100GE and 40GE PCS and MLD Proposal

IEEE 802.3ba March 2008 Orlando

Contributors and Supporters

David Law – 3com

Steve Trowbridge, Jesse Simsarian - Alcatel-Lucent

Brad Booth, Dimitrios Giannakopoulos, Francesco Caggioni, Keith Conroy – AMCC

Piers Dawe, Rita Horner – Avago

Howard Frazier - Broadcom

Arthur Marris – Cadence

Mike Shahine - Ciena

Mark Nowell, Gary Nicholl, Hugh Barrass - Cisco Systems

Steve Swanson - Corning

Med Belhadj – Cortina

Chris Cole - Finisar

Krishnamurthy Subramanian – Force10

Aris Wong, Shashi Patel, Bill Ryan – Foundry Networks

Ryan Latchman, Justin Abbott - Gennum

Hong Liu, Ashby Armistead – Google

Shinji Nishimura, Hidehiro Toyoda - Hitachi Ltd

Dan Dove – HP

Petar Pepeljugoski – IBM

John Jaeger, Andy Moorwood, Drew Perkins - Infinera

Jerry Pepper, Thananya Baldwin - Ixia

Faisal Dada, Jack Jewell, Mike Dudek - JDSU

Jeffery J. Maki, David Ofelt, Brad Turner - Juniper Networks

Martin White – Marvell

Pete Anslow, David W. Martin – Nortel

Osamu Ishida, Shoukei Kobayashi - NTT

Matt Traverso – Opnext

Farhad Shafai - Sarance Technologies

Farzin Firoozmand, Craig Hornbuckle – SMI

