A 400GbE PCS Option

IEEE 400 Gb/s Ethernet Study Group

November 2013 Dallas

Mark Gustlin – Xilinx
Gary Nicholl – Cisco
Dave Ofelt – Juniper
Jerry Pepper - Ixia
Andre Szczepanek – Inphi
Tongtong Wang - Huawei

Introduction

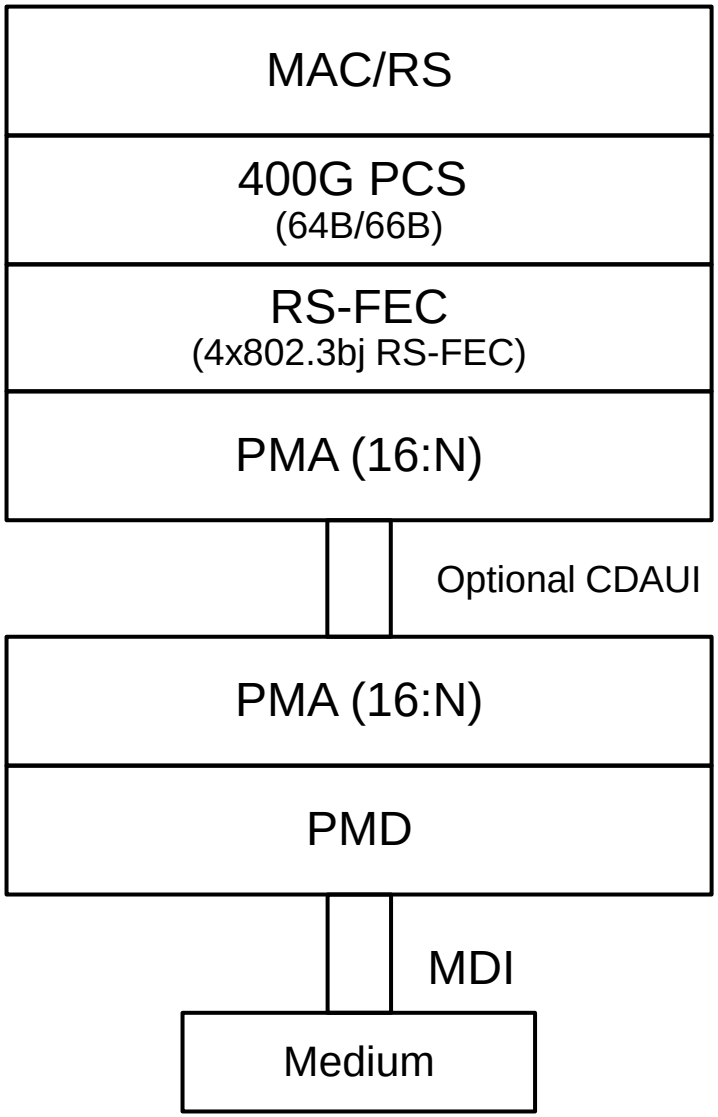
- The following slides explore the feasibility of a 400GbE PCS with RS-FEC as an integral and required portion of the architecture
- This architecture enables practical re-use of logic between 100GbE and 400GbE

References

- 400GbE PCS requirements :
  - http://www.ieee802.org/3/400GSG/public/13_09/gustlin_400_02_0913.pdf
- 400G PCS options:
  - http://www.ieee802.org/3/400GSG/public/13_09/wang_400_01_0913.pdf
  - http://www.ieee802.org/3/400GSG/public/13_09/begin_400_01_0913.pdf
  - http://www.ieee802.org/3/400GSG/public/13_09/ghiasi_400_01_0913.pdf
  - http://www.ieee802.org/3/400GSG/public/13_09/song_400_01_0913.pdf
  - http://www.ieee802.org/3/400GSG/public/13_09/wang_z_400_01_0913.pdf
  - http://www.ieee802.org/3/400GSG/public/13_07/gustlin_400_02_0713.pdf
  - http://www.ieee802.org/3/400GSG/public/13_07/wang_400_01_0713.pdf
  - http://www.ieee802.org/3/400GSG/public/13_07/ghiasi_400_01_0713.pdf
  - http://www.ieee802.org/3/400GSG/public/13_05/ghiasi_400_01a_0513.pdf

