Transmitter and Receiver Architecture Without Self-Synchronizing Rx Scrambler

IEEE 802.3bj interim meeting

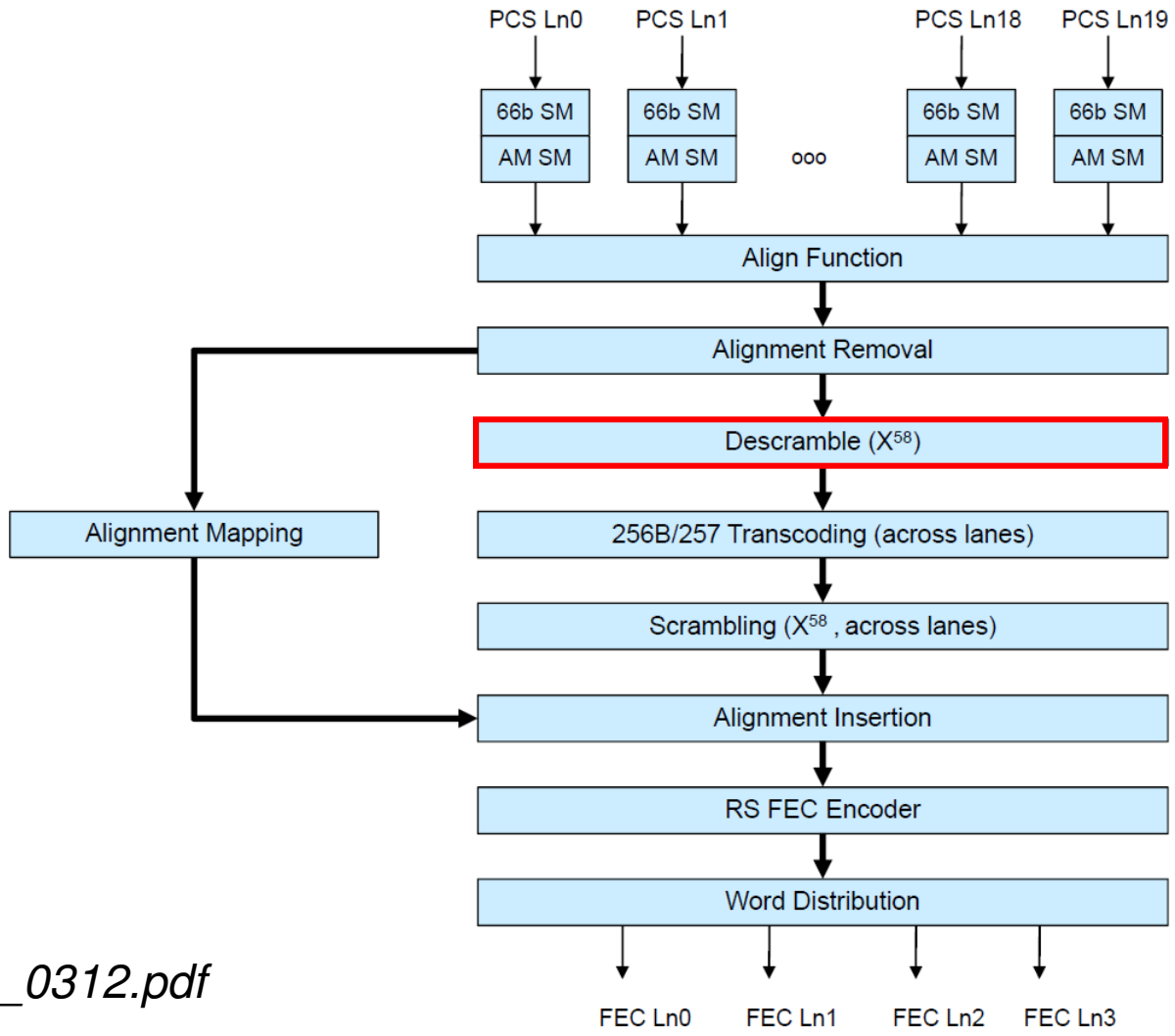
May 17-18, 2012, Minneapolis, MN

Roy Cideciyan - IBM

Introduction

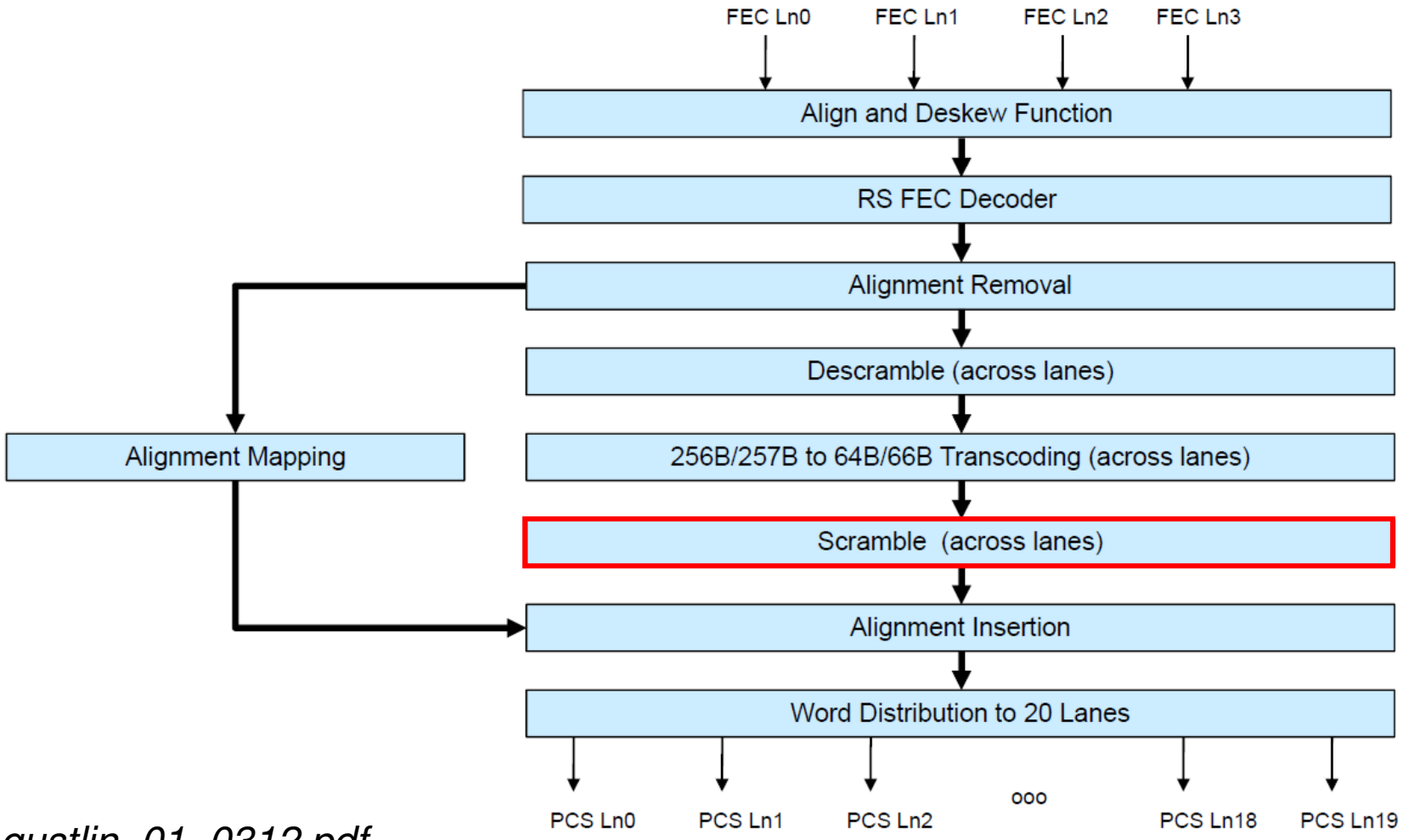
- Tx and Rx architectures for low-latency FEC have been adopted at the March plenary meeting: *gustlin_01_0312.pdf* and *brown_01a_0312.pdf*
- High-rate transcoding is performed on non-scrambled data. As data from PCS layer is already scrambled with a self-synchronizing scrambler, data in current Tx architecture is descrambled prior to 256b/257b transcoding.
- Descrambling in Tx requires the inverse operation at the receiver, i.e., data is scrambled with a self-synchronizing scrambler at Rx before it is sent to physical coding sublayer (PCS) where it is descrambled.
- Self-synchronizing scramblers in Rx are usually not desirable because of error multiplication. Although PCS descrambler should regenerate the possibly erroneous sequence at the Rx scrambler input, it is worthwhile to think whether a low-latency FEC architecture that avoids Rx scrambler is possible.
- A proposal for FEC architecture that avoids self-synchronizing Rx scrambler is shown.