Ted Seely - Sprint

Kengo Matsumoto - Sumitomo Electric

Shimon Muller - Sun

Andre Szczepanek – TI

Martin Carroll - Verizon

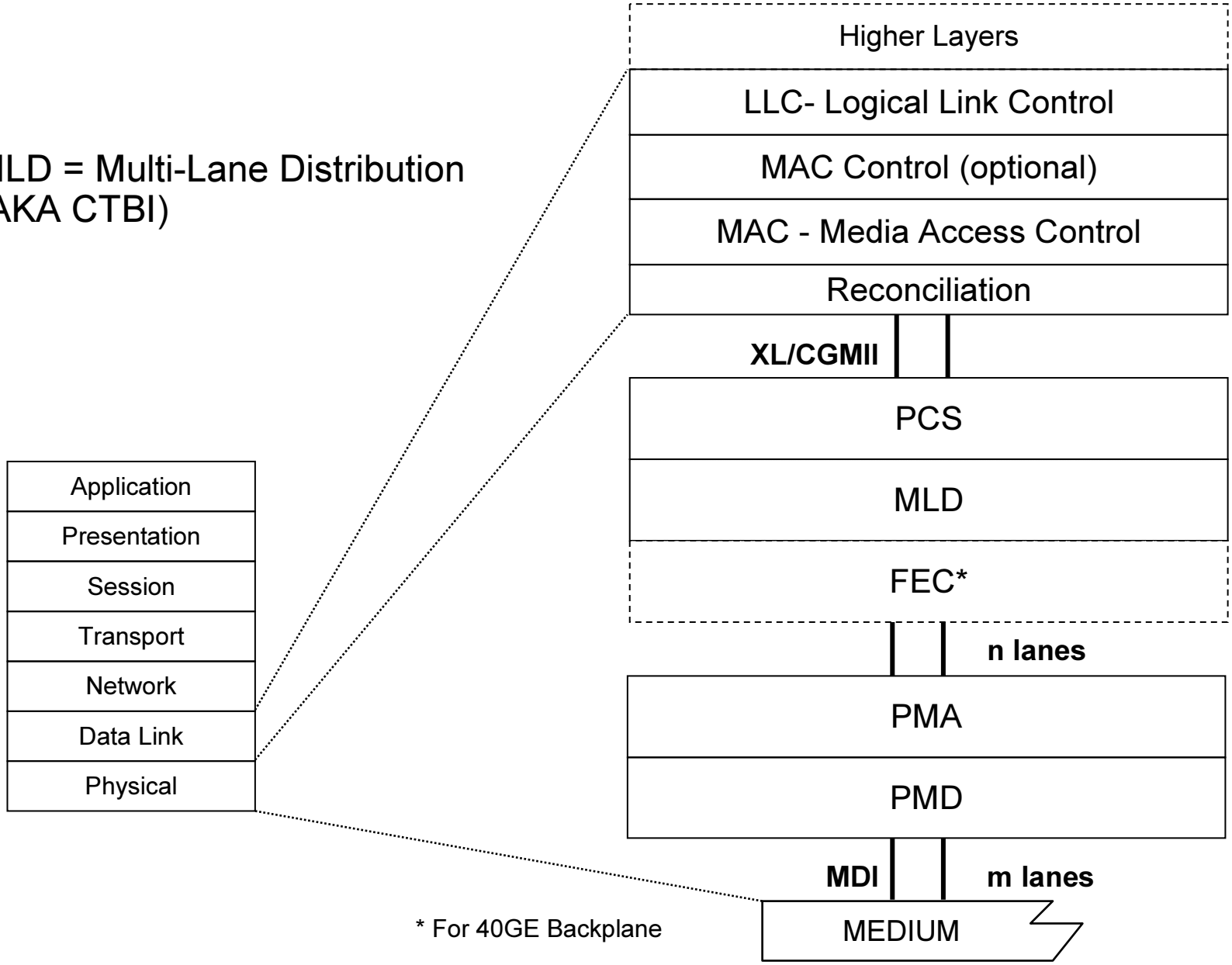
Frank Chang - Vitesse

Agenda

- 40GE/100GE Architecture
- PCS and MLD layer details
- Possible XL/CGMII Interface
- Alignment details
- Alignment performance metrics
- Clocking example
- Skew
- Future work items and summary

40GE/100GE Architecture

MLD = Multi-Lane Distribution
(AKA CTBI)

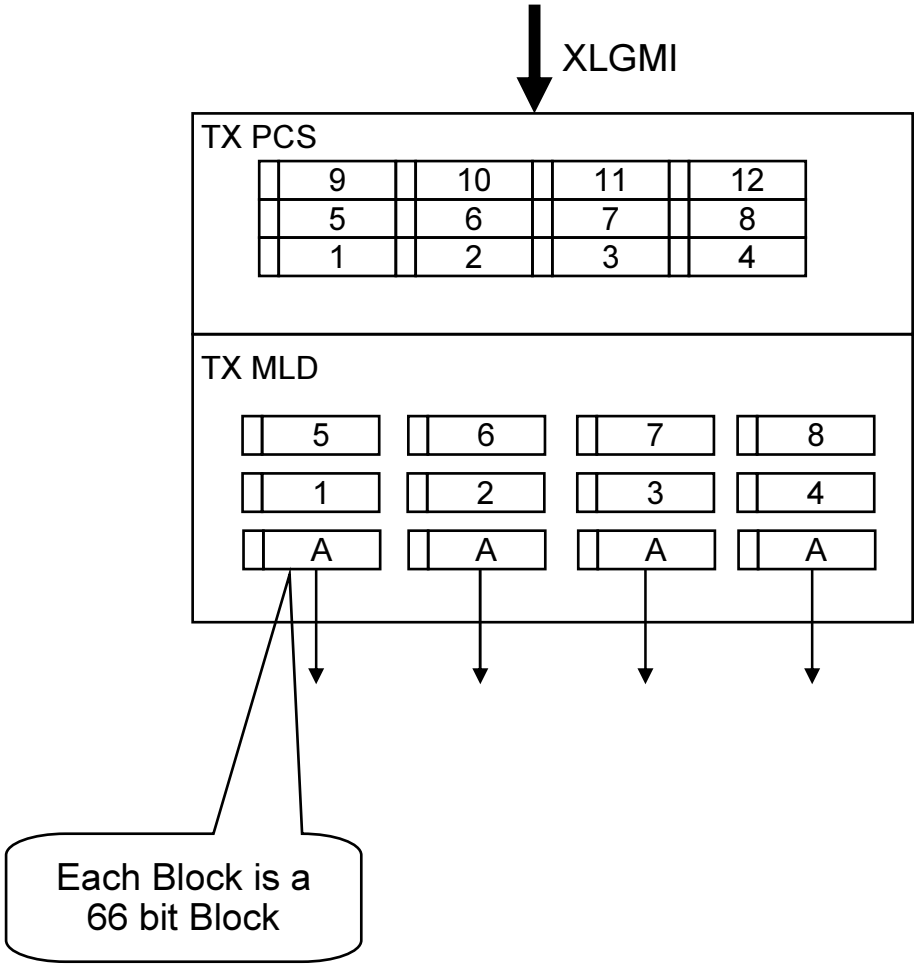


Proposed 100GE/40GE PCS and MLD Layer

- 10GBASE-R 64B/66B based PCS
  - Run at 100G or 40G aggregate rate
- n Lane MAC/PCS to PMA/PMD Electrical Interface
  - Ten Lanes for 100GE initially
  - Four Lanes for 40GE initially
  - Each lane runs at 10.3125G
  - Data is striped across the virtual lanes 66 bit blocks at a time (round robin)
  - Periodic alignment blocks are added to allow deskew
- Support m PMD lanes with the same PCS/MLD layer
- PMA maps n lane electrical interface to m lane PMD
  - PMA is simple bit level muxing
  - Does not know or care about PCS coding
- Alignment and skew compensation is done in the Rx MLD block only