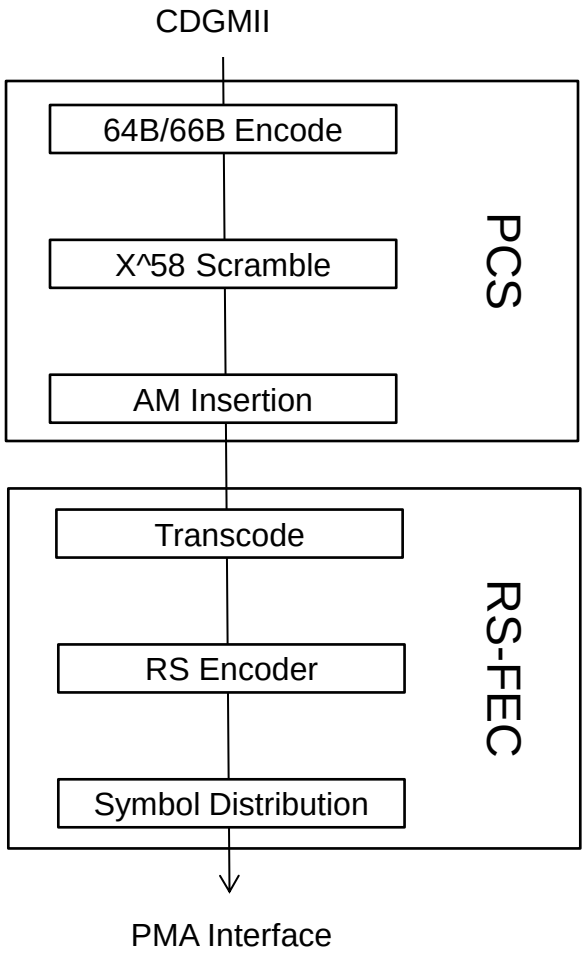
400GbE Architecture With RS-FEC

- PCS is 64B/66B based
- Required RS-FEC sublayer
- Interface between the PCS and RS-FEC is not exposed (no concept of PCS lanes!)
- 16 FEC lanes below the RS-FEC sublayer



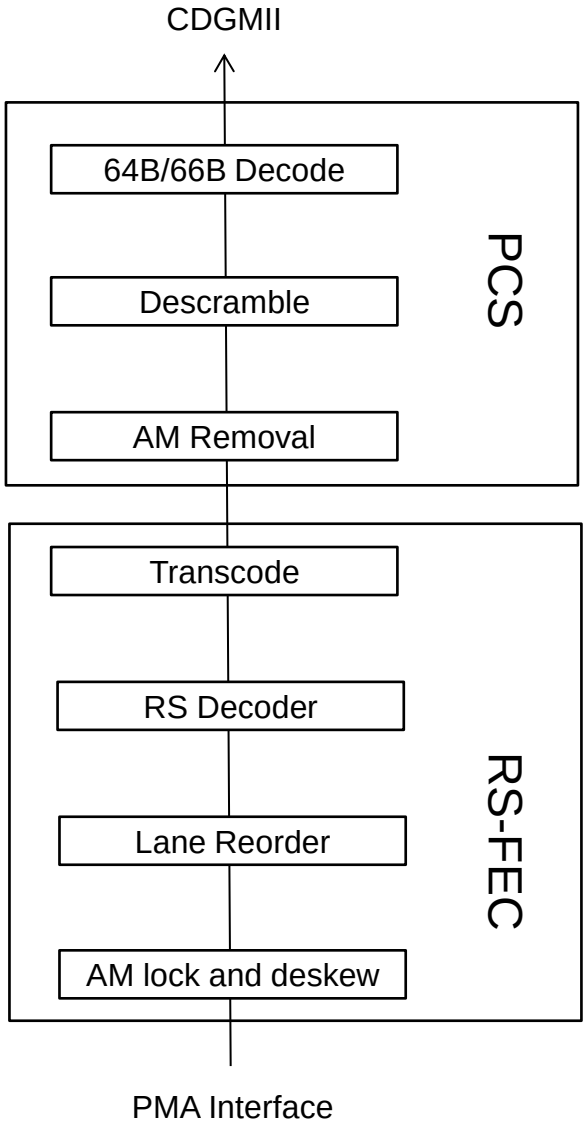
Data Flow - TX

- RS-FEC sublayer re-uses the transcoding function and the RS encoder from 802.3bj x 4