Adopted Tx Architecture for NRZ



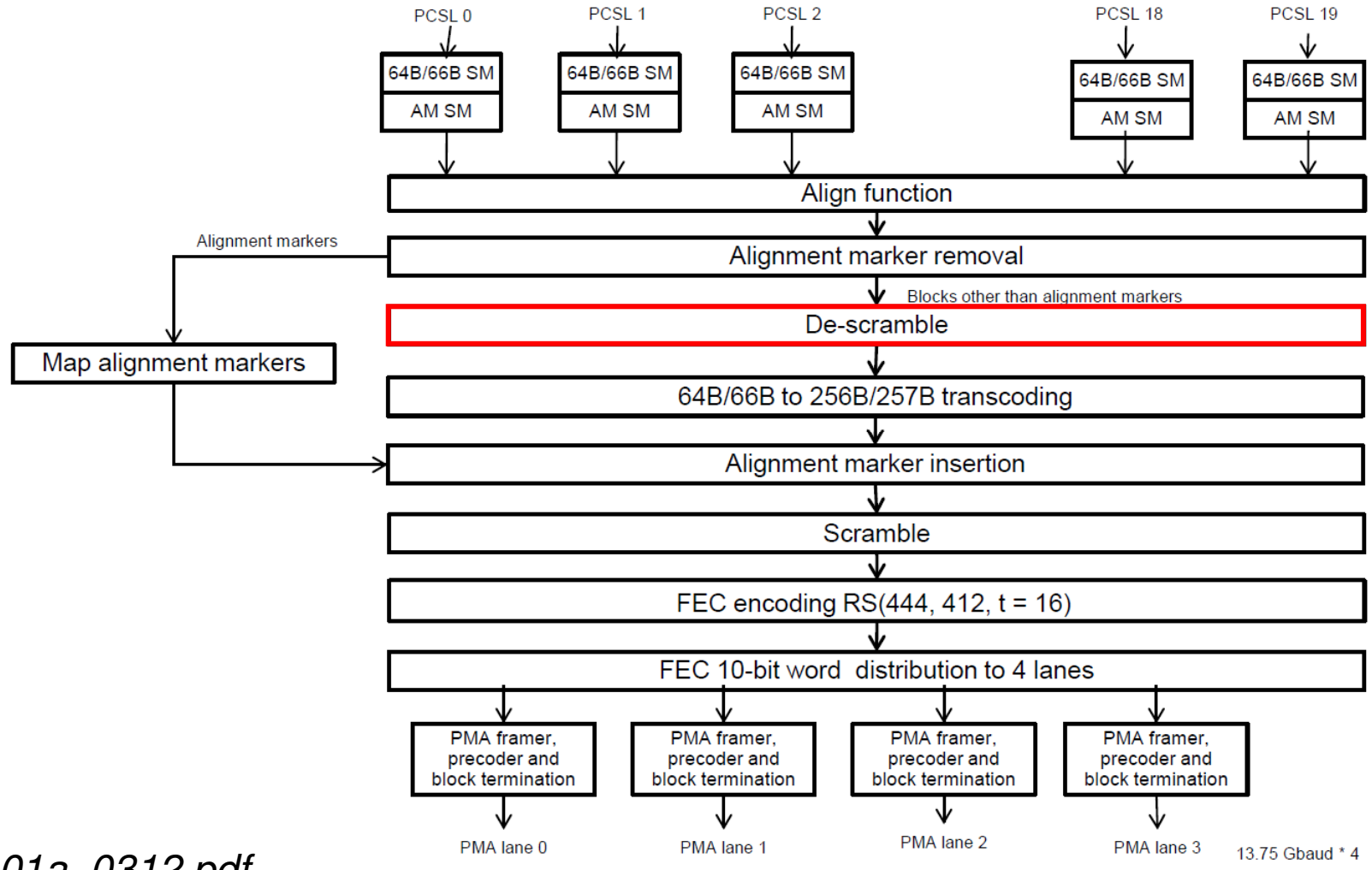
gustlin_01_0312.pdf

Adopted Rx Architecture for NRZ



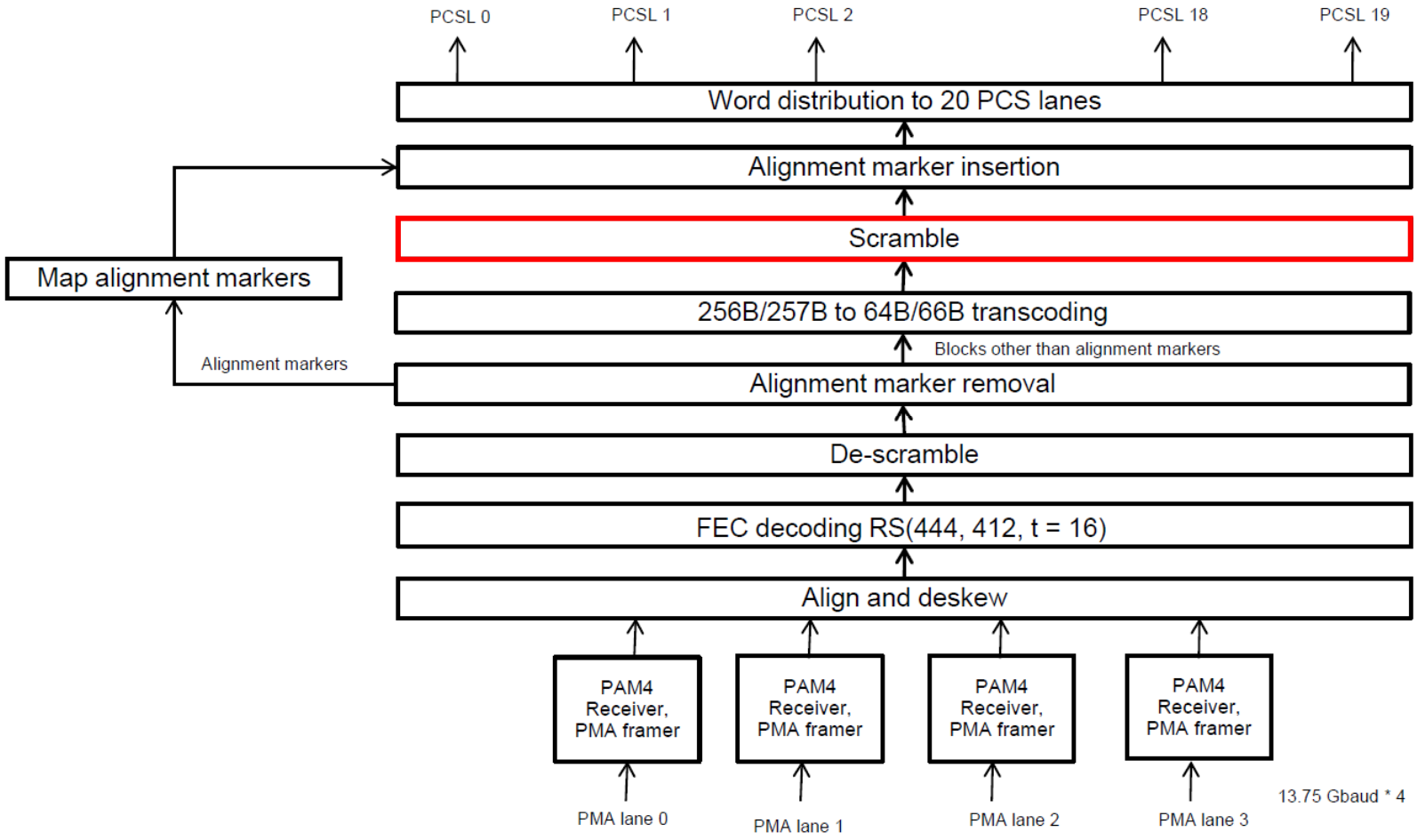
gustlin_01_0312.pdf

Adopted Tx Architecture for PAM4



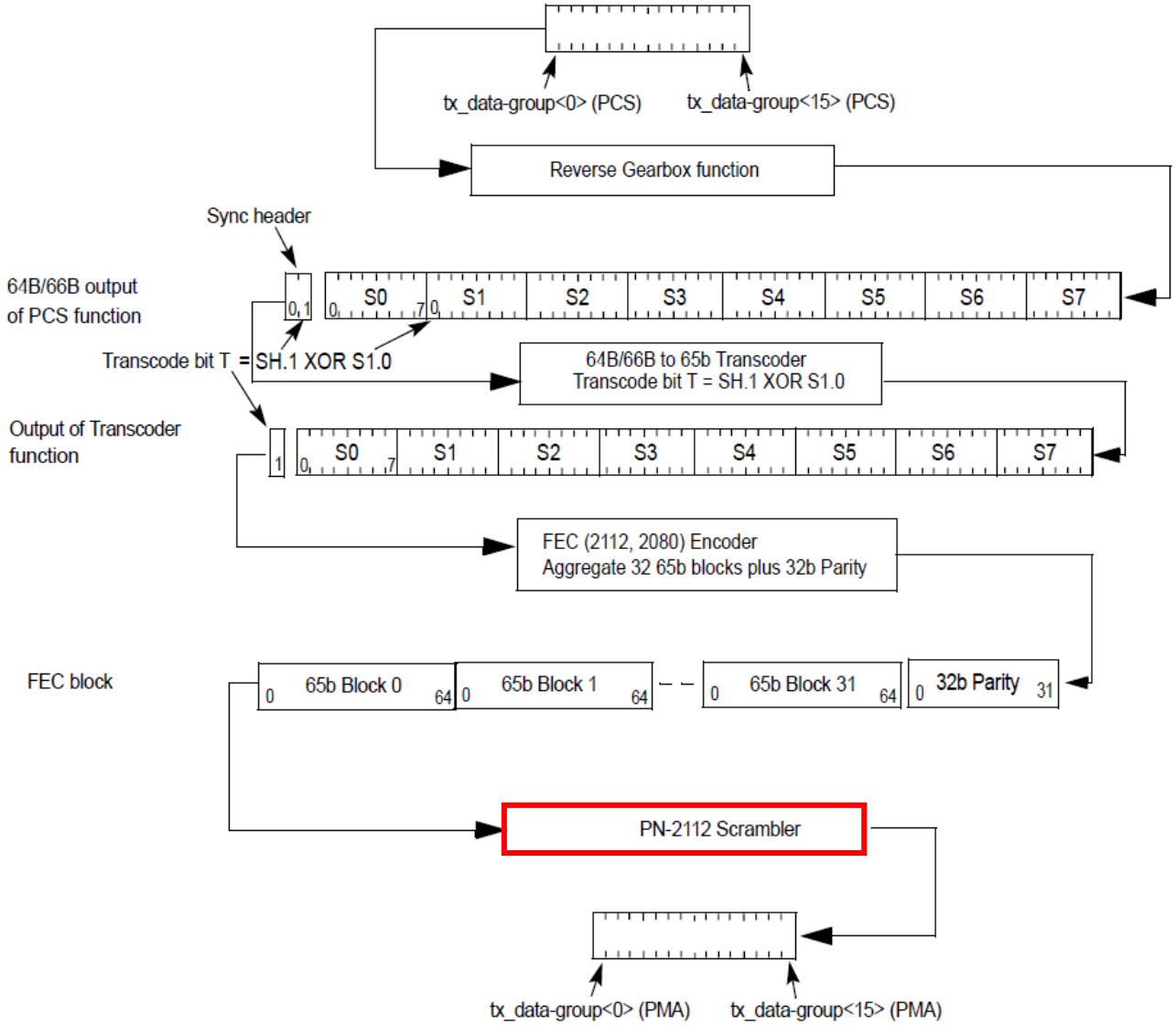
brown_01a_0312.pdf

Adopted Rx Architecture for PAM4



