The Technical Feasibility of a 100GE PCS and Electrical Interface

Oded Trainin, Mark Gustlin and Gary Nicholl - Cisco Systems

IEEE HSSG January 2007

Supporters

- Subi Krishnamurthy – Force10 Networks
- Chris Cole – Finisar
- Med Belhadj – Cortina Systems
- Brad Booth – AMCC
- Frank Chang – Vitesse
- Arthur Marris – Cadence

Outline

- **The following slides present a possible 100GE PCS and PMA/PMD interface which enables a simple bit muxing function at the PMA**
- **The presentation includes the following parts:**
  - **Motivation**
  - **Overview**
  - **Key concept of virtual lanes**
  - **100GE PCS concept**
  - **Alignment mechanism**

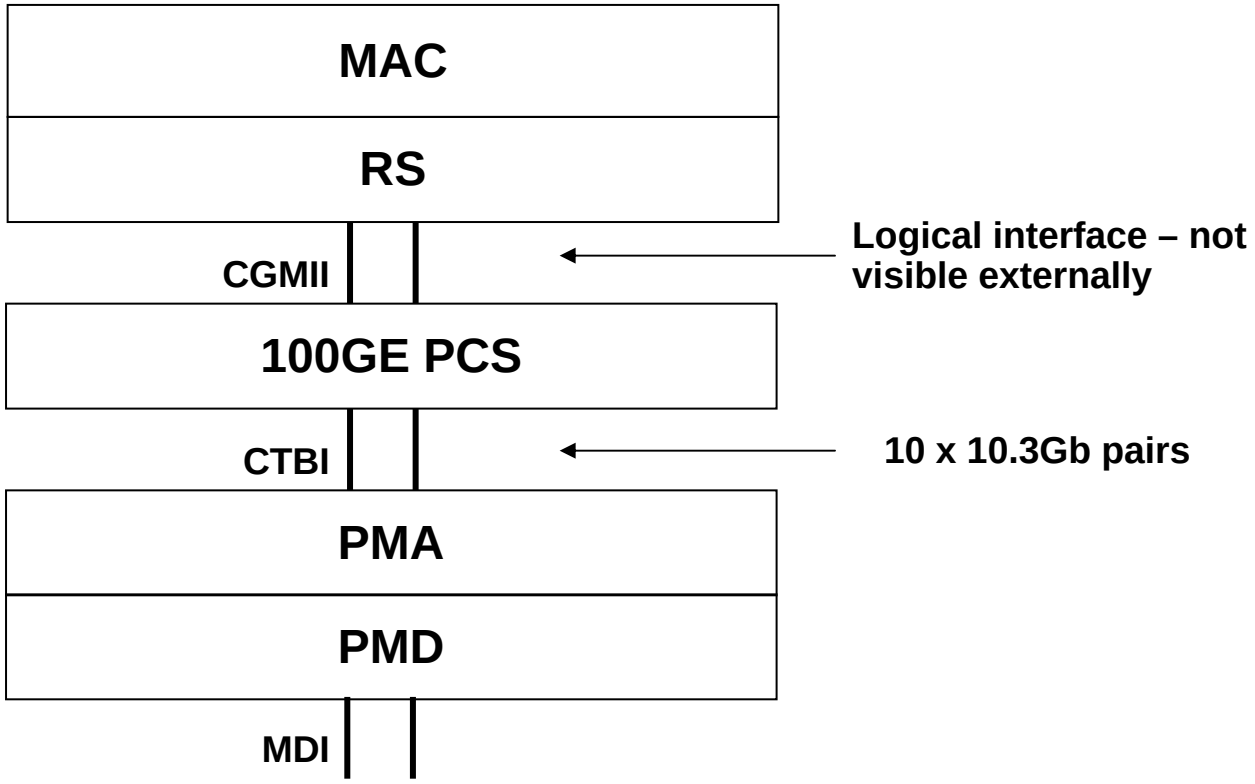
Motivation

- **Enable a simple Optical Module (PMA/PMD)**
  - **Allow for a simple PMA (bit level muxing only)**
    - **Enables non CMOS PMA (some PMD lane speeds likely faster than CMOS can handle)**
  - **Reduces the cost**
- **Architecture that is tolerant to PMD technology advances and maturity**
  - **Single PCS for current and most future PMDs**