Striping Mechanism

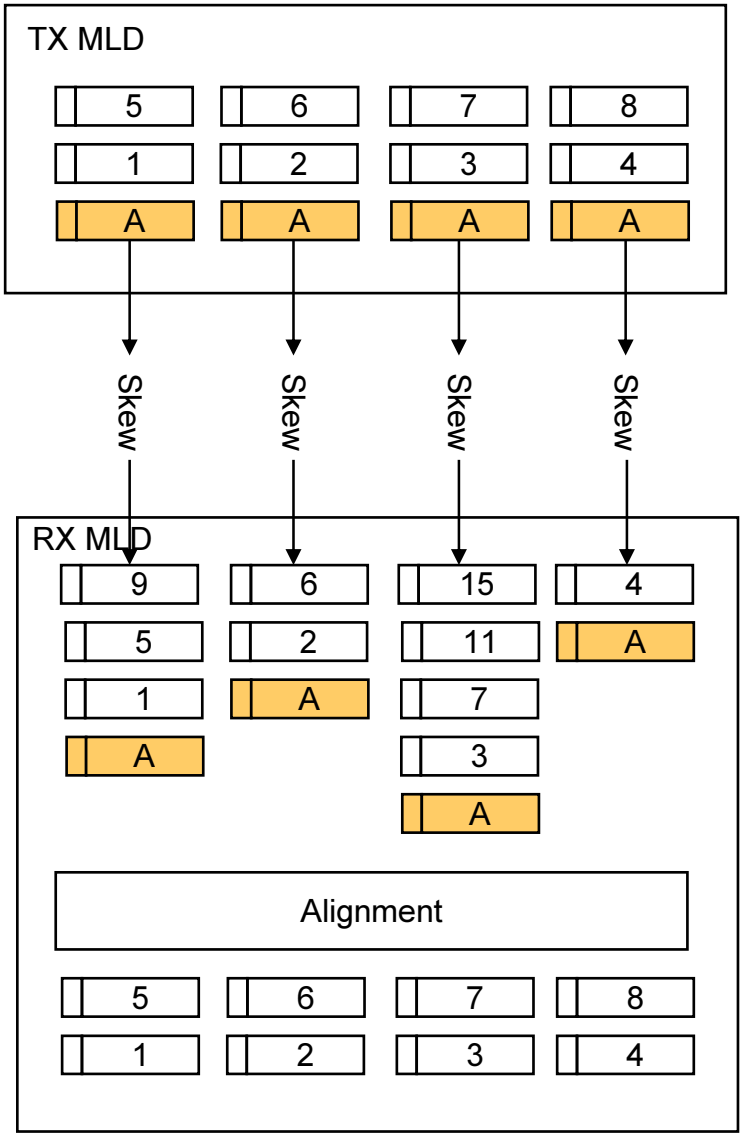
This example is 40GE with 4 electrical and 4 optical lanes



PCS Functions:
- 66 bit encoding
- Scrambling

MLD Functions:
- Alignment block addition periodically
- Round Robin block distribution

Alignment Mechanism – 40GE Example



RX MLD Functions:

- Re-Align 66 bit blocks
- Remove the Alignment blocks

Key Concept – Virtual Lanes

- **This is only needed when the number of Electrical (n) and PMD (m) lanes are not equal**

  If an interface will evolve so that n != m then VLs make the transition easy

- **Data from the MAC is first encoded into a continuous stream of 64B/66B blocks (100G or 40G aggregate stream).**
- **The 100G aggregate stream is split into a number of 'virtual lanes', also based on 64B/66B blocks**
- **An alignment block is added to each virtual lane**

