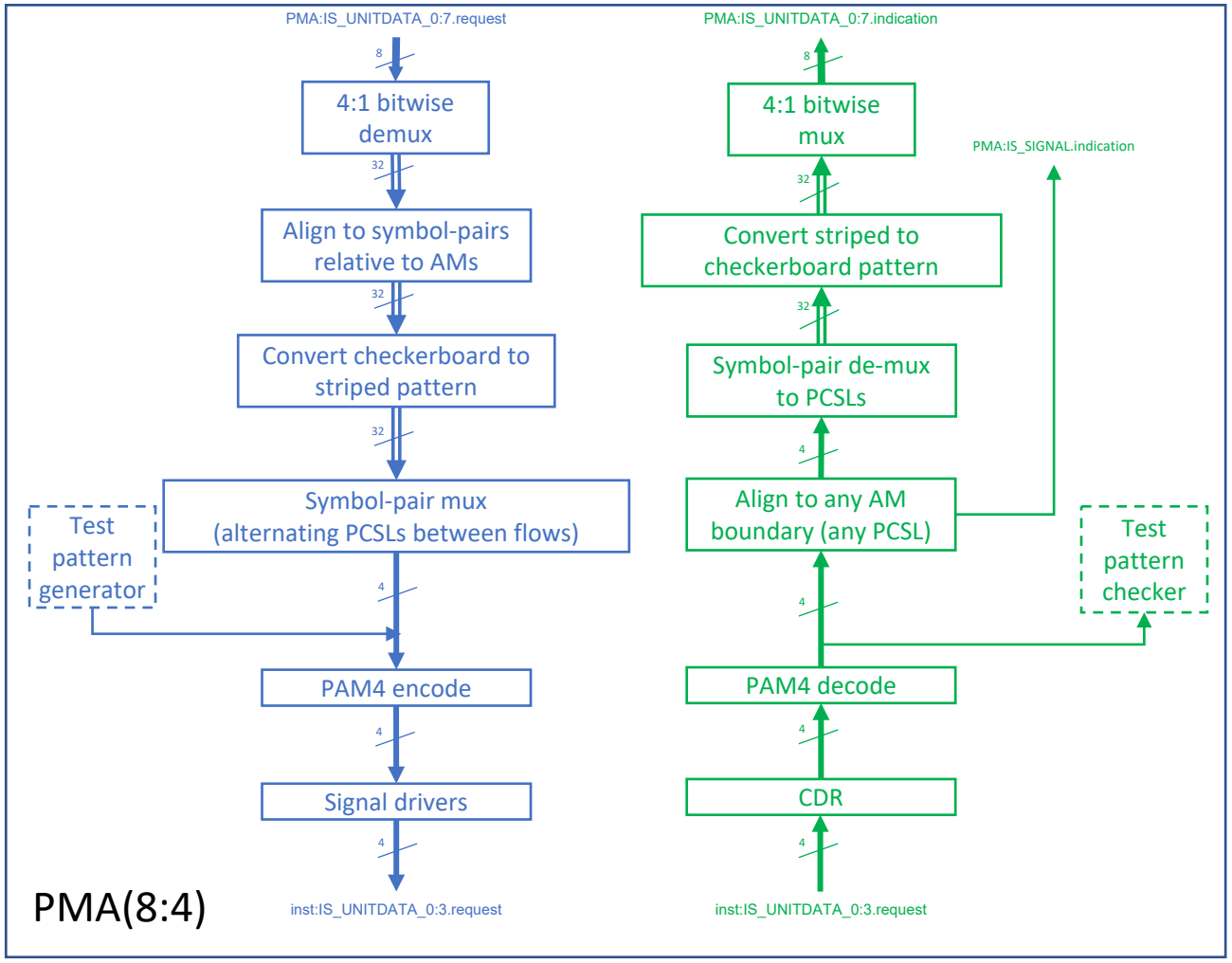


Figure 172–2—Functional block diagram



PMA(4:32) is identical to PMA(32:4) placed backwards (i.e., service interface has 4 lanes and interface below has 32 lanes).
Similarly for PMA(8:1) for 200GBASE-R and PMA(16:2) for 400GBASE-R and their inverses, except for appropriate width changes, and only one flow.

Functional block diagram: **PMA(8:4)**
(800GAUI-8 to 800GAUI-4 converter)



PMA(4:8) is identical to PMA(8:4) placed backwards (i.e., service interface has 4 lanes and interface below has 8 lanes).

Similarly for PMA(2:1) for 200GBASE-R and PMA(4:2) for 400GBASE-R and their inverses, except for appropriate width changes, and only one flow.

Converting checkerboard to striped symbol pattern

- Required for creating a round-robin codeword order on the 200G lanes, to maximize burst immunity of the FEC
- One method, illustrated in the backup section, is **swapping the symbols of each symbol pair on the odd-numbered PCSLs** (in both Tx and Rx)
  - Easy to do in a Tx that is co-located with a PCS
  - Easy to undo in the Rx
- There are other ways to achieve a striped pattern
  - Further study of different options is expected

Muxing constraints

- For 32:4 PMAs (800GBASE-R adjacent to a PCS):
  - Each lane contains symbol-pairs from PCSLs alternating between the two PCS flows
  - This is the same as in clause 173, so **no new constraints**
- For 16:2 and 8:1 PMAs (400GBASE-R / 200GBASE-R adjacent to a PCS):
  - **No constraints** – mux any group of 8 PCSLs in any order
- For 8:4, 4:2, and 2:1 PMAs (gearboxes)
  - Each pair of Tx input lanes is muxed into a single Tx output lane
    - In 8:4, PCSLs are taken alternately from the two flows
  - Each Rx input lane is de-muxed into two Rx output lanes
  - Essentially: shuffling PCSLs across physical lanes is not allowed
- For 4:4, 2:2 and 1:1 PMAs (retimers)
  - Similar to those of the 8:8 PMA (clause 173) (**no new constraints**)
- Corresponding requirements for all “backward” PMAs (e.g., 4:32)
- These rules, combined with symbol-pair muxing, result in **symbols from each of the codewords appearing on each lane with maximum separation from each other**

Specific PMA logic

- For PMAs co-located with a PCS:
  - Symbol alignment on the input lane using only common AM content
  - Undo the checkerboard
- For “external gearbox” (including gearbox modules):
  - Conversion from bit to symbol-pair muxing (100G to 200G per lane)
    - Align to symbol-pair boundary using AMs
    - Unique-AM identification is required to distinguish even and odd PCSLs
  - Conversion from symbol-pair to bit muxing (200G to 100G per lane)
    - Find symbol-pair boundary using any AM (common portion is enough)
    - Distinguish odd and even PCSLs and convert back to checkerboard
    - No unique-AM logic required
- No dependence between lanes that are not muxed together
- No PCSL re-ordering logic anywhere in any PMA
- Deskew buffers are not required
  - symbol alignment can be considered partial deskew, but is very small
- **Overall logic is small compared to DSP or FEC implementation**

PAM4 encoding

- All physical instantiations of the PMA service interface use PAM4 encoding.
- The 10 bits of each RS symbol shall be encoded into five PAM4 symbols:
  - The first transmitted PAM4 symbol is the result of encoding {b0, b1}
  - …
  - The fifth transmitted PAM4 symbol is the result of encoding {b8, b9}
- The encoding of each pair includes Gray mapping, as specified in 120.5.7.1
- 1/1+D Precoding:
  - **Required** on PMA lanes connected to AUIs or electrical PMDs (to prevent significant degradation in 200G/400G where the PCS interleaves only 2 codewords)
  - For interfaces to optical PMDs, may depend on presence and characteristics of inner code – for future discussion, not part of this proposal.

