IEEE 802.3by 25G Ethernet TF
A BASELINE PROPOSAL FOR RS, PCS, AND FEC

Eric Baden (ericb at broadcom com), presenting
Ankit Bansal (ankitb at broadcom com),
Cedric Begin (cbegin at cisco com),
Gary Nicholl (gnicholl at cisco com)

Supporters

Rob Stone, Broadcom

Howard Frazier, Broadcom

Yong Kim, Broadcom

Don Cober, CoMira Solutions

Ron Muir, JAE

MOTIVATION & GOALS

- Goal is to develop a simple & efficient PCS architecture with maximal re-use of current 802.3 specifications
- Two approaches available: scale up from 10G or scale down from 40G/100G
- Interest in supporting 3 FEC modes which cross the two available approaches
- Implementation complexity examined and reviewed
  - leads to conclusion that it is simplest to leverage the existing single-lane PCS architecture to develop this next-gen single-lane PCS architecture

HISTORY

This baseline proposal is the culmination of many joint presentations that have been presented and reviewed by the study group at regular and ad-hoc meetings

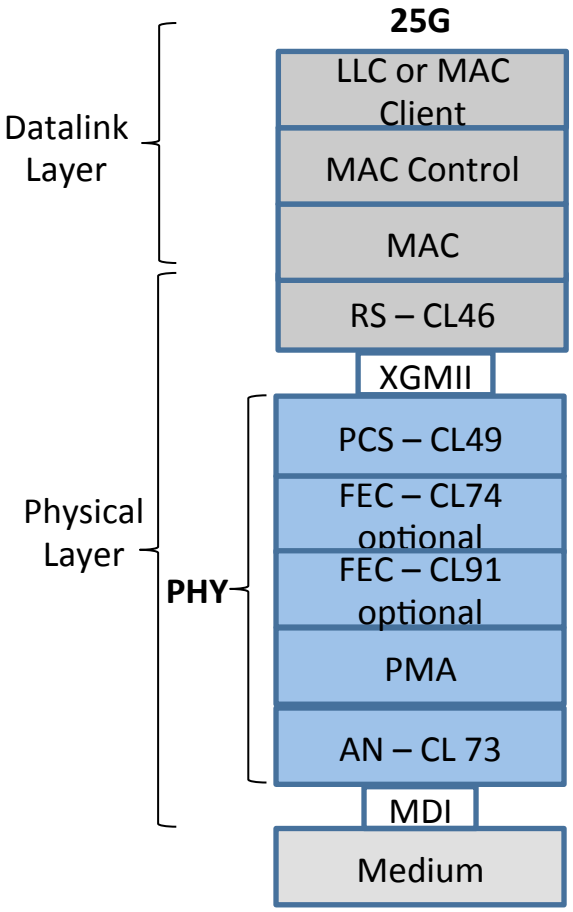
- "25GbE PCS Technical Feasibility" - gustlin_081214_25GE_adhoc.pdf (initial overview of options)
- "PCS Thoughts and Considerations" - kim_100114_25GE_adhoc.pdf
- "25G RS/PCS Considerations – A follow up" - kim_100814_25GE_adhoc.pdf
- "Layering and Gaps" - baden_102214_25GE_adhoc.pdf
- "Architectural Thoughts – 25G Interconnect" - booth_102914_25GE_adhoc.pdf
- "25G Ethernet Layering and Gaps" - baden_25GE_01a_1114.pdf
- “Architectural Thoughts – 25G Interconnect”- booth_25GE_01a_1114.pdf
-

This baseline proposal to follow is consistent with the work built up during these contributions

-
- Source:
- Plenary presentations: http://www.ieee802.org/3/25GSG/public/Nov14/index.html
- Ad hoc presentations: http://www.ieee802.org/3/25GSG/public/adhoc/architecture/index.html

