

# Aligning 800GBASE-ER1 logic to OIF 800ZR logic

Tom Huber, Nokia

# Supporters

- Mike Sluyski, Cisco
- Leon Bruckman, Huawei
- Ted Sprague, Infinera
- Eric Maniloff, Ciena
- Jeffery Maki, Juniper Networks

# 800GBASE-ER1 logic baseline cf. other SDOs

- It is convenient to align 800GBASE-ER1 logic with 800ZR logic so the same framers/DSPs can be used for both applications
- The adopted logic baseline for 800GBASE-ER1 (nicholl\_3dj\_02a\_2307) is based on asynchronous mapping (GMP) into a FEC frame and is architecturally similar to 802.3cw
  - As a result, 800GBASE-ER1 will have a distinct PCS and PMA from other 800GBASE PHYs, and will rely on the XS in cases where pluggable modules are used to support 800GBASE-ER1 PHYs
- The FEC frame is the same one used in OIF 800ZR, which in turn is based on ITU-T G.709.6
- 800ZR and ITU-T G.709.6 do not strictly adhere to the definition of the XS to avoid having to process 66b blocks in the mapping of clients into the FEC frame
  - Idles are not inserted to compensate for AM removal (because this cannot be done based on 257b blocks)
  - The 257b block streams from each flow of an 800GBASE-R client are interleaved to form the 800G signal that is input to the GMP mapper

# Discrepancy #1: rate matching

- There are two reasons that rate matching is necessary in a PCS
  - Compensating for the insertion or removal of alignment markers by removing or inserting Idle blocks
  - Compensating for transitions between clock domains (again by inserting/removing Idles)
- In the 800GBASE-R PCS and XS, all rate matching is part of the 66b encoding/decoding process
  - The PHY\_XS removes AMs, and the 66b decoding/rate matching process in the PHY\_XS would insert Idles to compensate for that so that the nominal clock rate at the MII is preserved
- In 800GBASE-ER1, GMP provides another tool for rate matching
- In 800ZR, GMP is the only tool available for rate matching
  - The 800ZR data path disables rate matching – so it will remove AMs, but will not insert Idles
  - As such, a similar issue to what was described in [bruckman 3cw 01a 2306](#) for 400GBASE-ZR exists between 800GBASE-ER1 and 800ZR

# Discrepancy #2: block interleaving

- The 800GBASE-R PCS distributes 66b blocks to two flows
  - Flow0 is blocks 1, 3, 5, 7, ...
  - Flow1 is blocks 2, 4, 6, 8, ...
  - 257b transcoding and subsequent PCS processes are done on each flow
- GMP requires a single 800G stream as input
  - The PHY\_XS 'unwinds' the 800GBASE-R PCS all the way to the MII; the 800GBASE-ER1 PCS would by default 66b-encode the MII and transcode it, producing the equivalent 66b block sequence 1, 2, 3, 4, 5, 6, 7, 8, ...
  - OIF 800ZR creates the 800G stream that will be input to the GMP mapper by interleaving the 257b blocks from the two flows, so the equivalent 66b block order in 800ZR is 1, 3, 5, 7, 2, 4, 6, 8, 9, 11, 13, 15, 10, 12, 14, 16, ...

# How to resolve the discrepancies

- To keep 800GBASE-ER1 and 800ZR the same, the input to the GMP mapper within the 800GBASE-ER1 PCS needs to have AMs removed, no Idles added to compensate, and the same equivalent block order as 800ZR
- However, the description of the 800GBASE-ER1 PCS must be compliant with the 800G XS, which is already specified in 802.3df
  - The PHY\_XS will insert Idles to compensate for removed Ams, reverse transcode to 66b, merge the two flows of 66b blocks, and decode to the MII
- The solution is for the 800GBASE-ER1 PCS to “undo” the operations that the PHY\_XS does and create the same signal that 800ZR uses as input to the GMP mapper
  - The 800GBASE-ER1 PCS needs to remove Idles as if it will later insert AMs and insert Idles as if it has removed AMs
    - This can be described explicitly in the 66b encode/rate matching process in the 800GBASE-ER1 PCS
  - The 800GBASE-ER1 PCS needs to shuffle 66b block order so that the 257b blocks that are presented to the GMP mapper are the same ones that would be presented by 800ZR
    - This can be easily described as distributing to two flows, transcoding each flow, and then merging the two flows
- This produces the correct input to the GMP mapper whether or not the XS is present