Summary

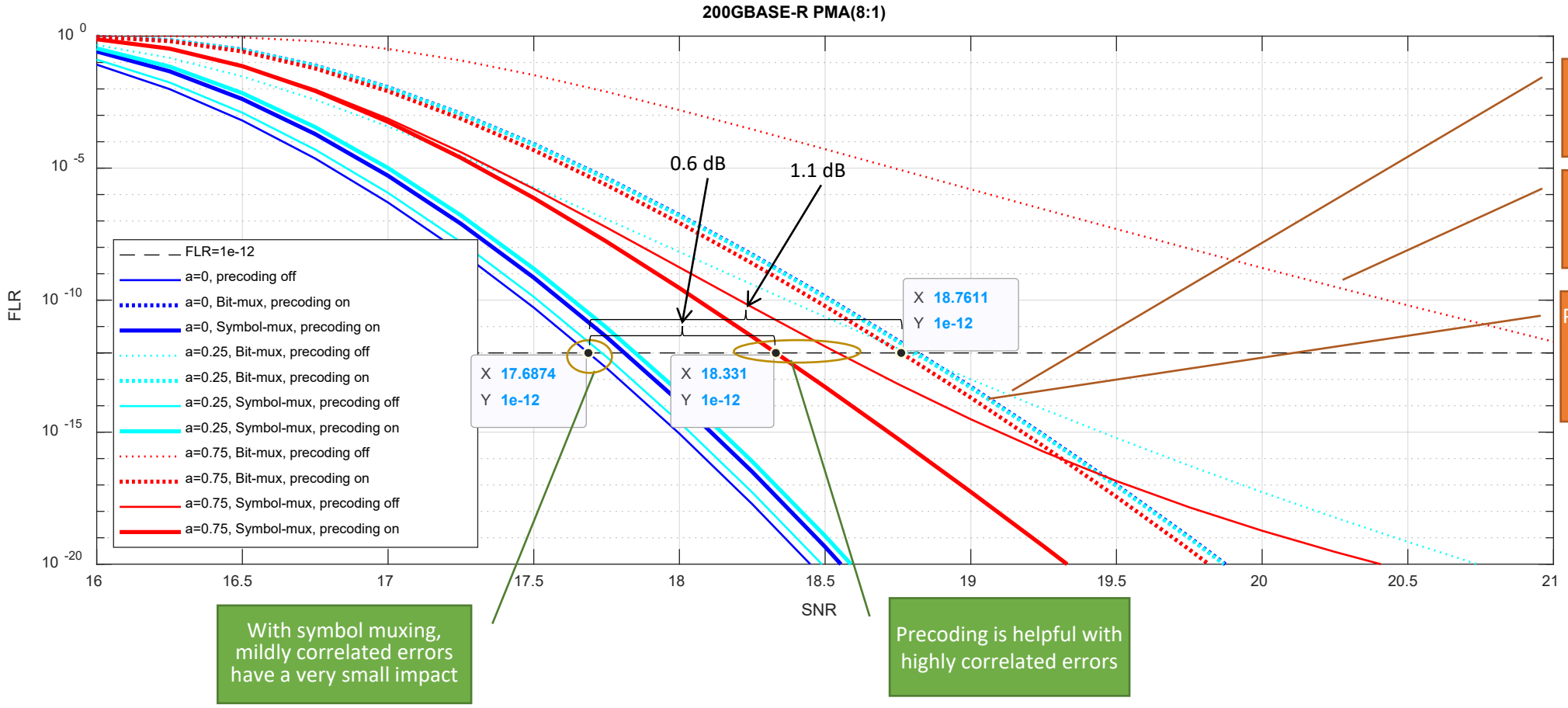
- **Bit muxing will cause severe degradation at 200 Gb/s per lane**
- PMA muxing should maximize temporal separation between symbols of the same codeword on the output lane
  - Round-robin (4 codewords) or alternating pattern (2 codewords)
  - This would optimize FEC performance
- **Symbol-pair muxing** (aka half-blind symbol muxing) **concept has been presented**
  - Achieves maximal separation for best FEC performance
  - Logic implementation is trivial for co-located PMAs and repeaters, and reasonable for gearbox PMAs
  - Method of converting checkerboard to striped symbol pattern and back is still TBD
- In addition to symbol-pair muxing, precoding support is required on PMA lanes connected to AUIs or electrical PMDs.

Backup sections

FEC performance

Backup

Worst case 8:1 muxing: 200G/400G (2 codeword interleaving)



Bit muxing without precoding is very sensitive to even mildly correlated errors

Highly correlated errors are disastrous

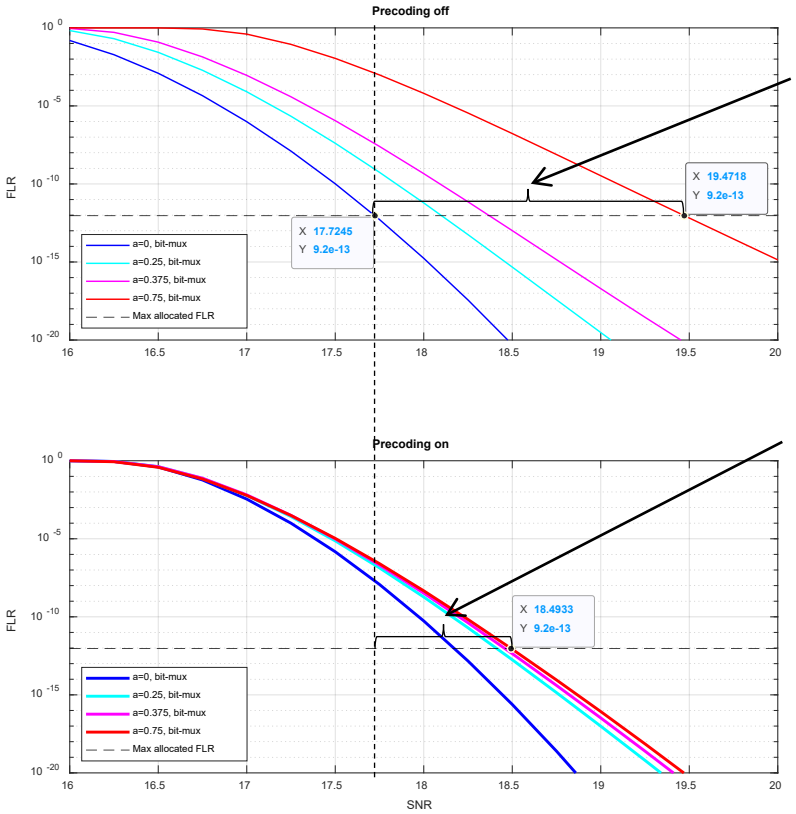
Precoding reduces the bit muxing penalty - but it is still larger than with any symbol muxing case

