Symbol-muxing PMA for 200 Gb/s per lane signaling

Towards a baseline proposal

Adee Ran, Cisco

Supporters

- Will Bliss, Broadcom
- Vasu Parthasarathy, Broadcom
- Cathy Liu, Broadcom

Background

- P802.3df has adopted PCS and PMAs for 8×100G PHYs
  - PCS with 32 logical lanes and 4 FEC engines
  - 4:1 bit muxing (32:8 lanes) in the PMA
- As next steps (possibly in P802.3dj) we should define **200G per lane AUIs: 800GAUI-4, 400GAUI-2 and 200GAUI-1**
- There is a preference to keep the same PCSs and the KP-FEC
  - This means the PMA muxing ratio will be 8:1
- Analysis of FEC performance with correlated errors shows symbol muxing has a significant advantage
  - 8:1 bit muxing causes unacceptable degradation
  - **See ran_3df_01_2211**

Goals of this presentation

- Describe a symbol-muxing PMA which will:
  - Leverage the existing sublayer architecture
  - Provide good FEC performance with correlated errors
  - Enable simple implementation and interoperability in various use cases

… towards a future baseline proposal

Symbol muxing at 200 Gb/s per lane

- A proposal should include support for all interfaces with 200 Gb/s per lane signaling (PMAs connected to either AUIs or PMDs).
- The following slides focus on the PMAs required for 800GBASE-R PHYs.
  - 200G and 400G PHY rates are expected to have similar PMAs with the appropriate changes (different number of lanes).
  - 1.6T architecture has not been adopted yet, so it is not addressed in this presentation. It could use a similar approach.

Basic concepts for 800GBASE-R

- Keep the same PCS (32 PCS lanes, 4 codewords)
- Define a new PMA with symbol-muxing functionality
  - 32:4 for a PMA co-located with an 800GBASE-R PCS or a DTE 800GXS
  - 8:4 for 800G “gearboxes”
  - 4:32 (for PHY 800GXS) and 4:8 – essentially the same, Tx and Rx directions swapped
  - 4:4 for 800G retimers
- For the 32:4 PMA, both sides are symbol-muxed
  - Symbol muxing requires alignment, but implementation is straightforward
- For the 8:4 PMA, the “8” side is bit muxed, while the “4” side is symbol muxed
  - Conversion requires some “protocol-aware” logic, but is relatively simple
  - Gearboxes typically exist in a transition phase, lower volume