Overview

- 64B/66B based PCS (100G aggregate)
- 10 Lane MAC/PCS to PMA/PMD Electrical Interface (CTBI)
  - Each lane runs at 10.3125G
  - 100G data is inverse muxed across the CTBI lanes
- Support 1-12 PMD lanes all with the same electrical Interface and PCS
- PMA maps 10 lane CTBI to n 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 PCS only
- Assumes the same coding can be used for the CTBI and the PMD

100GE Layering



Key Concept – Virtual Lanes

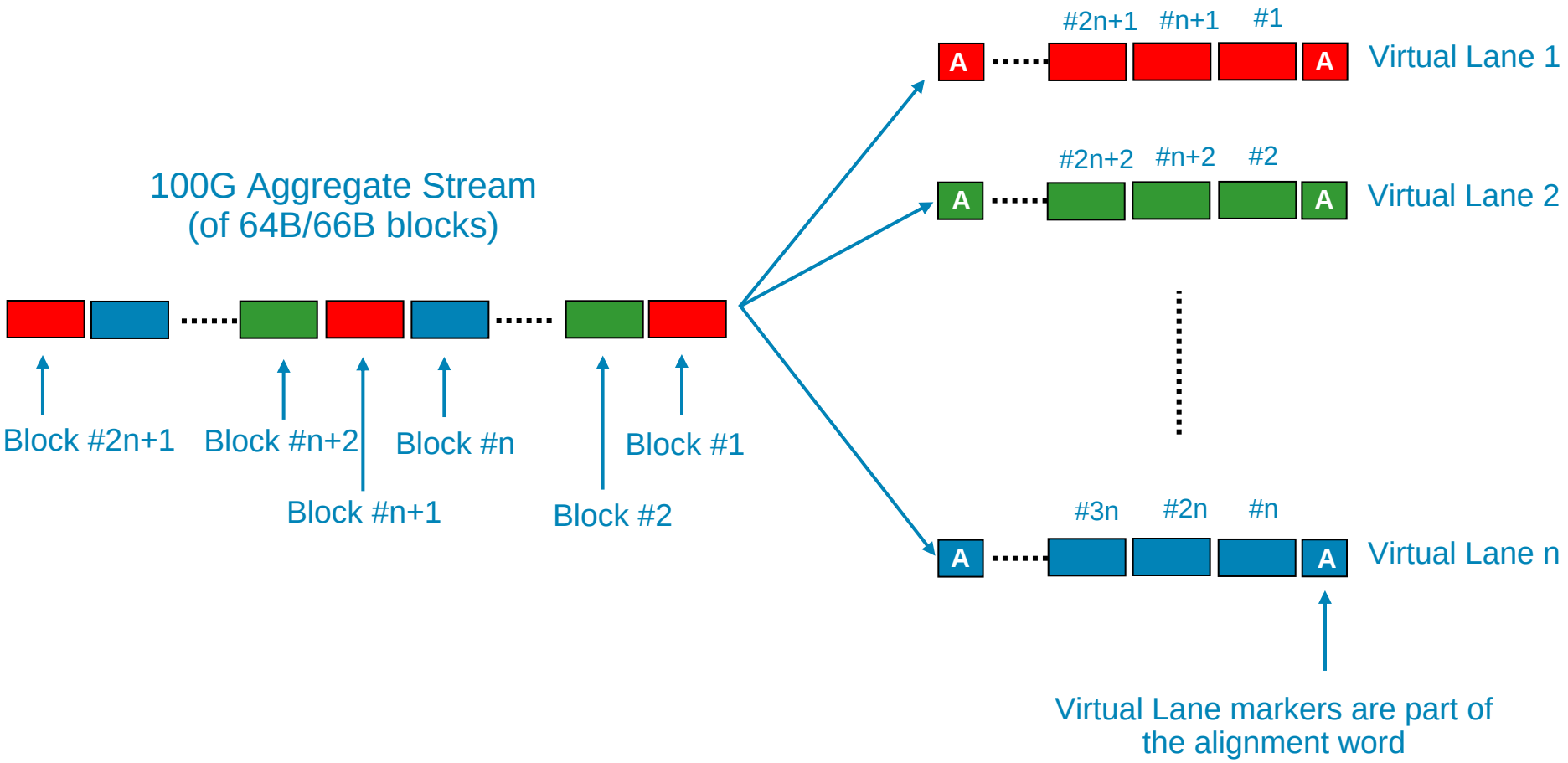
- Data from the MAC is first encoded into a continuous stream of 64B/66B blocks (100G aggregate stream).
- The 100G aggregate stream is split into a number of ‘virtual lanes’, also based on 64B/66B blocks
- Each virtual lane is assigned a unique marker (as part of an alignment block)
- The number of virtual lanes generated is scaled to the Least Common Multiple (LCM) of the 10 lane CTBI interface and the n lane PMD
  - This allows all data (bits) from one virtual lane to be transmitted over the same CTBI and PMD lane combination
  - This ensures that the data from a virtual lane is always received with the correct bit order at the Rx PCS
- Although the data for a particular virtual lane always follows the same path, different virtual lanes follow different paths, and therefore can arrive at the RX PCS skewed
- The virtual lane marking allows the Rx PCS to perform skew compensation, realign all the virtual lanes, and reassemble a single 100G aggregate stream (with all the 64B/66B blocks in the correct order)

