400GbE PCS Architectural Options

IEEE 400 Gb/s Logic Ad hoc

June 7, 2013

Mark Gustlin - Xilinx
Gary Nicholl - Cisco
Dave Ofelt - Juniper

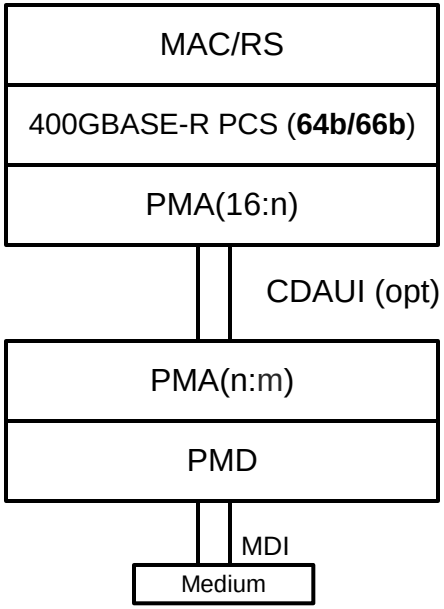
Introduction

- The following slides explore the feasibility of a 400GbE PCS
- Several PCS architectural options are shown at 400GbE, building on the 802.3ba PCS and the work that has been done within P802.3bj so far

400GbE Possible Architecture #1

- Based on a 16 Lane PCS with 64B/66B encoding (25 Gb/s per PCS Lane)
- Data is striped to PCS lanes 66-bit blocks at a time
- Alignment Markers are periodically added to all PCS lanes to enable alignment in the RX PCS
- PMAs do simple bit multiplexing to change lane widths
- Lane widths of 16, 8, 4, 2, 1 can all be easily supported

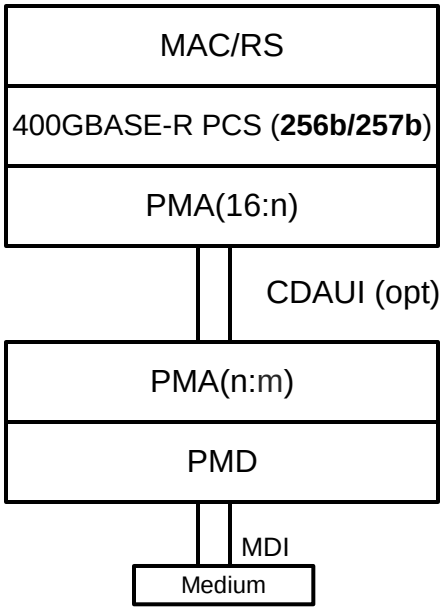
- Pros of this architecture
  - Very flexible, can support future lane widths without a PCS change
  - Most of the complexity is in the PCS, PMAs are very simple bit multiplexers
  - Low latency solution
- Cons of this architecture
  - No low latency FEC
  - If FEC is added on then it likely requires transcoding, similar to 802.3bj
  - Susceptible to burst error MTTFPA issues, when bit muxing PCS



400GbE Possible Architecture #2

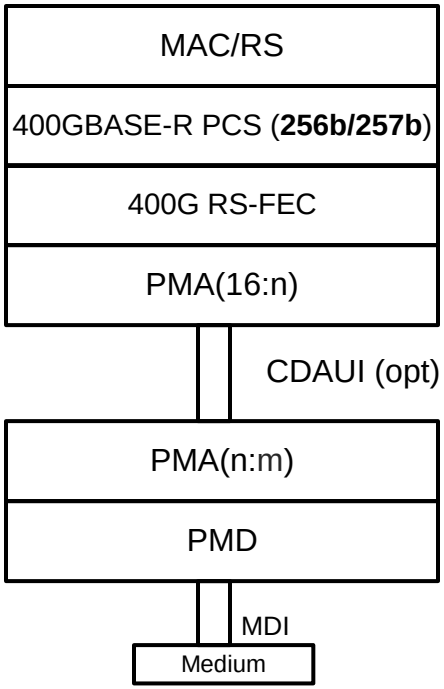
- Based on a 16 Lane PCS with 256B/257B encoding (25 Gb/s per PCS Lane)
- Data is striped to PCS lanes 257-bit blocks at a time
- Alignment Markers are periodically added to all PCS lanes to enable alignment in the RX PCS
- PMAs do simple bit multiplexing to change lane widths
- Lane widths of 16, 8, 4, 2, 1 can all be easily supported
- No FEC, but you can use the extra bits to add a robust checksum
  - 7bits per 257b block are available if running at 25.78125G per lane
  - Native rate without an additional checksum is 25.09765625G