PMA(32:4) functional block diagram

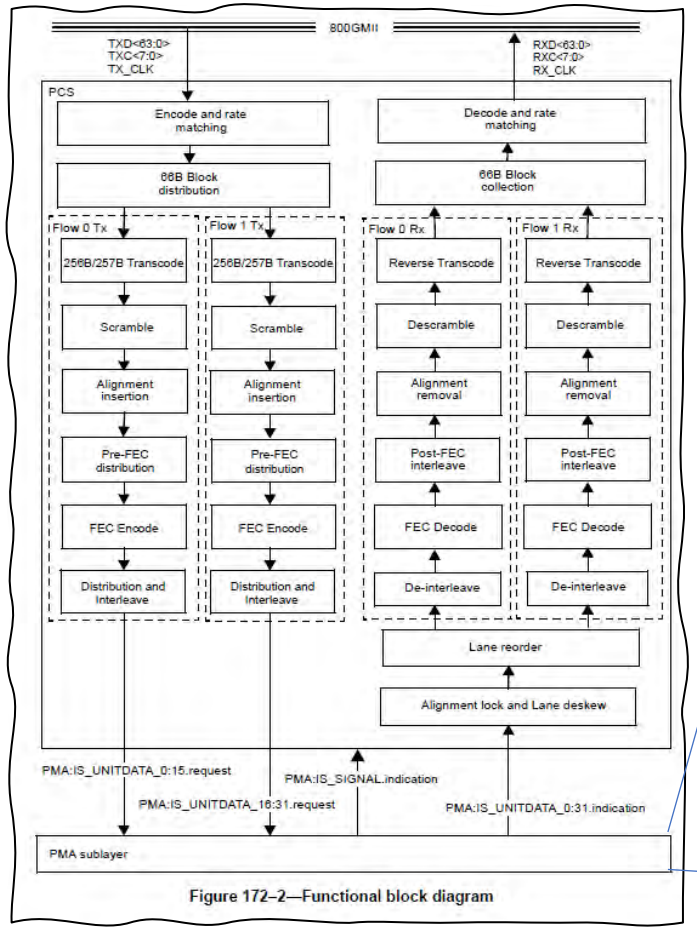
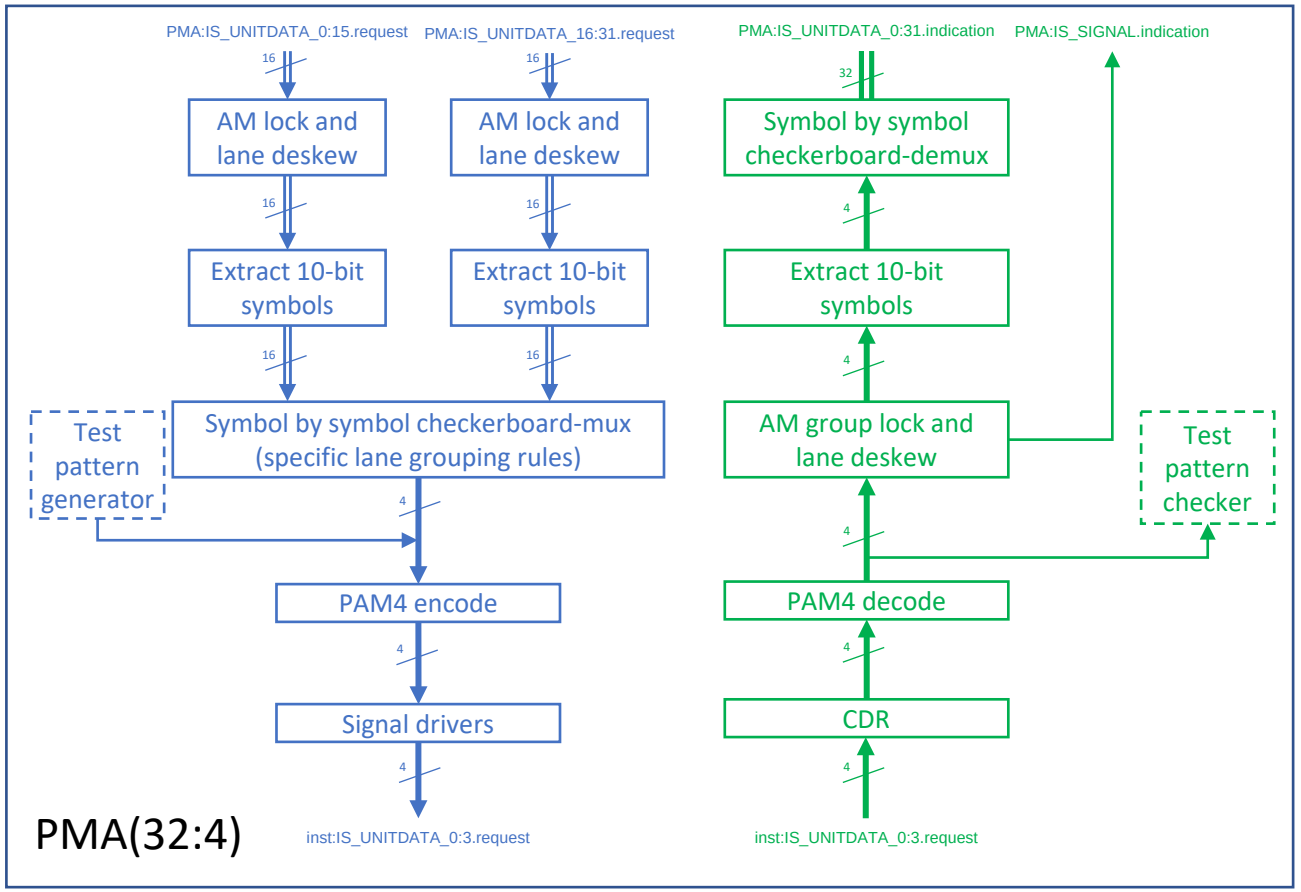