OVERVIEW

- **25G IS SINGLE LANE PCS**
- **LEVERAGE EXISTING SINGLE LANE PCS (CL49)**
- **START WITH 10G KR AND SPEED UP**
  - USE CL46 RS LAYER (XGMII)
  - USE CL49 PCS
- **FEC OPTIONS:**
  - NO FEC
  - USE CL74 FEC (speed up)
  - USE CL91 (equivalent) FEC



OVERVIEW with optional FEC(s)

PCS/FEC	25G without any FEC	25G with CL74 FEC	25G with CL91 RS FEC
Block Coding	64/66B		
Lanes	1	1	1
RS	CL46 (4B)	CL46 (4B)	CL46 (4B)
PCS	CL49	CL49	CL49
Codeword Markers	N	N	Y
Transcode	N/A	N/A	256/257B
Reach	TBD	3+ m	5+ m
Latency	Low	Medium	High

SOME POINTS ABOUT RS FEC

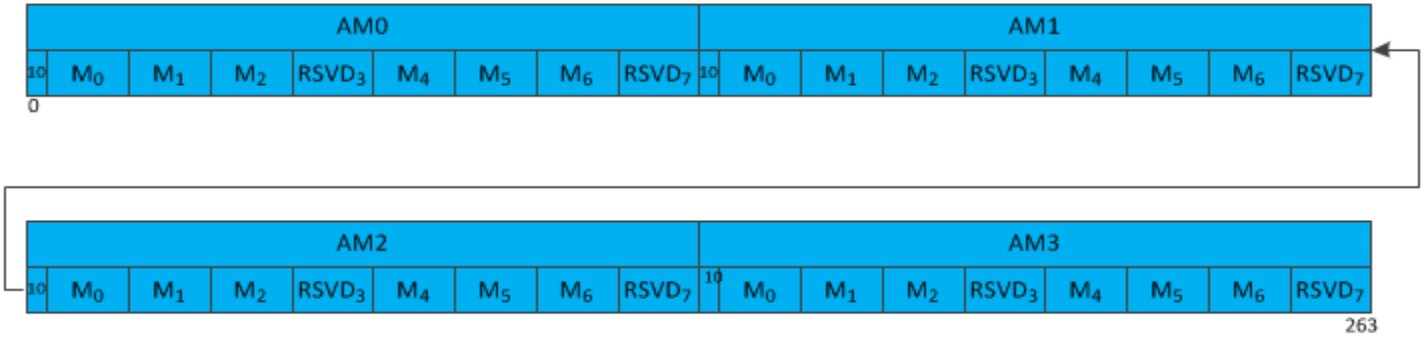
- THE RS FEC REQUIRES CODEWORD (CW) BOUNDARY IDENTIFICATION:

- UPDATE CL49 TO USE CW MARKERS (CWMs):
  - CL49 MODIFICATIONs
    - PERIODICALLY INSERT CWMs INTO PCS STREAM AT A CODEWORD BOUNDARY
    - DELETE EQUIVALENT IDLES
    - **CWMs ONLY** USED WHEN RS FEC IS ENABLED