  Sent infrequently

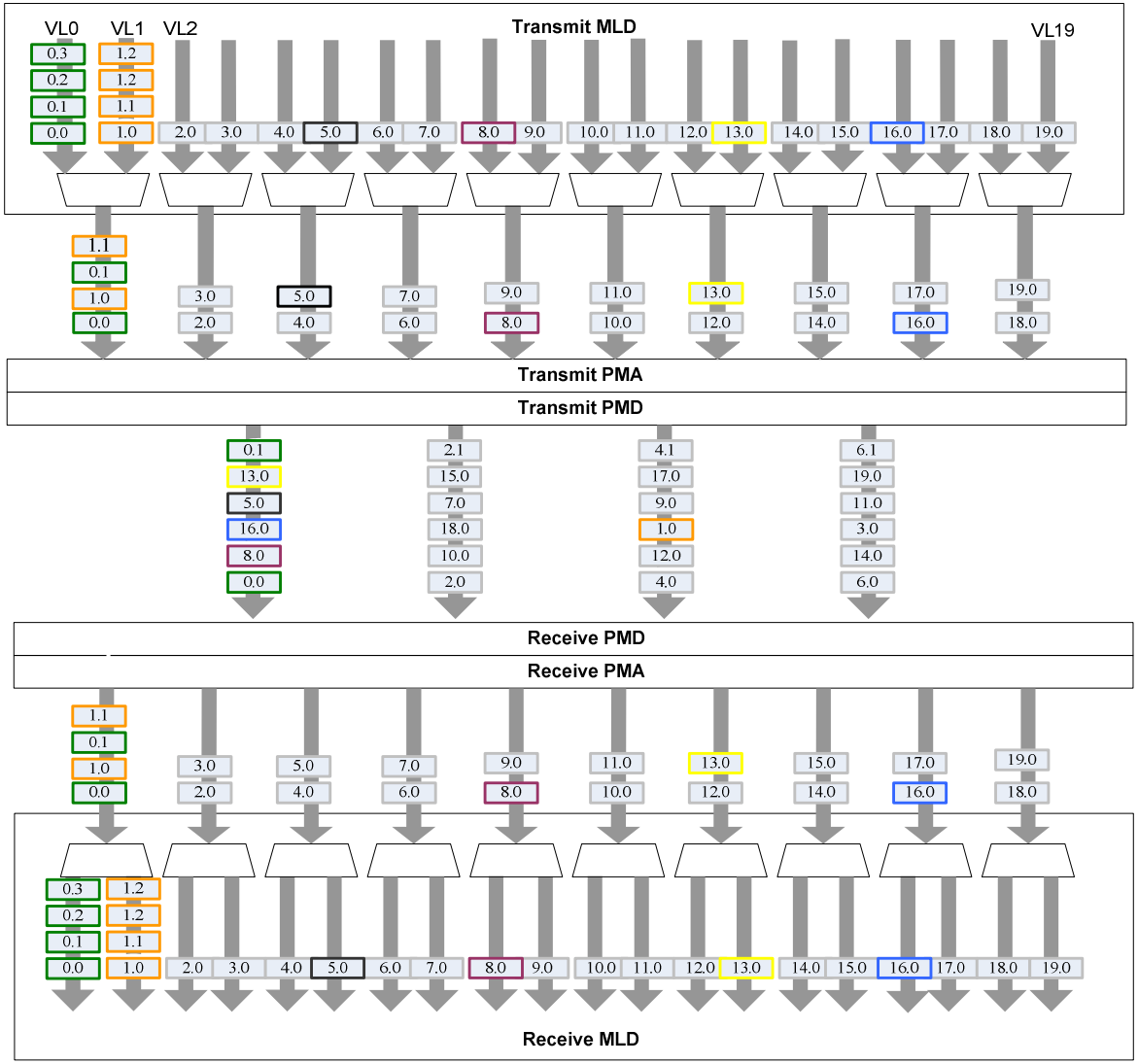
- **The number of virtual lanes generated is scaled to the Least Common Multiple (LCM) of the n lane electrical interface and the m lane PMD**

  This allows all data (bits) from one virtual lane to be transmitted over the same electrical and optical lane combination

  This ensures that the data from a virtual lane is always received with the correct bit order at the Rx MLD

- **The virtual lane marking allows the Rx MLD to perform skew compensation, realign all the virtual lanes, and reassemble a single 100G or 40G aggregate stream (with all the 64B/66B blocks in the correct order)**

Bit Flow Through – 100GE 4 lane PMD



- 20 VLs
- 10 Electrical lanes
- 4 Optical lanes
- With Skew, VLs move around
- RX MLD puts things back in order

How Many Virtual Lanes for 40GE?

- For each PMD objective, what is the number of lanes being considered?

  All can evolve to fewer lanes in the future

- Support at least 100m on MMF

  4 fibers

- Support at least 10m over a copper cable assembly

  4 lanes

- Support at least 1m over a backplane

  4 lanes

- Number of "Virtual Lanes" = Number of Lanes

  Just simple 66 bit block striping

Number of Electrical Lanes	Supportable PMDs	Virtual Lanes Needed
4, 2, 1	1, 2, 4	4

- With 4 VLs, all combinations of the above are possible

How Many Virtual Lanes for 100GE?

- For each PMD objective, what is the number of lanes being considered?

