Lane muxing constraints for 800GBASE-R PMA

(in support of comment #6, and pertaining to comments #166, #167)

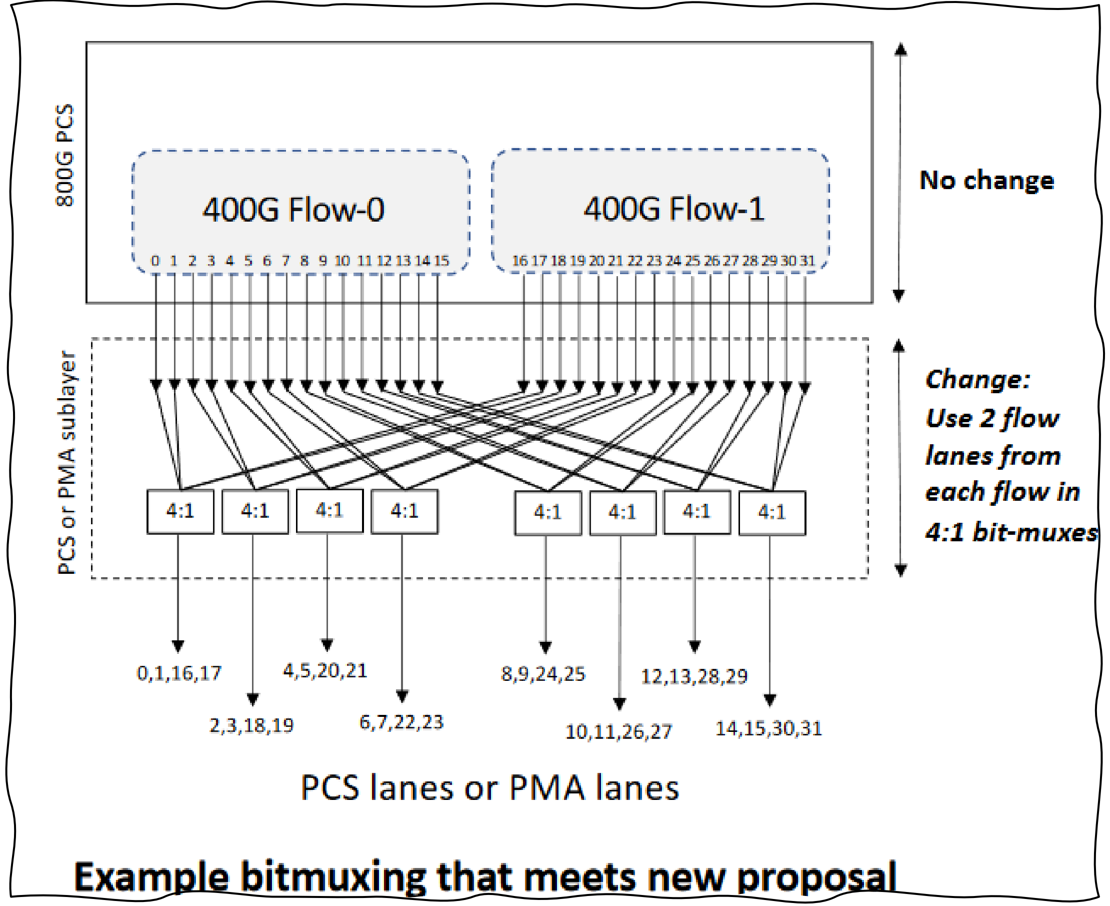
Adee Ran, Cisco

Support

- Xiang He, Huawei
- Hao Ren, Huawei
- Ali Ghiasi, Ghiasi Quantum

Restricted PCS lane muxing by the PMA

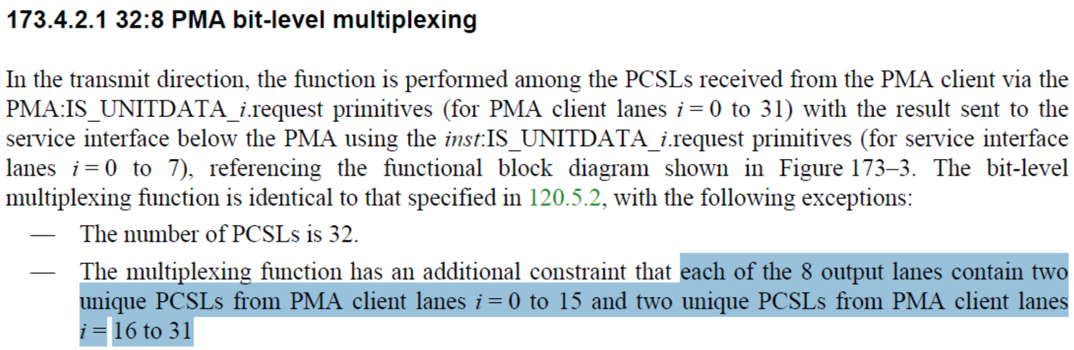
Page 18 of shrikhande_3df_01a_221004 (adopted baseline)



Example bitmuxing that meets new proposal

Below, codewords of flow 0 are denoted A/B, those of flow 1 are denoted C/D

This was implemented in D1.0 as part of the PMA sublayer, Clause 173:



173.4.2.1 32:8 PMA bit-level multiplexing

In the transmit direction, the function is performed among the PCSLs received from the PMA client via the PMA:IS_UNITDATA_*i*.request primitives (for PMA client lanes *i* = 0 to 31) with the result sent to the service interface below the PMA using the *inst*:IS_UNITDATA_*i*.request primitives (for service interface lanes *i* = 0 to 7), referencing the functional block diagram shown in Figure 173–3. The bit-level multiplexing function is identical to that specified in 120.5.2, with the following exceptions:

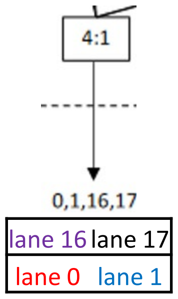
- The number of PCSLs is 32.
- The multiplexing function has an additional constraint that each of the 8 output lanes contain two unique PCSLs from PMA client lanes *i* = 0 to 15 and two unique PCSLs from PMA client lanes *i* = 16 to 31

- The purpose of the constraint is to have bits from all four codewords on each PMA lane
  - But… there is more than one way to do it