Key Concept – Virtual Lanes

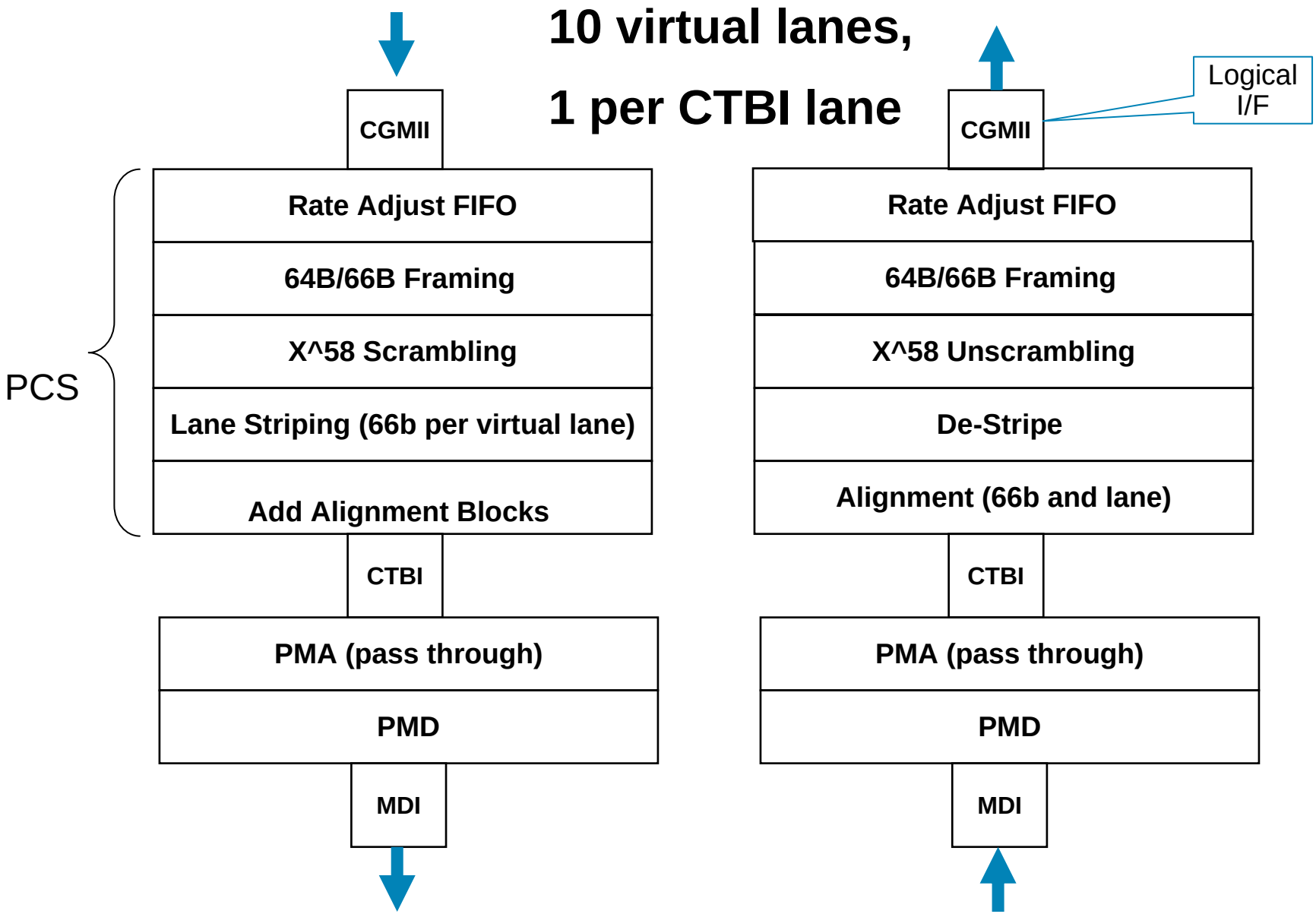
- In the Tx PCS the virtual lanes are distributed to the 10 CTBI lanes in a round robin fashion (i.e. for 10 virtual lanes there will be one per CTBI lane, for 20 virtual lanes there will be two per CTBI lane, for 30 virtual lanes there will be 3 per CTBI lanes, etc)
- In the TX PCS the virtual lanes assigned to a given CTBI lane are then bit interleaved before being transmitted.