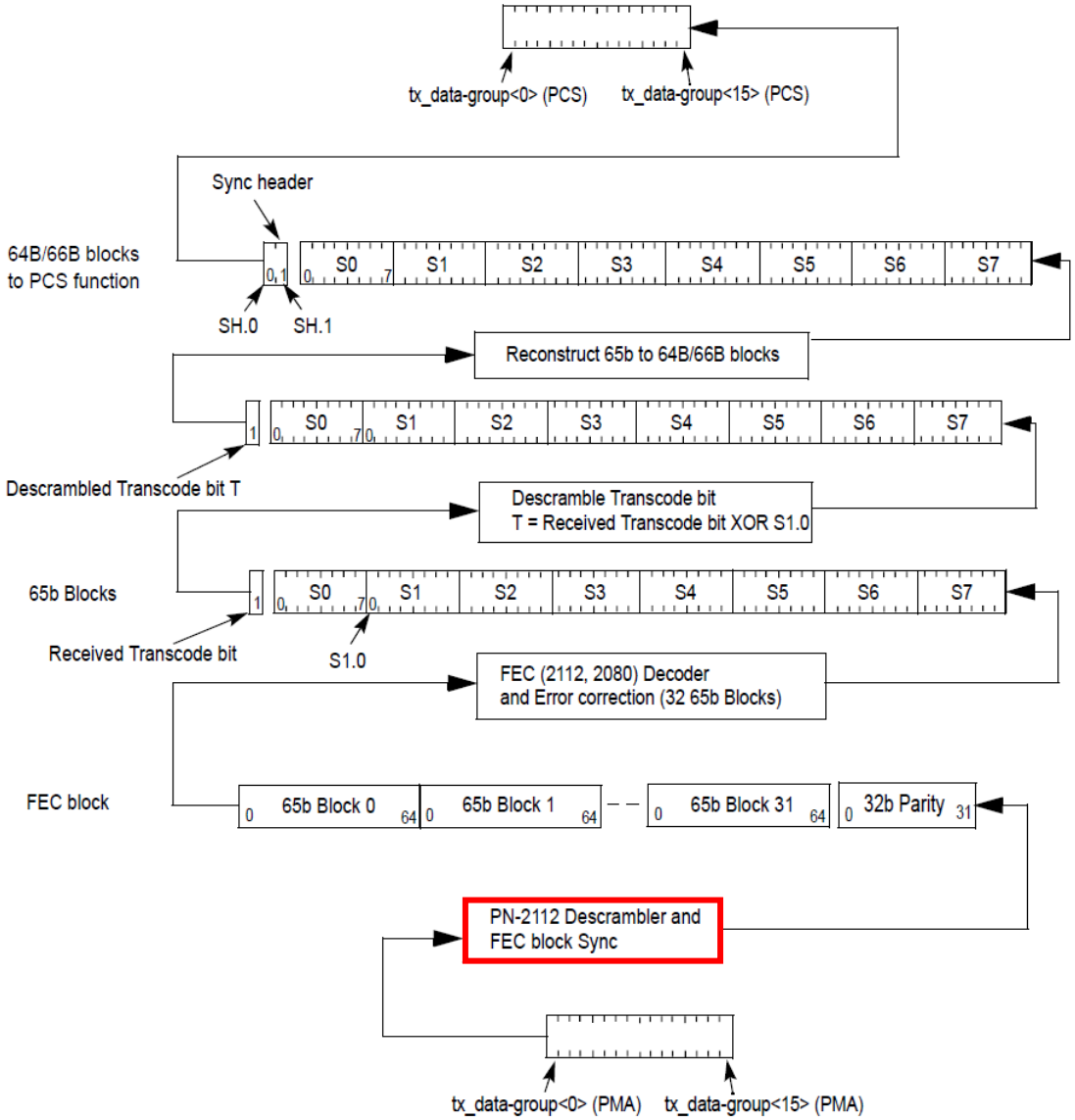
brown_01a_0312.pdf

Tx Architecture of FEC Sublayer for 10GBASE-R



Clause 74

Rx Architecture of FEC Sublayer for 10GBASE-R



Clause 74

64b/66b Coding in 100GBASE-R

- 64b/66b coding used in 100GBASE-R (IEEE 802.3ba-2010, Clause 82)
  - 1 type of data block (DB) with 2-bit header **01**