# Tx direction details

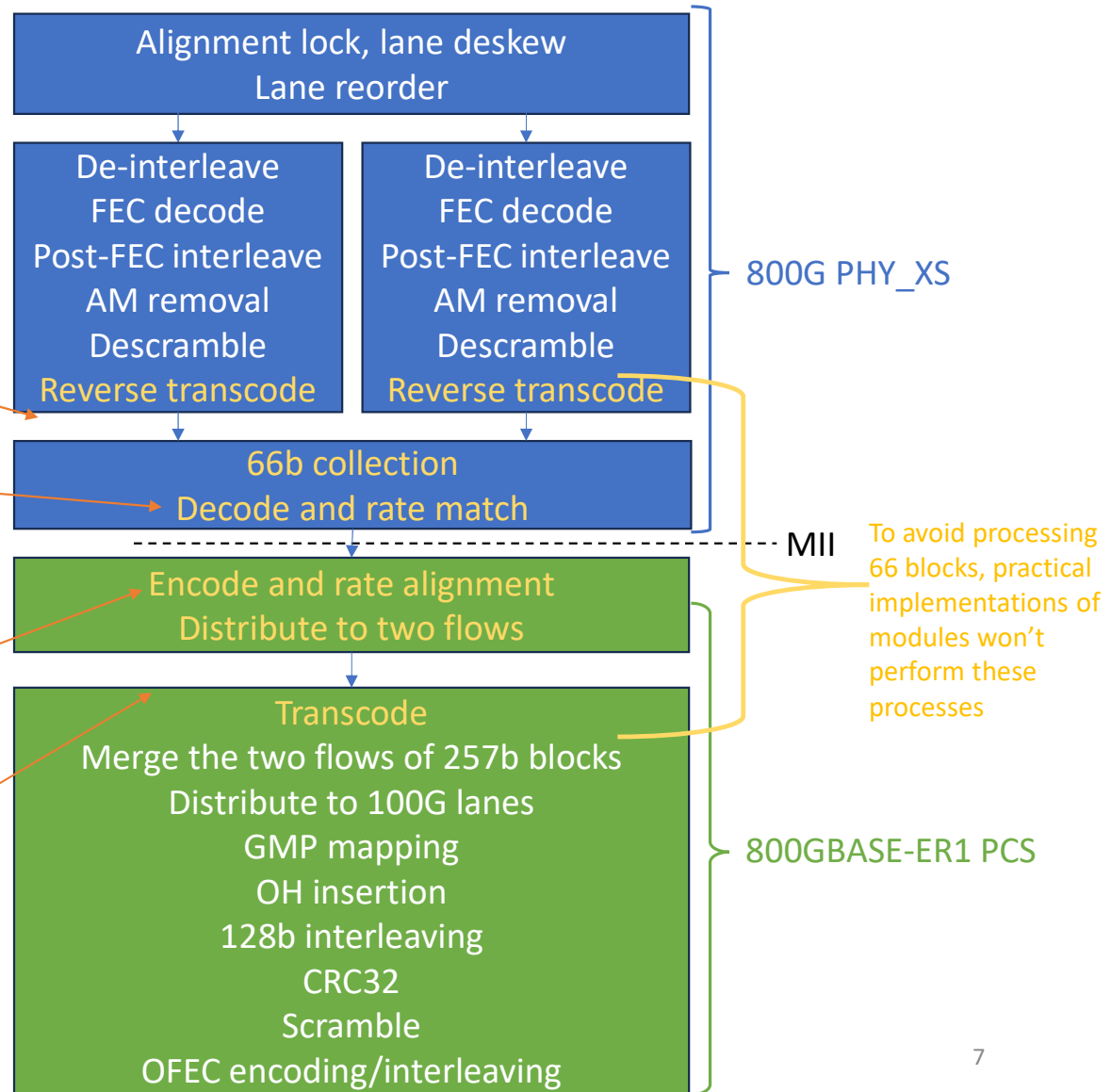
## 800GBASE-ER1

257b blocks are reverse transcoded and 66b blocks are re-interleaved

Idles are inserted to compensate the removed AMs and restore the MII clock rate

Idles are removed as if to compensate for AM insertion later in the flow  
 "Rate alignment" is used instead of "rate matching" to call attention to the difference in behavior from typical PCS encoding/rate matching processes