With symbol muxing, mildly correlated errors have a very small impact

Precoding is helpful with highly correlated errors

FEC performance with an 8:1 bit-muxing PMA with 4-way interleaved FEC

As a function of SNR



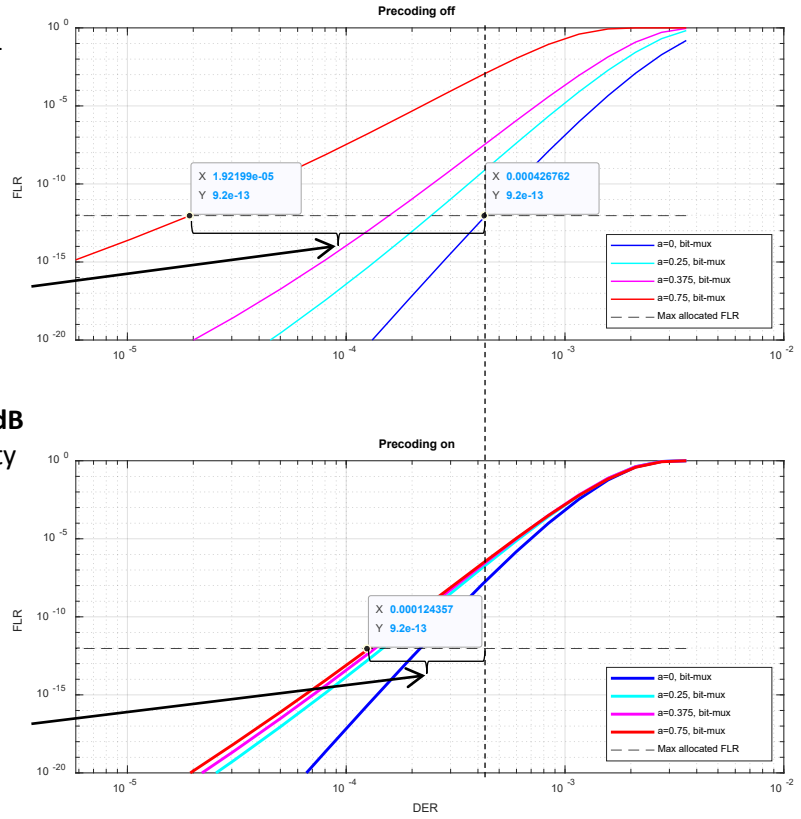
Without precoding, the worst-case penalty at 8:1 muxing (with a=0.75) is **1.75 dB** (much worse than 4:1)

A factor of 20 in BER!

Precoding reduces the worst-case penalty to **0.8 dB** but has a minimum penalty (again, worse than 4:1)

A factor of 4 in BER in almost all cases

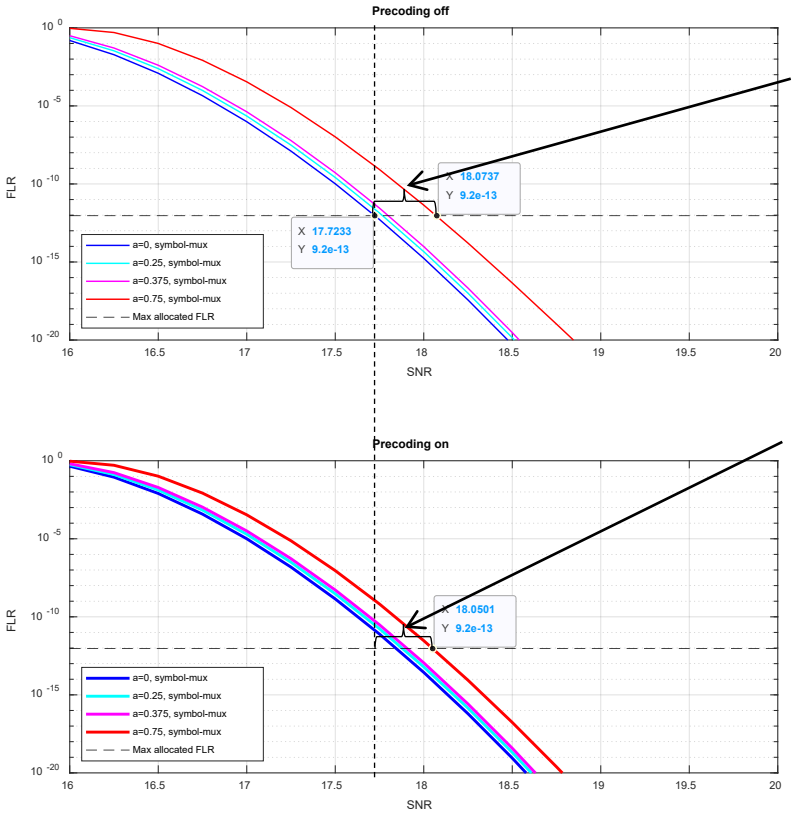
As a function of DER (2*BER)



Source: [ran_3df_01a_2211]

FEC performance with an 8:1 symbol-muxing PMA

As a function of SNR

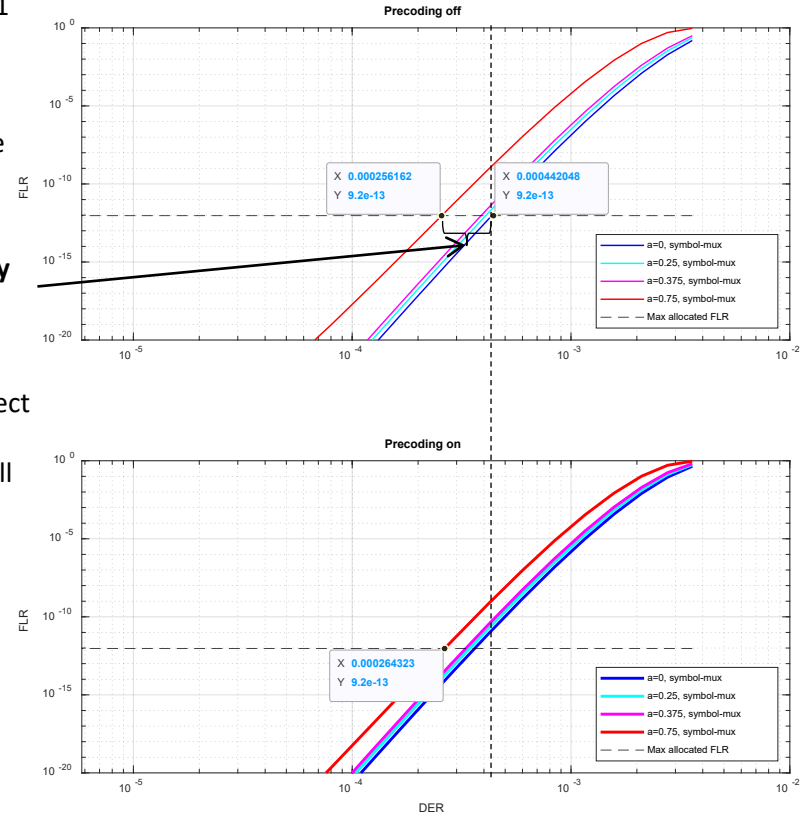


Without precoding, the worst-case penalty at 8:1 muxing (with α=0.75) is **0.35 dB**
(much better than 4:1)
Lower values of α create negligible penalty