Figure 172–2—Functional block diagram



A PMA(4:32) is identical to a PMA(32:4) placed backwards (i.e., service interface has 4 lanes and interface below has 32 lanes)

800GBASE-R symbol-muxing PMA(32:4): Functions in the transmit direction

- Lock on AMs
  - Identifying each of the 32 PCS lanes
- Deskew
  - May be limited to groups of lanes that go to the same physical lane
- Extract 10-bit symbols from each lane
- Symbol-wise mux to 4 lanes
  - Specific lane grouping (symbols from all codewords on each lane)
  - *Apply a checkerboard pattern (undoing the PCS alternating symbols)*
- Send to the next sublayer (AUI/PMD)

The operations marked in orange are intended to match the PCS specification, but may be implemented differently as part of a co-located PCS.

The text in the standard can make a note of that.

800GBASE-R symbol-muxing PMA(32:4): Functions in the receive direction

- Receive and decode PAM4 symbols
- Lock on AM groups
  - Identify each of the 4 PCS lane groups
- Deskew
- Extract 10-bit symbols from each lane
- Symbol-wise demux to 32 lanes
  - *Apply a checkerboard pattern*
- Send to the PCS

The operations marked in orange are intended to match the PCS specification, but may be implemented differently as part of a co-located PCS.

The text in the standard can make a note of that.

Proposed lane muxing rules in the transmit direction (800GBASE-R)

PMA lane	PCS lane source (after undoing the checkerboard pattern)							
	PCS symbol 8n	PCS symbol 8n+1	PCS symbol 8n+2	PCS symbol 8n+3	PCS symbol 8n+4	PCS symbol 8n+5	PCS symbol 8n+6	PCS symbol 8n+7
0	0	1	16	17	8	9	24	25
1	2	3	18	19	10	11	26	27
2	4	5	20	21	12	13	28	29
3	6	7	22	23	14	15	30	31

The 10 bits of each symbol are encoded into five PAM4 symbols, encoded as specified in 120.5.7.1:

- The first transmitted PAM4 symbol is the result of encoding {b0, b1}

  …
- The fifth transmitted PAM4 symbol is the result of encoding {b8, b9}

Notes

- Undoing the checkerboard pattern (or equivalently, re-applying it) means that the next group of 32 symbols swaps every symbol pair (2n+1 and 2n).
  - This creates a consistent symbol order on each physical lane, as illustrated in the next slide.
  - In the receive direction, functionally, the checkerboard pattern is restored.
  - A co-located PCS/PMA implementation can bypass the checkerboard pattern in both sublayers.
- A specific PCS lane grouping and order on each PMA lane is suggested.
  - Flexibility seems to have no benefit here; leaving the grouping unspecified will unnecessarily complicate receiver design and verification.
  - If the PMA is not co-located with a PCS, the AM lock and deskew function provides the required information to apply the specified muxing.
  - PMA lanes may be re-ordered, and the receiving PMA has to identify them using the AM groups.
  - The PAM4 patterns for the symbol-muxed AM groups have good properties (see backup).

32:4 symbol muxing illustrated

PCS lane →	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
Symbol # 1	B8	A8	B9	A9	B10	A10	B11	A11	B12	A12	B13	A13	B14	A14	B15	A15	D8	C8	D9	C9	D10	C10	D11	C11	D12	C12	D13	C13	D14	C14	D15	C15
0	A0	B0	A1	B1	A2	B2	A3	B3	A4	B4	A5	B5	A6	B6	A7	B7	C0	D0	C1	D1	C2	D2	C3	D3	C4	D4	C5	D5	C6	D6	C7	D7

1 FEC symbol on each PCS lane

PMA lane →	0	1	2	3
Symbol # 15	D12	D13	D14	D15
14	C12	C13	C14	C15
13	B12	B13	B14	B15
12	A12	A13	A14	A15
11	D8	D9	D10	D11
10	C8	C9	C10	C11
9	B8	B9	B10	B11
8	A8	A9	A10	A11
7	D4	D5	D6	D7
6	C4	C5	C6	C7
5	B4	B5	B6	B7
4	A4	A5	A6	A7
3	D0	D1	D2	D3
2	C0	C1	C2	C3
1	B0	B1	B2	B3
0	A0	A1	A2	A3

last

first

1 FEC symbol on each physical lane

PMA lane 0 muxes PCS lanes 0, 1, 16, 17, 8, 9, 24, 25.
PMA lanes 1, 2, 3 mux corresponding PCS lanes with offsets 2, 4, 6 respectively.

