100/40 GbE PMA Proposal

Mark Gustlin – Cisco
Steve Trowbridge – Alcatel-Lucent

IEEE P802.3ba July 2008 Denver

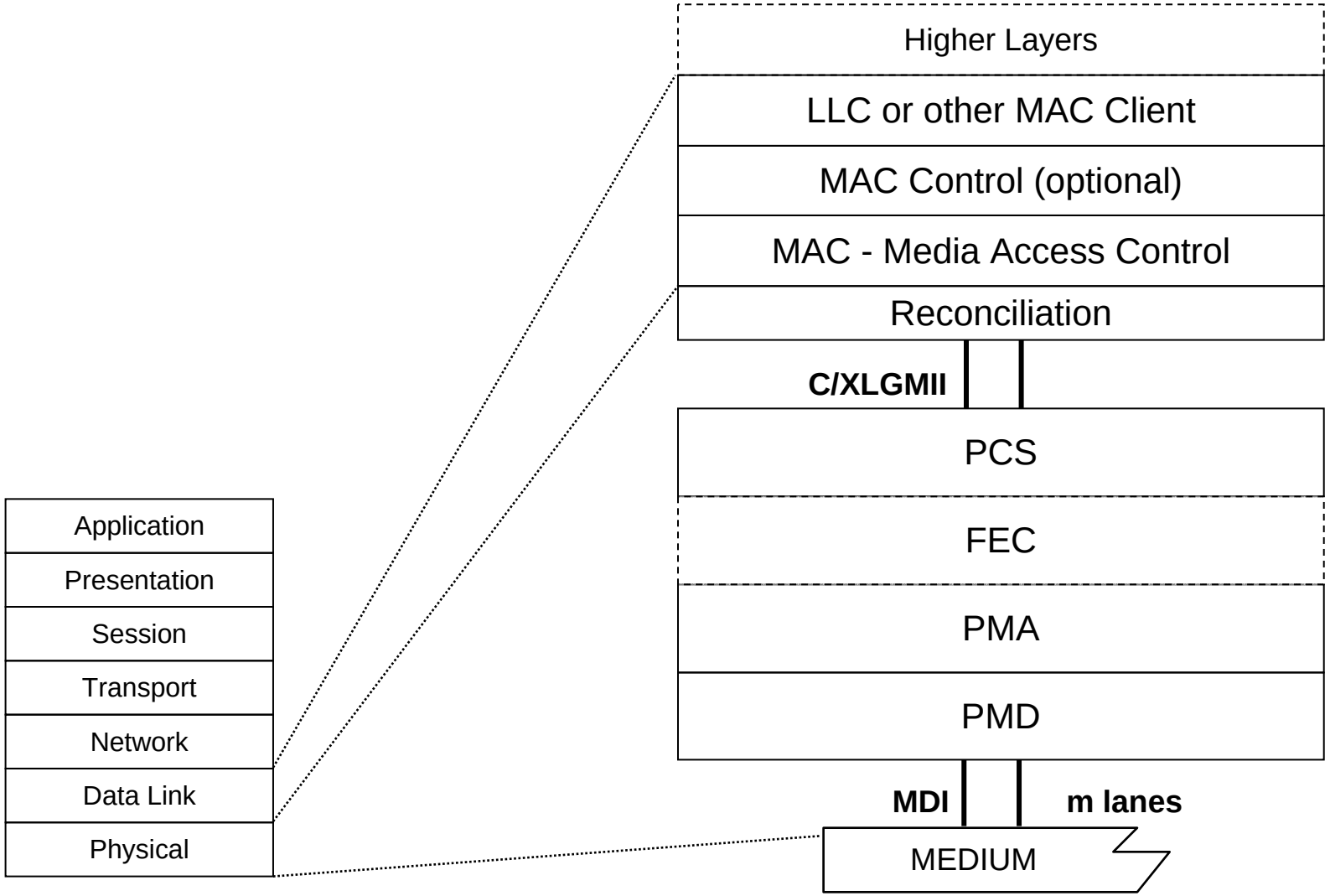
Contributors and Supporters

- Brad Booth - AMCC
- Gary Nicholl - Cisco
- Chris Cole – Finisar
- Subi Krishnamurthy - Force10 Networks
- Shashi Patel - Foundry
- Ryan Latchman – Gennum
- Shinji Nishimura - Hitachi
- Hidehiro Toyoda – Hitachi
- Pete Anslow - Nortel
- Farhad Shafai - Sarance