  - 11 types of control blocks (CB) with 2-bit header **10** where the first byte of the payload is rate-4/8 encoded (Hamming distance=4) **8-bit block type field** indicating the type of control block format

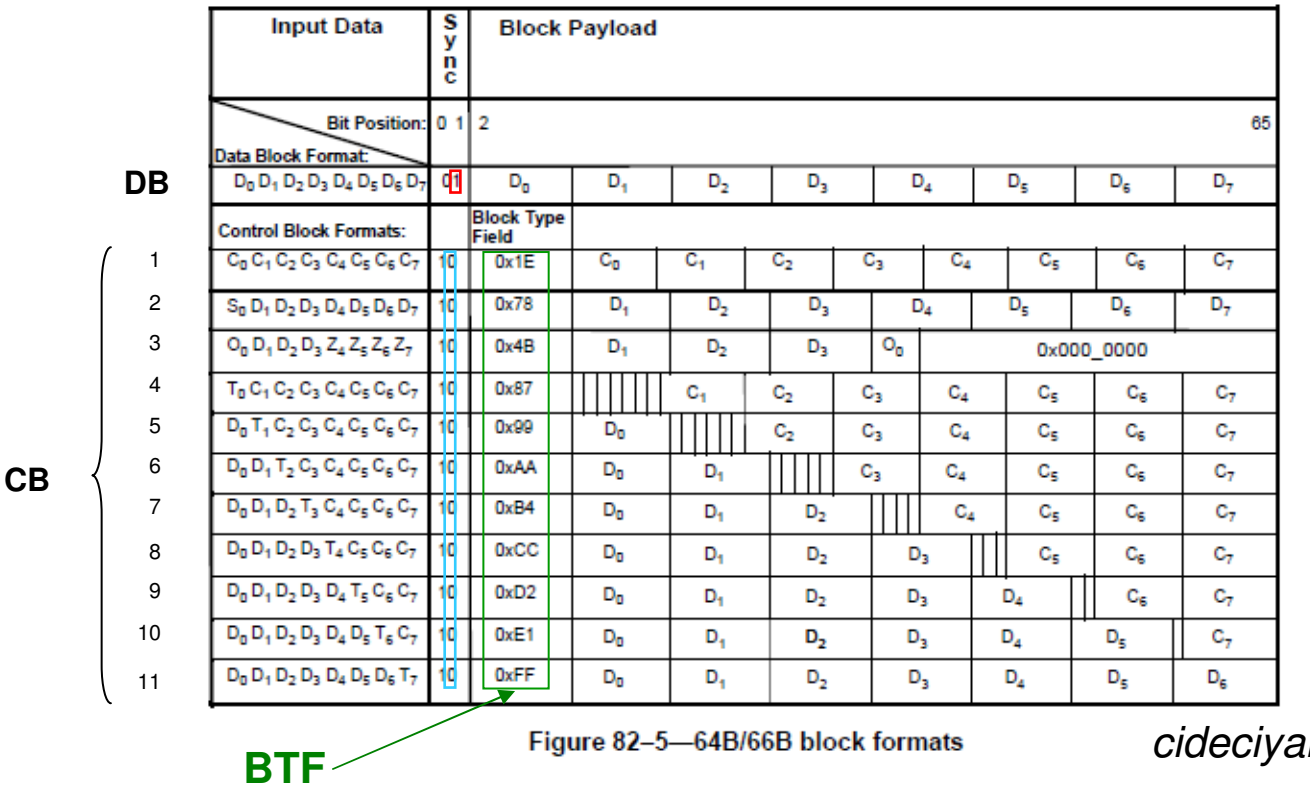
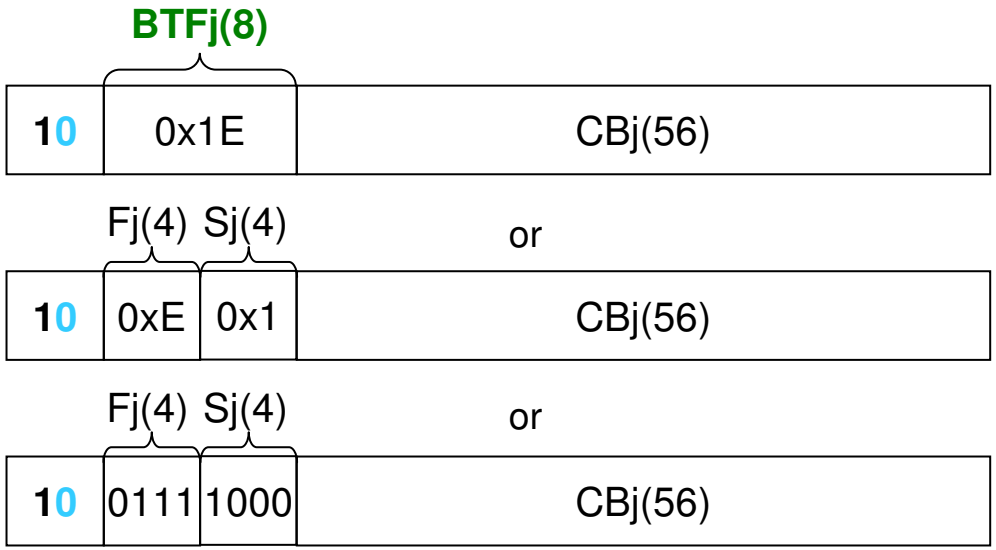