CW order is fixed and consistent – checkerboard pattern is undone as part of symbol muxing
PCS lanes on the same PMA lane are de-skewed, but skew may exist between PMA lanes (not shown)

Proposed lane muxing rules in the transmit direction (400GBASE-R and 200GBASE-R)

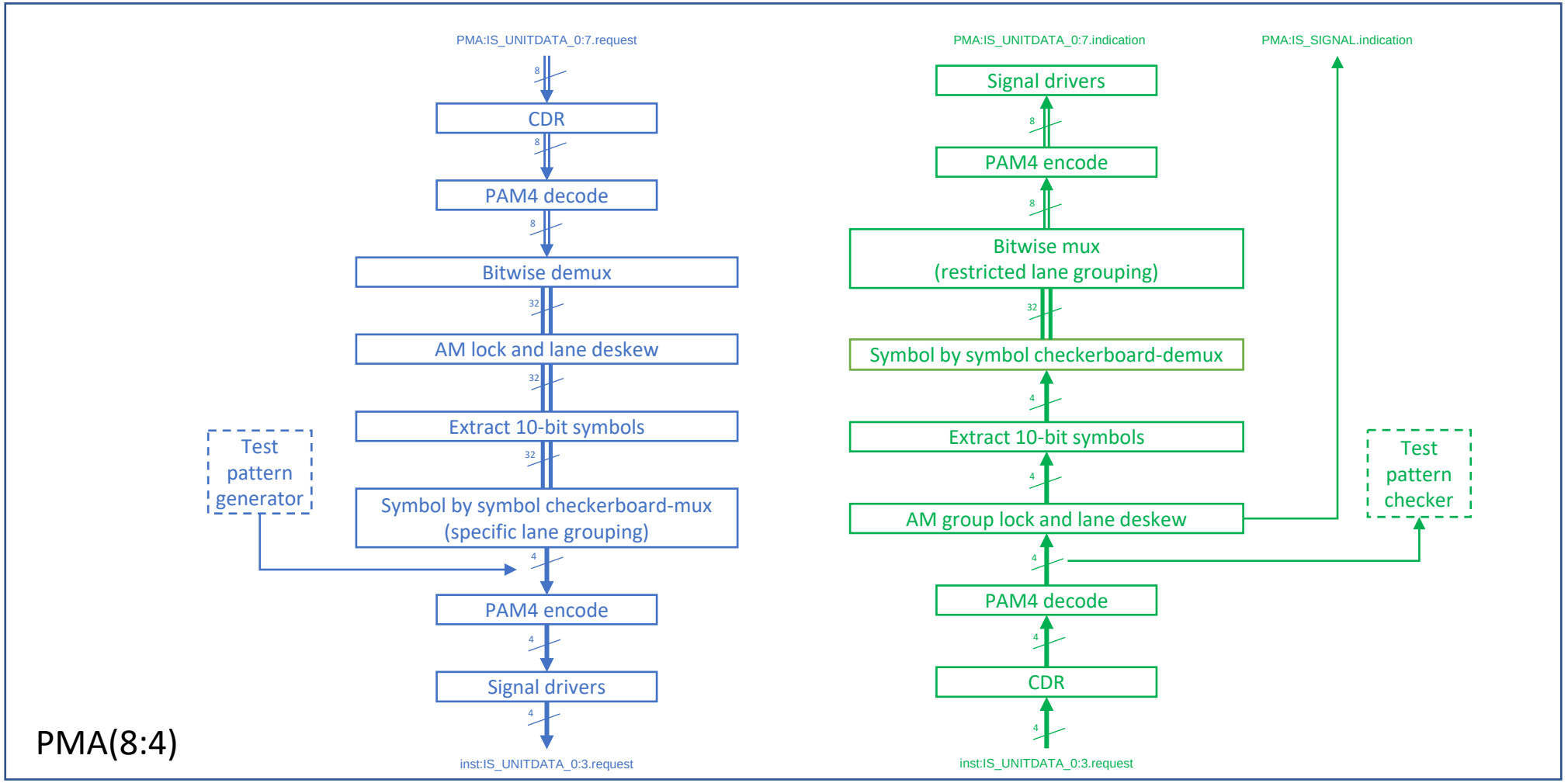
400GBASE-R (16 PCS lanes)