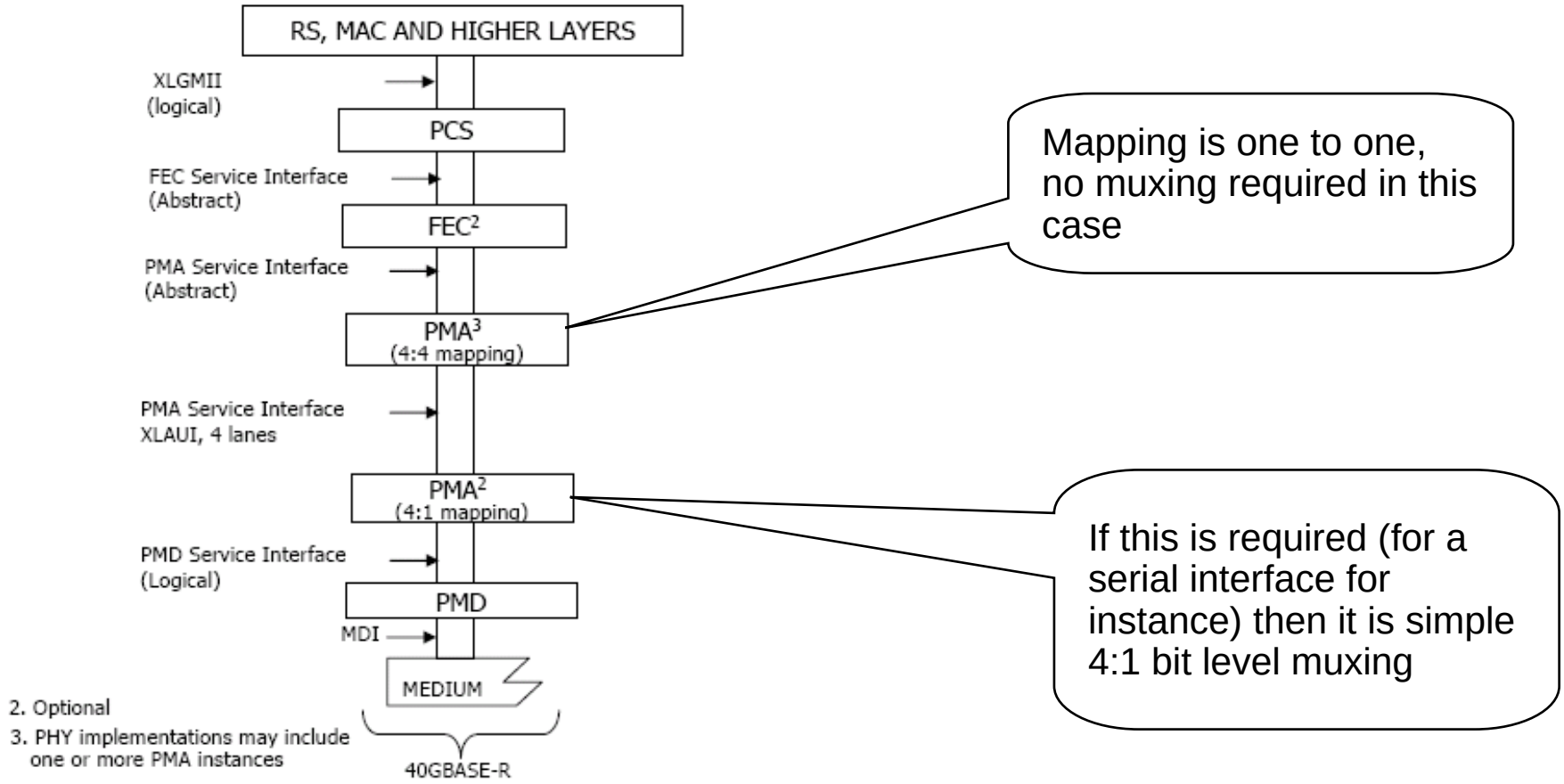
Overview

- This is a proposal for the logical portion on the PMA for 100GE and 40GE
- It does not cover the electrical interfaces (CAUI/XLAUI)

40GE/100GE Generic Architecture

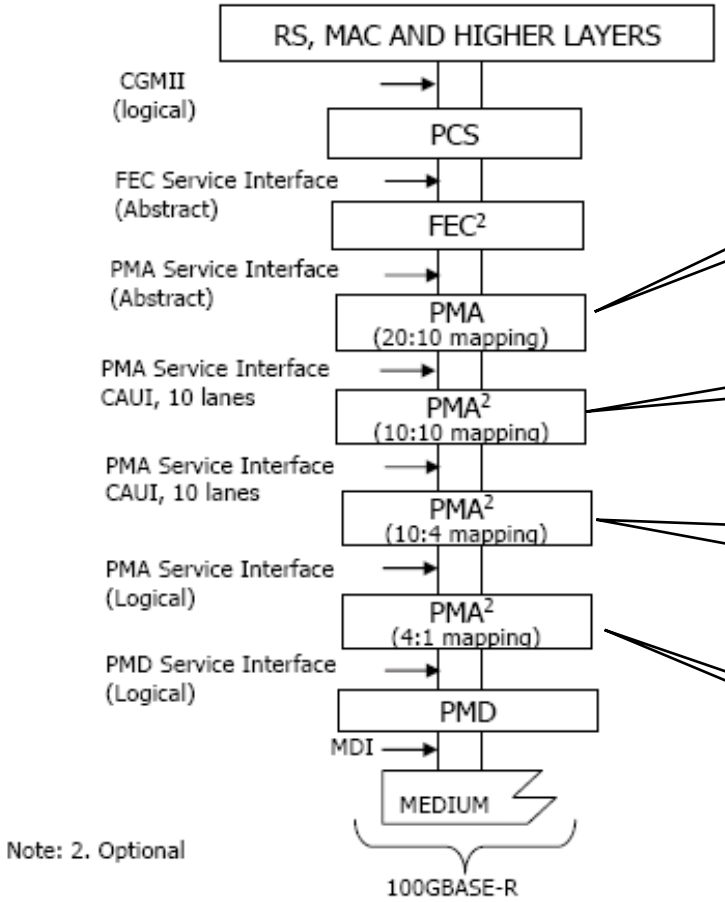


40GE PMA Variants



From ganga_01_0508

100GE PMA Variants



Mapping is two to one, Simply done at the bit level with 2:1 muxes.