- Pros of this architecture
  - Very flexible, can support future lane widths without a PCS change
  - Most of the complexity is in the PCS, PMAs are very simple bit multiplexers
  - Low latency solution
  - Robust error detection in the face of errors
- Cons of this architecture
  - No low latency FEC, but one can be added without transcoding



400GbE Possible Architecture #3

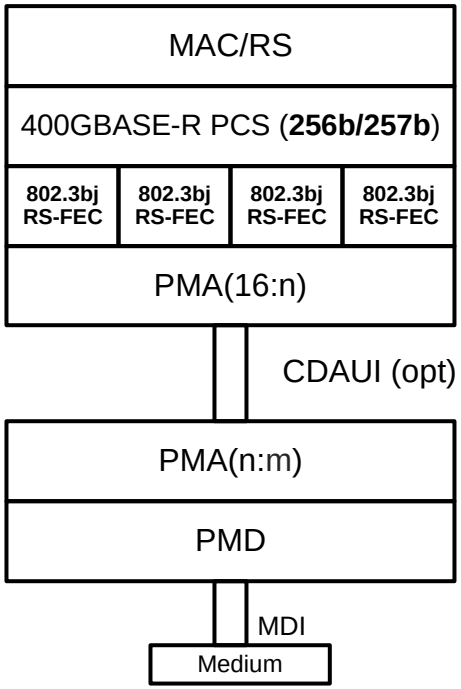
- Based on a 16 Lane PCS with 256B/257B encoding (25 Gb/s per PCS Lane)
- Data is striped to PCS lanes 257-bit blocks at a time?
  - Or distributed in RS symbol boundaries (10b for instance)?
- Alignment Markers are periodically added to all PCS lanes to enable alignment in the RX FEC block and PCS
- 400 Gb/s RS-FEC is added, no transcoding
  - Other FEC options should be explored also, what error signatures do we expect for electrical and optical lanes?
- PMAs do simple block multiplexing to change lane widths to preserve the error detection capability in the face of burst errors
- Lane widths of 16, 8, 4, 2, 1 can all be supported



- Pros of this architecture
  - Very flexible, can support future lane widths without a PCS change
  - A lot of the complexity is in the PCS, PMAs though do have to find AM lock before muxing to preserve error correction capability
  - Pretty low latency solution (~25ns of added latency due to FEC, depends on block size though)
  - Robust error detection correction
- Cons of this architecture
  - Limited re-use from 100GbE

400GbE Possible Architecture #4

- Based on a 16 Lane PCS with 256B/257B encoding (25 Gb/s per PCS Lane)
- Data is striped to PCS lanes 257-bit blocks at a time
  - Or distributed in RS symbol boundaries (10b for instance)?
- Alignment Markers are periodically added to all PCS lanes to enable alignment in the RX FEC block and PCS
  - Need 16 unique AMs, unlike 802.3bj
- A portion of the 100 Gb/s RS-FEC x 4 is added, no transcoding
- PMAs do simple block multiplexing to change lane widths to preserve the error detection capability in the face of burst errors
- Lane widths of 16, 8, 4, 2, 1 can all be supported



- Pros of this architecture
  - Very flexible, can support future lane widths without a PCS change
  - A lot of the complexity is in the PCS, PMAs though do have to find AM lock before muxing to preserve error correction capability
  - Re-use of some of the 802.3bj RS-FEC
  - Robust error detection correction
- Cons of this architecture
  - Higher latency than the other options, ~100ns with correction

Stronger FEC

- With option 2,3 or 4, if a stronger FEC is needed than the base FEC (or no FEC in the case of option 2), you simply add the FEC on top of what is already there, no transcoding is needed
- You can also strip off the current FEC and then add a stronger FEC to the PCS encoded data, again without having to do transcoding

