Views on the FEC Architecture Decision

802.3bs – 2015-07

David Ofelt – Juniper Networks

Supporters

- Pete Anslow - Ciena
- Thananya Baldwin - Ixia
- Mark Gustlin – Xilinx
- Martin Langhammer - Altera
- Mike Peng Li – Altera
- Jeffery Maki - Juniper
- Gary Nicholl – Cisco
- Mark Nowell – Cisco
- Jerry Pepper - Ixia
- Steve Trowbridge - ALU

Introduction

- There is a difference between architecture and implementation
  - Most of the (excellent) analysis so far has been on implementation
  - Few real high-level architectural differences between the approaches
- The standard documents the architecture
  - We attempt to place as few limitations on implementation
- The physical layer represents the most cost, difficulty, and opportunity
- Systems and Chips are distinct
  - A system frequently can make an assumption about what is and isn’t supported
  - A chip frequently supports multiple systems in multiple markets

Breakout

- Breakout is extremely important!
  - But breakout is actually more of a module topic
  - Real issue for host chips is co-existence of multiple rates
- Not sure why folks just focus on 4x100GbE & 1x400GbE
- Reality is:
  - 16x10GbE, 4x40GbE, Nx40GbE, 16x25GbE, 8x50GbE, 4x100GbE, 2x200GbE, 1x400GbE, Nx10GbE, Mx40GbE MLG1,2,3, etc, etc, (not exhaustive)
  - May also have OTN, FlexE, fibre channel, Interlaken, and/or fabric!
- A perfect choice for clocking and datapath width almost certainly doesn’t exist.