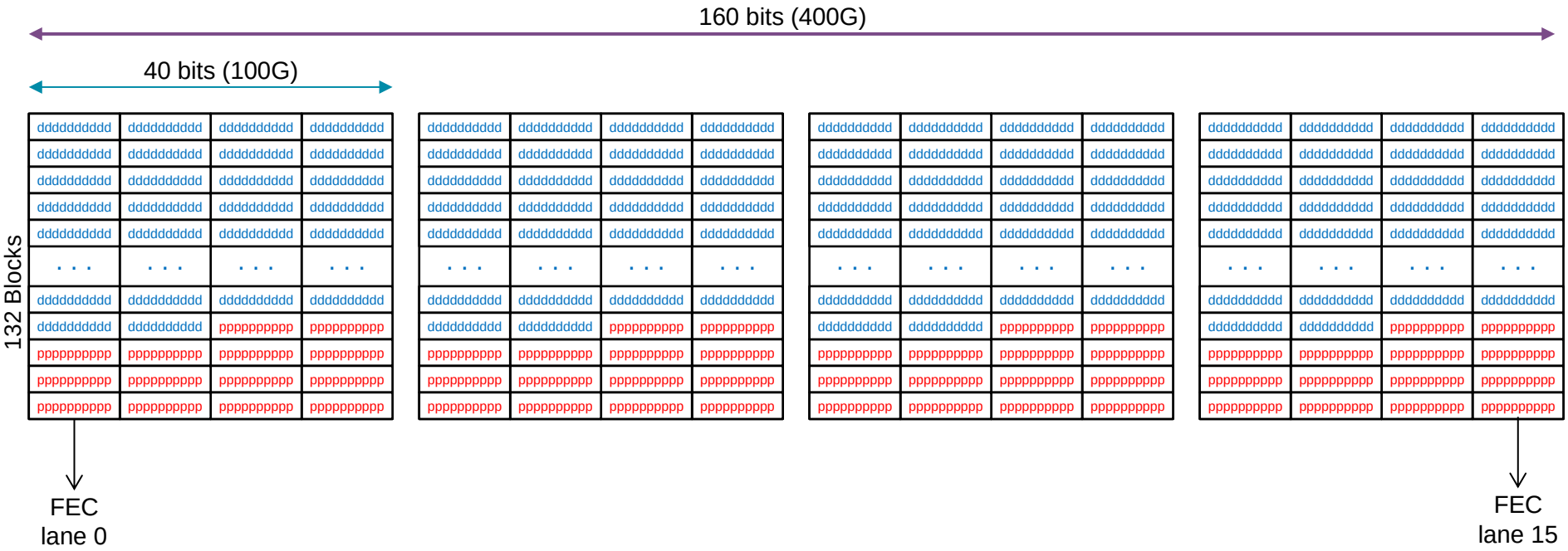
Data Flow - RX

- RS-FEC sublayer re-uses the transcoding function and the RS decoder from 802.3bj x 4



400GbE Data Distribution

- Below the RS-FEC sublayer, with using 4x802.3bj FEC, you would naturally have 16 FEC lanes



