100G-CR1/KR1 PCS, FEC and PMA Baseline Proposal

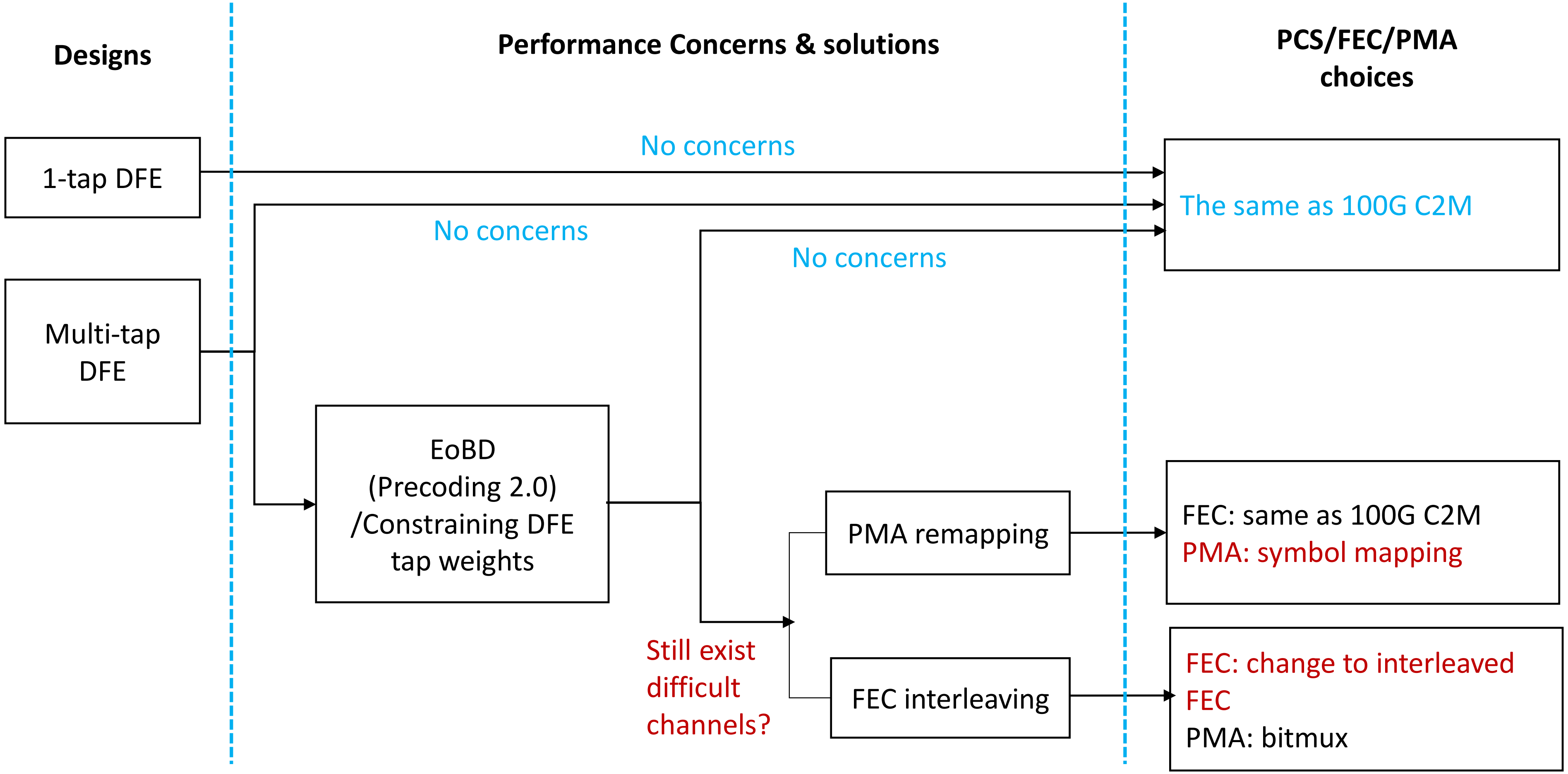
Yan Zhuang, Huawei
Yuchun Lu, Huawei

IEEE P802.3ck Task Force Meeting, Salt Lake, May 2019

Background

- FEC performance concern for 100GE-CR1/KR1 multi-tap DFEs with 4:1 bitmux PMA was shown and interleaved FEC was proposed in gustlin_3ck_01_1118.
- Interleaved FEC will introduce more latency and complicated CDR is needed to address the interoperability and compatibility issues. Both the latency and the complicated CDR are not affordable in some applications. In-depth analysis was given lu_3ck_adhoc_01_022719.
- Further, analysis of potential solutions for 100G-CR1/KR1 multi-tap DFE error propagation was given in lu_3ck_02_0319 including PMD, PMA and FEC sublayer solutions.
- Constraining DFE weights for multi-tap DFE is also investigated in lyubomirsky_3ck_01a_0319.
- Detailed proposal of a PMA solution which introduces a new optional AUI extender sublayer to implement PMA remapping was discussed in lu_3ck_adhoc_01_041019.
- With this, this presentation is a baseline proposal for 100G-CR1/KR1 PCS, FEC and PMA options, including no changes for designs with DFE constrains or EoBD (lu_3ck_01_0319) and an optional PMA remapping function to support symbol mapping if there exists difficult channels of multi-tap DFE.

Decision tree for the KR/CR PCS/FEC/PMA choice



Designs

Performance Concerns & solutions

PCS/FEC/PMA choices

1-tap DFE

Multi-tap DFE

No concerns

No concerns

No concerns

EoBD (Precoding 2.0) /Constraining DFE tap weights

Still exist difficult channels?