  All can evolve to less lanes in the future

- Support at least 10km on SMF

  4 wavelengths

- Support at least 100 meters on OM3 MMF

  10 fibers

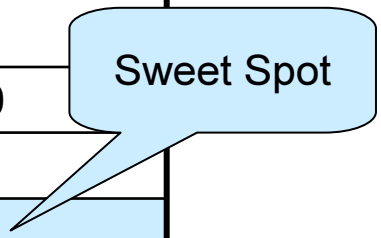
- Support at least 40-km on SMF

  4 wavelengths

- Support at least 10m over a copper cable assembly

  10 lanes

Number of Electrical Lanes	Supportable PMD Lane Widths	Virtual Lanes Needed
10, 5, 4, 2, 1	1, 2, 3, 4, 5, 6, 8, 10, 12	120
10, 5, 4, 2, 1	1, 2, 3, 4, 5,10	60
10, 5, 4, 2, 1	1, 2, 4, 5, 10	20
10, 5, 2, 1	1, 2, 5, 10	10



Do We Always Need Alignment Markers?

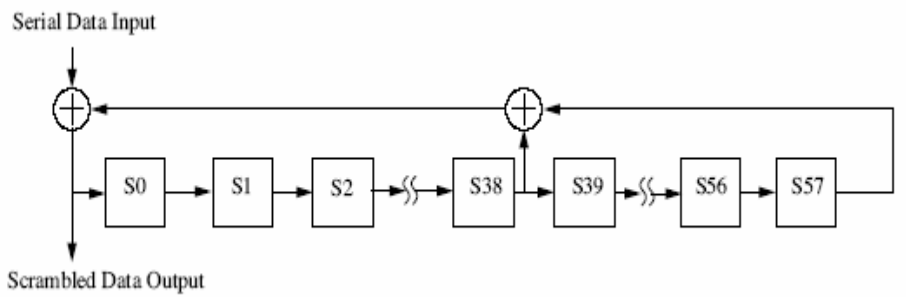
- For interfaces where we have little skew, and where the number of electrical lanes always equals the number of PMD lanes; can we get by with alignment based on the 66b sync field only?
- A prime example might be 40GE backplane
- Trade off to be made: is it easier to always have the alignment markers, or to control the skew tightly enough (32b)?
- Also is it easier to always have the alignment markers instead of supporting multiple modes in the same device?
  - Multiple modes actually might complicate the standard, and might complicate some implementations with multiple modes to test and verify?
- In today's FPGAs we likely can't constrain the skew enough to work with just sync field deskew?

PCS

- Same 10GBASE-R PCS (Clause 49), just running at 40Gbps or 100Gbps
- Same Control Block encoding, same scrambler

  With 8B alignment we don't use all of the block types

Input Data	Sync	Block Payload							
Bit Position: / Data Block Format:	0 1	2							65
$D_0 D_1 D_2 D_3/D_4 D_5 D_6 D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		Block Type Field							
$C_0 C_1 C_2 C_3/C_4 C_5 C_6 C_7$	10	0x1e	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$C_0 C_1 C_2 C_3/O_4 D_5 D_6 D_7$	10	0x2d	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
$C_0 C_1 C_2 C_3/S_4 D_5 D_6 D_7$	10	0x33	C_0	C_1	C_2	C_3		D_5	D_6	D_7
$O_0 D_1 D_2 D_3/S_4 D_5 D_6 D_7$	10	0x66	D_1	D_2	D_3	O_0		D_5	D_6	D_7
$O_0 D_1 D_2 D_3/O_4 D_5 D_6 D_7$	10	0x55	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
$S_0 D_1 D_2 D_3/D_4 D_5 D_6 D_7$	10	0x78	D_1	D_2	D_3	D_4		D_5	D_6	D_7
$O_0 D_1 D_2 D_3/C_4 C_5 C_6 C_7$	10	0x4b	D_1	D_2	D_3	O_0	C_4	C_5	C_6	C_7
$T_0 C_1 C_2 C_3/C_4 C_5 C_6 C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0 T_1 C_2 C_3/C_4 C_5 C_6 C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0 D_1 T_2 C_3/C_4 C_5 C_6 C_7$	10	0xaa	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0 D_1 D_2 T_3/C_4 C_5 C_6 C_7$	10	0xb4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0 D_1 D_2 D_3/T_4 C_5 C_6 C_7$	10	0xcc	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0 D_1 D_2 D_3/D_4 T_5 C_6 C_7$	10	0xd2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0 D_1 D_2 D_3/D_4 D_5 T_6 C_7$	10	0xe1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0 D_1 D_2 D_3/D_4 D_5 D_6 T_7$	10	0xff	D_0	D_1	D_2	D_3	D_4	D_5	D_6	