Figure 82–5—64B/66B block formats

cideciyan_01a_0312.pdf

Block Type Field

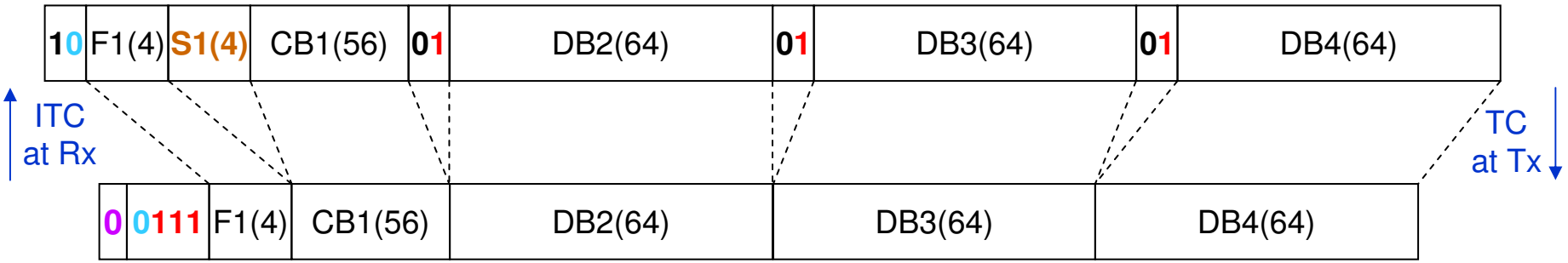
- According to clause 82.2.3.1 the LSB of the block type field (BTF) represented as a hexadecimal value is the first transmitted bit. For example, the block type field 0x1E is sent from left to right as 01111000.
- BTFj(8)=BTFj<7:0> where BTFj<0> is the first transmitted bit. We represent BTFj(8) as the concatenation of the first nibble Fj(4)=BTFj<3:0> and the second nibble Sj(4)=BTFj<7:4>. For example, for BTFj(8)=0x1E, we obtain Fj(4)=0xE sent from left to right as 0111 and Sj(4)=0x1 sent from left to right as 1000.



cideciyan_01a_0312.pdf

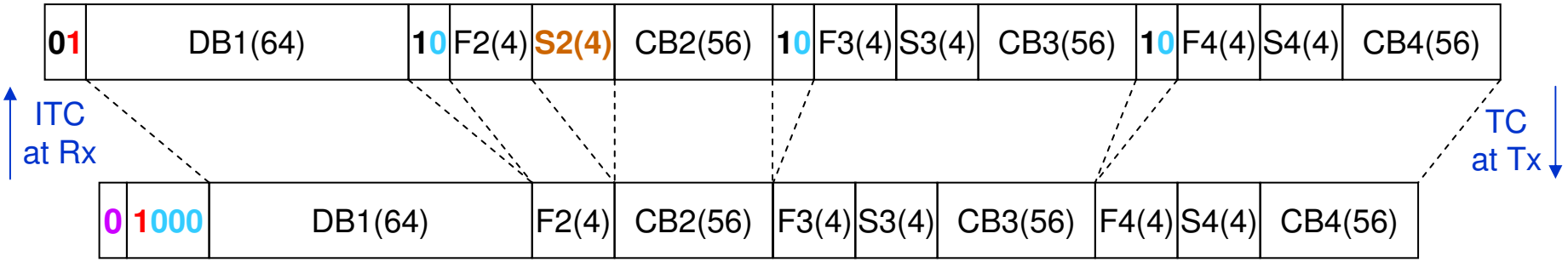
Two Examples of 256b/257b Transcoding

Case 8: CB #1, DB #2, DB #3 and DB #4



TC: Transcoding
ITC: Inverse Transcoding

Case 9: DB #1, CB #2, CB #3 and CB #4



cideciyan_01a_0312.pdf

TC and ITC of Scrambled Data

- Transcoding (TC) operation of scrambled 66b blocks from PCS sublayer can be performed exactly the same way as TC of descrambled 66b blocks by performing the same operations described in *cideciyan_01a_0312.pdf*, e.g., deleting the scrambled second 4-bit nibble in the first scrambled byte of the first scrambled control block in a transcoded block that contains at least one control block. Note that two header bits of 66b blocks from PCS are not scrambled; only the 64-bit payload is scrambled.
- Inverse transcoding (ITC) of descrambled data according to *cideciyan_01a_0312.pdf* regenerates second nibble of block type field, which has been deleted during TC, by using a one-to-one mapping between first nibbles (1st) and second nibbles (2nd) of BTF. This is only possible because BTF is a codeword from (8,4) Hamming code with Hamming distance 4.