- Comments #166, #167 suggest that this constraint become only a recommendation
- Below I compare three options A, B, and X

Option A (constrained muxing)

Specific example of muxing lanes per the diagram in slide 3

UI\Lane	0	1	2	3	4	5	6	7
22	B81A81	B91A91	B101A101	B111A111	B121A121	B131A131	B141A141	B151A151
21	D80C80	D90C90	D100C100	D110C110	D120C120	D130C130	D140C140	D150C150
20	B80A80	B90A90	B100A100	B110A110	B120A120	B130A130	B140A140	B150A150
19	C9D9	C19D19	C29D29	C39D39	C49D49	C59D59	C69D69	C79D79
18	A9B9	A19B19	A29B29	A39B39	A49B49	A59B59	A69B69	A79B79
17	C8D8	C18D18	C28D28	C38D38	C48D48	C58D58	C68D68	C78D78
16	A8B8	A18B18	A28B28	A38B38	A48B48	A58B58	A68B68	A78B78
15	C7D7	C17D17	C27D27	C37D37	C47D47	C57D57	C67D67	C77D77
14	A7B7	A17B17	A27B27	A37B37	A47B47	A57B57	A67B67	A77B77
13	C6D6	C16D16	C26D26	C36D36	C46D46	C56D56	C66D66	C76D76
12	A6B6	A16B16	A26B26	A36B36	A46B46	A56B56	A66B66	A76B76
11	C5D5	C15D15	C25D25	C35D35	C45D45	C55D55	C65D65	C75D75
10	A5B5	A15B15	A25B25	A35B35	A45B45	A55B55	A65B65	A75B75
9	C4D4	C14D14	C24D24	C34D34	C44D44	C54D54	C64D64	C74D74
8	A4B4	A14B14	A24B24	A34B34	A44B44	A54B54	A64B64	A74B74
7	C3D3	C13D13	C23D23	C33D33	C43D43	C53D53	C63D63	C73D73
6	A3B3	A13B13	A23B23	A33B33	A43B43	A53B53	A63B63	A73B73
5	C2D2	C12D12	C22D22	C32D32	C42D42	C52D52	C62D62	C72D72
4	A2B2	A12B12	A22B22	A32B32	A42B42	A52B52	A62B62	A72B72
3	C1D1	C11D11	C21D21	C31D31	C41D41	C51D51	C61D61	C71D71
2	A1B1	A11B11	A21B21	A31B31	A41B41	A51B51	A61B61	A71B71
1	C0D0	C10D10	C20D20	C30D30	C40D40	C50D50	C60D60	C70D70
0	A0B0	A10B10	A20B20	A30B30	A40B40	A50B50	A60B60	A70B70