XL/CGMII Interface

- Leverage XGMII

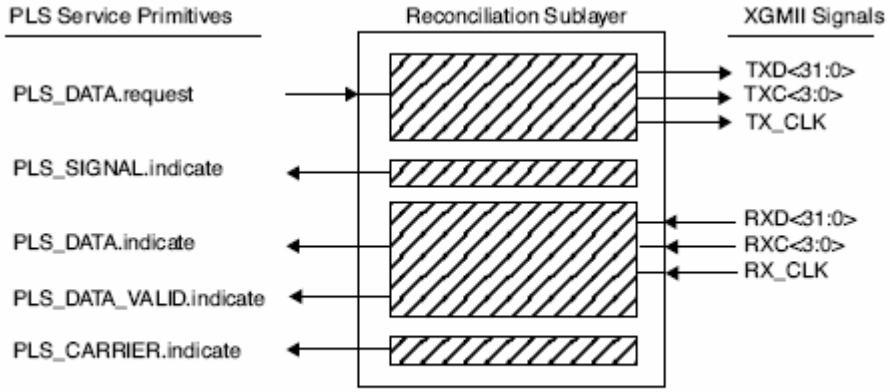


Figure 46–2—Reconciliation Sublayer (RS) inputs and outputs

- Make it scalable
- Proposal: 64 bit interface, logical interface
- TXD<63:0>, TXC<7:0>, RXD<63:0>, RXC<7:0>
- Interface is naturally scaled based on speed targets in an implementation
- New Requirement: Only start packets on 8B boundary

  Simplifies MAC design, idle deletion etc.
- Use deficit counter to adjust idle to an average of 12B

Alignment Proposal

- Send alignment on a fixed time basis
- Alignment word also identifies virtual lanes
- Sent every 16384 66bit blocks on each virtual lane at the same time
  - ~216usec for 20 VLs @ 100G
  - ~108usec for 4 VLs @ 40G
- It temporarily interrupts packets
- Takes only 0.006% (60PPM) of the Bandwidth
- Rate Adjust FIFO will delete enough IPG so that the MAC still runs at 100.000G or 40.000G with the interface running at 10.3125G

Alignment Word Proposal

Requirements:

- Significant transitions and DC balanced – word is not scrambled
- Keep in 66 bit form, but no relation to 10GBASER is needed
- But why not keep it close? – Because of the clock wander concerns
- Contains Virtual Lane Identifier

Proposed Alignment Word

- This is DC balanced
- No relationship to the normal 10GBASE-R blocks
- Added after and removed before 64/66 processing
- Alignment block is periodic, no Hamming distance concerns with 64/66 block types

Alignment Word Proposal

The encoding of the VL markers is as follows (based on $x^{58} + x^{39} + 1$ scrambler output):

VL Number	32 Bit encoding	VL Number	32 Bit encoding
0	C1,68,21,F4	10	FD, 6C, 99, DE
1	9D, 71, 8E, 17	11	B9, 91, 55, B8
2	59, 4B, E8, B0	12	5C, B9, B2, CD
3	4D, 95, 7B, 10	13	1A, F8, BD, AB
4	F5, 07, 09, 0B	14	83, C7, CA, B5
5	DD, 14, C2, 50	15	35, 36, CD, EB
6	9A, 4A, 26, 15	16	C4, 31, 4C, 30
7	7B, 45, 66, FA	17	AD, D6, B7, 35
8	A0, 24, 76, DF	18	5F, 66, 2A, 6F
9	68, C9, FB, 38	19	C0, F0, E5, E9

Finding VL Alignment

- After reception in the rx MLD, you have x VLs, each skewed and transposed
- First you find 66bit alignment on each VL

  Each VL is a stream of 66 bit blocks

  Same mechanism as 10GBASE-R (64 valid 2 bit frame codes in a row)
- Then you hunt for alignment on each VL

  Look for one of the 20 VL patterns repeated and inverted
- Alignment is declared on each VL after finding 2 consecutive non-errored alignment patterns in the expected locations (16k words apart)
- Out of alignment is declared on a VL after finding 4 consecutive errored frame patterns
- Once the alignment pattern is found on all VLs, then the VLs can be aligned

Alignment Performance Parameters – 100GE

- Mean Time To Alignment (MTTA)

  Mean time it takes to gain Alignment on a lane or virtual lane for a given BER

  Nominal time = 314usec

- Mean Time To Loss of Alignment (MTTLA)