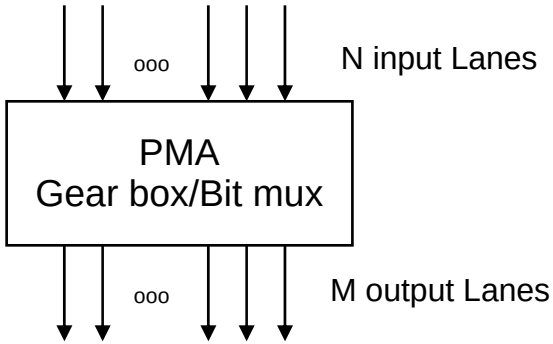
Mapping is one to one, no muxing required in this case

If this is required (for the single mode interface for example) then it is 10:4 gearbox. It could also be implemented as 2x 5:2 muxes

If this is required (for a future serial interface for example) then it is a 4:1 mux.

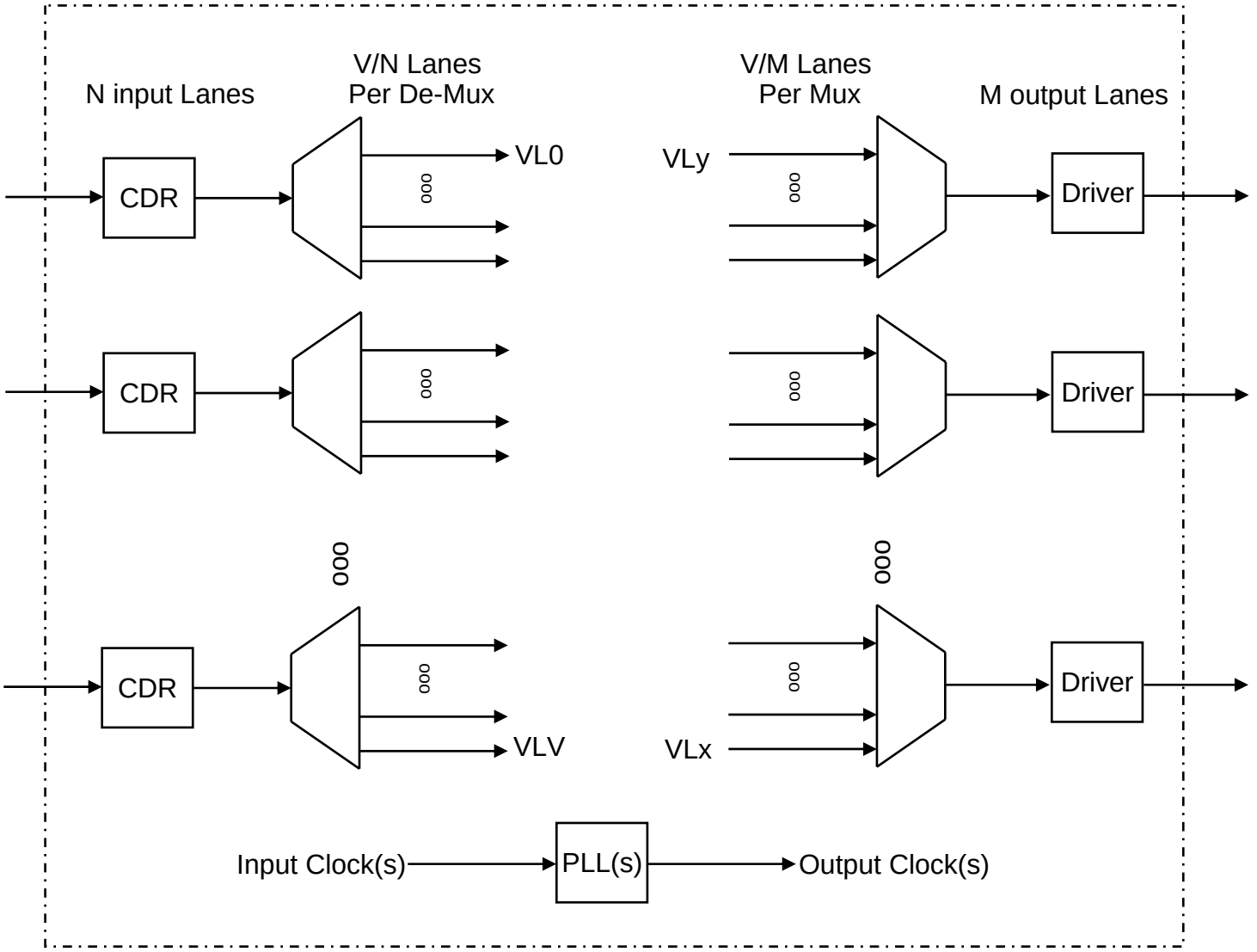
From ganga_01_0508

A Parameterized PMA



- V is the number of Virtual Lanes
- N input lanes, each with V/N virtual lanes
- M output lanes, each with V/M virtual lanes
- The muxing/gearboxing follows the rules as stated later in this presentation