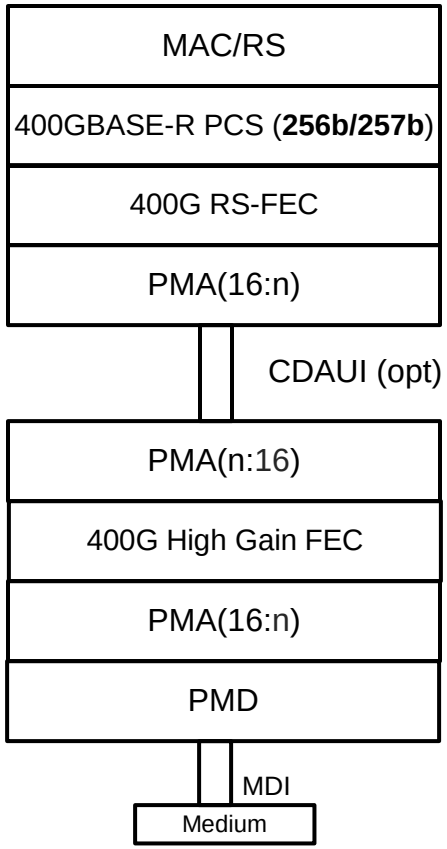


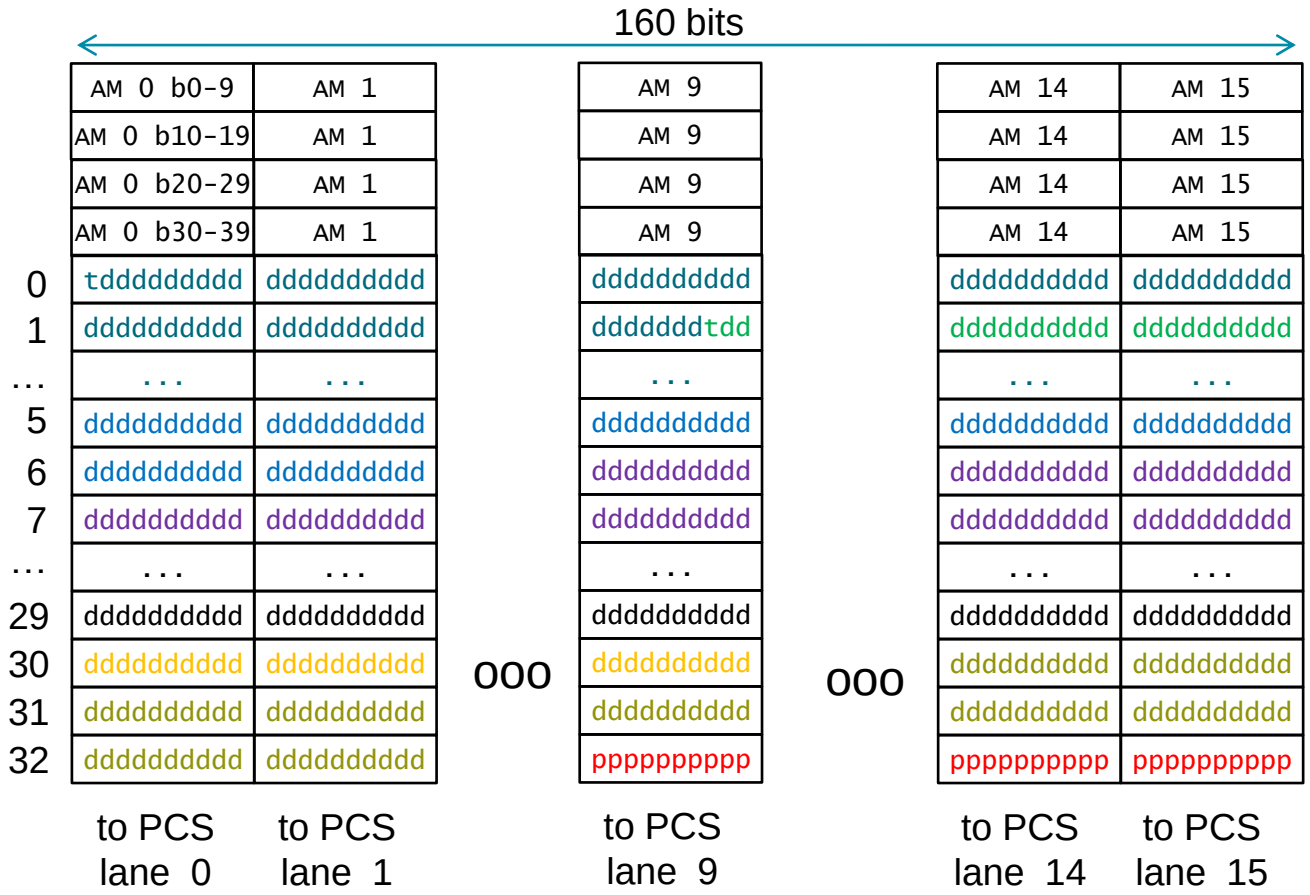
Table of Options

Option	Encoding	FEC?	PMA Width Change	Added Latency	MTTFPA concerns?
1	64B/66B	None, adding FEC likely requires transcoding	Bit muxing	0	Yes when bit muxing + burst errors
2	256B/257B	None, but ready for FEC	Bit muxing	~5ns	No
3	256B/257B	400G FEC	Block muxing	~25-50ns?	No
4	256B/257B	4xRS-FEC	Block muxing	~100ns	No