PCS checkerboard symbol pattern

Each PAM4 symbol contains two bits from the same flow (either flow 0 – with codewords A and B, or flow 1 – with codewords C and D)

Consecutive PAM4 symbol are from alternate flows

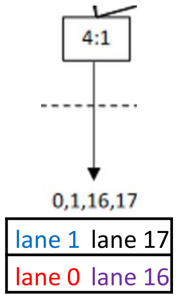
This muxing adheres to the constraint

Combined with the checkerboard pattern, this muxing **allocates LSBs and MSBs of the PAM4 symbols equally among the four codewords**

Option B (constrained muxing, alternative)

Another example of muxing lanes per the diagram in slide 3

UI\Lane	0	1	2	3	4	5	6	7
22	B81 D81	B91 D91	B101 D101	B111 D111	B121 D121	B131 D131	B141 D141	B151 D151
21	A80 C80	A90 C90	A100 C100	A110 C110	A120 C120	A130 C130	A140 C140	A150 C150
20	B80 D80	B90 D90	B100 D100	B110 D110	B120 D120	B130 D130	B140 D140	B150 D150
19	B9 D9	B19 D19	B29 D29	B39 D39	B49 D49	B59 D59	B69 D69	B79 D79
18	A9 C9	A19 C19	A29 C29	A39 C39	A49 C49	A59 C59	A69 C69	A79 C79
17	B8 D8	B18 D18	B28 D28	B38 D38	B48 D48	B58 D58	B68 D68	B78 D78
16	A8 C8	A18 C18	A28 C28	A38 C38	A48 C48	A58 C58	A68 C68	A78 C78
15	B7 D7	B17 D17	B27 D27	B37 D37	B47 D47	B57 D57	B67 D67	B77 D77
14	A7 C7	A17 C17	A27 C27	A37 C37	A47 C47	A57 C57	A67 C67	A77 C77
13	B6 D6	B16 D16	B26 D26	B36 D36	B46 D46	B56 D56	B66 D66	B76 D76
12	A6 C6	A16 C16	A26 C26	A36 C36	A46 C46	A56 C56	A66 C66	A76 C76
11	B5 D5	B15 D15	B25 D25	B35 D35	B45 D45	B55 D55	B65 D65	B75 D75
10	A5 C5	A15 C15	A25 C25	A35 C35	A45 C45	A55 C55	A65 C65	A75 C75
9	B4 D4	B14 D14	B24 D24	B34 D34	B44 D44	B54 D54	B64 D64	B74 D74
8	A4 C4	A14 C14	A24 C24	A34 C34	A44 C44	A54 C54	A64 C64	A74 C74
7	B3 D3	B13 D13	B23 D23	B33 D33	B43 D43	B53 D53	B63 D63	B73 D73
6	A3 C3	A13 C13	A23 C23	A33 C33	A43 C43	A53 C53	A63 C63	A73 C73
5	B2 D2	B12 D12	B22 D22	B32 D32	B42 D42	B52 D52	B62 D62	B72 D72
4	A2 C2	A12 C12	A22 C22	A32 C32	A42 C42	A52 C52	A62 C62	A72 C72
3	B1 D1	B11 D11	B21 D21	B31 D31	B41 D41	B51 D51	B61 D61	B71 D71
2	A1 C1	A11 C11	A21 C21	A31 C31	A41 C41	A51 C51	A61 C61	A71 C71
1	B0 D0	B10 D10	B20 D20	B30 D30	B40 D40	B50 D50	B60 D60	B70 D70
0	A0 C0	A10 C10	A20 C20	A30 C30	A40 C40	A50 C50	A60 C60	A70 C70