PMA remapping

FEC interleaving

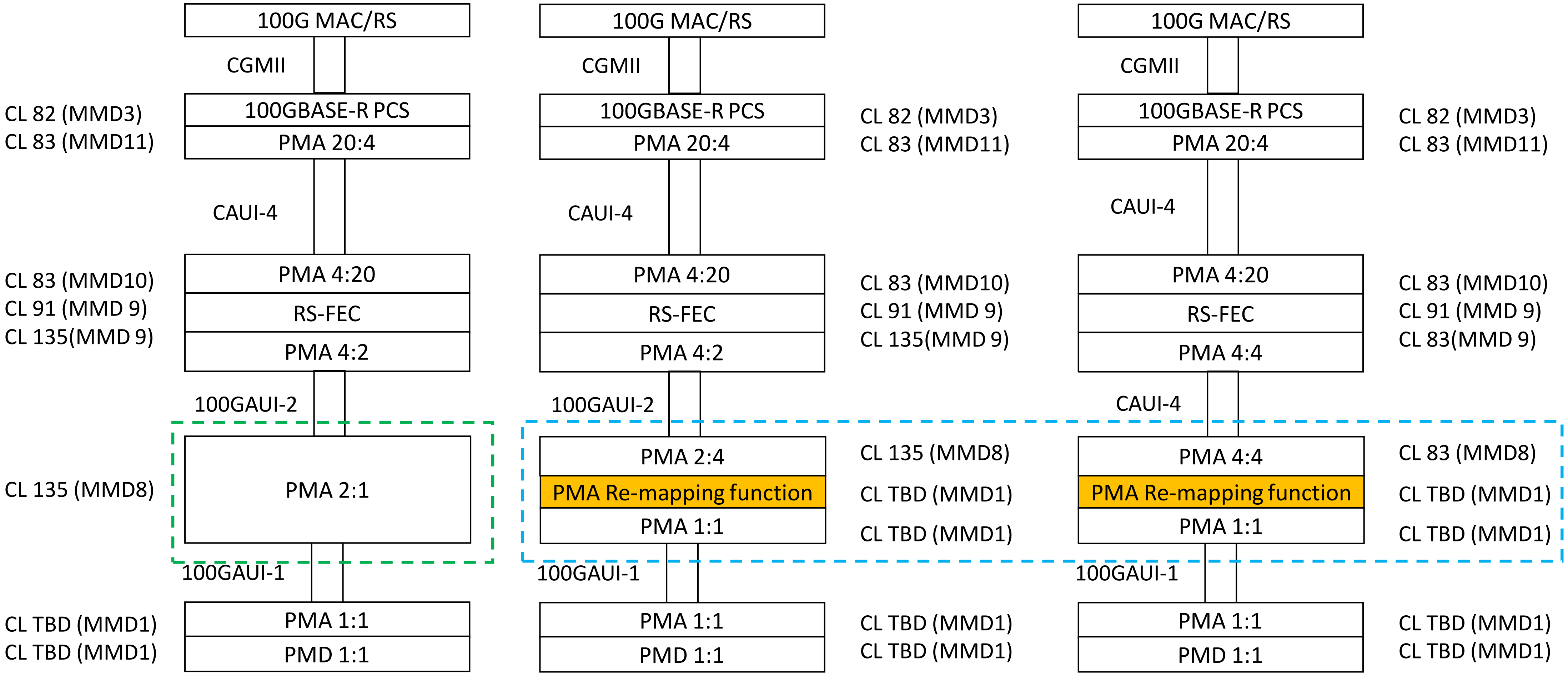
The same as 100G C2M

FEC: same as 100G C2M
PMA: symbol mapping

FEC: change to interleaved FEC
PMA: bitmux

System concerns should be also considered.

Examples of PMA remap function for 100GBASE-CR1/KR1