A factor of 2 in BER, only in the worst case

Precoding has a minor effect (worst-case 0.33 dB)
Minimum penalty is small

As a function of DER (2*BER)



Source: [ran_3df_01a_2211]

Summary

Compare muxing options for 800G: SNR [dB] and DER for meeting FLR=9.2e-13

Scenario	8-lane AUI/PMD 4:1 bit muxing	4-lane AUI/PMD 8:1 bit muxing	4-lane AUI/PMD 8:1 symbol muxing
Uncorrelated errors	17.7 (reference) 4.3e-4		
Limited DFE, α=0.375	18.05 (Δ=0.35 dB) 2.7e-4	18.4 (Δ=0.6 dB) 1.6e-4	17.8 (Δ=0.1 dB) 3.9e-4
Unlimited DFE, α=0.75	18.7 (Δ=1 dB) 8.9e-5	19.5 (Δ=1.75 dB) 1.9e-5	18.07 (Δ=0.35 dB) 2.6e-4
Unlimited DFE, α=0.75 + precoding	18.3 (Δ=0.6 dB) 1.8e-4	18.5 (Δ=0.75 dB) 1.2e-4	18.05 (Δ=0.33 dB) 2.6e-4
Overall	Acceptable for PMD where precoding can be negotiated AUI and optics assumed not to have α=0.75	Not acceptable unless precoding is negotiated	Minimal degradation in all cases Precoding not required*

* Precoding may be needed for 400G and 200G with only 2-way codeword interleaving

Source: ran_3df_01a_2211

Striping the checkerboard

Backup

Detailed description of symbol-pair muxing with symbol swapping (striping method “A”)

- In the transmit direction:
  - Align to symbol-pair (20-bit) boundary and process a symbol-pair at a time
    - Either using information from a co-located PCS, or using AMs
  - In odd-numbered PCSLs, swap the two 10-bit symbols in each symbol pair.
    - This turns the “checkerboard” into a striped pattern
    - The swapping is also applied to the symbols included in AMs of odd PCSLs
  - Mux the symbol-pairs from each PCSL into the output lane.
    - In 800G, alternate symbol-pairs between flow 0 and flow 1
    - In 200G/400G – any order.
- In the receive direction, on each input lane:
  - Find at least one AM
    - Can be found using just the common part of the AMs
    - Due to symbol-pair muxing, each AM appears as six symbol pairs; pairs can be either swapped or in the original order; pairs separated by three other symbol pairs
  - Distribute symbol pairs to PCSLs based on the AM alignment
  - On each PCSL, find the AM and determine if it has swapped symbol pairs or not
    - If an AM in a PCSL has swapped symbol pairs, swap each symbol pair in that PCSL (undo the Tx swapping)
  - If the output lanes are connected to a 100G/lane AUI, bitwise-mux groups of 4 PCSLs.
- The examples in the next slides use this striping method.

32:4 (800GBASE-R) symbol-pair-muxing: Example 1

Subsequent symbol pairs from a PCSL appear every 16 symbols on the output lane

PCSL #

Symbol transmit order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
last 4	A32	B32	A33	B33	A34	B34	A35	B35	A36	B36	A37	B37	A38	B38	A39	B39	C32	D32	C33	D33	C34	D34	C35	D35	C36	D36	C37	D37	C38	D38	C39	D39
3	B24	A24	B25	A25	B26	A26	B27	A27	B28	A28	B29	A29	B30	A30	B31	A31	D24	C24	D25	C25	D26	C26	D27	C27	D28	C28	D29	C29	D30	C30	D31	C31
2	A16	B16	A17	B17	A18	B18	A19	B19	A20	B20	A21	B21	A22	B22	A23	B23	C16	D16	C17	D17	C18	D18	C19	D19	C20	D20	C21	D21	C22	D22	C23	D23
1	B8	A8	B9	A9	B10	A10	B11	A11	B12	A12	B13	A13	B14	A14	B15	A15	D8	C8	D9	C9	D10	C10	D11	C11	D12	C12	D13	C13	D14	C14	D15	C15
first 0	A0	B0	A1	B1	A2	B2	A3	B3	A4	B4	A5	B5	A6	B6	A7	B7	C0	D0	C1	D1	C2	D2	C3	D3	C4	D4	C5	D5	C6	D6	C7	D7

PCSL grouping:

```
Lane 0:  0, 16 |  8, 24 |  1, 17 |  9, 25
Lane 1:  2, 18 | 10, 26 |  3, 19 | 11, 27
Lane 2:  4, 20 | 12, 28 |  5, 21 | 13, 29
Lane 3:  6, 22 | 14, 30 |  7, 23 | 15, 31
```

Symbol pair from an odd PCSL, swapped

Symbol pair from an even PCSL

lane #

Symbol transmit order	0	1	2	3
last 25	B16	B17	B18	B19
24	A24	A25	A26	A26
23	D28	D29	D30	D31
22	C20	C21	C22	C23
21	B28	B29	B30	B31
20	A20	A21	A22	A23
19	D24	D25	D26	D27
18	C16	C17	C18	C19
17	B24	B25	B26	B27
16	A16	A17	A18	A19
15	D4	D5	D6	D7
14	C12	C13	C14	C15
13	B4	B5	B6	B7
12	A12	A13	A14	A15
11	D0	D1	D2	D3
10	C8	C9	C10	C11
9	B0	B1	B2	B3
8	A8	A9	A10	A11
7	D12	D13	D14	D15
6	C4	C5	C6	C7
5	B12	B13	B14	B15
4	A4	A5	A6	A7
3	D8	D9	D10	D11
2	C0	C1	C2	C3
1	B8	B9	B10	B11
first 0	A0	A1	A2	A3

32:4 (800GBASE-R) symbol-pair-muxing: Example 2

Subsequent symbol pairs from a PCSL appear every 16 symbols on the output lane

PCSL #

Symbol transmit order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
last 4	A32	B32	A33	B33	A34	B34	A35	B35	A36	B36	A37	B37	A38	B38	A39	B39	C32	D32	C33	D33	C34	D34	C35	D35	C36	D36	C37	D37	C38	D38	C39	D39
3	B24	A24	B25	A25	B26	A26	B27	A27	B28	A28	B29	A29	B30	A30	B31	A31	D24	C24	D25	C25	D26	C26	D27	C27	D28	C28	D29	C29	D30	C30	D31	C31
2	A16	B16	A17	B17	A18	B18	A19	B19	A20	B20	A21	B21	A22	B22	A23	B23	C16	D16	C17	D17	C18	D18	C19	D19	C20	D20	C21	D21	C22	D22	C23	D23
1	B8	A8	B9	A9	B10	A10	B11	A11	B12	A12	B13	A13	B14	A14	B15	A15	D8	C8	D9	C9	D10	C10	D11	C11	D12	C12	D13	C13	D14	C14	D15	C15
first 0	A0	B0	A1	B1	A2	B2	A3	B3	A4	B4	A5	B5	A6	B6	A7	B7	C0	D0	C1	D1	C2	D2	C3	D3	C4	D4	C5	D5	C6	D6	C7	D7

