

FEC Striping Options for 100 Gb/s Backplane and Copper Study Group

100 Gb/s Backplane and Cable Study Group
IEEE 802.3
Incline Village May 2011

Mark Gustlin – Cisco Systems

Supporters and Contributors

- Sudeep Bhoja - Broadcom
- Andre Szczepanek - Inphi

How to Stripe FEC Across PCS Lanes

- If FEC is part of the 100 Gb/s backplane and copper solution, and if we decide to stripe the FEC blocks across lanes in order to reduce the latency penalty of FEC, then we need a mechanism to stripe the data and align the data before decoding the FEC blocks
- If we do FEC across multiple physical lanes, how do we re-align data before processing the FEC?

You could do training like 10GBASE-T, but could we use the alignment markers that are already there in the 100GE data stream?

Clause 74 FEC (KR FEC)

- Today KR FEC maps 32x65 bit blocks into a FEC block adding a 32 bit checksum at the end (redundant sync bits are removed)
- The whole block is scrambled for DC balance (mainly sync bit issue) and to make sure the checksum is not spoofed.
- Today KR FEC is only run across a single lane
 - 10GBASE-KR is only a single lane interface
 - 40GBASE-KR4/CR4 has 4 lanes, with FEC running independently on each lane
 - 100GBASE-CR10 has 10 lanes, with 2 interleaved instances of FEC running per physical Lane (1 instance per PCS lane)

Clause 74 FEC (KR FEC)

Because Clause 74 FEC runs across a single lane, the receive locking mechanism is straight forward (described below, straight from clause 74):

Receive FEC block synchronization is achieved using conventional n/m serial locking techniques as described as follows:

- a) Test a potential candidate block start position
 - 1) Descramble block using PN-2112 Generator per 74.7.4.4.1
 - 2) Evaluate parity for the potential block
 - i) If the parity does not match (i.e., the received parity does not match the computed parity), shift candidate start by one bit position and try again.
- b) Validate potential block start position has good parity for “n” consecutive blocks
 - 1) If any of them fail shift candidate start one bit position and start again
 - 2) If “n” consecutive blocks are received with good parity, report Block Sync
- c) Block Sync is established.
- d) If “m” consecutive blocks are received with bad parity, drop Block Sync and restart again at item a).

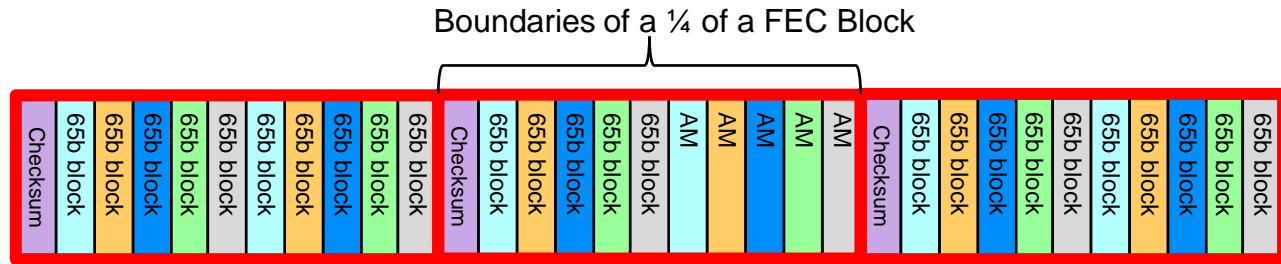
FEC Across Multiple Lanes

- If a FEC block is striped across multiple lanes, it seems that a simple n/m serial locking mechanism would not work since you can have differential skew across the multiple lanes
- A mechanism to deskew the lanes is needed first before the FEC blocks are decoded
- Two options come to mind:
 1. Re-use the Alignment Markers that are already part of the data stream
 2. Introduce some sort of training pattern to deskew the lanes on startup
- The rest of this presentation investigates option #1

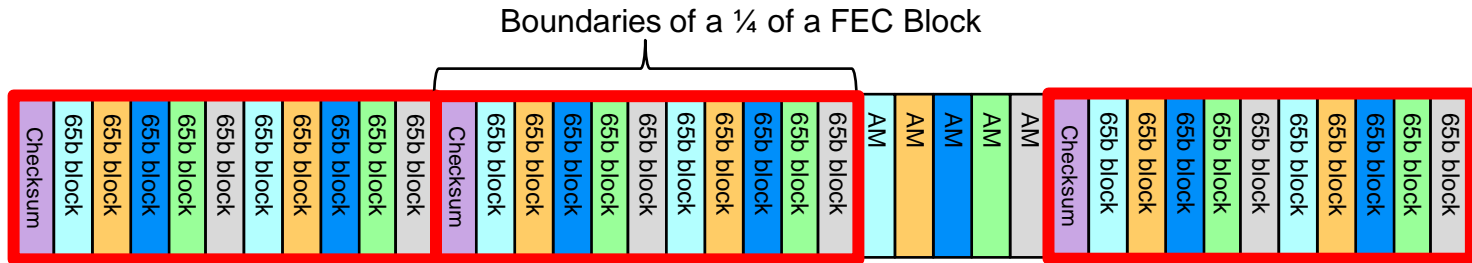
FEC Across Multiple Lanes

At least two implementations are possible:

1. Alignment Markers are included in the FEC blocks



2. Alignment Markers are not included in the FEC blocks



For either case, the Alignment Markers must repetitively be in the same location relative to FEC block starts:

1. $(AM\ Spacing * \#\ PCS\ Lanes * block\ size)$ must be divisible by $(FEC\ block\ size)$
2. $((AM\ Spacing - 1) * \#\ PCS\ Lanes * block\ size)$ must be divisible by $(FEC\ block\ size)$

One Example

- Lets look at RS(276,260) for one example

$260 * 10 = 2600$ bits of payload / 65b = 40 blocks can be carried per FEC block

That means 10 (65b) blocks + $\frac{1}{4}$ checksum (32 bits) per physical lane

Alignment markers appear once every

$$16384 * 65 * 20 / (260 * 10) = 8192 \text{ FEC blocks}$$

Receiver can lock onto the alignment markers, not the FEC checksums, performs block alignment by looking for the alignment markers, they are always at the beginning of an RS block.

We could allow PCS lanes anywhere (0,4,7,9,11 together on a lane for instance).

We could even keep the bit interleaving, but since you have to shift to align would that make sense?

Likely still need to scramble the sync bits, but only those, just use a sync scrambler

Every 8192 blocks the alignment markers repeat

CS 0	PCS LN 4	PCS LN 3	PCS LN 2	PCS LN 1	PCS LN 0	AM 4	AM 3	AM 2	AM 1	AM 0	Lane 0
CS 1	PCS LN 9	PCS LN 8	PCS LN 7	PCS LN 6	PCS LN 5	AM 9	AM 8	AM 7	AM 6	AM 5	Lane 1
CS 2	PCS LN 14	PCS LN 13	PCS LN 12	PCS LN 11	PCS LN 10	AM 14	AM 13	AM 12	AM 11	AM 10	Lane 2
CS 3	PCS LN 19	PCS LN 18	PCS LN 17	PCS LN 16	PCS LN 15	AM 19	AM 18	AM 17	AM 16	AM 15	Lane 3

Thanks!