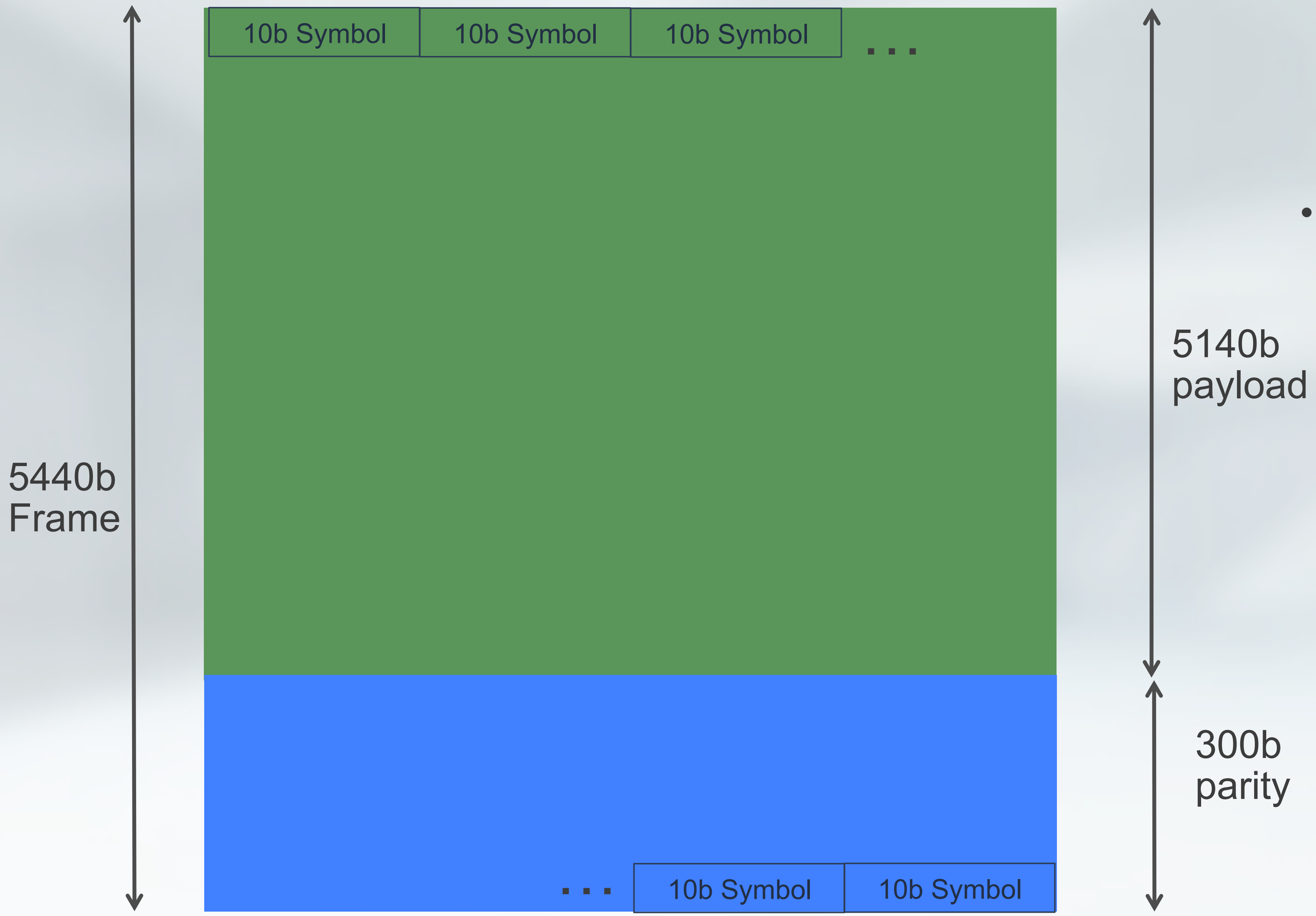
Architectural differences

- Not many fundamental architectural differences between the proposals
  - What is the flexibility to reorder lanes?
  - What is the base latency?
  - Where does the FEC frame’s data come from?
  - What is the FEC performance?
  - Can you can build a 400GE using 100Gb/s external FECs?

Lane Reordering & Latency

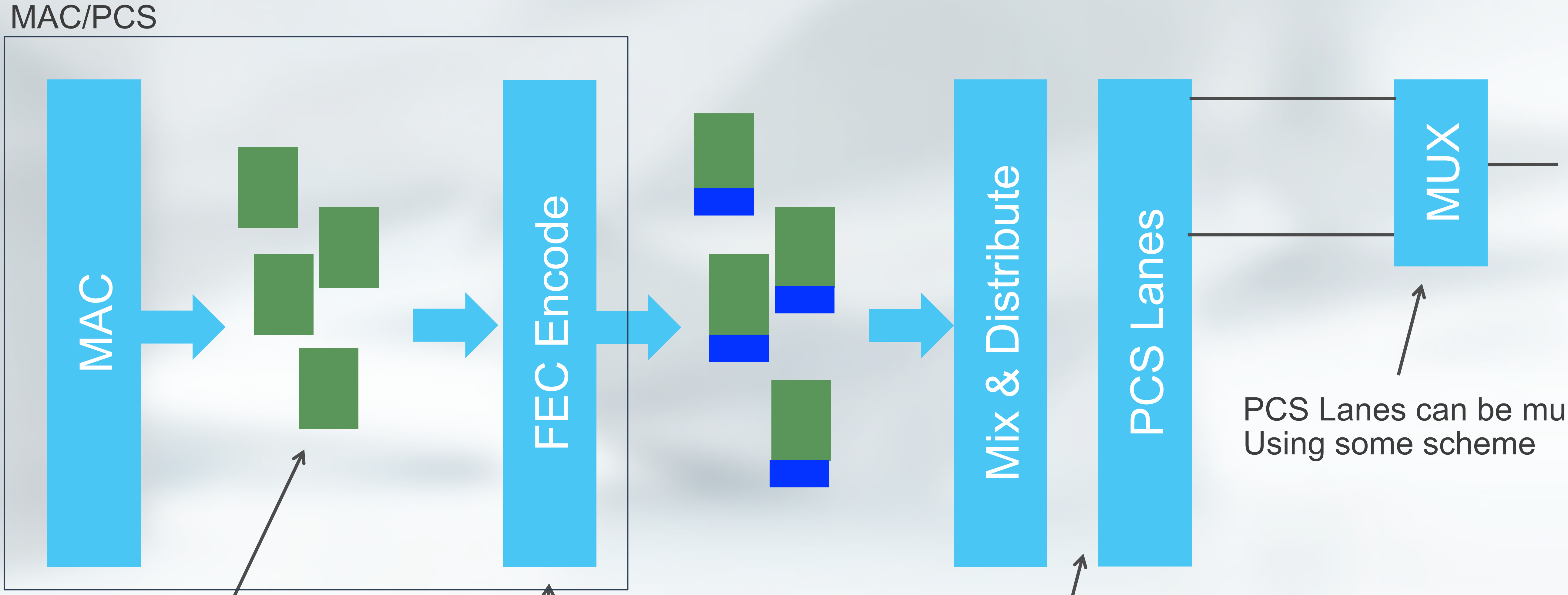
- Few restrictions on lane reordering allows for more freedom for the:
  - ASIC
  - Board
  - Module
  - Optics
- Many of the presentations gloss over the lane reordering limitations
  - Which SERDES contribute to which FEC frames?
  - In the 4x100G case- can the 100G slices be in any order?
  - Data from all SERDES needs to converge on the MAC
- Latency
  - There is a difference of ~40ns between 1x400 and 4x100
  - Less important than for slower interfaces