PCSL grouping:

```
Lane 0: 0, 16 | 8,  24 | 4, 20 | 12, 28
Lane 1: 1, 17 | 9,  25 | 5, 21 | 13, 29
Lane 2: 2, 18 | 10, 26 | 6, 22 | 14, 30
Lane 3: 3, 19 | 11, 27 | 7, 23 | 15, 31
```

Symbol pair from an odd PCSL, swapped

Symbol pair from an even PCSL

lane #

Symbol transmit order	0	1	2	3
last 25	B26	B18	B27	B19
24	A18	A26	A19	A27
23	D28	D20	D29	D21
22	C20	C28	C21	C29
21	B28	B20	B29	B21
20	A20	A28	A21	A29
19	D24	D16	D25	D17
18	C16	C24	C17	C25
17	B24	B16	B25	B17
16	A16	A24	A17	A25
15	D14	D6	D15	D7
14	C6	C14	C7	C15
13	B14	B6	B15	B7
12	A6	A14	A7	A15
11	D10	D2	D11	D3
10	C2	C10	C3	C11
9	B10	B2	B11	B3
8	A2	A10	A3	A11
7	D12	D4	D13	D5
6	C4	C12	C5	C13
5	B12	B4	B13	B5
4	A4	A12	A5	A13
3	D8	D0	D9	D1
2	C0	C8	C1	C9
1	B8	B0	B9	B1
first 0	A0	A8	A1	A9

PMA(8:4) example

(Combining two bit-muxed 800GAUI-8 lanes into one 800GAUI-4 symbol-pair-muxed lane)

Bit patterns on two input lanes (with skew)
Every cell represents a PAM4 symbol (2 bits)

UI\Lane	0	4
22	b81a81	d122c122
21	d80c80	b122a122
20	b80a80	d121c121
19	c9d9	b121a121
18	a9b9	d120c120
17	c8d8	b120a120
16	a8b8	c49d49
15	c7d7	a49b49
14	a7b7	c48d48
13	c6d6	a48b48
12	a6b6	c47d47
11	c5d5	a47b47
10	a5b5	c46d46
9	c4d4	a46b46
8	a4b4	c45d45
7	c3d3	a45b45
6	a3b3	c44d44
5	c2d2	a44b44
4	a2b2	c43d43
3	c1d1	a43b43
2	a1B1	c42d42
1	c0d0	a42b42
0	a0b0	c41d41

1:4 bit-demux each lane
Deskew to 2-symbol boundary

Bit patterns on the original 8 PCSLs (deskewed)
Every cell represents one bit

Bit \ PCSL	0	1	16	17	8	9	24	25
19	b89	a89	d89	c89	b129	a129	d129	c129
18	b88	a88	d88	c88	b128	a128	d128	c128
17	b87	a87	d87	c87	b127	a127	d127	c127
16	b86	a86	d86	c86	b126	a126	d126	c126
15	b85	a85	d85	c85	b125	a125	d125	c125
14	b84	a84	d84	c84	b124	a124	d124	c124
13	b83	a83	d83	c83	b123	a123	d123	c123
12	b82	a82	d82	c82	b122	a122	d122	c122
11	b81	a81	d81	c81	b121	a121	d121	c121
10	b80	a80	d80	c80	b120	a120	d120	c120
9	a9	b9	c9	d9	a49	b49	c49	d49
8	a8	b8	c8	d8	a48	b48	c48	d48
7	a7	b7	c7	d7	a47	b47	c47	d47
6	a6	b6	c6	d6	a46	b46	c46	d46
5	a5	b5	c5	d5	a45	b45	c45	d45
4	a4	b4	c4	d4	a44	b44	c44	d44
3	a3	b3	c3	d3	a43	b43	c43	d43
2	a2	b2	c2	d2	a42	b42	c42	d42
1	a1	b1	c1	d1	a41	b41	c41	d41
0	a0	b0	c0	d0	a40	b40	c40	d40

Symbol-swap odd PCSLs
8:1 mux symbol pairs
alternate flows

Bit pattern on one 800GAUI-4 lane
Every cell represents a PAM4 symbol (2 bits)

UI\Lane	0
39	d128d129
38	d126d127
37	d124d125
36	d122d123
35	d120d121
34	c48c49
33	c46c47
32	c44c45
31	c42c43
30	c40c41
29	b128b129
28	b126b127
27	b124b125
26	b122b123
25	b120b121
24	a48a49
23	a46a47
22	a44a45
21	a42a43
20	a40a41
19	d88d89
18	d86d87
17	d84d85
16	d82d83
15	d80d81
14	c8c9
13	c6c7
12	c4c5
11	c2c3
10	c0c1
9	b88b89
8	b86b87
7	b84b85
6	b82b83
5	b80b81
4	a8a9
3	a6a7
2	a4a5
1	a2a3
0	a0a1

25 (d40-d49)
25 (c120-c129)
9 (b40-b49)
9 (a120-a129)
17 (d0-d9)
17 (c80-c89)
1 (b0-b9)
1 (a80-a89)

24 (D)

24 (C)

8 (B)

8 (A)

16 (D)

16 (C)

0 (B)

0 (A)

PMA(2:1) example

(Combining two bit-muxed 200GAUI-2 lanes into one 200GAUI-1 symbol-pair-muxed lane)

Bit patterns on two input lanes (with skew)
Every cell represents a PAM4 symbol (2 bits)

UI\Lane	0	1
19	b44 a44	b72 a72
18	b63 a63	b52 a52
17	b43 a43	b71 a71
16	b62 a62	b51 a51
15	b42 a42	b70 a70
14	b61 a61	b50 a50
13	b41 a41	a39 b39
12	b60 a60	a19 b19
11	b40 a40	a38 b38
10	a29 b29	a18 b18
9	a9 b9	a37 b37
8	a28 b28	a17 b17
7	a8 b8	a36 b36
6	a27 b27	a16 b16
5	a7 b7	a35 b35
4	a26 b26	a15 b15
3	a6 b6	a34 b34
2	a25 b25	a14 b14
1	a5 b5	a33 b33
0	a24 b24	a13 b13