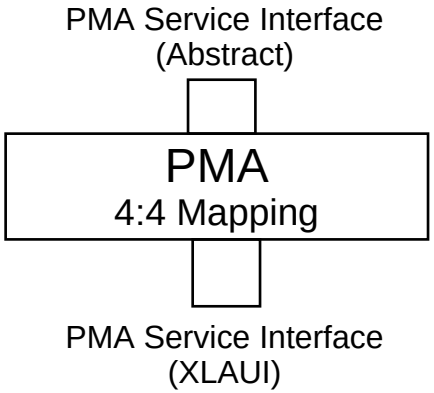
A Parameterized PMA - Details



All implementations that map every input VL to an output VL position are valid, even if they do not completely demux and remux the VLs

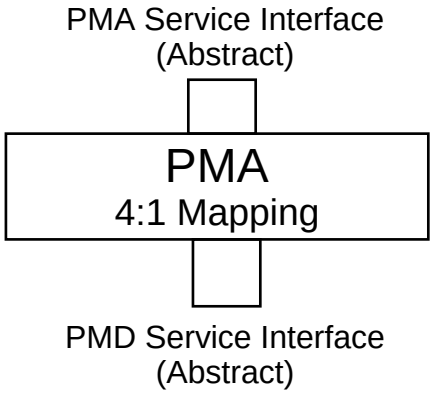
40GE PMA Variant #1

- In the transmit direction the following is provided:
  1. Direct 1:1 mapping of the interface.
  2. Transmission of parallel data to PMD.
- In the receive direction the following is provided:
  1. Direct 1:1 mapping of the interface
  2. Transmission of parallel data to PMA client.
  3. Provide link status information.



40GE PMA Variant #2

- In the transmit direction the following is provided:
  1. Bit level multiplexing of the 4 input lanes into a single output lane
  2. Provide a clock source to the PMA client.
  3. Transmission of serial data to the PMD.
- In the receive direction the following is provided:
  1. Reception of serial data from the PMD
  2. Provides receive clock to PMA client
  3. Bit level de-multiplexing of the serial data into four output lanes
  4. Transmission of parallel data to PMA client.
  5. Provide link status information.



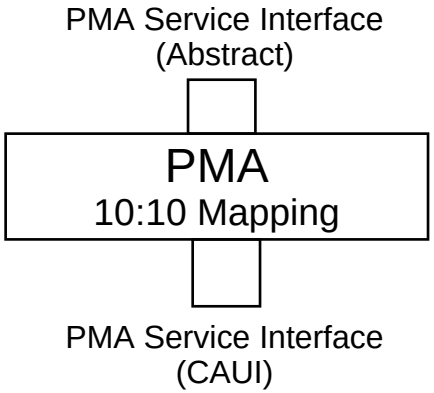
PSI 0	PSI 1	PSI 2	PSI 3
8	9	10	11
4	5	6	7
0	1	2	3

Serial I/F
5
4
3
2
1
0

One possible bit muxing order. PSI = PMA Service Interface

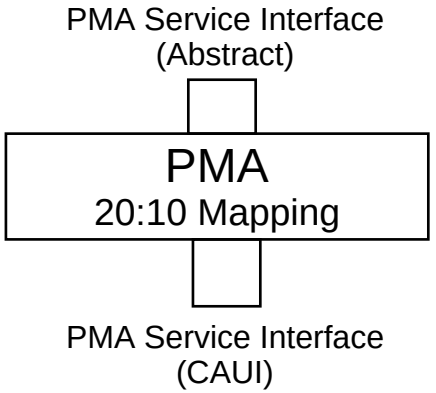
100GE PMA Variant #1

- In the transmit direction the following is provided:
  1. Direct 1:1 mapping of the interface.
  2. Transmission of parallel data to PMD.
- In the receive direction the following is provided:
  1. Direct 1:1 mapping of the interface
  2. Transmission of parallel data to PMA client.
  3. Provide link status information.



100GE PMA Variant #2

- In the transmit direction the following is provided:
  1. Bit level multiplexing of the 20 input lanes into a 10 output lanes
  2. Provide a clock source to the PMA client.
  3. Transmission of parallel data to the PMD.
- In the receive direction the following is provided:
  1. Reception of parallel data from the PMD
  2. Bit level de-multiplexing of the 10 input lanes into 20 output lanes
  3. Provides receive clock to PMA client
  4. Transmission of parallel data to PMA client.
  5. Provide link status information.