PMAs without re-mapping
The same as C2M

Add an optional PMA remapping function for difficult channels of multi-tap DFE.

RS/MII Baseline

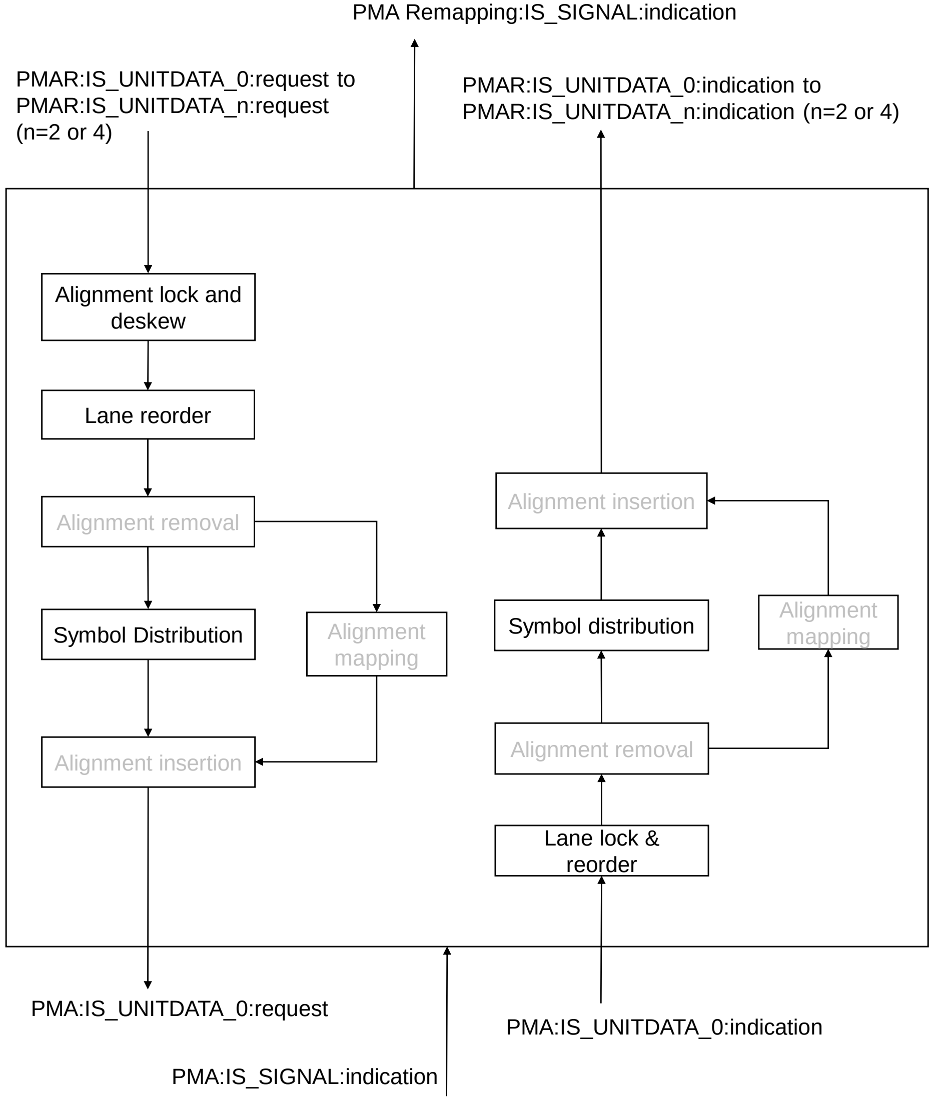
- 100G RS and MII are already defined in Clause 81.