802.3bj AMs

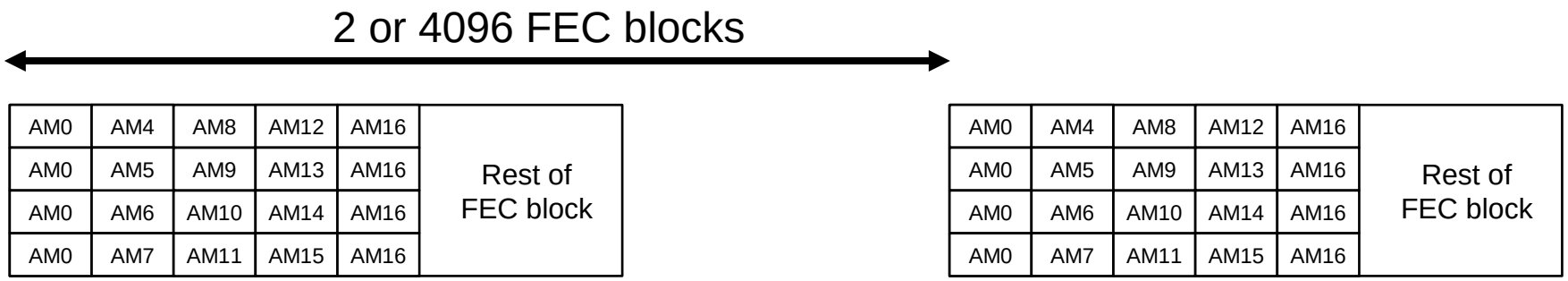
- Clause 91 defines how Alignment Markers are mapped when sent across the 4 FEC lanes
  - They are re-mapped to the FEC lanes so they appear consecutively on a given FEC lanes
  - A 5b pad is added to the end to round make them fit within a even number of 257b blocks (20*64+5 = 257*5)
  - AM0 and AM16 are repeated on all 4 FEC lanes to make it less logic intensive to find block alignment
  - The remaining AMs uniquely identify the 4 FEC lanes

FEC Lane	Reed-Solomon symbol index (10 bit symbols																																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
0	AM0 (0 … 63)										AM4					AM8					AM12					AM16								
1	AM0										AM5					AM9					AM13					AM16								
2	AM0										AM6					AM10					AM14					AM16								
3	AM0										AM7					AM11					AM15					AM16								

5b pad

802.3bj AM Distance

- AMs are always aligned to the beginning of an RS-FEC block
- The repetition distance between AMs for normal operation in 802.3bj is once every 4096 FEC blocks
- When sending rapid alignment markers, they are sent every 2 FEC blocks for EEE support



Possible 400Gb/s AMs

- Re-use many of the AMs from 802.3ba to allow common lane processing between 100GbE and 400GbE, add unique 400G AM also with TBD functionality
- Note that a given combination 320b creates a unique FEC AM for each FEC lane

FEC Lane	Reed-Solomon symbol index (10 bit symbols)						
0	0 AM0 63	AM4	AM8	400G AM0	AM16		
1	AM0	AM5	AM8	400G AM1	AM16		
2	AM0	AM6	AM8	400G AM2	AM16		
3	AM0	AM7	AM8	400G AM3	AM16		
4	0 AM0 63	AM4	AM9	400G AM4	AM16		
5	AM0	AM5	AM9	400G AM5	AM16		
6	AM0	AM6	AM9	400G AM6	AM16		
7	AM0	AM7	AM9	400G AM7	AM16		
8	0 AM0 63	AM4	AM10	400G AM8	AM16		
9	AM0	AM5	AM10	400G AM9	AM16		
10	AM0	AM6	AM10	400G AM10	AM16		
11	AM0	AM7	AM10	400G AM11	AM16		
12	0 AM0 63	AM4	AM11	400G AM12	AM16		
13	AM0	AM5	AM11	400G AM13	AM16		
14	AM0	AM6	AM11	400G AM14	AM16		
15	AM0	AM7	AM11	400G AM15	AM16		

100G RS-FEC Instance 0

100G RS-FEC Instance 1

100G RS-FEC Instance 2

100G RS-FEC Instance 3

400 Gb/s AM Distance

- AMs are always aligned to the beginning of an RS-FEC block
- Keep the same repetition distance between AMs for normal operation as in 802.3bj, once every 4096 FEC blocks
- When sending rapid alignment markers, they are sent every 2 FEC blocks for EEE support