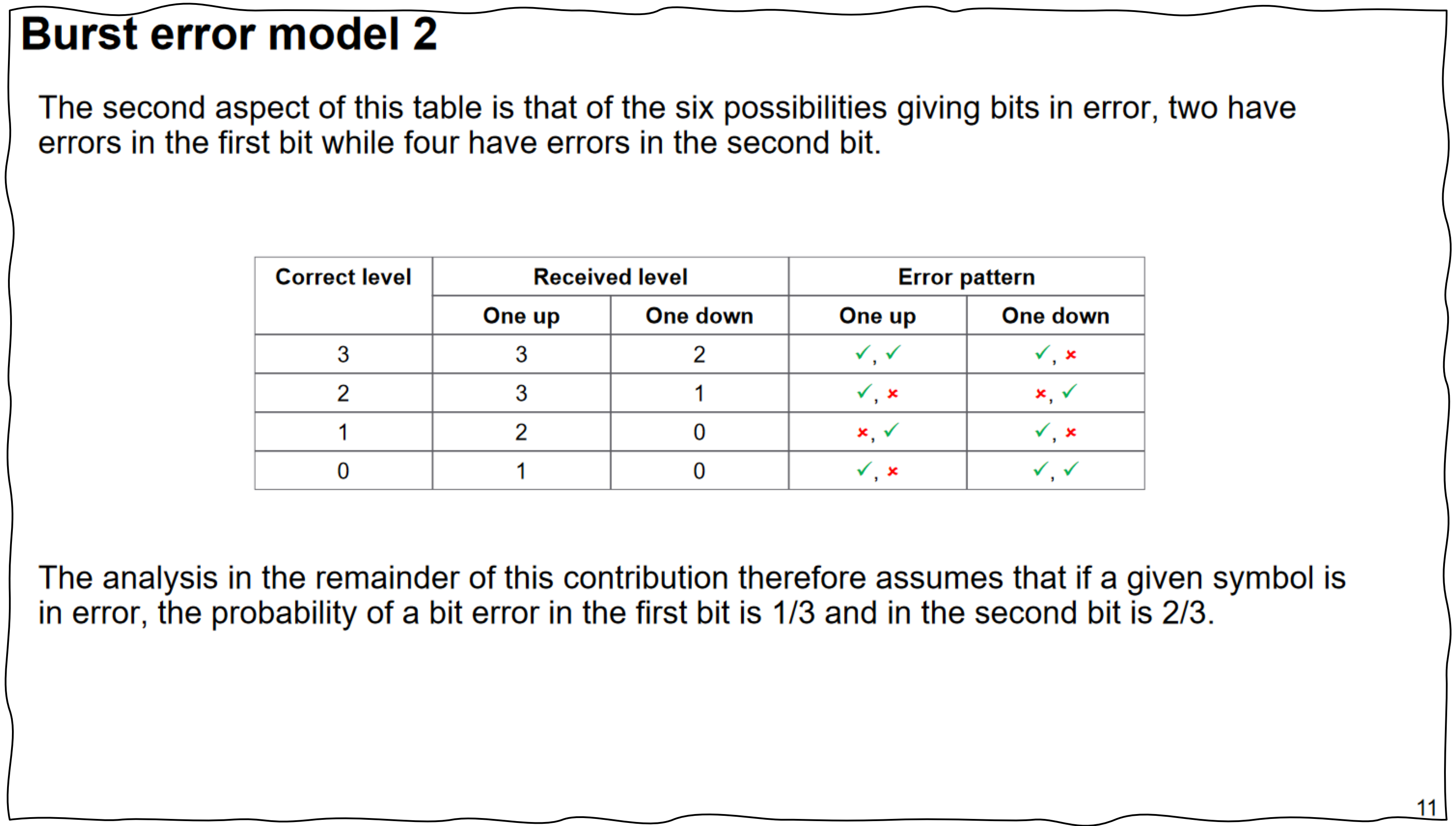
PCS checkerboard symbol pattern

Each PAM4 symbol contains two bits not from the same flow (in this example, one PAM4 symbol has AC and the other has BD; AD+BC also possible)

This muxing still adheres to the constraint

However, despite the checkerboard pattern, this **always allocates MSBs to two of the codewords** (here A and B) **and LSBs to the other codewords** (here C and D)

2/3 of random errors occur in the LSB



Burst error model 2

The second aspect of this table is that of the six possibilities giving bits in error, two have errors in the first bit while four have errors in the second bit.