FEC Frame Review



- KP4
  - 5140b of payload
  - 300b of parity
  - 5440b total frame
  - Can correct any 15 10b symbols
- FEC Encode/Decode operation is datastream agnostic

Block Diagram



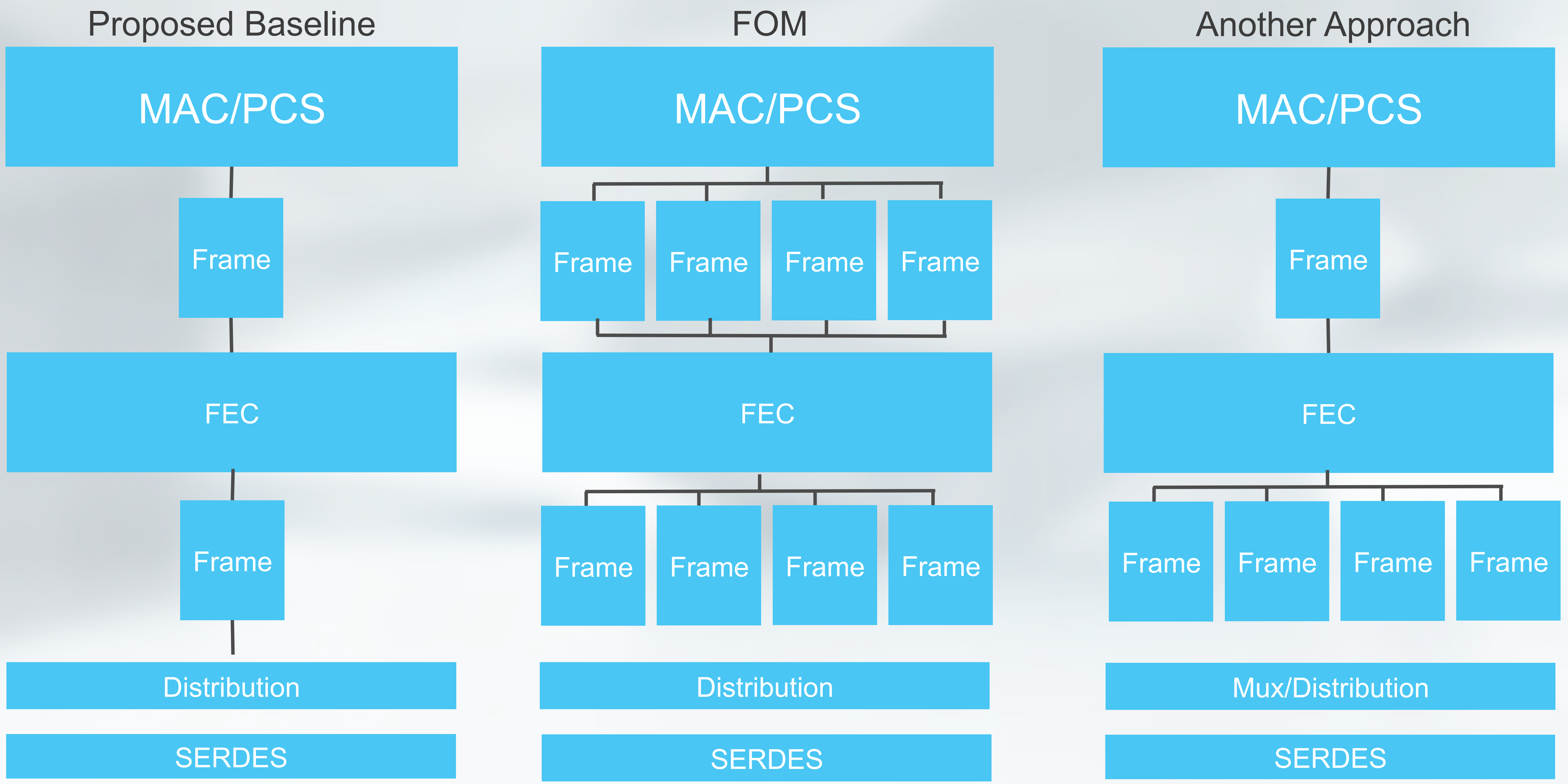
PCS Datastream Broken
into FEC Blocks

FEC Blocks Encoded and
parity added
at 78M frames/s

Some number of frames
Mixed and distributed to
PCS lanes

PCS Lanes can be multiplexed
Using some scheme

Differences between 1x400 and 4x100



Where does the FEC frame's data come from?