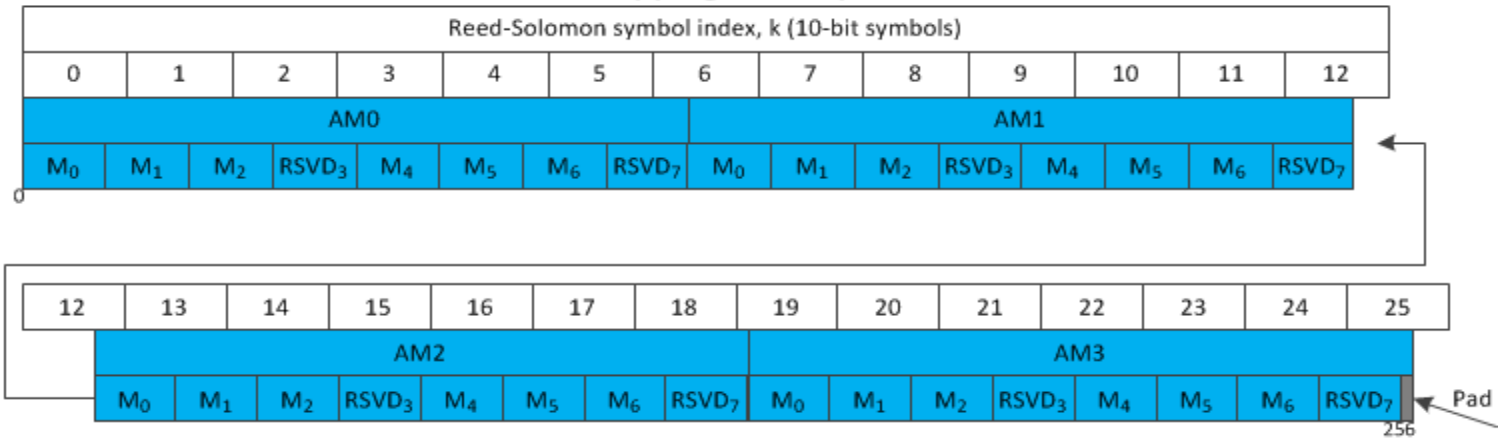
RS FEC Codeword Markers

- Codeword Markers (CWM):
  - Constructed from the concatenation of CL82 MLD 4 Alignment Markers 0 thru 3

25G W/CL91 FEC Code Word Marker (CWM)



CWM Mapping to RS Symbols



RS FEC CWM (cont.)

- CW MARKERS (CWM):
  - FORMAT
    - Almost identical to CL82 40G MLD4 AMs
    - BIP is not used and forced to fixed (tbd) values.
  - INSERTION RATE
    - Every 16384***5***66 = 1024 Code Words.
  - DELETE EQUIVALENT IDLES
    - Re-use CL49 Clock Compensation Mechanism (Section **49.2.4.7 Idle (/I/)**)
  - **ONLY** INSERTED WHEN RS FEC IS ENABLED
  - MAINTAINs BASICS of:
    - CL91 FEC SYNCHRONIZATION STATE FSM (**Figure 91-8)**
    - CL91 FEC ALIGNMENT STATE FSM (**Figure 91-9)**

RELATED CLAUSES

- MULTIPLE CHANGES REQUIRED:
  - CL45 – MDIO (addressable register set)
  - CL49 – Single Lane 64b/66b PCS
  - CL91 – RS FEC for 100G
- MINIMAL CHANGES REQUIRED:
  - CL46 – XGMII Reconciliation Layer – Speed up only
  - CL74 – KR FEC – Speed up only

CL45 (MDIO) CHANGES

- **CL49** Related Register Changes:
  - Add CL49 10GBASE-R PCS control for CWM insertion and IDLE deletion
    - Use PCS Control 2 Reg?
- **CL91** Related Register Changes:
  - Maintain 100GBASE-R FEC Control/Status bits
    - Still per 'PHY'
    - PCSLane alignment status register applicable, but for one lane only.
    - FECLane alignment register applicable, but for one lane only.
    - RS FEC lane mapping register is not applicable.
    - Maintain RS FEC symbol errors counter for lane 0 only.
    - RS FEC BIP Error Counters are not applicable.
    - RS FEC PCS Alignment status register is not applicable.
- **GENERAL** Changes:
  - Add 25G CR4/KR4 to Negotiated Port Type
  - Add 25G CR4/KR4 to Backplane Ethernet, BARE-R copper status register
  - OTHER?

CL49 (PCS) CHANGES

- **Add/update sections for the insertion of CWMs and deletion of equivalent IDLEs or ordered sets.**
  - Update Figure 49-4 (Functional Block Diagram) to include IDLE Deletion and CWM Insertion functions
  - Add paragraph in Section 49.2.2 (Functions within the PCS) to include IDLE deletion function and CWM Insertion functions.
  - Update Figure 49-5 (PCS Transmit bit ordering) to include IDLE Deletion and CWM Insertion functions.
  - Update paragraph 49.2.4.7 (IDLE /I/) to indicate IDLEs may be deleted to offset CWM insertion.
  - Update paragraph 49.2.4.10 (ordered_set /O/) to indicate ordered sets may be deleted to offset CWM insertion.
  - Update paragraph 49.2.5 (Transmit Process) to include IDLE Deletion and CWM Insertion functions
  - Add section 49.2.65 (Alignment Marker (CWM) Insertion) to describe details of AM insertion function.