Correct level	Received level		Error pattern	
	One up	One down	One up	One down
3	3	2	✓, ✓	✓, ×
2	3	1	✓, ×	×, ✓
1	2	0	×, ✓	✓, ×
0	1	0	✓, ×	✓, ✓

The analysis in the remainder of this contribution therefore assumes that if a given symbol is in error, the probability of a bit error in the first bit is 1/3 and in the second bit is 2/3.

11

Source: anslow_3ck_adhoc_01_072518

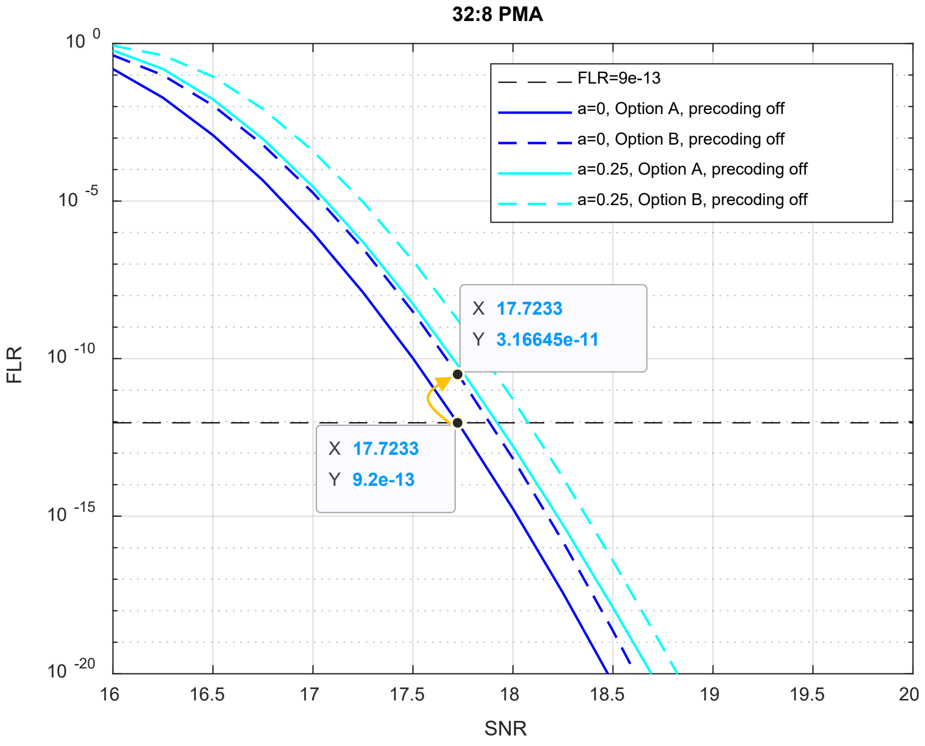
This means that, with option B:

- The two codewords that get the MSBs (A/B) have 2/3 of the average BER
- The two codewords that get the LSBs (C/D) have 4/3 of the average BER
- Uncorrectable errors occur more often in C and D
- Any uncorrectable error corrupts all four codewords