  Mean time it takes to lose Alignment on a lane or virtual lane for a given BER

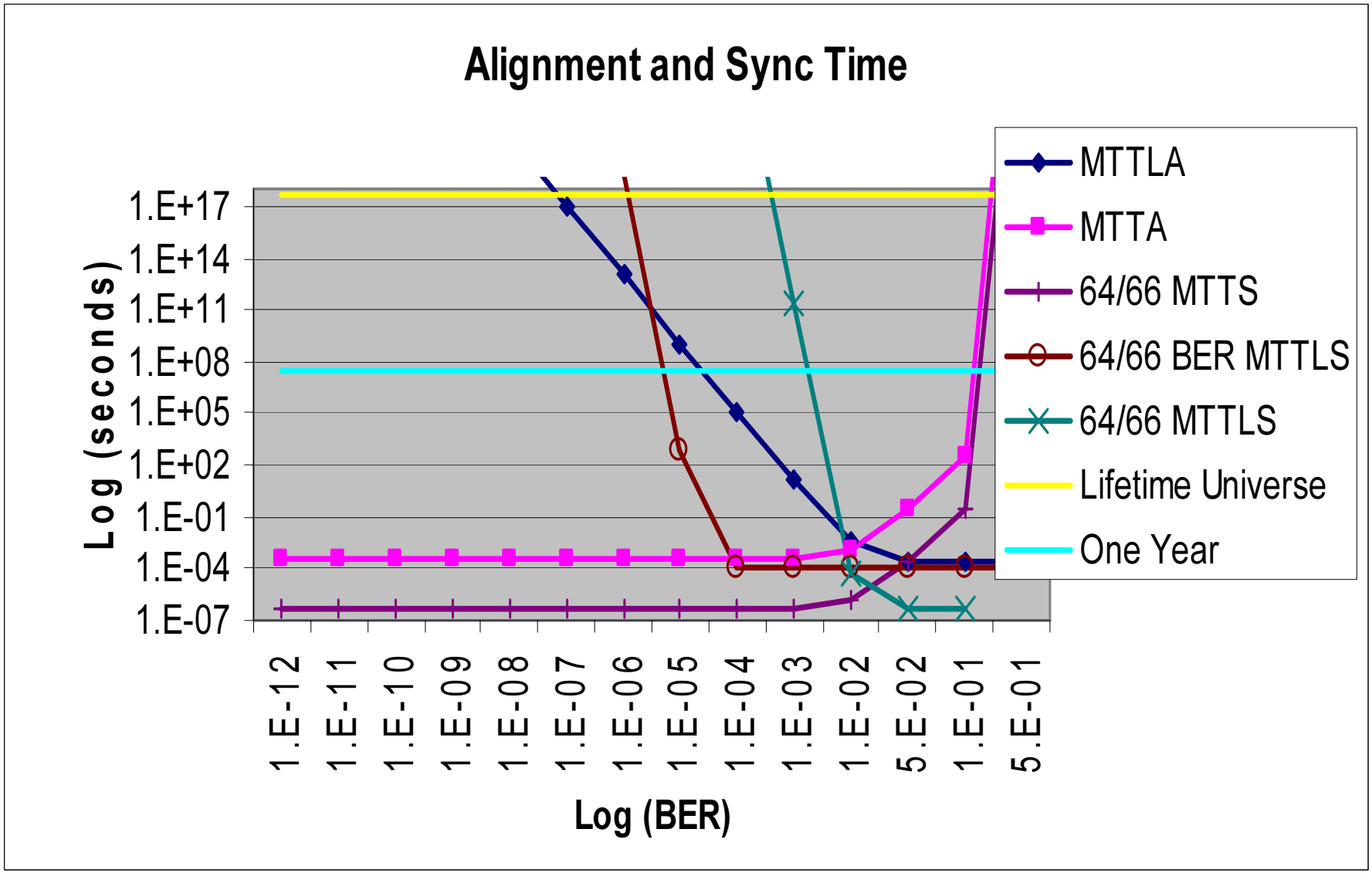
- Probability of False Alignment (PFA) = 3 E-40
- Probability of Rejecting False Alignment (PRFA) = ~1
- Also have 64/66 sync stats on the graph for comparison

  MTTS – Mean Time To Sync (64 non errored syncs in a row)