66b blocks are distributed to two flows and transcoded



# Rx direction details

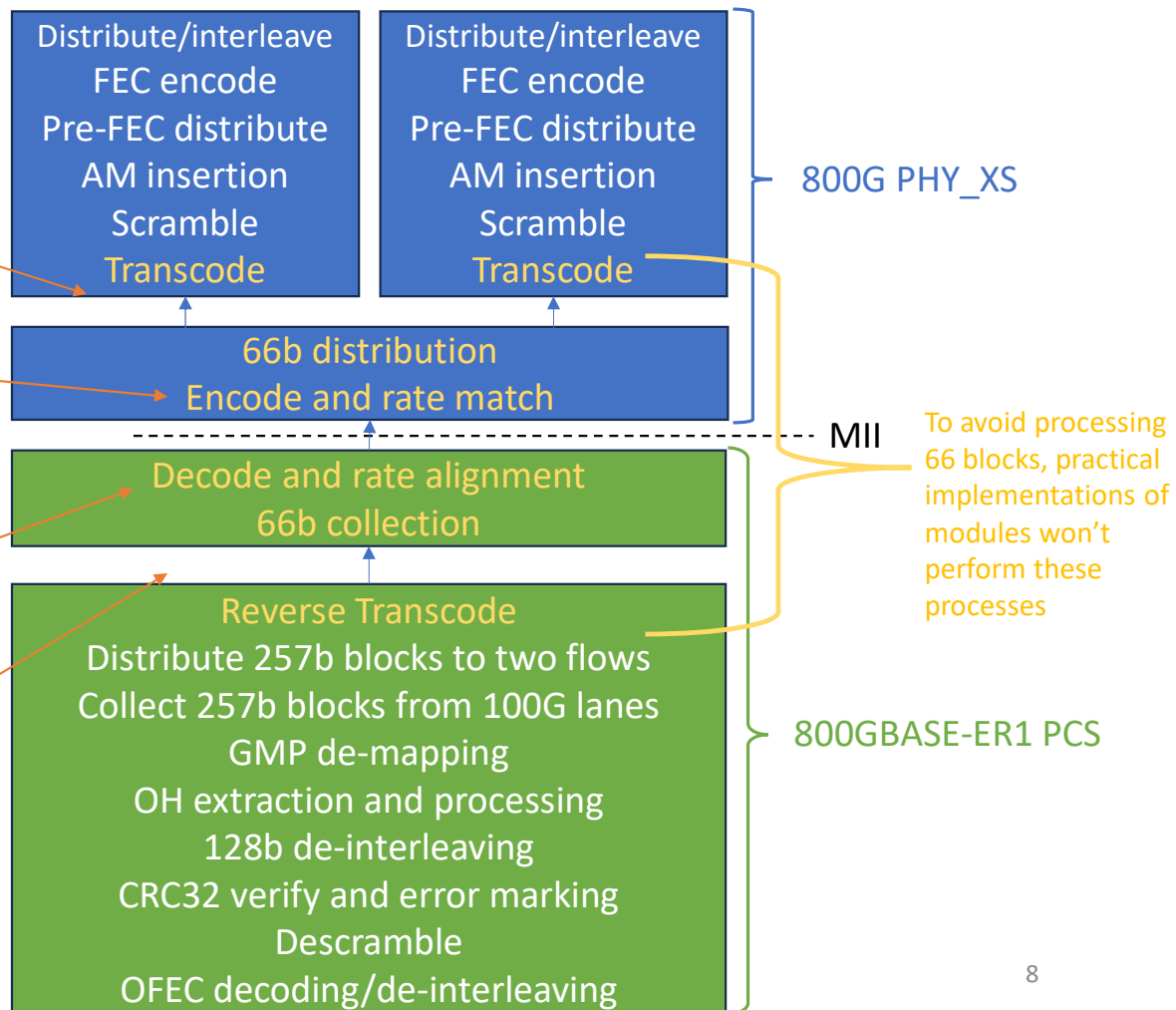
## 800GBASE-ER1

66b blocks are distributed and transcoded

Idles are removed to compensate the AMs that will be inserted and restore the MII clock rate