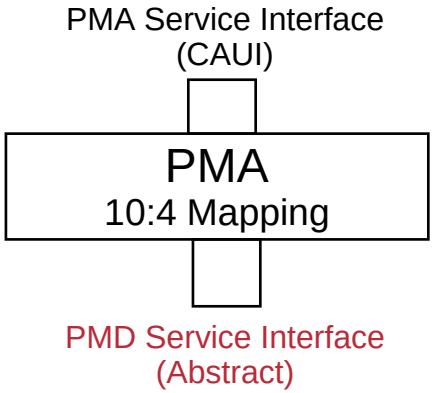
PSI 0	PSI 1	PSI 2	PSI 3	PSI 4	PSI 5	PSI 6	PSI 7	PSI 8	PSI 9	PSI 10	PSI 11	PSI 12	PSI 13	PSI 14	PSI 15	PSI 16	PSI 17	PSI 18	PSI 19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

CI 0	CI 1	CI 2	CI 3	CI 4	CI 5	CI 6	CI 7	CI 8	CI 9
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

Here is one bit muxing order. PSI = PMA Service I/F, CI = CAUI I/F. Others are ok also, rx must expect any lane to show up anywhere.

100GE PMA Variant #3

- In the transmit direction the following is provided:
  1. Bit level gearboxing of the 10 input lanes into a 4 output lanes
  2. Provide a clock source to the PMA client.
  3. Transmission of parallel data to the PMD.
- In the receive direction the following is provided:
  1. Reception of parallel data from the PMD
  2. Bit level gearboxing of the 4 input lanes into 10 output lanes
  3. Provides receive clock to PMA client
  4. Transmission of parallel data to PMA client.
  5. Provide link status information.



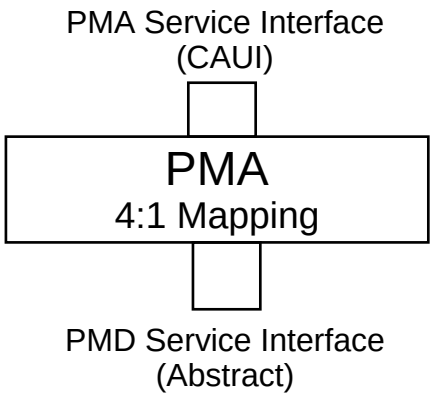
CI 0	CI 1	CI 2	CI 3	CI 4	CI 5	CI 6	CI 7	CI 8	CI 9
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

PI 0	PI 1	PI 2	PI 3
12	13	14	15
8	9	10	11
4	5	6	7
0	1	2	3

Here is one possible bit gearbox order. PI = PMD I/F, CI = CAUI I/F. Others are possible and supportable.

100GE PMA Variant #4

- Same as 40GE PMA variant #2, except for a faster speed
- In the transmit direction the following is provided:
  1. Bit level multiplexing of the 4 input lanes into a single output lane
  2. Provide a clock source to the PMA client.
  3. Transmission of serial data to the PMD.
- In the receive direction the following is provided:
  1. Reception of serial data from the PMD
  2. Bit level de-multiplexing of the serial data into four output lanes
  3. Provides receive clock to PMA client
  4. Transmission of parallel data to PMA client.
  5. Provide link status information.



PSI 0	PSI 1	PSI 2	PSI 3
8	9	10	11
4	5	6	7
0	1	2	3

Serial I/F
5
4
3
2
1
0

One of the possible bit muxing orders. PSI = PMA Service Interface.

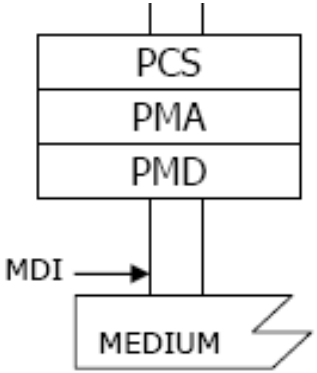
A Note on Bit Muxing Requirements

- All PCS receivers must support receiving a virtual lane on any physical lane
  - This allows flexibility now and in the future for how we bit mux and what widths of the interfaces we have today and tomorrow
- This means that there is more than one valid way to multiplex the virtual lanes at all stages
- All are supportable, with the requirement that:
  - When multiplexing from n to m lanes, any given virtual lane is always only sent on one physical lane, which particular lane does not matter
  - On each of M output lanes, every nth bit on a given physical lane must be a given Virtual Lane, where n = V/M (where V = number of total Virtual Lanes)
- With the above multiplexing rules, and with the requirement that the number of virtual lanes is the Least Common Multiple of all the to be supported lane widths, then everything works
  - For 100GE we can support any combination of lane widths of: 20, 10, 5, 4, 2, 1 with 20 VLs
  - For 40GE we can support any combination of lane widths of: 4, 2, 1 with 4 VLs

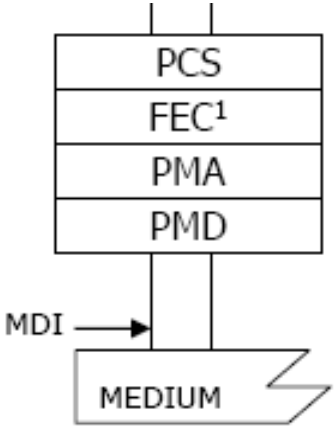
Key Differences from Clause 51 (802.3ae PMA)

- Parameterized Specification
  - Same specification covers both rates, input/output lane counts
- Layer Adjacency
  - In clause 51, PCS is above the PMA and PMD is below the PMA
  - For 802.3ba, above could be PCS, FEC, or another (stacked) PMA. Below could be another PMA or PMD. An appropriate naming scheme for primitives will be introduced to describe this
- Unidirectional Specification
  - Clause 51 documents bi-directional behavior (Tx direction from XSBI to PMD service interface, Rx direction from PMD service interface to XSBI)
  - 802.3ba will use a different instance of the single parameterized specification in each direction, e.g., 10:4 in the Tx direction and 4:10 in the Rx direction
- Independent bit arrival per lane
  - XSBI uses a vector of aligned lanes. For 802.3ba, Dynamic skew results in varying independent arrival of bits on each lane (even though all lanes originate at the Tx with the same clock)