Note: At this point supporting 10x40G lanes are not addressed with these options

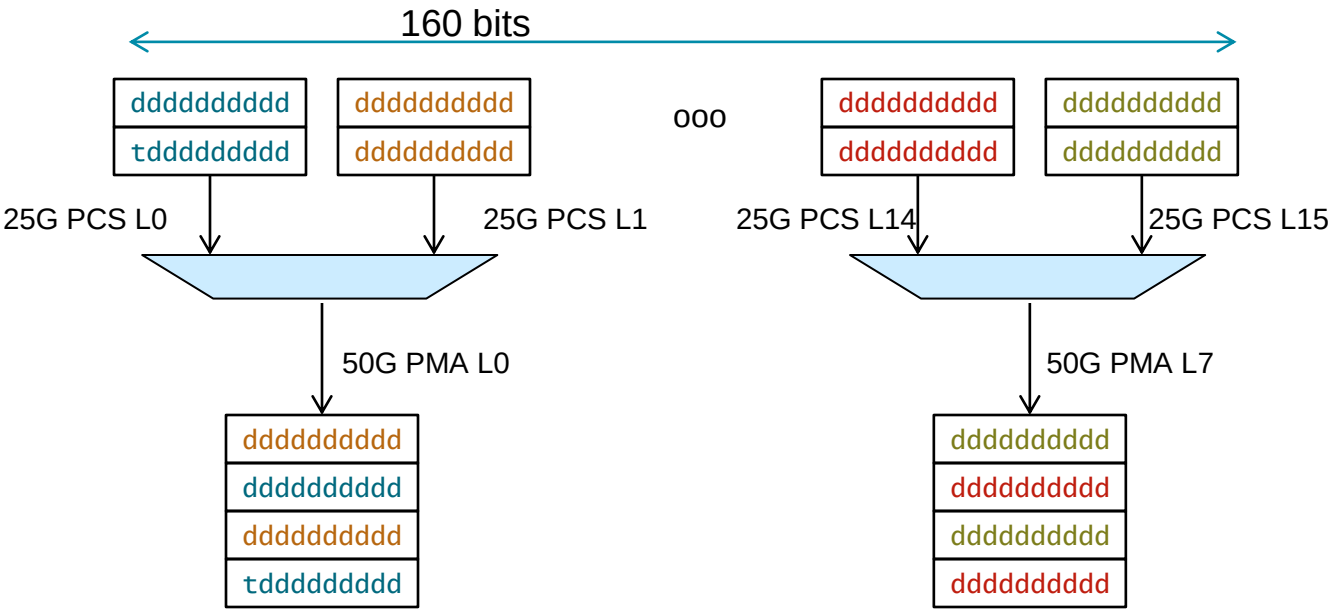
Alignment Markers

- Add an alignment marker to each PCS lane periodically, it does not need to be part of the FEC blocks, and it seems to make it easier if they are not part of the FEC block (so you don't have Alignment issues)
- Below the AMs are 40 bits each, but this is flexible, just must be nx10-bit



Multiplexing

- With 16 PCS lanes, you can multiplex down to 8, 4, 2, or 1 lane(s)
- All multiplexing must be on RS boundaries (10-bit in the case shown)
  - To preserve error correction capability in the face of burst errors
- First you must find alignment marker lock to find 10-bit boundaries, then you multiplex on RS boundaries
  - No need to deskew the various lanes
- Below shows muxing from 16 lanes down to 8 lanes



Summary

- There are many possible solutions for a 400GbE PCS, this paper shows a couple of options that are feasible with today's technology (either ASIC or FPGA)
- One simple option is scaling the 802.3ba PCS up in speed
- But if there will be interfaces that require FEC, and low latency is important, then a PCS could be defined that incorporates a low latency FEC from the start
  - This applies to both electrical and optical interfaces

Thanks!