1:4 bit-demux each lane
Deskew to 2-symbol boundary

Bit patterns on the original 8 PCSLs (deskewed)
Every cell represents one bit

Bit \ PCSL	0	1	4	5	2	3	6	7
19	b49	a49	b69	a69	b59	a59	b79	a79
18	b48	a48	b68	a68	b58	a58	b78	a78
17	b47	a47	b67	a67	b57	a57	b77	a77
16	b46	a46	b66	a66	b56	a56	b76	a76
15	b45	a45	b65	a65	b55	a55	b75	a75
14	b44	a44	b64	a64	b54	a54	b74	a74
13	b43	a43	b63	a63	b53	a53	b73	a73
12	b42	a42	b62	a62	b52	a52	b72	a72
11	b41	a41	b61	a61	b51	a51	b71	a71
10	b40	a40	b60	a60	b50	a50	b70	a70
9	a9	b9	a29	b29	a19	b19	a39	b39
8	a8	b8	a28	b28	a18	b18	a38	b38
7	a7	b7	a27	b27	a17	b17	a37	b37
6	a6	b6	a26	b26	a16	b16	a36	b36
5	a5	b5	a25	b25	a15	b15	a35	b35
4	a4	b4	a24	b24	a14	b14	a34	b34
3	a3	b3	a23	b23	a13	b13	a33	b33
2	a2	b2	a22	b22	a12	b12	a32	b32
1	a1	b1	a21	b21	a11	b11	a31	b31
0	a0	b0	a20	b20	a10	b10	a30	b30

Symbol-swap odd PCSLs
8:1 mux symbol pairs

Bit pattern on one 800GAUI-4 lane
Every cell represents a PAM4 symbol (2 bits)

UI\Lane	0
39	b78 b79
38	b76 b77
37	b74 b75
36	b72 b73
35	b70 b71
34	a38 a39
33	a36 a37
32	a34 a35
31	a32 a33
30	a30 a31
29	b68 b69
28	b66 b67
27	b64 b65
26	b62 b63
25	b60 b61
24	a28 a29
23	a26 a27
22	a24 a25
21	a22 a23
20	a20 a21
19	b58 b59
18	b56 b57
17	b54 b55
16	b52 b53
15	b50 b51
14	a18 a19
13	a16 a17
12	a14 a15
11	a12 a13
10	a10 a11
9	b48 b49
8	b46 b47
7	b44 b45
6	b42 b43
5	b40 b41
4	a8 a9
3	a6 a7
2	a4 a5
1	a2 a3
0	a0 a1

7 (b30-b39)
7 (a70-a79)
5 (b20-b29)
5 (a60-a69)
3 (b10-b19)
3 (a50-a59)
1 (b0-b9)
1 (a40-a49)

6 (B)
6 (A)
4 (B)
4 (A)
2 (B)
2 (A)
0 (B)
0 (A)

400GAUI-4 to 400GAUI-2 conversion is similar

Detailed description of symbol-pair muxing with symbol delaying (striping method “B”)

- In the transmit direction:
  - Align to symbol-pair (20-bit) boundary
    - Either using information from a co-located PCS, or using AMs
  - In odd-numbered PCSLs, delay the symbol sequence by one symbol, such that the symbols are paired in the same order as in even-numbered PCSLs
    - This turns the “checkerboard” into a striped pattern
    - The shifted pairing is also applied to the symbols included in AMs of odd PCSLs
  - Mux the symbol-pairs from each PCSL into the output lane.
    - In 800G, alternate symbol-pairs between flow 0 and flow 1
    - In 200G/400G – any order.
- In the receive direction, on each input lane:
  - Find at least one AM
    - Can be found using just the common part of the AMs
    - Due to symbol-pair muxing, each AM appears as six symbol pairs on even PCSLs, and as five symbol-pairs and two separate symbol on odd PCSLs; pairs separated by three other symbol pairs
  - Distribute symbol pairs to PCSLs based on the AM alignment
  - If the output lanes are connected to a 100G/lane AUI, bitwise-mux groups of 4 PCSLs.

Previously considered methods

Backup

Options considered in internal discussions for converting from bit to symbol muxing

Bit mux ➔ symbol mux direction

- **“Full”**: Bitwise de-mux; AM lock and fully align all input PCSLs; re-mux with specific lane constraints (for example, ran_3df_02a_2211)
  - Requires alignment between all lanes and large deskew buffers
- **“Blind”**: Bitwise de-mux; on each pair of input lanes, AM lock and align PCSLs to 1-symbol boundary; re-mux with no constraints
  - Allows per-lane implementation, minimum deskew buffers; but muxed symbol order is not optimal
- **“Half-blind”*** : Bitwise de-mux; on each pair of input lanes, AM lock and align PCSLs to 2-symbol boundary; re-mux with constraint to create a repeating codeword order (e.g., ABCDABCD)
  - Allows per-lane implementation, slightly larger buffers; muxed symbol order is optimal
- **“10-bit muxing”**: Bitwise de-mux; mux 10-bit symbols with any alignment
  - Simplest “bit ➔ symbol” direction, but “symbol ➔ bit” direction is challenging (must recover symbol boundaries and align in order to be PCS-compatible)
- **“Hybrid”**: 4:1 symbol-mux and then 2:1 bit-mux
  - Not compatible with existing 100 Gb/s per lane AUIs that use 4:1 bit-mux

* The term “half blind” was used in several offline discussions. It is essentially identical to the “symbol pair muxing” concept in this presentation.