Overview of 100G-CR1/KR1 PCS, FEC & PMA with PMA re-mapping

- PCS
  - Re-use existing 100GbE(Clause 82) PCS
  - No changes proposed.
- FEC
  - Re-use existing 802.3bj RS(544,514) FEC (Clause 91)
  - No new Alignment Markers (AM) are needed to be defined.
  - No changes proposed.
- PMA
  - Leverage with CL135.
  - Add an optional PMA remapping function

  "PMA remapping": Reverse 2 lane bitmux to 4 lane FEC symbols and map FEC symbols to a single lane.

PMA Re-mapping Functional Block Diagram

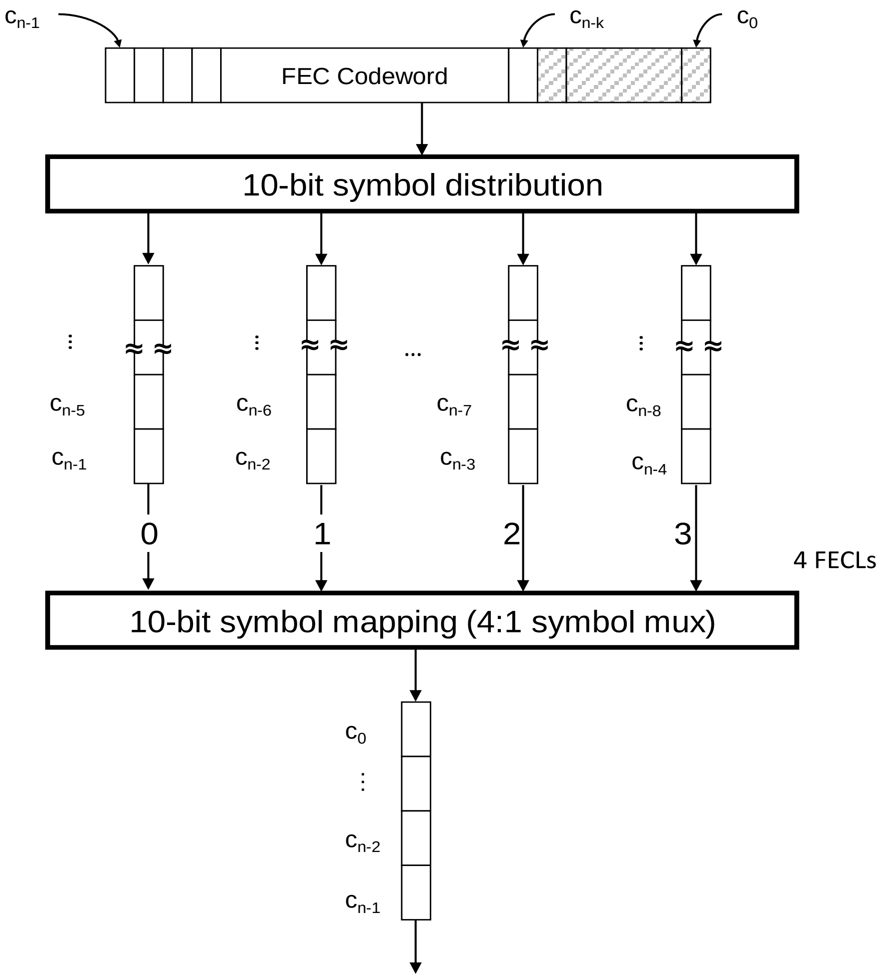

PMA Remapping:IS_SIGNAL:indication
PMAR:IS_UNITDATA_0:request to PMAR:IS_UNITDATA_n:request (n=2 or 4)
PMAR:IS_UNITDATA_0:indication to PMAR:IS_UNITDATA_n:indication (n=2 or 4)
Alignment lock and deskew
Lane reorder
Alignment removal
Symbol Distribution
Alignment mapping
Alignment insertion
Alignment insertion
Symbol distribution
Alignment mapping
Alignment removal
Lane lock & reorder
PMA:IS_UNITDATA_0:request
PMA:IS_UNITDATA_0:indication
PMA:IS_SIGNAL:indication