One-to-one mapping between 1st and 2nd nibbles of BTFs in hexadecimal notation

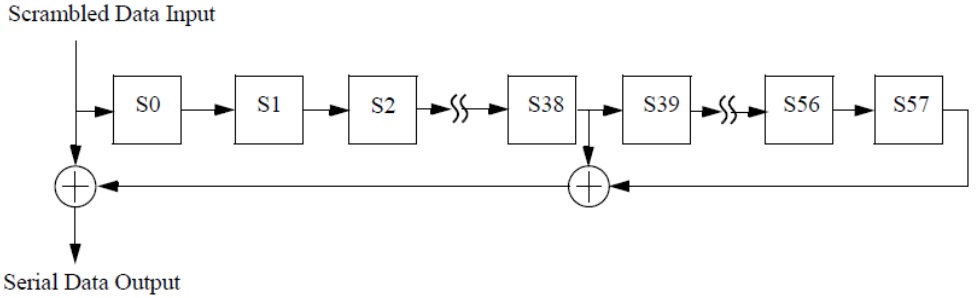
1st	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2nd	0	E	D	3	B	5	6	8	7	9	A	4	C	2	1	F

11 BTFs used in 100GBASE-R

4 additional BTFs used in 10GBASE-R

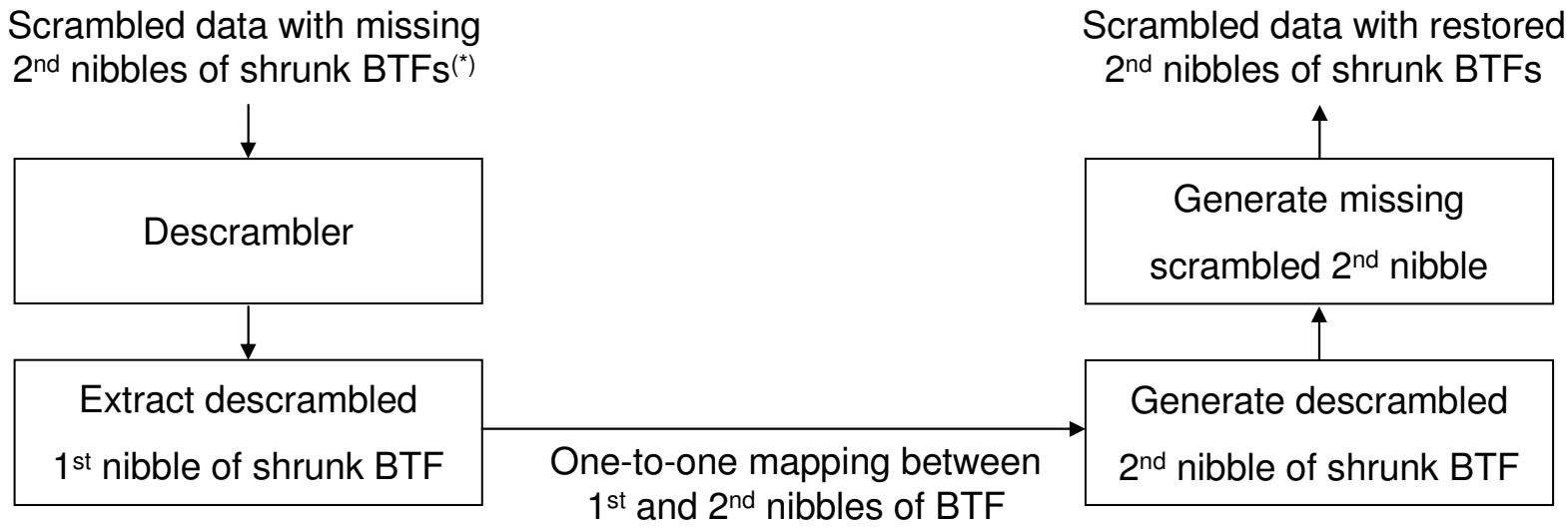
- ITC of scrambled data requires a descrambler running in parallel

Descrambler-Aided ITC of Scrambled Data



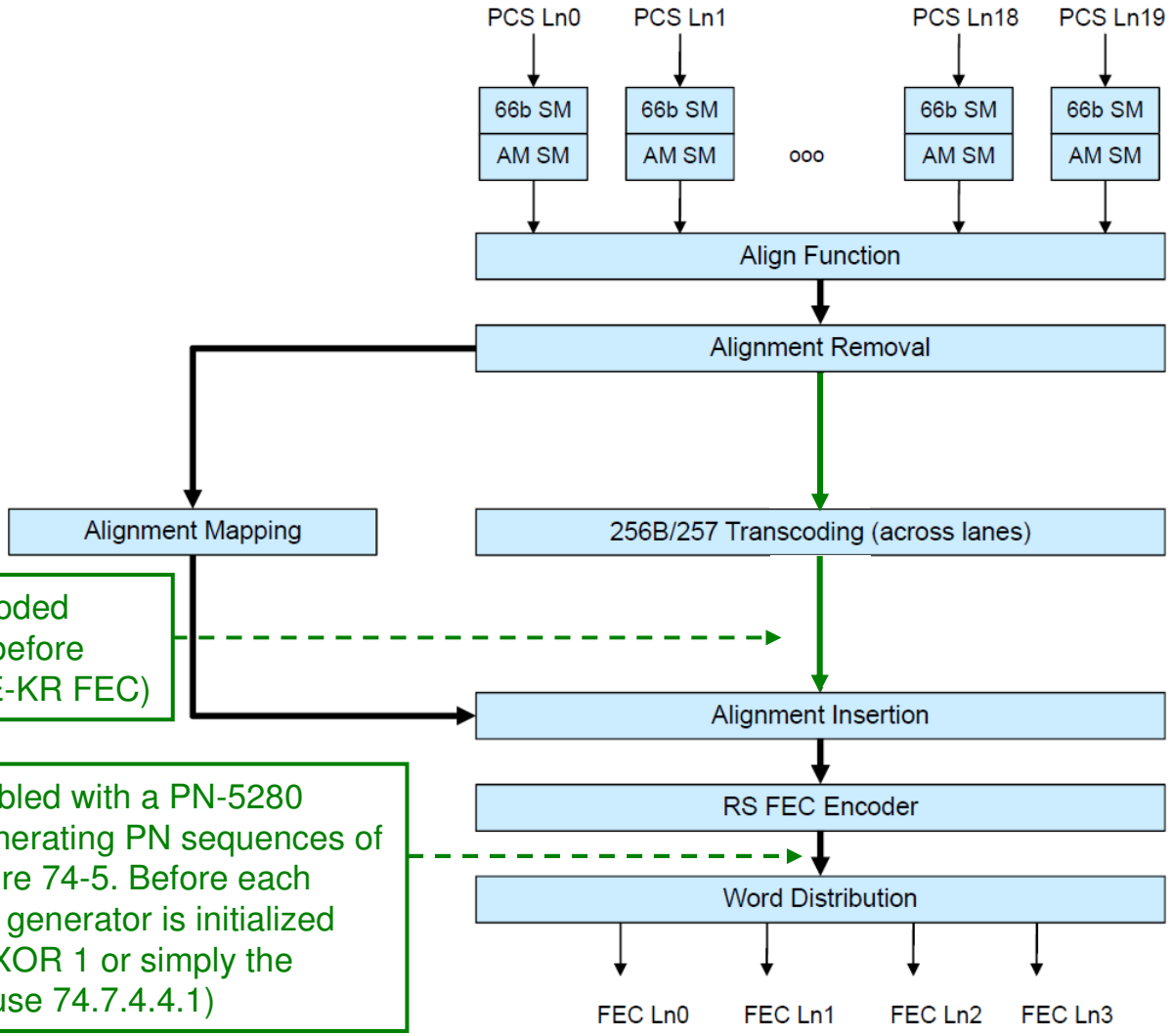
Descrambler defined in clause 49.2.10

Figure 49–10—Descrambler



(*): Shrunk BTF (4 bit) is a block type field whose second nibble was deleted during transcoding