PMA lane	PCS lane source (after undoing the checkerboard pattern)							
	PCS symbol 8n	PCS symbol 8n+1	PCS symbol 8n+2	PCS symbol 8n+3	PCS symbol 8n+4	PCS symbol 8n+5	PCS symbol 8n+6	PCS symbol 8n+7
0	0	1	4	5	8	9	12	13
1	2	3	6	7	10	11	14	15

200GBASE-R (8 PCS lanes)

PMA lane	PCS lane source (after undoing the checkerboard pattern)							
	PCS symbol 8n	PCS symbol 8n+1	PCS symbol 8n+2	PCS symbol 8n+3	PCS symbol 8n+4	PCS symbol 8n+5	PCS symbol 8n+6	PCS symbol 8n+7
0	0	1	2	3	4	5	6	7

PMA(8:4) block diagram



A PMA(4:8) is identical to a PMA(8:4) placed backwards (i.e., service interface has 4 lanes and interface below has 8 lanes)

Gearbox PMA(8:4) and PMA(4:8)

Only the operations marked in orange are new and "protocol-aware". Their implementation is simple.

- Functionality:
  - Extract and decode PAM4 symbols
  - On the "8" side:
    - Bitwise-demux the input stream
    - Lock on AMs
    - Recover the 32 PCS lanes and find symbol boundaries
    - Deskew groups of input lanes
  - On the "4" side:
    - Lock on AM groups in the input stream
    - Recover the 32 PCS lanes and find symbol boundaries
    - Symbol-wise demux
  - Apply checkerboard pattern to the symbols on the PCS lanes (in both directions)
    - Data coming from the "8" side already has a checkerboard pattern, so this operation undoes it when sending to the "4" side
  - Bitwise mux on the "8" side, symbol-wise mux on the "4" side
  - PAM4 encode and send

Non-gearbox PMA(4:4)

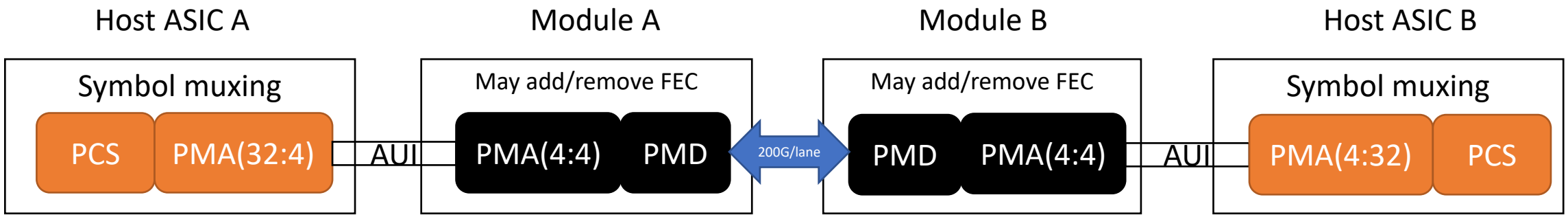
- In a retimer that has the same width on both interfaces, the PMA is not required to know anything about FEC symbols.
- In these cases, the PMA is specified as a lane-by-lane “CDR”.
  - Forwarding PAM4 symbols between input and output interfaces
  - No need for any inter-lane or symbol-oriented logic (AM, deskew, etc.)
- Re-ordering of physical lanes is allowed
- Changing bit/symbol order within each lane is not allowed

Use cases, symbol muxed PMA(n:m)