CL91 CHANGES - OVERVIEW

- Maintain the same register set as the 100G RS-FEC
- Add/Update sections to show the following:
  - Remove MLD functions:
    - TX:
      - Block Lock
      - AM lock except for one FEC lane
      - Deskew
      - Re-order
      - Symbol distribution
    - RX:
      - Alignment lock except for one FEC lane
      - Deskew
      - Block Distribution
  - Change AM removal, mapping, and insertion for both RX and TX
    - Different number of AMs (CWM) inserted and deleted.
    - Single FEC lane (0) and new AM to FEC lane mapping.
      - For TX and for RX (removal)
  - Transcoding changes
    - Extending the control blocks to support CL49 as well as CL82 types

CL91 CHANGES - DETAILS

- **Lane block synchronization** (91.5.2.1)
  - Change to Block Synchronization
  - Change text to indicate a single lane (FEC:IS_UNITDATA_i.request) provides a stream of data to this function. It obtains lock to the 66-bit blocks in the bit stream using the sync headers, and outputs 66-bit blocks. Block lock is obtained as specified in the block lock state diagram in Figure 82-10.
- **Alignment lock and deskew** (91.5.2.2)
  - Change to Codeword Marker Lock
  - Change text to indicate Codework Marker lock is obtained by using the alignment marker lock state diagram in Figure 82-11, with the following changes:
    - X has a value of 0 only, representing the 40G MLD AM0 encoding (only lane 0 is considered)
    - Change am_counter variable in 82.2.18.2.4 Counters to indicate the AMs are separated by 16383*5 66 bit blocks
- **Lane Reorder** (91.5.2.3)
  - Remove this section.

CL91 CHANGES – MORE DETAILS

- **Alignment marker removal** (91.5.2.4)
  - Change to Codeword Marker removal
  - Change text to indicate that once Codeword Marker lock is obtained (as indicated by am_lock), the Codework Marker is removed from the data stream.
- **Figure** 91-2 (Functional block diagram):
  - Change TX interface to include FEC:IS_UNIDATA_0.request only
  - Change “Lane block synchronization” to “Block Synchronization”
  - Change TX direction “Alignment lock and deskew” to “CWM lock”
  - Remove TX direction “Lane reorder”
  - Remove “Symbol distribution”
  - Change RX direction “Alignment lock and deskew” to “CWM lock”
  - Remove RX direction “Lane reorder”
  - Remove “Block distribution”

CL91 CHANGES - MORE DETAILS

- **Alignment marker mapping and insertion** (91.5.2.4)
  - Change to CWM mapping and insertion
  - Change text to the following:

    The codeword markers that were removed per 91.5.2.4 are re-inserted after being processed by the codeword marker mapping function.

    The RS-FEC receive function uses knowledge of this mapping to identify RS-FEC codeword boundaries.

    The codeword marker mapping function operates on a group of four codeword markers. Let am_tx_x<65:0> be the codeword marker "x", x=0 to 3, where bit 0 is the first bit transmitted. The codeword markers shall be mapped to am_txmapped<256:0> in a manner that yields the same result as the process defined below.

    a) am_txmapped<64x+63:64x> = am_tx_x<65:2> for x = 0 to 3

    b) amp_txmapped<256> = 1

    One group of codeword markers are mapped every 5 x 16384 66-bit blocks. This corresponds to 1024 Reed-Solomon codewords. The mapped codeword markers, am_txmapped<256:0> shall be inserted as the first 257 message bits to be transmitted from every 1024th codeword.