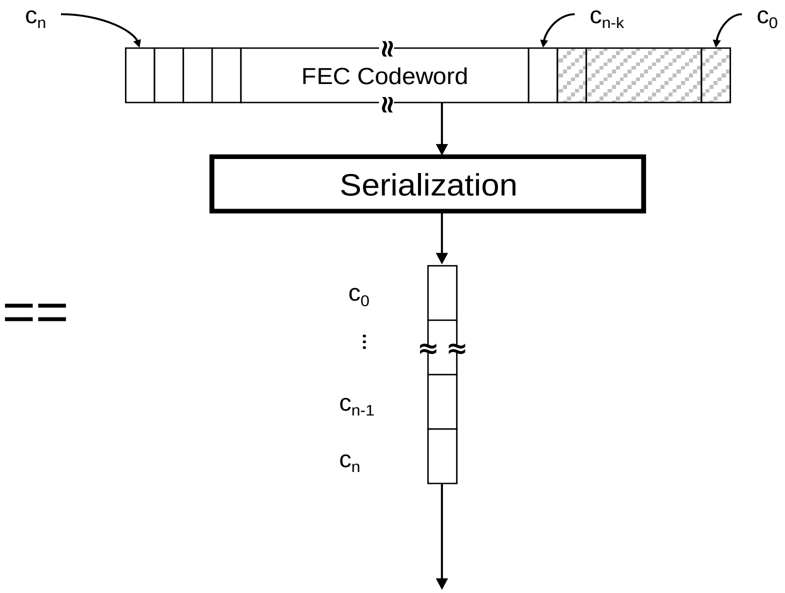
- All the function blocks and procedure can reuse Clause 91. No new functions will be introduced.

- Alignment mapping is actually not needed.

- No new alignment markers are needed to be defined. The remapping Alignment Markers can be reused as new Alignment Markers, or the FEC can be self-synchronized.

Symbol mapping relationships





“Keep all the FECLs in order and use 4:1 symbol mux.” is equivalent to direct “symbol mapping” from the FEC codeword to 1 PMD lane.

If we choose to work with 4 FECLs directly, the symbol mapping is the only thing we should add.

Baseline Options

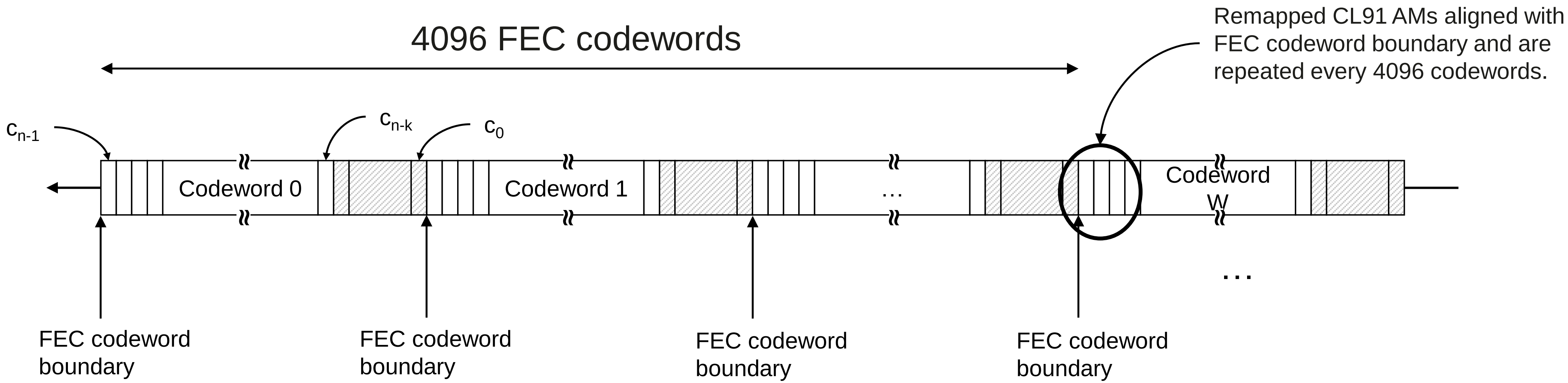
- Option 1: no performance concerns for 1-tap DFE and multi-tap DFE (if needed) with constrained weights/EoBD.
  - Adopt Clause 82 as the PCS, Clause 91 as the FEC, and Clause 135 as the PMA for 100Gb/s Attachment Unit interface C2C for this project. The same as 100Gb/s AUI C2M.
- Option 2: if there exist minor difficult channels for multi-tap DFE solutions even with constrained DFE taps/EoBD
  - Adopt Clause 82 as the PCS, Clause 91 as the FEC, and Clause 135 as the PMA for 100Gb/s Attachment Unit interface C2C for this project.
  - Adopt optional PMA remapping function as defined on page 7 for 100Gb/s AUI C2C for this project.