Idles are inserted as if to compensate for AM removal earlier in the flow  
 "Rate alignment" is used instead of "rate matching" to call attention to the difference in behavior from typical PCS encoding/rate matching processes

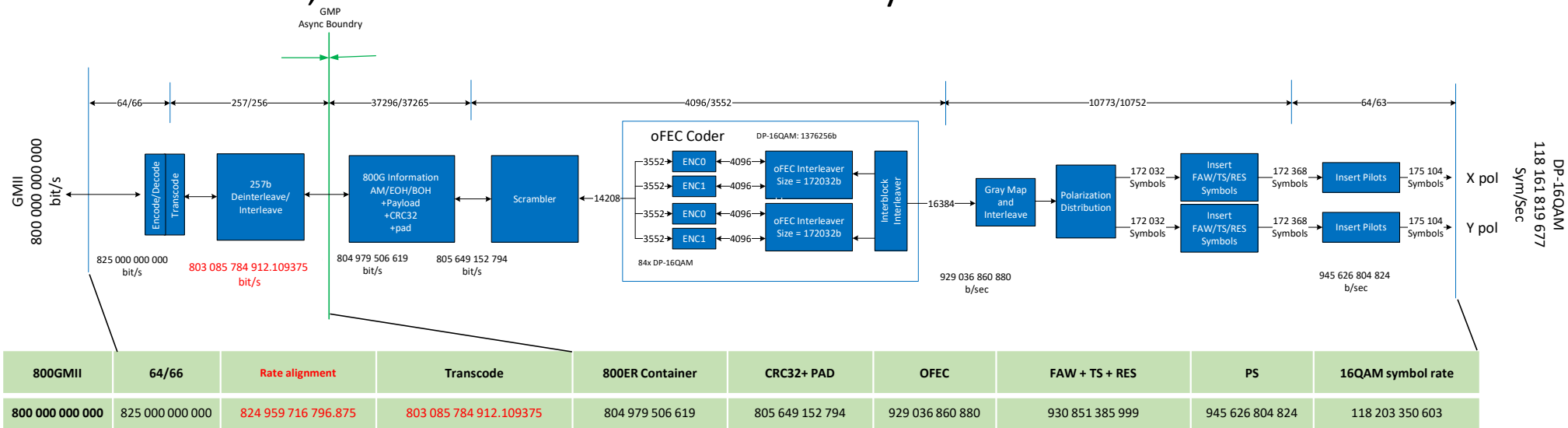
257b blocks are reverse transcoded and 66b blocks collected to a single flow





# Impacts to the 800GBASE-ER1 baseline

- The proposed additional processes are details of how the rate matching and 257b transcoding are done in this PCS, so they don't materially change the baseline
- The bit rate of the input to the GMP mapper is different than what is shown in the baseline, but there is no difference to the symbol rate



# Comparing to P802.3cw

- The same rate matching discrepancy existed between OIF 400ZR and 400GBASE-ZR, but was noticed later in the process
- The solution in P802.3cw was to include rate matching in the description of the GMP mapper/demapper in order to minimize text changes
  - The process for the GMP mapper (155.2.5.3) removed Idle blocks to align the bit rate with the 400ZR bit rate
  - The process for the GMP demapper (155.2.6.8) added Idle blocks to align the bit rate with the MII
- The approach proposed here is simpler
  - The GMP mapping process is exactly the same as it is in OIF and ITU documents
  - Describing the “extra” processes separately will make it easier for implementers to omit the processes that are not strictly necessary (i.e., there won't be a need to partially implement any processes specified in 802.3)

# 800GBASE-LR1 is not impacted

- 800GBASE-LR1 aligns to OIF 800LR
- The 800LR IA is based on the full XS and uses synchronous mapping into the inner FEC frame

# Proposal

- Adopt the PCS processes on slides 6-7 for the 800GBASE-ER1 PCS (without the notation about which ones will be skipped in implementations that use the XS)

Thank you!