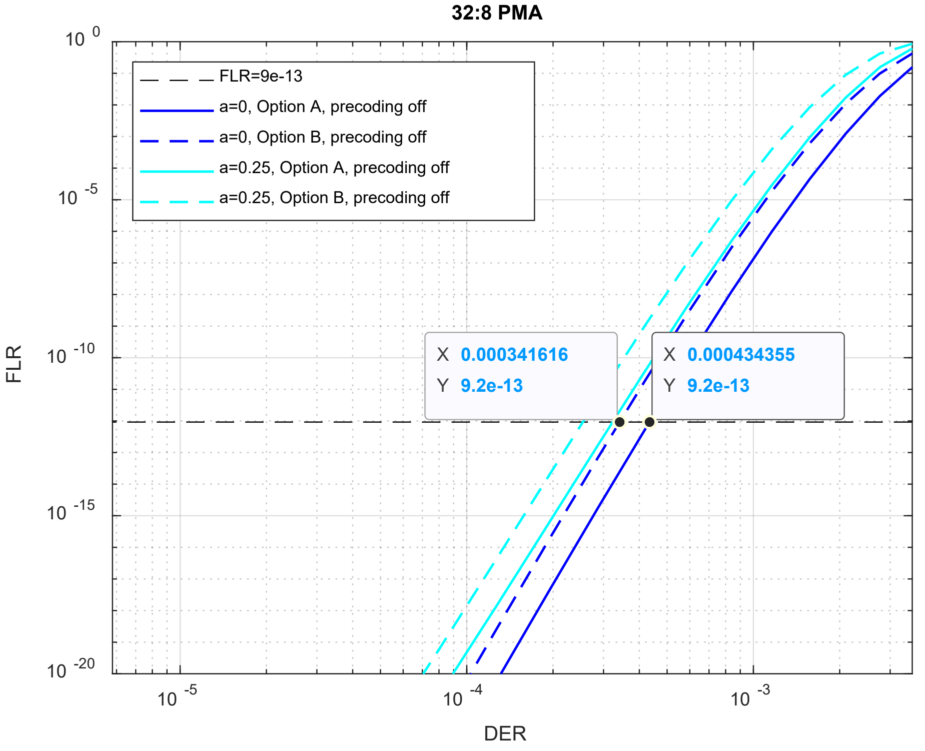
Note: if precoding is used, the decoding operation spreads errors equally across MSB and LSB, so this only applies to the non-precoded case

FLR effect of option B with low error correlation (optics or C2M)

Factor of 34 (1.5 orders of magnitude) increase in FLR, or ~0.15 dB penalty



Reduction of 25% in pre-FEC BER is required



Notes

- The FLR penalty is almost constant (1.5 order of magnitude) regardless of error correlation, so only shown at two values of *a*
    - Higher values of *a* will likely cause precoding to be used anyway
- “Option B” does not exist in any 200G/400GBASE-R PHY or AUI
    - The FLR effect (×34) is worse than that of having 4 instead of 2 codewords (×2)
    - Existing links will have a higher FLR at 800G (with option B) than at 200G/400G
- “Option B” may be susceptible to the “low clock content” issue because the LSBs always come from the same flow (and thus the same scrambler), as in clause 119
    - In “Option A” the LSBs alternate between two flows, which seems to solve the issue (the probability of having correlated outputs from two separate scramblers is negligible).
- A PMA(8:8) should be prevented from permuting PCSLs such that “option A” at the input is converted to “option B” at the output.

Suggested remedy (per comment #6)

173.4.2.1 32:8 PMA bit-level multiplexing

Change the second list item as shown:

– The multiplexing function has an additional constraint that each of the 8 output lanes contain two unique PCSLs from PMA client lanes *i* = 0 to 15 encoded as one PAM4 symbol, and two unique PCSLs from PMA client lanes *i* = 16 to 31 encoded as the subsequent PAM4 symbol (see 173.4.7).

173.4.2.2 8:32 PMA bit-level multiplexing

Change the second list item as shown:

– The multiplexing function has an additional constraint that each of the 8 output lanes contain two unique PCSLs from service interface lanes i = 0 to 15 encoded as one PAM4 symbol, and two unique PCSLs from service interface lanes i = 16 to 31 encoded as the subsequent PAM4 symbol (see 173.4.7).

173.4.2.3 8:8 PMA bit-level multiplexing

Change the second list item as shown:

– The 4 PCSLs received on any input lane shall be mapped together to an output lane, maintaining the bit pairs encoded on each PAM4 symbol. Other than that, The order of PCSLs from an input lane does not have to be maintained on the output lane.

Suggested remedy (modified) – part 1

173.4.2.1 32:8 PMA bit-level multiplexing

Change the second list item as shown:

– The multiplexing function has an additional constraint that each of the 8 output lanes contain two unique PCSLs from PMA client lanes *i* = 0 to 15 ~~and~~ followed by two unique PCSLs from PMA client lanes *i* = 16 to 31

173.4.2.2 8:32 PMA bit-level multiplexing

Change the second list item as shown:

– The multiplexing function has an additional constraint that each of the 8 output lanes contain two unique PCSLs from service interface lanes i = 0 to 15 ~~and~~ followed by two unique PCSLs from service interface lanes i = 16 to 31.