Questions?

Thank you!

FEC self-synchronization and Alignment Markers-1

- No new Alignment Markers (AM) are needed to be defined. Two ways to achieve alignment
  1. The RS(544, 514) can be self-synchronized.



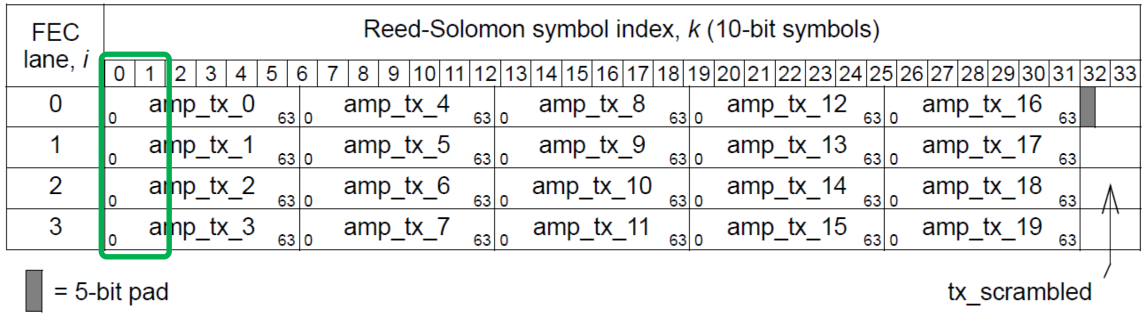
The CL91 Alignment Markers are aligned with RS(544, 514) FEC boundary. As long as the FEC boundary is founded by the self-synchronization algorithm, the FECLs can be easily recovered.

FEC self-synchronization and Alignment Markers-2

- No new Alignment Markers (AM) are needed to be defined. Two ways to achieve alignment

    2. Reuse the remapped alignment markers of Clause 91.

amp_tx_x={M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}



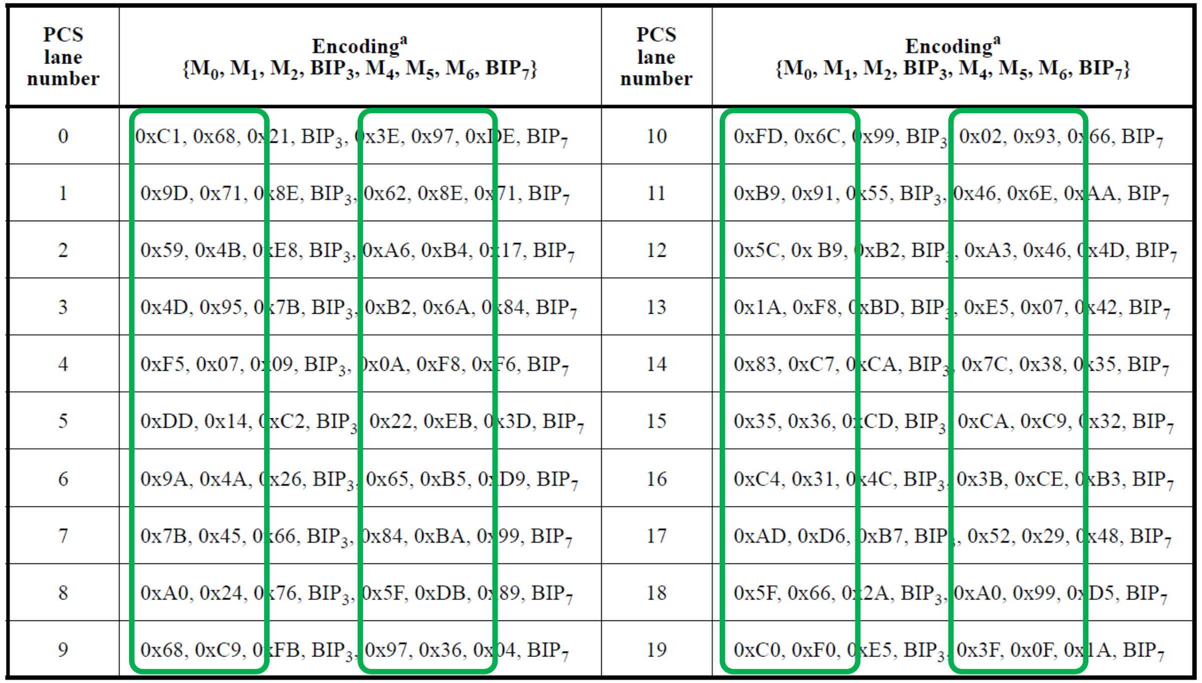
Reed-Solomon symbol index, *k* (10-bit symbols): 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