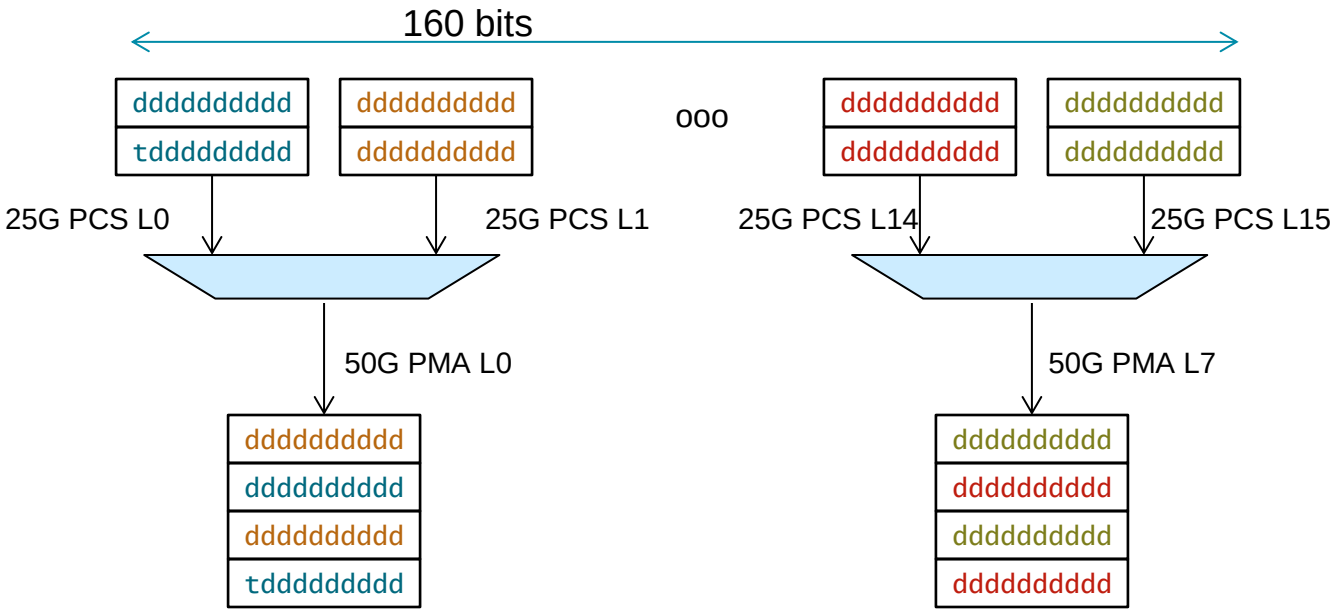
2 or 4096 FEC blocks

AM0	AM4	AM8	400G AM0	AM16	Rest of FEC block
AM0	AM5	AM8	400G AM1	AM16	
AM0	AM6	AM8	400G AM2	AM16	
AM0	AM7	AM8	400G AM3	AM16	
AM0	AM4	AM9	400G AM4	AM16	Rest of FEC block
AM0	AM5	AM9	400G AM5	AM16	
AM0	AM6	AM9	400G AM6	AM16	
AM0	AM7	AM9	400G AM7	AM16	
AM0	AM4	AM10	400G AM8	AM16	Rest of FEC block
AM0	AM5	AM10	400G AM9	AM16	
AM0	AM6	AM10	400G AM10	AM16	
AM0	AM7	AM10	400G AM11	AM16	
AM0	AM4	AM11	400G AM12	AM16	Rest of FEC block
AM0	AM5	AM11	400G AM13	AM16	
AM0	AM6	AM11	400G AM14	AM16	
AM0	AM7	AM11	400G AM15	AM16	

AM0	AM4	AM8	400G AM0	AM16	Rest of FEC block
AM0	AM5	AM8	400G AM1	AM16	
AM0	AM6	AM8	400G AM2	AM16	
AM0	AM7	AM8	400G AM3	AM16	
AM0	AM4	AM9	400G AM4	AM16	Rest of FEC block
AM0	AM5	AM9	400G AM5	AM16	
AM0	AM6	AM9	400G AM6	AM16	
AM0	AM7	AM9	400G AM7	AM16	
AM0	AM4	AM10	400G AM8	AM16	Rest of FEC block
AM0	AM5	AM10	400G AM9	AM16	
AM0	AM6	AM10	400G AM10	AM16	
AM0	AM7	AM10	400G AM11	AM16	
AM0	AM4	AM11	400G AM12	AM16	Rest of FEC block
AM0	AM5	AM11	400G AM13	AM16	
AM0	AM6	AM11	400G AM14	AM16	
AM0	AM7	AM11	400G AM15	AM16	