Comparison of muxing options considered

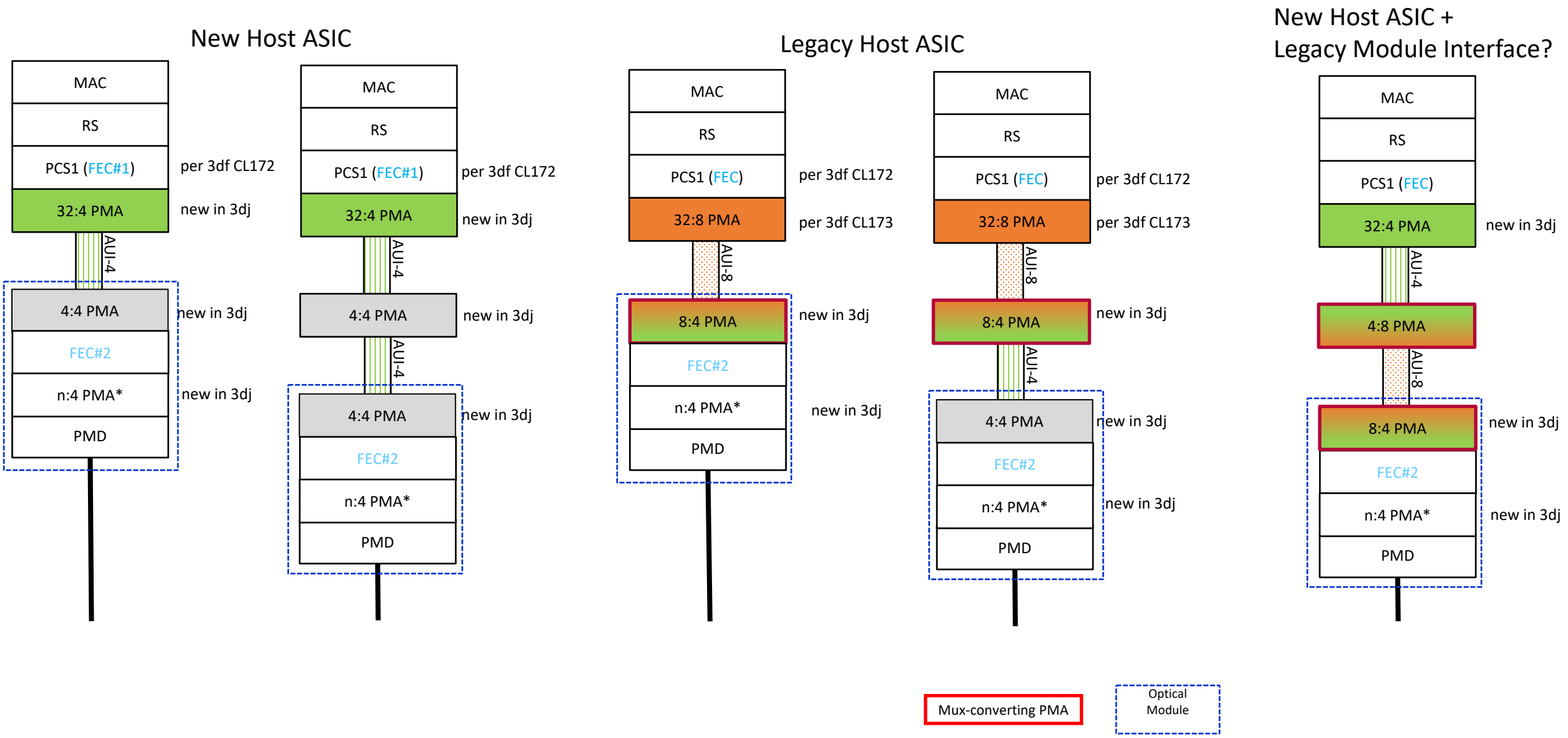
Module Muxing Scheme		Performance	Complexity	TX module AM Lock	RX module AM lock	Deskew?	Breakout Friendly?
Symbol Mux (800 GbE and 1.6 TbE)	Specific/Optimal symbol mux (Fully aligned codewords)	High	High	Full AM	Full AM	Full deskew of AUI lanes	No?
	Half-blind symbol mux (Aligned to 2x RS-symbol boundary)	High	Medium	Full AM	Common part of AM	20-b deskew	Yes
	Blind symbol mux (Aligned to 1x RS-symbol boundary)	Medium	Low	Common part of AM	Common part of AM	10-b deskew	Yes
	Blind 10-bit/40-bit mux (Un-aligned, arbitrary position)	Medium	Lower	None	Common part of AM	No	Yes
Hybrid (800 GbE only)	4:1 symbol mux + 2:1 bit mux (802.3df Mux changed)	Low	Lower	None	None	No	Yes
Bit Mux (800 GbE and 1.6 TbE)	2:1 bit mux (1.6TbE assuming 16x100G PCS lanes)	Low	Lower	None	None	No	Yes
	8:1 bit mux (Same as in 802.3df)	Lowest	Lowest	None	None	No	Yes

