Alignment Marker Lock/Unlock Schemes for 400GE

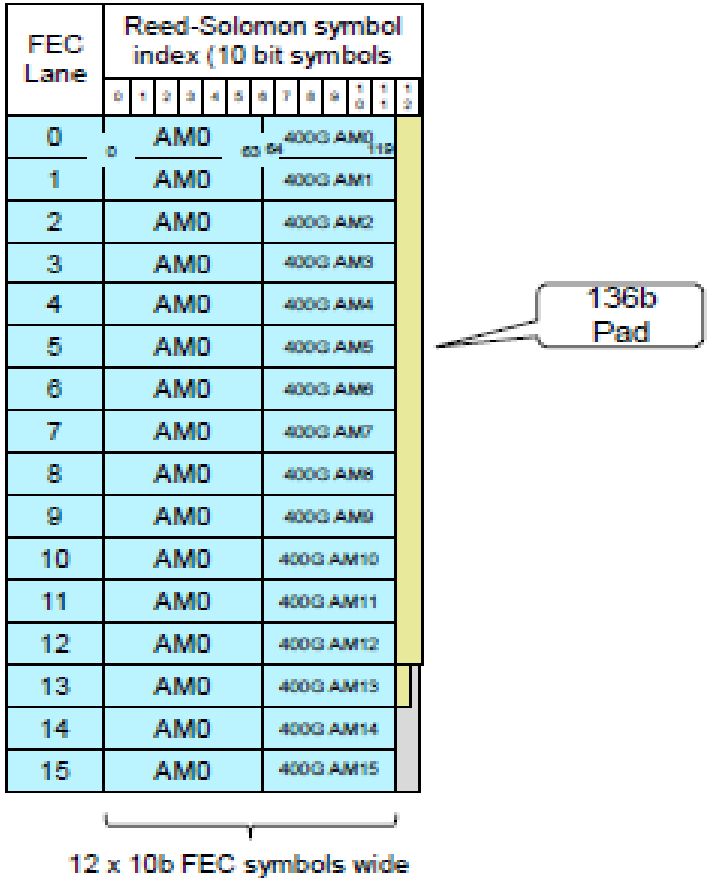
IEEE P802.3bs, Logic Ad Hoc
Dec. 2015

Zhongfeng Wang, Broadcom
Phil Sun, Credo
Mark Gustlin, Xilinx

Current Proposal for AM Patterns

Proposed 400Gb/s AMs

- Re-use 100G AM0 from 802.3ba to allow common block lock between lanes of 100G and 400G, the rest is unique to 400GbE
- Have a 56b 400G unique AM per lane also
  - 56+64 = 120b, putting 120b on each FEC lane after RS symbol distribution requires 8x257b AM blocks
  - Content of 400G AMx is TBD



AM Lock Scheme in 100G-KR4

- Input BER: 2.3e-5
- Assume error propagation: 0.42
- If no more than 3 nibbles errors happen in 48-b segment, we claim a “block match”.
- If two heading AM blocks in 2 consecutive AM groups are matched by definition, we will enter the “Lock” state.
- Mean time of lock:
  Tm=1.5 + 4e-7 (unit: AM group elapse time)

- False alignment probability:
  P(false align) ~1.0e-17 → mean time of 1.7e5 yrs.

- Worst case lock time:
  P(not lock in 6 group delays) < 1.0e-21
  → mean time > 1.7e9 years
  The worst scenario: ---v---x---v---x---v---v

Locking Analyses in 400GE

- Assume input BER: ~1e-6
- Assume error propagation probability (EPP): 0.75
- There're 16 nibbles in a 64-b data segment. Excluding the fields that were BIP fields in 802.3bj, we will only match 12 nibbles to keep the lock circuitry consistent between 100G and 400G".

- Block match:: there are no more than 2 nibble errors per 64-b block.
- Lock definition:: 2 consecutive block matches
- Mean time of lock:
  Tm=1.5 + 7e-3 (unit: AM group elapse time)
- False alignment probability:
  P(false align) ~= 3.6e-21
  → mean time ~6e7 years.

- Worst case lock time:
  P(not lock in 12 group delays) < 6.8e-21
  → mean time > 3e7 years
  The worst scenario: ---v---x---v---x---v---x---v—x---v----x---v---v

Note: when BER=2e-4, EPP=0.65, we get similar results

Use FEC Status for Unlock Detection

- Assume all 16 (or 8) lanes are correctly aligned. But FEC decoding status may provide misleading information.
  - Consider FLER=6e-11. Thus it is practically impossible to have 3 consecutive un-decodable FEC blocks when correctly aligned.
  - Therefore, it is reasonable to claim “unlock” when detecting 3 undecodable FEC blocks.

Use FEC Status for Unlock Detection (II)

- If there's any false alignment, .e.g., offset by 1 bit or 1 symbol for one lane, it will lead to 100+ symbol errors with high probability. In this case, a false decoding may happen. But the probability of false decoding will be much less than 1e-13. A worst case ($p < 1e-26$) is given as follows:
  ---Ud----Ud----Fd---Ud---Ud---Fd---Ud---Ud---Ud,
  where "Fd" denotes "false decoding of a FEC block", "Ud" stands for a undecodable FEC block.

- In the above case, we take 9 FEC blocks of transmission time to enter "unlock" state when false aligned. This is many orders of magnitudes shorter than using normal AM blocks.

- Note: if setting 2-pair of undecodable FEC blocks as condition, we will have same best case, but worst case will be:
  --"Ud--Ud-"---"Fd---Ud"---"Ud---Ud"---"Ud---Fd"---"Ud ---Ud"--"Ud –Ud"

Summary

- Block match using only 48-b pattern is defined to make it consistent with 100G.
- The proposed lock scheme is targeted at a good balance between *low false alignment probability* and *short worst-case lock time*.
- Using FEC decoder status for unlock detection is stated in the current draft.

Appendix: Locking Scheme B

- Assume input BER: ~2e-4
- Assume error propagation probability: 0.65 (<0.75)
- There're 16 nibbles in a 64-b data segment. Excluding BIP fields, we will only match 12 nibbles (i.e., 48 bits out of 64-b segment)

- Block match:: there are no more than 3 nibble errors per 64-b block.
- Lock definition:: 2 consecutive block matches
- Mean time of lock:
  Tm=1.5 + 9.3e-5 (unit: AM group elapse time)

- False alignment probability:
  P(false align) ~= 1.0e-17
  → mean time > 20,000 years.

- Worst case lock time:
  P(not lock in 12 group delays) < 6.8e-22
  → mean time > 3e8 years
  The worst scenario: ---v---x---v---x---v---x---v—x---v----x---v---v