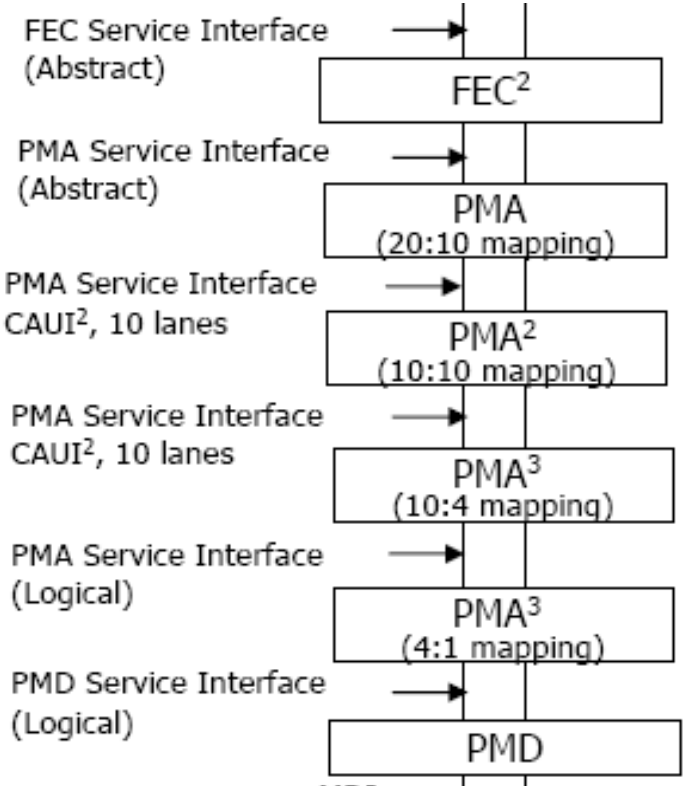
Layer Adjacency aspects



Classical:
-PCS above PMA
-PMD below PMA



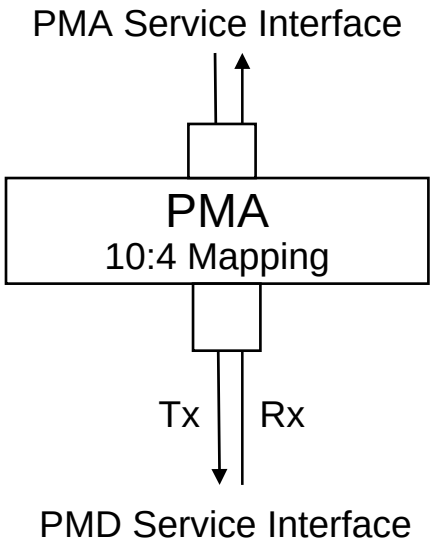
Variant –
- PMA client could be FEC rather than PCS



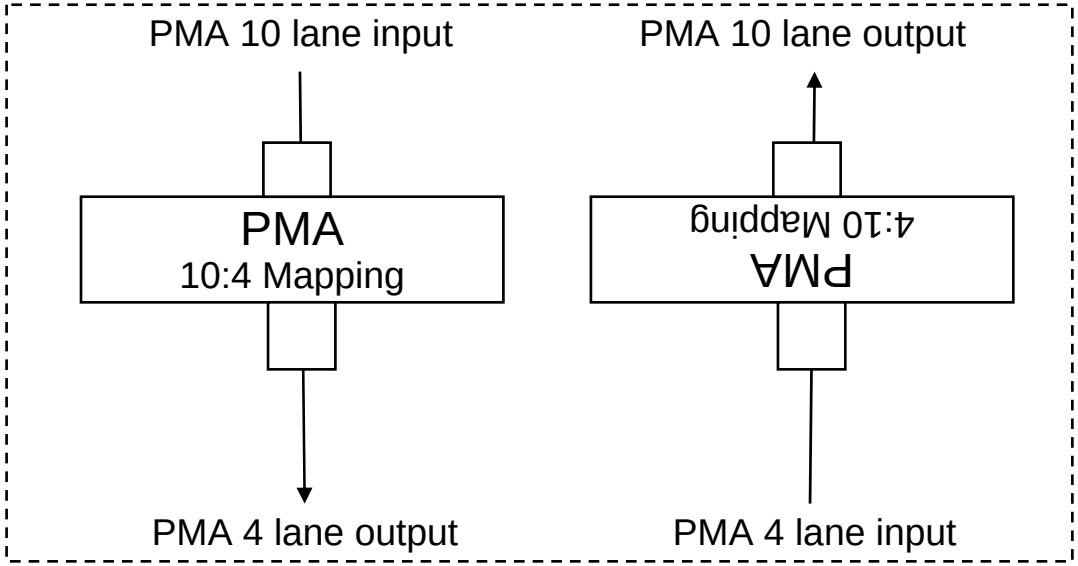
Variant –
- Can also have PMA as client or server for another PMA