Suggested remedy (modified) – part 2

173.4.2.3 8:8 PMA bit-level multiplexing

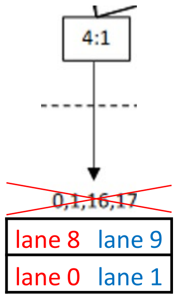
Change the second list item as shown:

– The 4 PCSLs received on any input lane shall be mapped together to an output lane such that the Gray-coded PAM4 symbol sequence on the output is identical to the Gray-coded PAM4 symbol sequence on the input (see 173.4.7.1). ~~The order of PCSLs from an input lane does not have to be maintained on the output lane.~~

Option X (unconstrained muxing)

Specific example of muxing lanes not according to the diagram in slide 2

UI\Lane	0	1	2	3	4	5	6	7
22	B81 A81	D81 C81	B91 A91	D91 C91	B101 A101	D101 C101	B111 A111	D111 C111
21	B120 A120	D120 C120	B130 A130	D130 C130	B140 A140	D140 C140	B150 A150	D150 C150
20	B80 A80	D80 C80	B90 A90	D90 C90	B100 A100	D100 C100	B110 A110	D110 C110
19	A49 B49	C49 D49	A59 B59	C59 D59	A69 B69	C69 D69	A79 B79	C79 D79
18	A9 B9	C9 D9	A19 B19	C19 D19	A29 B29	C29 D29	A39 B39	C39 D39
17	A48 B48	C48 D48	A58 B58	C58 D58	A68 B68	C68 D68	A78 B78	C78 D78
16	A8 B8	C8 D8	A18 B18	C18 D18	A28 B28	C28 D28	A38 B38	C38 D38
15	A47 B47	C47 D47	A57 B57	C57 D57	A67 B67	C67 D67	A77 B77	C77 D77
14	A7 B7	C7 D7	A17 B17	C17 D17	A27 B27	C27 D27	A37 B37	C37 D37
13	A46 B46	C46 D46	A56 B56	C56 D56	A66 B66	C66 D66	A76 B76	C76 D76
12	A6 B6	C6 D6	A16 B16	C16 D16	A26 B26	C26 D26	A36 B36	C36 D36
11	A45 B45	C45 D45	A55 B55	C55 D55	A65 B65	C65 D65	A75 B75	C75 D75
10	A5 B5	C5 D5	A15 B15	C15 D15	A25 B25	C25 D25	A35 B35	C35 D35
9	A44 B44	C44 D44	A54 B54	C54 D54	A64 B64	C64 D64	A74 B74	C74 D74
8	A4 B4	C4 D4	A14 B14	C14 D14	A24 B24	C24 D24	A34 B34	C34 D34
7	A43 B43	C43 D43	A53 B53	C53 D53	A63 B63	C63 D63	A73 B73	C73 D73
6	A3 B3	C3 D3	A13 B13	C13 D13	A23 B23	C23 D23	A33 B33	C33 D33
5	A42 B42	C42 D42	A52 B52	C52 D52	A62 B62	C62 D62	A72 B72	C72 D72
4	A2 B2	C2 D2	A12 B12	C12 D12	A22 B22	C22 D22	A32 B32	C32 D32
3	A41 B41	C41 D41	A51 B51	C51 D51	A61 B61	C61 D61	A71 B71	C71 D71
2	A1 B1	C1 D1	A11 B11	C11 D11	A21 B21	C21 D21	A31 B31	C31 D31
1	A40 B40	C40 D40	A50 B50	C50 D50	A60 B60	C60 D60	A70 B70	C70 D70
0	A0 B0	C0 D0	A10 B10	C10 D10	A20 B20	C20 D20	A30 B30	C30 D30



PCS checkerboard symbol pattern

Each PAM4 symbol contains two bits from the same flow, but each lane has only one flow.