- This is the actual architectural difference between 1x400 and 4x100 choices
- FEC encode/decode is just a block that handles N frames/second
  - The 4x100 option can be implemented by a 1x400 FEC
  - The 1x400 option can be implemented by a 4x100 FEC
  - Both can be implemented by a 3x133 FEC
  - Your personal implementation constraints determine the best approach for your design
- Real difference is how the FEC frames are built
  - Data from which SERDES comes together to form a frame?
  - How is that data interleaved to form the frame?
  - How are the frames reassembled to form the PCS stream?

FEC Performance

- FEC is the single most costly part of the MAC/PCS logic
  - Need to make the best use of the investment
  - Simplifications that make the host design easier that waste FEC performance is a very poor tradeoff.
- anslow_3bs_05_0715 shows significant benefit to using a single FEC
  - BER Headroom above 1e-13 is critical for many markets
  - May allow for relaxation for PMD parameters
  - May allow for more interesting PMD implementations

Implementation Discussion

- Several vendors already shipping 400Gb/s NPUs in 28nm
  - Main NPU forwarding & datapaths are significantly harder than the MAC/PCS
- Very good work showing microarchitectural and implementation details
  - Narrow implementation choices analyzed – results generalized
    - Real design space is significantly larger – ex. Channelized MAC/PCS make for different results
  - My conclusion is that there isn't a significant difference in cost or complexity between the options.
    - Magnitude of differences are what I'd expect between two different designers
- Can easily implement both options in a mid-range current-generation FPGA
  - This means ASIC implementations are trivial
  - FPGAs have ~1year process advantage but
    - Have 5-10x gate density disadvantage
    - Have a 2-4x clock frequency disadvantage
  - Speed-grade differences either nonexistent or minor

Implementation Discussion Cont.

- KP4 is a superset of KR4
  - Supporting KR4 with a KP4 FEC is essentially free
  - So no implication on breakout nor multi-rate co-existence
- We are on our 4th generation of 100GbE designs
  - All differ in many areas due to design constraints of each device
  - Clocking, partitioning, datapath sizing, etc frequently differ
  - All are a mess due to the variety of interfaces that need to be supported
  - None had “half cycle issues” that weren’t rounding error
  - Clock frequency and datapath widths are often constraints rather than free variables

Implementation Discussion Cont. Cont.

- Fewer, larger things can evolve better then a collection of smaller things
- If 25GbE existed before we did the 802.3bj 100GbE interfaces…
  - Current arguments would lead to a call for 100GbE FEC to be 4x25Gb/s
  - So 400GbE FEC would then be 16x25Gb/s
- 800GbE Generation would have 32x25Gb/s
- 1.6TbE generation would have 64x25Gb/s
- Structure necessary even if 25GbE not implemented
- If instead, we define each generation as a monolithic FEC:
  - Finer-grained versions only necessary if implemented
  - Older structures fall off the end

Thought Process

- Silicon is (very) cheap – physical layer devices aren't
  - Push as much complexity into the host chip as possible
  - Leave as much freedom as possible to the physical layer device
  - Provide for as many futures as practical
  - Use the logic provided to the fullest possibility
- Future is hard to predict & implementations vary dramatically
  - Architect in as few constraints as possible
  - An individual's view of how things must be built is likely wrong for all other parties
- First generation implementations should be possible
  - Future generations should be cheap

Summary of Choices

- Proposed Baseline : gustlin_3bs_02_0715.pdf
  - Single FEC frame distributed to all 16 SERDES
  - Good random error BER performance
  - Bit muxing between lanes
  - Lane order independence
- FOM : wang_400_01a_0114.pdf
  - Four FEC frames interleaved to subset of SERDES
  - Good burst error BER performance
  - Bit muxing between appropriate lanes
  - Some lane order limitations
- Other :
  - 4 (or 2) FEC frames interleaved and sent to all SERDES
  - Good burst error BER performance
  - FECs symbol interleaved to each PCS lane
    - Bit interleaving after initial symbol interleaving may work – analysis needs to be done
  - No lane order limitations

Recommendation

- Adopt the current baseline proposal gustlin_3bs_02_0715.pdf
- This:
  - Specifies the 802.3bs FEC as a monolithic 1x400Gb/s FEC
  - Distributes a single frame to all 16 PCS lanes
- Which:
  - leads to the fewest constraints on PCS implementations
  - leads to the greatest freedom lane ordering
  - makes the best use of the FEC "gain"
  - provides the simplest structure for dealing with future rates

Thank You!