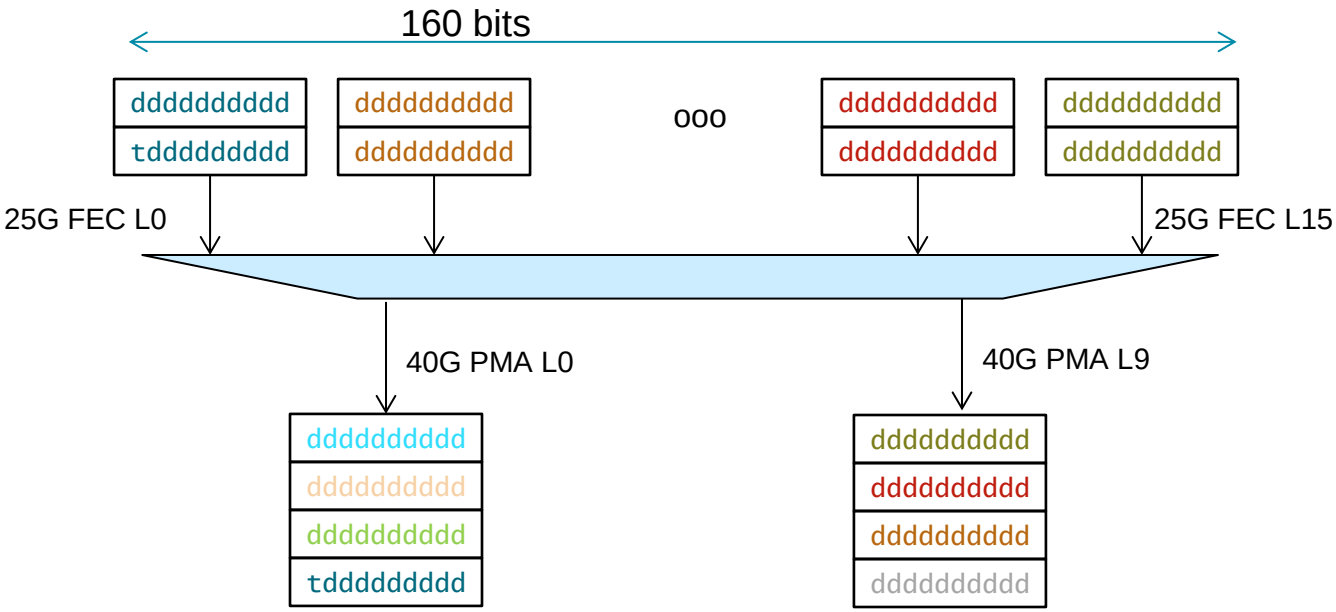
Multiplexing

- With 16 FEC lanes, you can multiplex down to 8, 4, 2, or 1 lane(s)
- Multiplexing is typically done on RS boundaries (10-bit in the case shown)
  - To preserve error correction capability in the face of burst errors
- If you are running across a medium that only has uncorrelated errors, then bit multiplexing is fine
- First you must find block lock to find 10-bit boundaries (using AM0/AM16), then you multiplex on RS boundaries
  - No need to deskew the various lanes
- Below shows muxing from 16 lanes down to 8 lanes