CTBI Lanes	PMD Lanes	LCM	Virtual Lanes
10	12	60	60
10	10	10	10
10	8	40	40
10	5	10	10
10	4	20	20
10	3	30	30
10	2	10	10
10	1	10	10

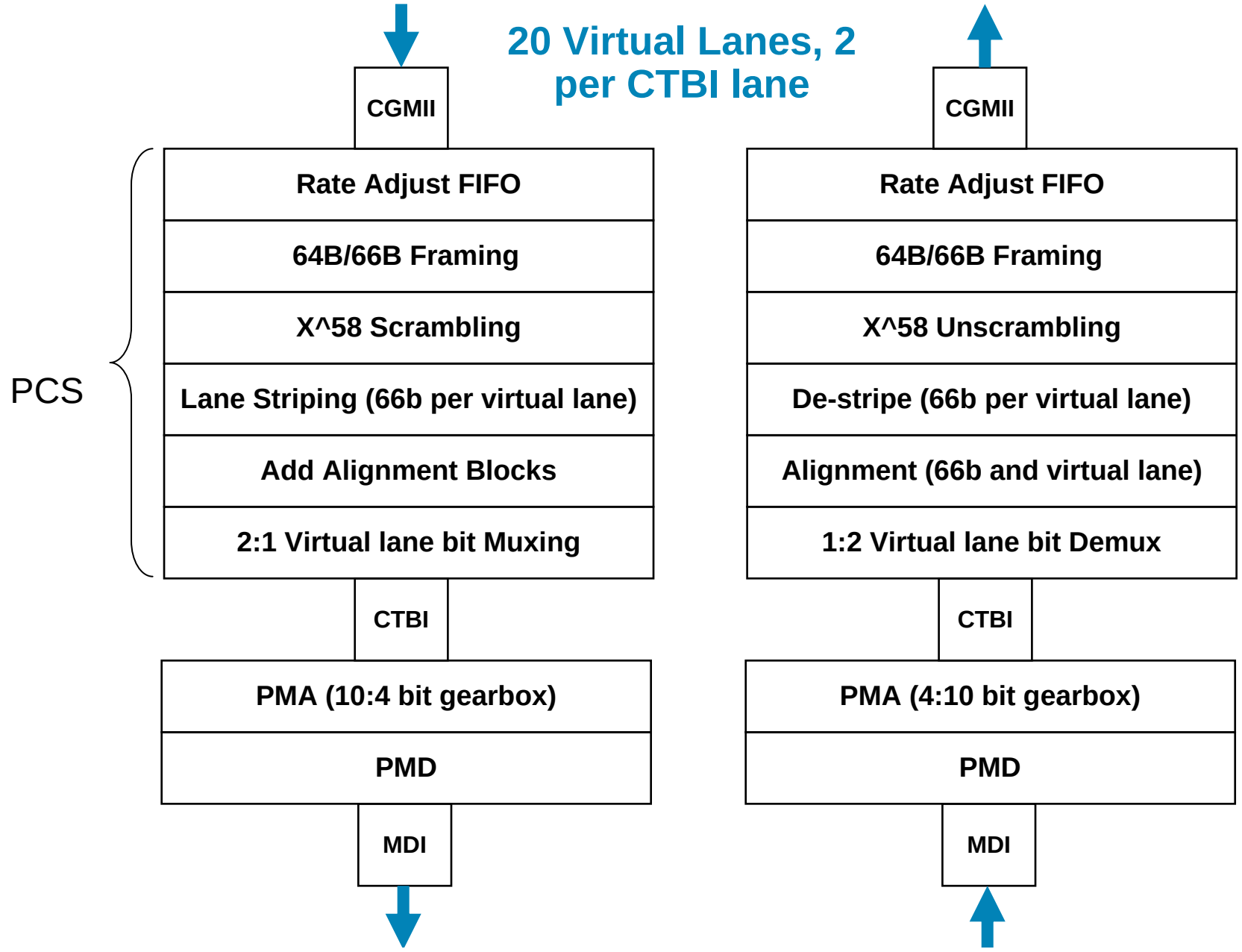
How we generate a Virtual Lane



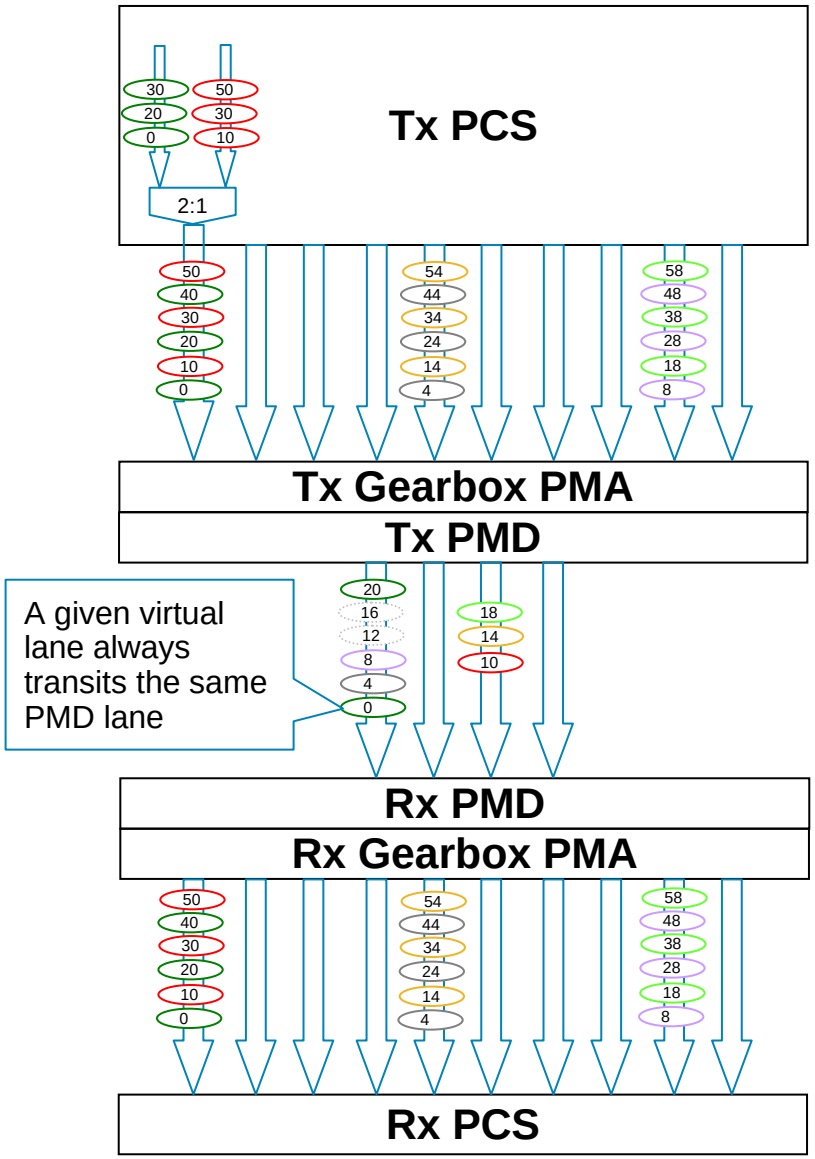
10 Lane PMD (Easy Case)