More protocol agnostic

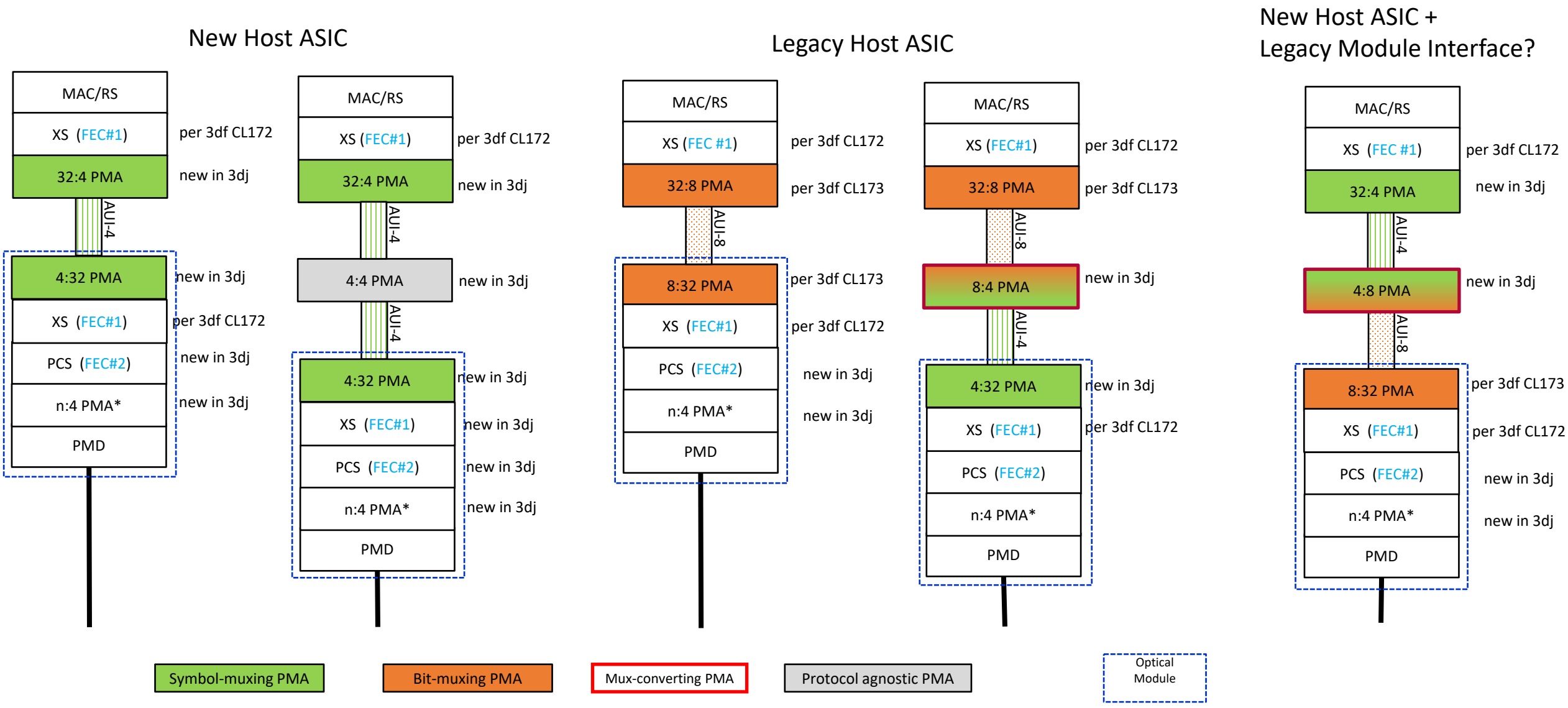
PMA partitioning examples

Backup

Sublayer stacks for 800GBASE-R with concatenated FEC



Sublayer stacks for 800GBASE-R with segmented FEC

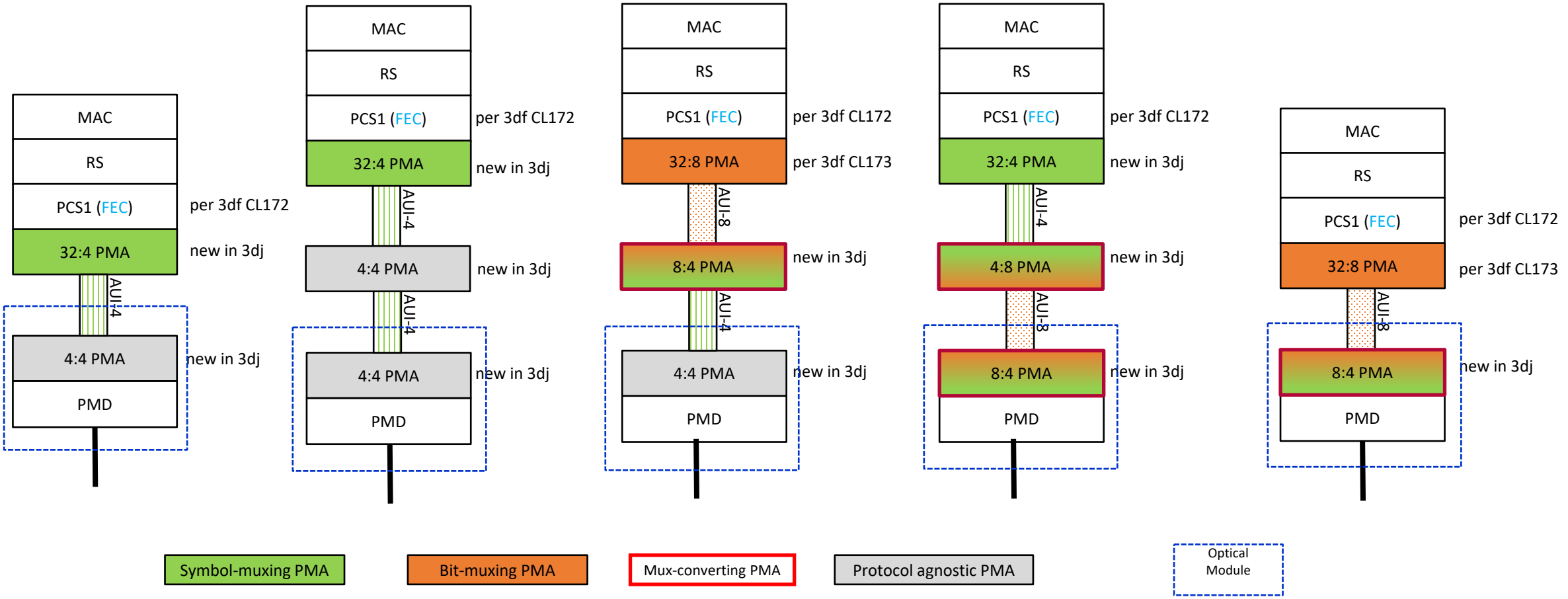


Sublayer stacks for 800GBASE-R with end-to-end FEC

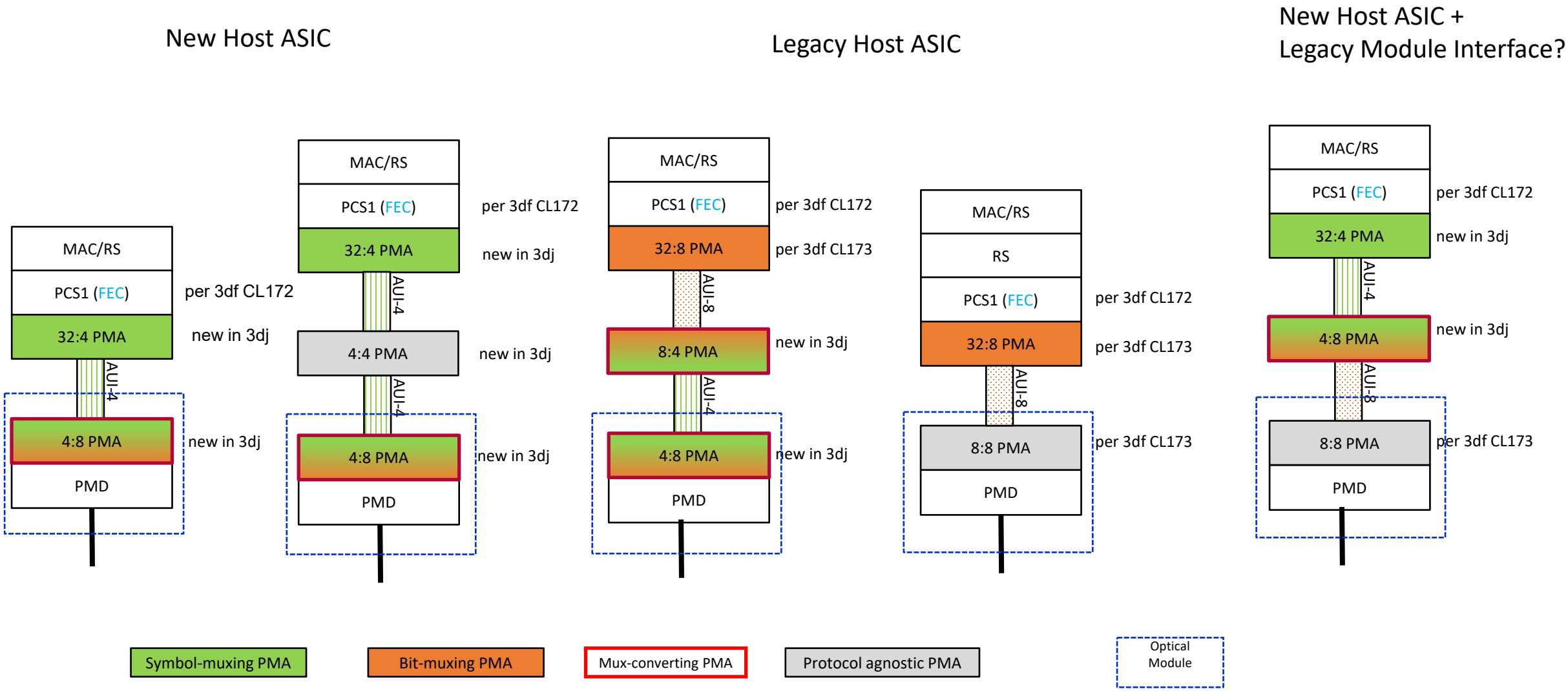
New Host ASIC

Legacy Host ASIC

New Host ASIC +
Legacy Module Interface?



Sublayer stacks for 800GBASE-R with end-to-end FEC and 100G/lane optics



New Clause and Annex structure

Backup

New Clause and Annex structure

- Clause 1XX: Physical Medium Attachment (PMA) sublayers for 200 Gb/s per lane signaling
  - 1XX.1 Overview – as in 173
  - 1XX.2 PMA service interface – as in 173
  - 1XX.3 Service interface below the PMA – as in 173
  - 1XX.4 Functions within the PMA
    - **Subclauses for New content – symbol-pair muxing, AM lock, checkerboard to striped, diagrams… – per baseline proposal**
  - 1XX.5 PMA MDIO function mapping
  - 1XX.6 PICS
- Annex 1XXa: PMA partitioning examples – per baseline proposal