FEC lane, *i*						
0	0 amp_tx_0 63	0 amp_tx_4 63	0 amp_tx_8 63	0 amp_tx_12 63	0 amp_tx_16 63	▮
1	0 amp_tx_1 63	0 amp_tx_5 63	0 amp_tx_9 63	0 amp_tx_13 63	0 amp_tx_17 63	
2	0 amp_tx_2 63	0 amp_tx_6 63	0 amp_tx_10 63	0 amp_tx_14 63	0 amp_tx_18 63	
3	0 amp_tx_3 63	0 amp_tx_7 63	0 amp_tx_11 63	0 amp_tx_15 63	0 amp_tx_19 63	

▮ = 5-bit pad

Figure 91–4—Alignment marker mapping to FEC lanes

Table 82–2—100GBASE-R Alignment marker encodings



PCS lane number	Encoding[a] {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}	PCS lane number	Encoding[a] {M_0, M_1, M_2, BIP_3, M_4, M_5, M_6, BIP_7}
0	0xC1, 0x68, 0x21, BIP_3, 0x3E, 0x97, 0xDE, BIP_7	10	0xFD, 0x6C, 0x99, BIP_3, 0x02, 0x93, 0x66, BIP_7
1	0x9D, 0x71, 0x8E, BIP_3, 0x62, 0x8E, 0x71, BIP_7	11	0xB9, 0x91, 0x55, BIP_3, 0x46, 0x6E, 0xAA, BIP_7
2	0x59, 0x4B, 0xE8, BIP_3, 0xA6, 0xB4, 0x17, BIP_7	12	0x5C, 0x B9, 0xB2, BIP_3, 0xA3, 0x46, 0x4D, BIP_7
3	0x4D, 0x95, 0x7B, BIP_3, 0xB2, 0x6A, 0x84, BIP_7	13	0x1A, 0xF8, 0xBD, BIP_3, 0xE5, 0x07, 0x42, BIP_7
4	0xF5, 0x07, 0x09, BIP_3, 0x0A, 0xF8, 0xF6, BIP_7	14	0x83, 0xC7, 0xCA, BIP_3, 0x7C, 0x38, 0x35, BIP_7
5	0xDD, 0x14, 0xC2, BIP_3, 0x22, 0xEB, 0x3D, BIP_7	15	0x35, 0x36, 0xCD, BIP_3, 0xCA, 0xC9, 0x32, BIP_7
6	0x9A, 0x4A, 0x26, BIP_3, 0x65, 0xB5, 0xD9, BIP_7	16	0xC4, 0x31, 0x4C, BIP_3, 0x3B, 0xCE, 0xB3, BIP_7
7	0x7B, 0x45, 0x66, BIP_3, 0x84, 0xBA, 0x99, BIP_7	17	0xAD, 0xD6, 0xB7, BIP_3, 0x52, 0x29, 0x48, BIP_7
8	0xA0, 0x24, 0x76, BIP_3, 0x5F, 0xDB, 0x89, BIP_7	18	0x5F, 0x66, 0x2A, BIP_3, 0xA0, 0x99, 0xD5, BIP_7
9	0x68, 0xC9, 0xFB, BIP_3, 0x97, 0x36, 0x04, BIP_7	19	0xC0, 0xF0, 0xE5, BIP_3, 0x3F, 0x0F, 0x1A, BIP_7

[a] Each octet is transmitted LSB to MSB.

An example of 80bits remapped AM is

amp_tx_0{ 9: 0}, amp_tx_1{ 9: 0}, amp_tx_2{ 9: 0}, amp_tx_3{ 9: 0}, amp_tx_0{19:10}, amp_tx_1{19:10}, amp_tx_2{19:10}, amp_tx_3{19:10}.

These bits are aligned with RS(544, 514) FEC codeword and repeat every 4096 FEC code words.

More bits are available for alignment except for the BIP bits.