4 Lane PMD Case



Bit Flow Through



4 PMD Lanes - Two Virtual Lanes per CTBI Lane, zero skew

10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

16	17	18	19
12	13	14	15
8	9	10	11
4	5	6	7
0	1	2	3

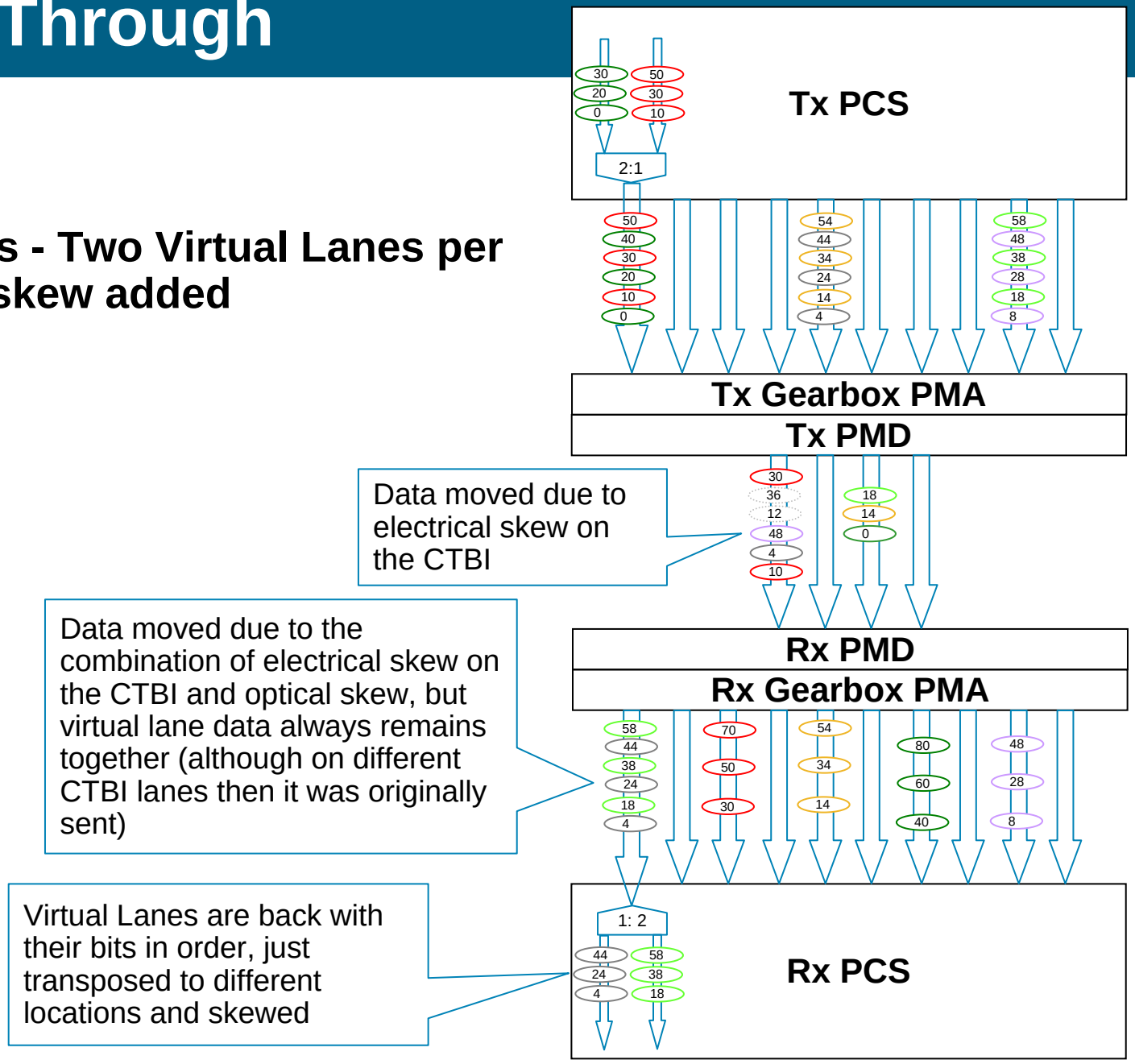
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