Unidirectional Specification

Classical:
- Tx and Rx specified as different aspects of the same PMA



Instead we propose to describe in this way:



Possible Nomenclature

- ***r*PMA*d_n_m* is a PMA for rate *r*=C or XL with *n* input lanes ane *m* output lanes in direction *d* (T or R)**
- **CPMA*d_n_m* has the following characteristics:**
  - **20 VLs at 5.15625 Gbit/s ±100ppm**
  - ***n* and *m* are divisors of 20**
  - **Each of the *n* input lanes carries 20/*n* bit-muxed VLs at 103.125/*n* Gbit/s ±100ppm**
  - **Each of the *m* output lanes carries 20/*m* bit-muxed VLs at 103.125/*m* Gbit/s ±100ppm**
  - **in direction *d*=T, must tolerate 2ui of dynamic skew between VLs**
  - **in direction *d*=R, must tolerate 15ui of dynamic skew between VLs**
- **XLPMA*d_n_m* has the following characteristics:**
  - **4 VLs at 10.3125 Gbit/s ±100ppm**
  - ***n* and *m* are divisors of 4**
  - **Each of the *n* input lanes carries 4/*n* bit-muxed VLs at 41.25/*n* Gbit/s ±100ppm**
  - **Each of the *m* output lanes carries 4/*m* bit-muxed VLs at 41.25/*m* Gbit/s ±100ppm**
  - **in direction *d*=T, must tolerate 2ui of dynamic skew**
  - **in direction *d*=R, must tolerate 30ui of dynamic skew**

PMA Primitives

- ***r*PMA*d_n_m*_UNITDATA.request (*lane, bit*) is used to indicate the arrival of a bit with value *bit* on one of the *n* input lanes.**

  Note: in 802.3ae, UNITDATA.request would only come from the PMA client, but as we are using the same specification in both directions, it could be either a request from above in the transmit direction or, e.g., a PMD_UNITDATA_indication in the receive direction

- ***r*PMA*d_n_m*_UNITDATA.indication (*lane, bit*) is used to emit a bit with value *bit* on one of the *m* output lanes**

  Note: in 802.3ae, UNITDATA.indication would only be sent to the PMA client above, but as we are using the same specification in both directions, it could either be an indication to the PMA client, or a bit sent to the server below, e.g., a PMD_UNITDATA.request in the transmit direction

PMA primitives - continued

- ***r*PMA*R*_*n*_*m*_SIGNAL.request (SIGNAL_OK) is used to indicate whether the received signal from the server layer is OK**

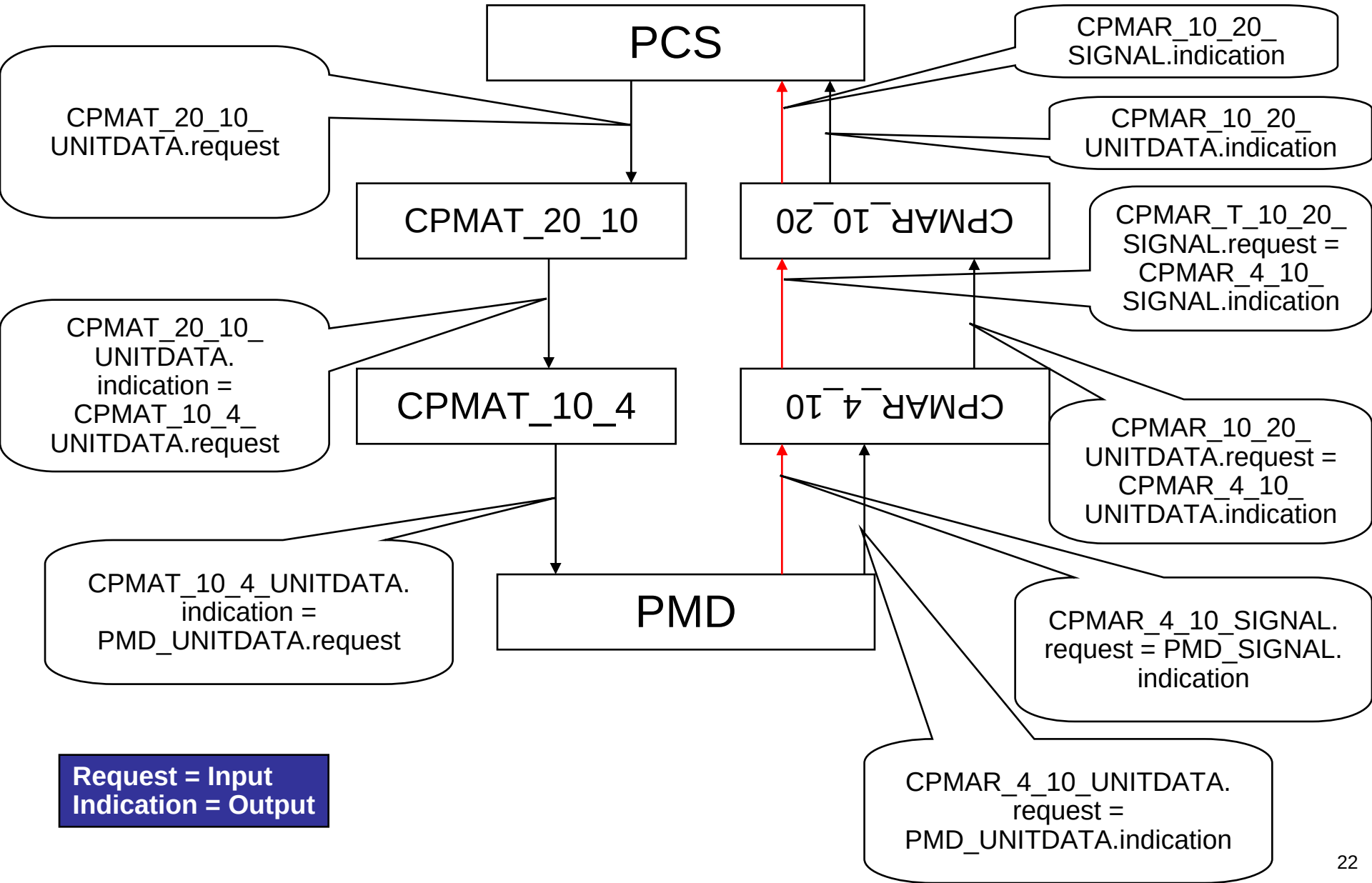
  Note: in 802.3ae, this would be a PMD_SIGNAL.indication, but for 802.3ba it could come from a PMD or another stacked PMA. Should we define this signal in both directions? 802.3ae only defines in receive direction. Initial assumption is that this only applies in the receive direction.