CL91 CHANGES - MORE DETAILS

- **Reed Solomon Encoder** (91.5.2.7)
  - Add a paragraph at the end:
  - The output of the encoder is connected to the PMA:_IS_UNITDATA_0.request input one 10-bit symbol at a time, in a concatenated order.
- **Alignment lock and deskew** (91.5.3.1)
  - Change to Codeword Marker lock
  - Change text to indicate the RS-FEC receive function forms a bit stream from the PMA:IS_UNITDATA_0.indication primitive. It obtains codeword marker logic and performs codeword validity checks as indicated by the FEC synchronization state diagram shown in Figure 91-8 and the FEC alignment state diagram show in Figure 91-9
- **FEC synchronization state diagram** (Figure 91-8)
  - Remove FEC_lane_mapping and fec_lane variables.
  - Redefine amp_valid: Boolean variable that is set to true if the received 64-bit block is a valid alignment marker payload. The alignment marker payload consists of 48 known bits and 16 variable bits (the BIP3 and CD3 field and it's compliment BIP7 and CD7, see 82.2.7). The bits of the candidate block that are in the positions of the known bits in the alignment marker payload are compared on a nibble-wise basis (12 comparisons). If no more than 3 nibbles in the candidate block fail to match the corresponding known nibbles in the alignment marker payload, the candidate block is considered a valid alignment marker payload. For the normal mode of operation, the lane compares the candidate block to the alignment marker payload for PCS lane 0, from the 40G MLD 4 specification.
  - Change all_locked to indicate x=0 only.
  - Change amp_counter to count 1024 FEC codewords that separate the ends of two consecutive, normal codeword marker payload sequences.

CL91 CHANGES - MORE DETAILS

- **FEC alignment state diagram** (Figure 91-9)
  - Remove DESKEW, DESKEW_FAIL, and ALIGN_ACQUIRED states.
  - Replace test_cw from ALIGN_ACQUIRED with all_locked
- **Lane Reorder** (91.5.3.2)
  - Remove this section
- **FIGURE 91-6** (Transmit bit ordering)
  - Remove symbol distribution
  - Remove PMA_UNIDATA_[{1,2,3}].request interfaces
  - Distribute all symbols in concatenated order to PMA_UNIDATA_0.request
- **Alignment marker removal** (91.5.3.4)
  - Change text to indicate the first 257 bits in every 1024th codeword is the vector am_rxmapped<256:0>.
- **256B/257B to 64B/66B transcoder** (91.5.3.5)
  - Change section **f2** to refer to block types using Figure 49-7
- **Block distribution** (91.5.3.6)
  - Remove this section

CL91 CHANGES - MORE DETAILS

- **Alignment marker mapping and insertion** (91.5.3.7)
  - Change to CWM mapping and insertion
  - Change text to the following:

    The codeword marker mapping function derives the codeword markers, am_rx_x<65:0> for x = 0 to 3, from am_rxmapped<256:0>.

    The codeword markers shall be derived from am_rxmapped<256:0> in a manner that yields the same result as the following process.

    For x=0 to 3, am_rx_x<65:0> is constructed as follows:

    am_rx_x<0>=1 and am_rx_x<1>=0.

    am_rx_x<25:2> is set to M_0, M_1, M_2 as shown in Figure 82-9 using the values in Table 82-3 for PCS lane number x.

    am_rx_x<33:26> = am_rxmapped<64x+31:64x+24>

    am_rx_x<57:34> is set to M_4, M_5, M_6 as shown in Figure 82-9 using the values in Table 82-3 for PCS lane number x.

    am_rx_x<65:58> = am_rxmapped<64x+63:64x+56>
- **Receive bit ordering** (Figure 91-7)
  - Remove PMA_UNITDATA_[{1,2,3}].indication interfaces.
  - Show that all symbols are received via the PMA_UNIDATA_0.indicate interface.
  - Change “Alignment lock, deskew, and lane reorder” to “CWM lock”
  - Change arrow at top from “To Block distribution” to “To Alignment Inerstion”

THANK YOU!