Alternative Tx Architecture for NRZ

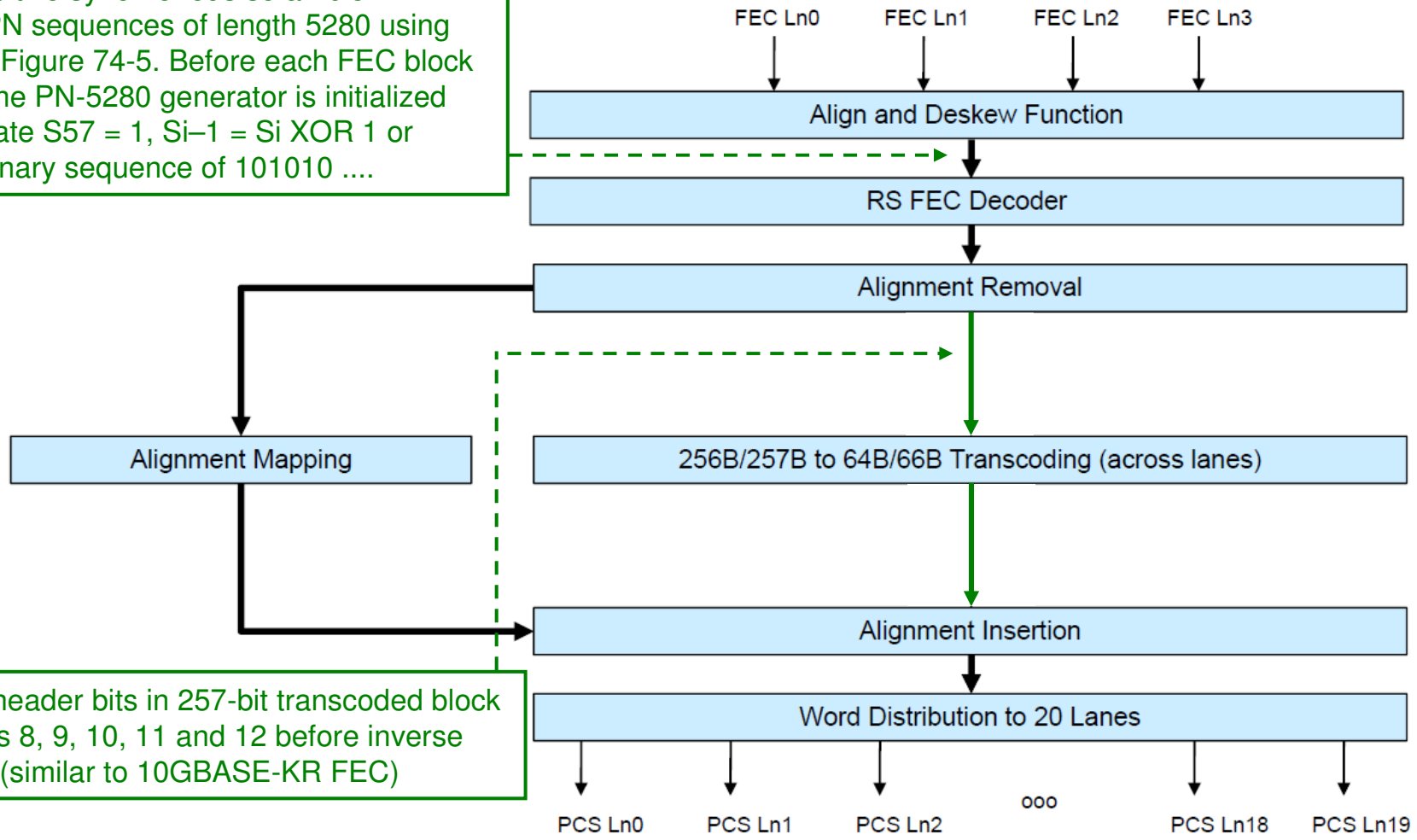


XOR first 5 header bits in 257-bit transcoded block with data bits 8, 9, 10, 11 and 12 before alignment insertion (similar to 10GBASE-KR FEC)

RS FEC encoded blocks are scrambled with a PN-5280 additive synchronous scrambler generating PN sequences of length 5280 using generator in Figure 74-5. Before each FEC block is encoded the PN-5280 generator is initialized with initial state S57 = 1, Si–1 = Si XOR 1 or simply the binary sequence of 101010 .... (clause 74.7.4.4.1)

Alternative Rx Architecture for NRZ

Received RS FEC blocks are scrambled with a PN-5280 additive synchronous scrambler generating PN sequences of length 5280 using generator in Figure 74-5. Before each FEC block is decoded the PN-5280 generator is initialized with initial state S57 = 1, Si–1 = Si XOR 1 or simply the binary sequence of 101010 ....