- ***r*PMA*R*_*n*_*m*_SIGNAL.indication (SIGNAL_OK) is used indicate to the PMA client whether the received signal is OK**

  Semantics are similar to 802.3ae, PMA_SIGNAL.indication if we only define this in the receive direction. The signal is indicated as OK if both SIGNAL_OK is being received from the server layer below AND data is being successfully recovered by the PMA on all lanes, PMA fifos are in limits, and therefore data is being transmitted on the lanes of the PMA output

Nomenclature/Primitives Example

Same interface may be known by different names by sender/receiver



Other PMA Aspects

Given that PMA is a logical interface, these next several items could be left as exercises for the designer. But they could be addressed at a high level in the standard.

- **PMA partitioning: If *n* and *m* share any common factors, an implementation may partition the PMA into several smaller PMAs. For example, a CPMAT_20_10 could be partitioned into ten CPMAT_2_1s, or CPMAT_10_4 could be partitioned into two CPMAT_5_2s. Then:**
  - **Virtual lane rearrangement input to output is only within a partition (although input and output lanes might be rearranged, e.g., for routing convenience knowing that the receive logic is general)**
  - **Clocking architecture is local to each partition**
  - **FIFO management for dynamic skew compensation is local to a partition**
  - **If m and n are equal, the partition size is one**

Other PMA Aspects - continued

- **Clocking Architecture – Several options based on context:**
  - **Single reference clock, e.g., for a transmit direction PMA implemented synchronously in the same device as the PCS**
  - **Per-input lane clock recovered in an upstream layer implemented in the same device**
  - **Per-input lane CDR in the PMA itself**
- **The PMA output clock is derived from the reference clock or the input clock on one of the input lanes (within a partition) using an m/n clock multiplier/divider circuit.**

Other PMA Aspects - Continued

- **Dynamic skew compensation and FIFO management**
  - **Amount of skew to be tolerated and/or compensated depends on context:**
    - **A transmit side PMA implemented synchronously in the same device as the PCS may not experience any dynamic skew**
    - **Skew compensation only required within a partition. If m and n are equal, no skew compensation is required since the partition size is one**
    - **2ui per VL dynamic skew tolerance in transmit direction**
    - **15ui per VL dynamic skew tolerance in receive direction (100 GbE)**
    - **30ui per VL dynamic skew tolerance in receive direction (40 GbE)**
  - **Requirement is that after startup, bit rotation order is maintained on each output lane as long as dynamic skew budget is not exceeded on input lanes**

Other PMA Aspects - Continued

- **Dynamic skew compensation implementation possibilities**
  - **FIFOs/buffers are needed between the PMA input and output to compensate dynamic skew (within a partition) where m and n are not equal. Equivalent implementations could use appropriately sized input lane FIFOs, output lane buffers, or per VL FIFOs. The FIFO/buffer depth (per VL) needs to be double the dynamic skew tolerance amount as the clock might be derived from a lane that is leading or lagging**
  - **The PMA output (within a partition) doesn't start until all input lanes have recovered clock and data and FIFOs/buffers (within the partition) are centered.**

Summary

- PMA Functions include:
  - Clock and data recovery
  - Bit level multiplexing/gearboxing
  - Clock generation
  - Signal drivers
- Key differences from Clause 51 (802.3ae) PMA description
  - Unified parameterized specification covers all rates and input/output lane counts
  - Matching interface descriptions may have different names at sending and receiving layers to allow layer stacking flexibility
  - Uni-directional specification uses same text to describe Tx and Rx directions
  - Independent bit arrival per lane (even though lanes originate from common Tx clock) rather than a bus (e.g., XSBI) operating on a common clock