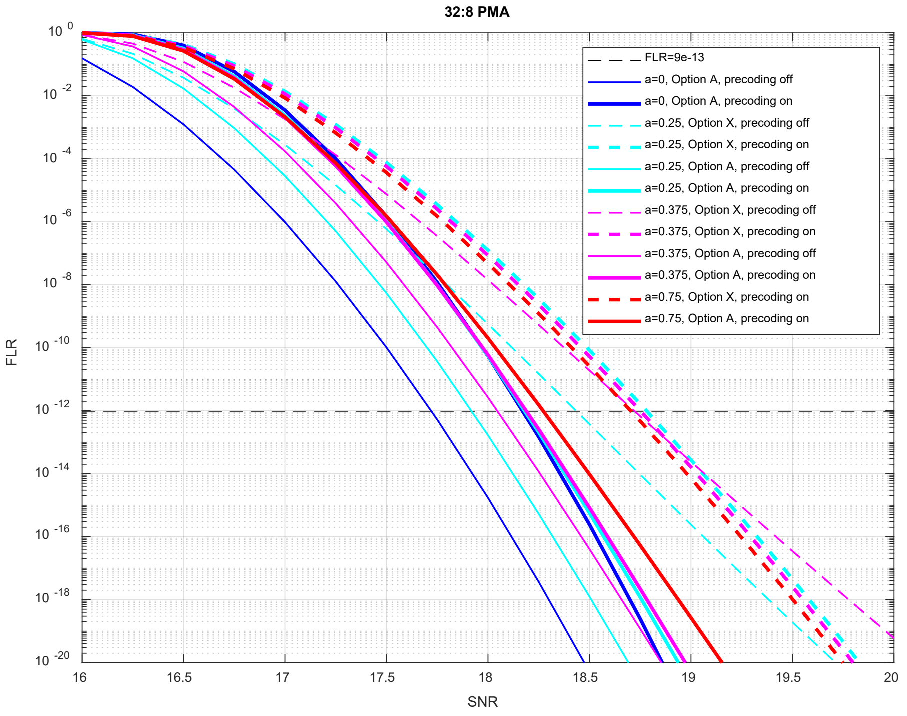
This muxing does not adhere to the constraint.

Short bursts can corrupt 4 symbols in the same codeword

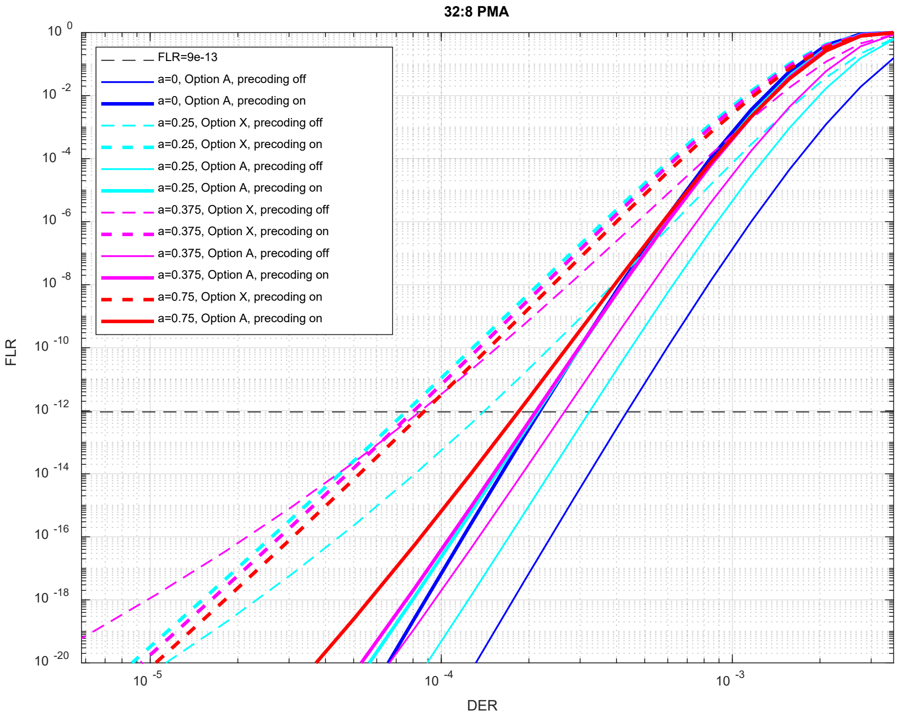
Immunity to correlated errors is worse than in 200G/400G

Effect of option X
(Compare dashed lines to solid lines)

Large SNR penalty for any *a*>0



Large minimum DER effect for any *a*>0



Summary

- The constraints on lane muxing should be retained.
- Additional constraints are proposed to prevent degraded performance due to bad muxing choice.