  BER MTTLS – With the 125usec BER window, what is the Mean Time To Lose Sync

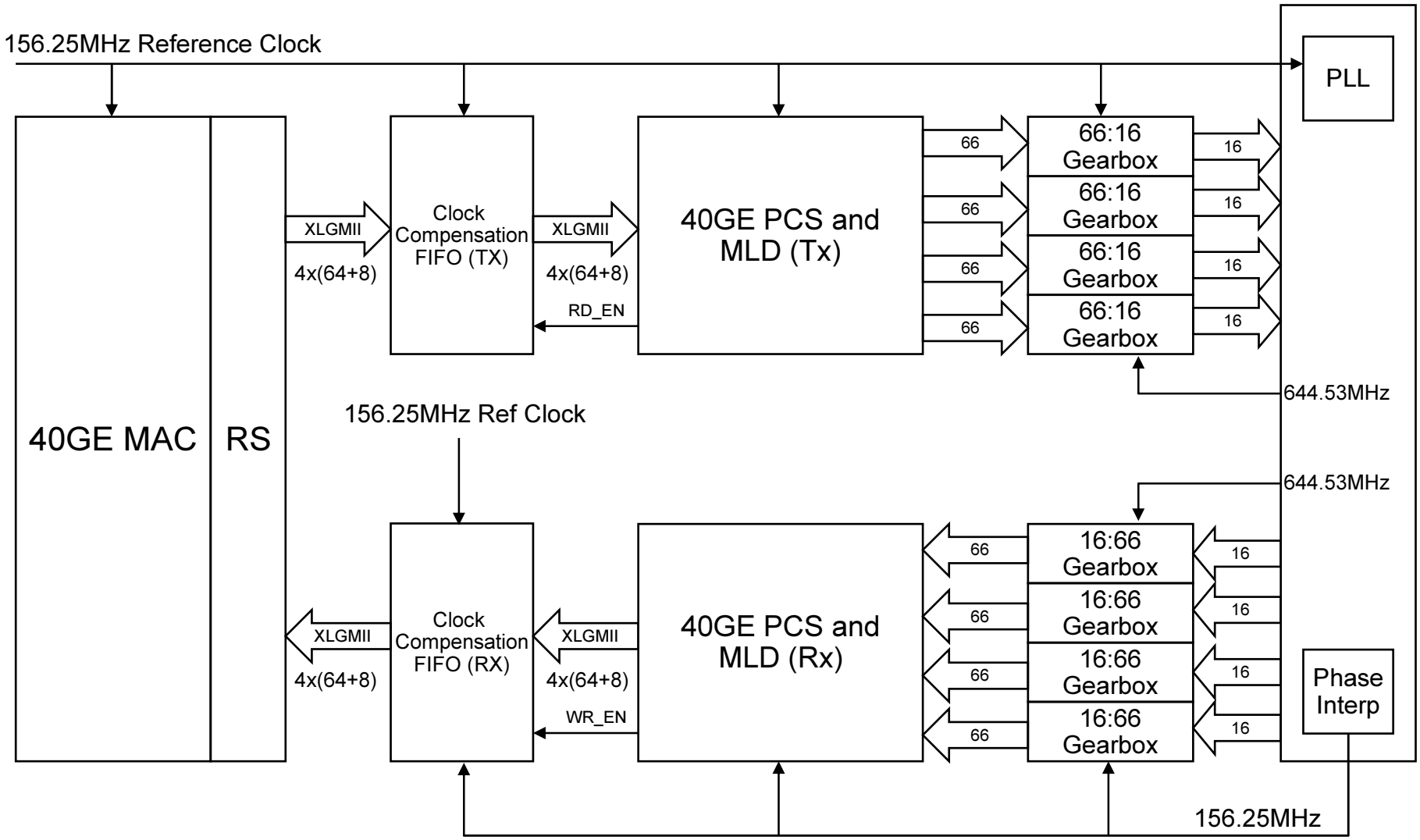
  MTTLS - Mean Time To Lose Sync

Alignment Performance Parameters – 100GE



40GE Alignment Performance will be similar

Clocking Example – 40GE



Dynamic Skew Handling

- This is a concern only for PMDs where we multiplex data when n != m
- We need to Find the maximum dynamic skew for the applicable PMDs
- For these PMDs, retiming buffers are used to handle the variable skew in the TX PMA and RX PMA
- Value of the dynamic skew is dependent on the technology, from a few bits to 10s of bits
- Anslow_01_0907:
  - 0.5 to 18 bits for 10km (depending on LAN WDM vs. CWDM)
  - 2 to 5 bits for 40km (LAN WDM)
- Add in a few more bits for electrical functions

Total Static Skew Numbers

- We need to add up all of the skew to see how much total static skew must be compensated for at the receiver
- TX Electrical
- TX PMD
- Optical Medium
- RX PMD
- RX Electrical

- Note that 10nsec at 100Gbps = 1kbit of memory
- Numbers depend on what technology is ultimately used!
- We should allow for FPGAs in our skew numbers

Work Items

- Determine Total Skew budget
- Determine Dynamic Skew budget
- Complete the MTTFPA analysis
- State Machines
- Fault Indications

Summary

- Simple 10GBASE-R based PCS
- MLD layer to support multiple physical lanes/lambdas
- Complexity is low within the MLD layer
  - Simple block data striping
- Complexity in the optical module is low
  - Simple bit muxing even when m != n
- Based on proven 64B/66B framing and scrambling
- Electrical interface is feasible at 10x10G or 4x10G
- Allows for a MAC rate of 100.000G or 40.000G
  - Overhead very low and independent of packet size
- Supports an evolution of optics and electrical interfaces