* Each Box/Circle is 1 bit

A note on Bit numbering: Bits (0, 20, 40, 60 …, 1280,1300) make up one 66 bit word in virtual lane 0

Bit Flow Through

4 PMD Lanes - Two Virtual Lanes per CTBI Lane, skew added



PCS Concept

Tx PCS

- **The number of virtual lanes is scaled to support all possible PMD configurations (LCM of combinations)**
- **An Alignment block is sent on each virtual lane at the same time with a unique identifier per virtual lane**
- **Virtual lanes are bit interleaved on each CTBI lane (n:1 bit mux)**

Tx PMA

- **Gear box to go from 10 CTBI lanes to n PMD lanes**

Rx PMA

- **Gear box to go from n PMD lanes to 10 CTBI lanes**

Rx PCS

- **The receive bit streams are bit de-muxed to the constituent virtual lanes per CTBI lane (1:n)**
- **64B/66B block alignment is found on each virtual lane**
- **Virtual lane alignment is done by using alignment blocks**
- **Skew is compensated for using pointers and buffering**

How Many Virtual Lanes are Needed?

- **The number of Virtual Lanes needed is the LCM of all of the PMD options**
- **Other option is to allow configuration of the PMD type, this will reduce the number of Virtual Lanes that are needed**

Number of Electrical Lanes (CTBI)	Supportable PMDs	Virtual Lanes Needed
10	1, 2, 3, 4, 5, 6, 8, 10, 12	120
10	1, 2, 3, 4, 5,10	60
10	1, 2, 4, 5, 10	20
10	1, 2, 5, 10	10

Sweet Spot?

CTBI Evolution

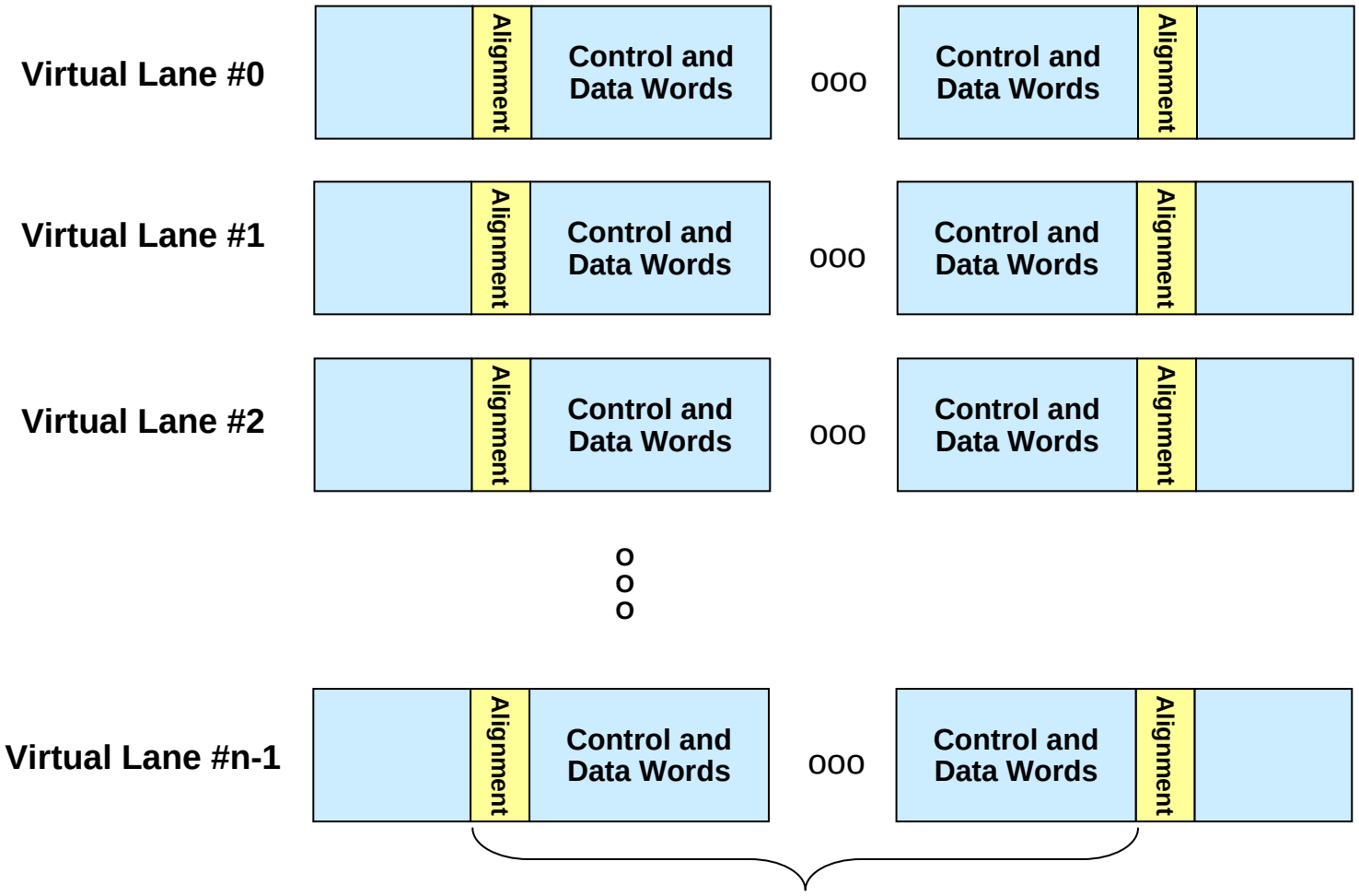
- What happens when we want to shrink the CTBI to 4 or 5 lanes (20 or 25G per lane)?
- Could change the number of CTBI lanes and with 20 virtual lanes still cover the likely cases.
- Virtual lane concept can also work for a x8 interface etc…