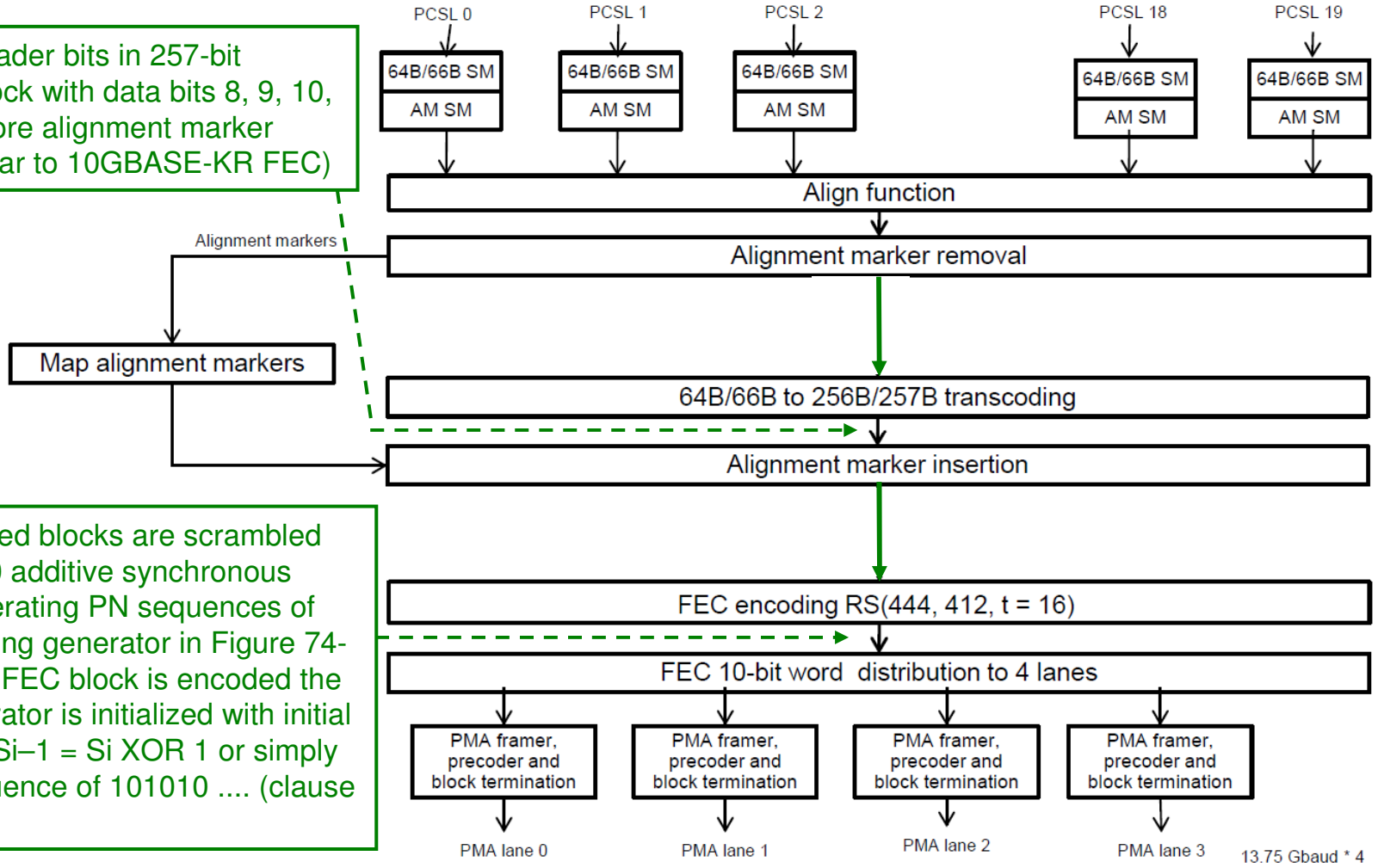


XOR first 5 header bits in 257-bit transcoded block with data bits 8, 9, 10, 11 and 12 before inverse transcoding (similar to 10GBASE-KR FEC)

Alternative Tx Architecture for PAM4

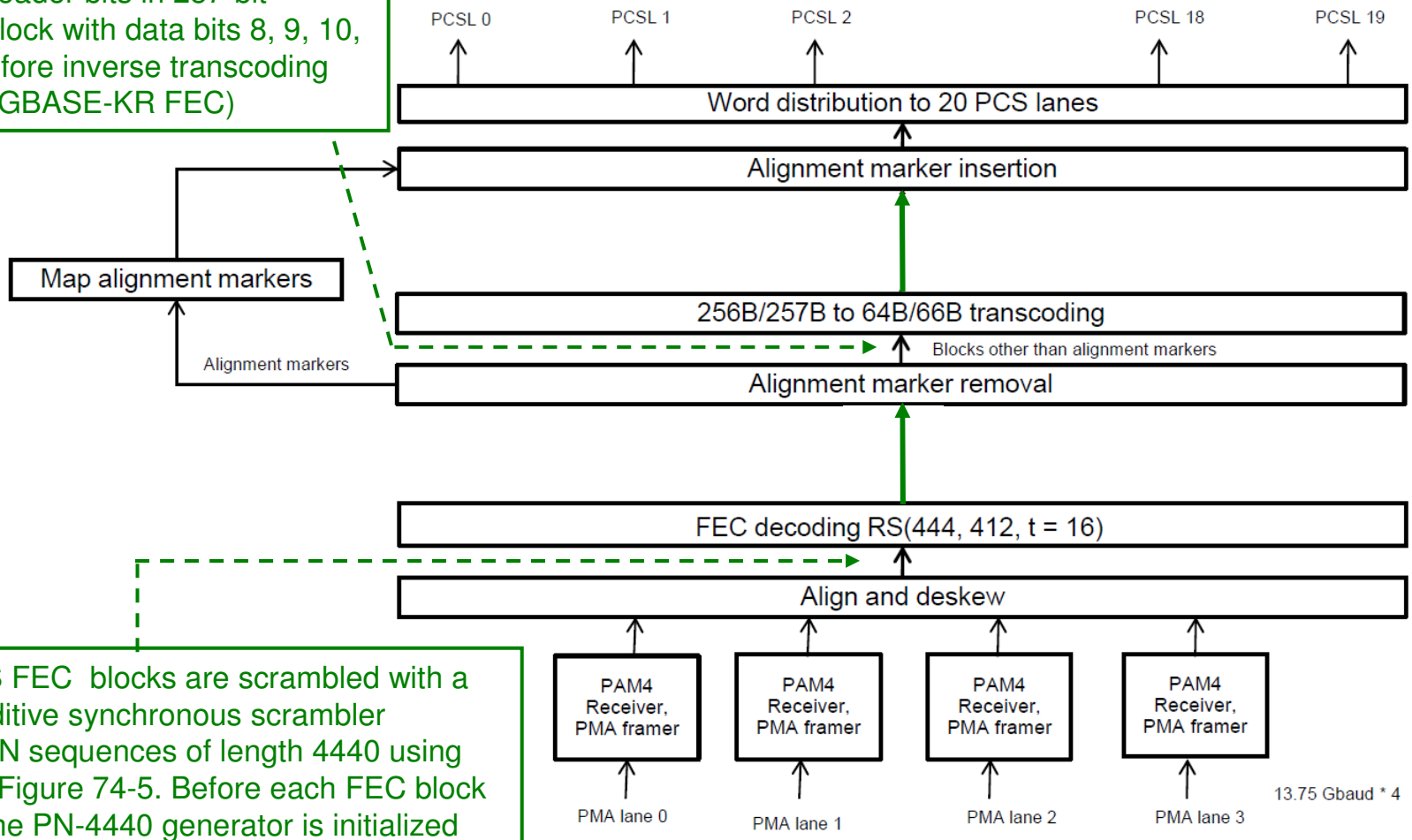
XOR first 5 header bits in 257-bit transcoded block with data bits 8, 9, 10, 11 and 12 before alignment marker insertion (similar to 10GBASE-KR FEC)

RS FEC encoded blocks are scrambled with a PN-4440 additive synchronous scrambler generating PN sequences of length 4440 using generator in Figure 74-5. Before each FEC block is encoded the PN-4440 generator is initialized with initial state S57 = 1, Si–1 = Si XOR 1 or simply the binary sequence of 101010 .... (clause 74.7.4.4.1)



Alternative Rx Architecture for PAM4

XOR first 5 header bits in 257-bit transcoded block with data bits 8, 9, 10, 11 and 12 before inverse transcoding (similar to 10GBASE-KR FEC)



Received RS FEC blocks are scrambled with a PN-4440 additive synchronous scrambler generating PN sequences of length 4440 using generator in Figure 74-5. Before each FEC block is decoded the PN-4440 generator is initialized with initial state S57 = 1, Si–1 = Si XOR 1 or simply the binary sequence of 101010 ....

Final Remarks

- In current adopted Tx and Rx architectures, if PCS and TC/FEC sublayer are implemented in one layer, scrambling and descrambling operation prior to transcoding at Tx and Rx could be merged, i.e., deleted. This would avoid self-synchronizing Rx scrambler and requires less implementation complexity than proposed alternative Tx and Rx architecture.
- Proposed Rx architecture requires a descrambler running in parallel and aiding inverse transcoding operaton. This requires slight increase in implementation complexity.
- One could replace post-FEC synchronous (additive) scrambler by pre-FEC self-synchronizing (multiplicative) scrambler.
- If desired, alignment markers can be prevented from being scrambled by pseudo-noise (PN) generator PN-5280 (NRZ) or PN-4440 (PAM4).

Thank You