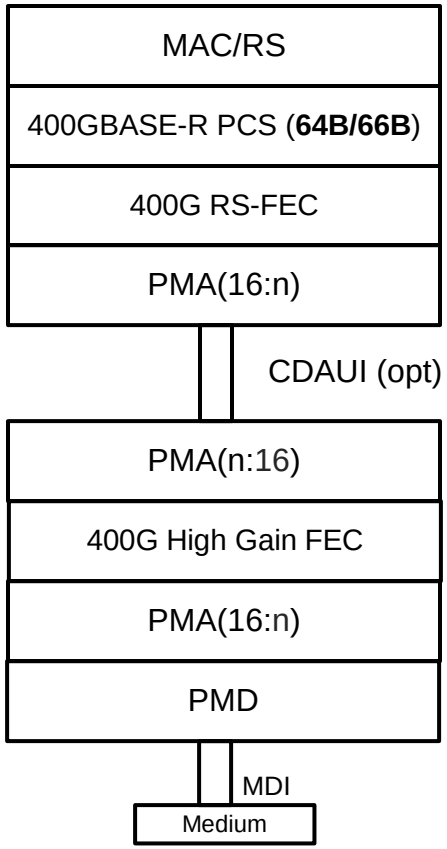
Supporting 10 Physical Lanes

- Can 10 physical lanes be supported with 16 FEC lanes?
  - Yes, but it would be a point to point interface, you have to deskew when you change lane widths
- You could stripe at the block level, or bit level; as long as you define a fixed mapping, with enough alignment marker information per physical lane to be able to align the 10 physical lanes
- Note that this is different than MLD where you can mux, and remux endlessly without doing deskew at intermediate points



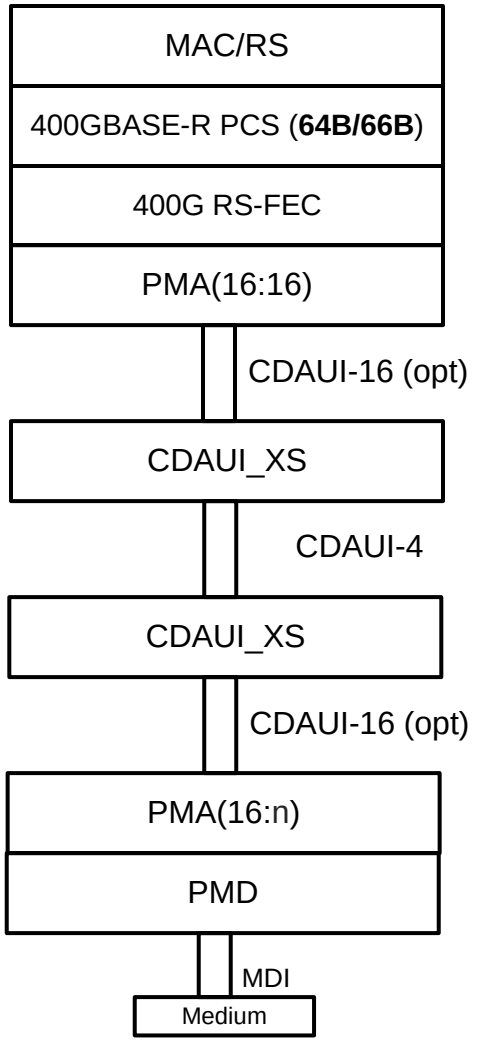
Stronger FEC

- With this architecture, if a stronger FEC is needed than the base FEC, you simply add the FEC on top of what is already there, no additional 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 again



Future AUI Example

- An extender sublayer can be used to add AUI specific FEC for a future AUI interface which requires a stronger FEC
- Extender sublayer would remove the RS-FEC parity bits, leave the transcoded data as is, and add a stronger FEC



Pros/Cons

- **Pros**
  - A lot of re-use from 100GbE 802.3bj, allows for compact 1x400GbE and 4x100GbE designs
  - Able to support 4xSR4 or other PMDs that require a medium weight FEC

- **Cons**
  - Always ~100ns of latency, and optimized 400G specific RS-FEC with similar gain could achieve ~ 50ns of latency

- **Reality**
  - This architecture won't cover all FEC or coding needs in the future, there will be PMD and AUI specific FECs and coding sublayers to be defined in the future (in a subordinate PCS sublayer)

Things to Look Into

- EEE interactions
- Support for OTN, what does that mean to the various options
- How to address 10x40G lanes? Is it needed?
- How does this fit in with the MLG protocol?

- Look more into the AM mapping functions…
- What is the BIP strategy?

Summary

- To be added

Thanks!