Physical CTBI Lanes	PMD Lanes	LCM	Minimum Number of Virtual Lanes
4	10	20	20
4	5	20	20
4	4	4	4
4	2	4	4
4	1	4	4
5	10	10	10
5	5	10	10
5	4	20	20
5	2	10	10
5	1	10	10

Alignment Proposal

- Send alignment on a fixed time basis
- Alignment word also identifies virtual lanes
- Sent every 16k 66bit blocks (104.8usec) on each virtual lane
  - Limits max skew, could also make it programmable
- It interrupts packets
- Takes only 0.006% (61PPM) of the Bandwidth (for 10 virtual lanes, scales linearly for more virtual lanes)
- Rate Adjust FIFO will delete enough IPG so that the MAC still runs at 100.000G with the CTBI at 10.3125G

Alignment Mechanism



16K 66 bit blocks (1 block = 66 bits)

Alignment Mechanism

Generic 10GBASE-R Control Block

10	Block Type = 0x1e	**C_0**	**C_1**	**C_2**	**C_3**	**C_4**	**C_5**	**C_6**	**C_7**

Proposed Alignment Word

10	Block Type = 0x1e	**0x4b**	**0x55**	**ox4b**	**0x55**	**0x4b**	**0x55**	**0x4b**	**VL#**

VL# = Virtual Lane Number (0-n)

Alignment blocks are not scrambled

Issues/Next Steps

- **Investigate the ‘randomness’ of the scrambled data after muxing**
- **Investigate error conditions**
  - **Scrambler error multiplication**
  - **Alignment errors etc.**
- **Look into the issue of multiple alignment words aligning**
  - **Can cause runs of 1’s or 0’s**
- **Can one PCS be used for all PMDs?**
  - **Long haul interfaces might require FEC?**
  - **If a particular PMD requires a different PCS then the CTBI becomes more like XAUI for that case (an extender layer)**
- **Estimate the complexity of the PCS**

Summary

- **The proposed 100GE PCS allows support for most PMD configurations**
- **Complexity is reasonable within the PCS**
- **Complexity in the optical module is very low**
- **Uses a single 10 lane electrical interface for all PMDs**
- **Based on proven 64B/66B framing and scrambling**
- **Electrical interface is very feasible at 10x10.3125G**
- **Allows for a MAC rate of 100.000G**
- **PCS and alignment overhead very low and independent of packet size**