PMA description	Service interface (n)	Service interface below (m)
Below an 800GBASE-R PCS or DTE 800GXS	32 lanes (bits)	4 lanes, symbol muxed
Below a 400GBASE-R PCS or DTE 400GXS	16 lanes (bits)	2 lanes, symbol muxed
Below a 200GBASE-R PCS or DTE 200GXS	8 lanes (bits)	1 lane, symbol muxed
800GAUI-8 to 800GAUI-4 gearbox (or an equivalent optical module)	8 lanes, bit muxed	4 lanes, symbol muxed
400GAUI-n to 400GAUI-2 gearbox, n>2	n lanes, bit muxed	2 lanes, symbol muxed
200GAUI-n to 200GAUI-1 gearbox, n>1	n lanes, bit muxed	1 lane, symbol muxed
Above a PHY 800GXS	4 lanes, symbol muxed	32 lanes (bits)
Above a PHY 400GXS	2 lanes, symbol muxed	16 lanes (bits)
Above a PHY 200GXS	1 lane, symbol muxed	8 lanes (bits)

Transition from 100 Gb/s to 200 Gb/s per lane

End goal: 200G/lane electrical and optical



- Hosts A and B use RS544 FEC with symbol muxing in both directions (mandatory for 200G/lane)
- Modules A and B may add another FEC (no need for symbol alignment or deskew logic), or terminate and re-generate the RS FEC
- Symbol muxing persists over all segments.

FEC Symbol-aware function

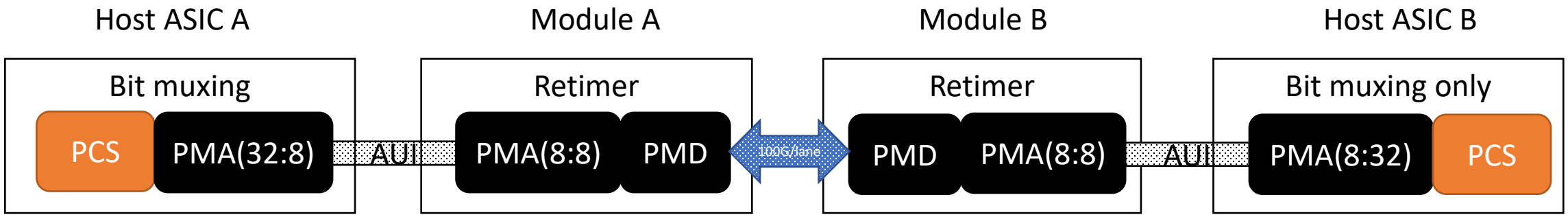
Bit muxed

Symbol muxed

Bit/PAM4 only function

Modules do not care about RS-FEC symbol boundaries

First generation: 100G/lane electrical and optical



- Hosts A and B have bit muxing PMAs
- Modules A and B are lane-by-lane retimers (no symbol alignment or deskew logic)
- Bit muxing persists over all segments

FEC Symbol-aware function

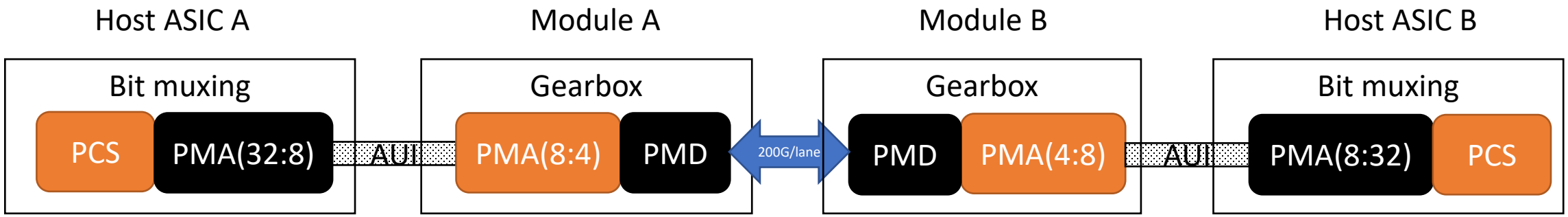
Bit muxed

Symbol muxed

Bit/PAM4 only function

Modules do not care about RS-FEC symbol boundaries

Transition: 100G/lane electrical, 200G/lane optical



- Hosts A and B have bit muxing PMAs
- Modules A and B are 8:4 gearboxes (with additional logic)
- Symbol muxing is applied only over the optical segment

FEC Symbol-aware function

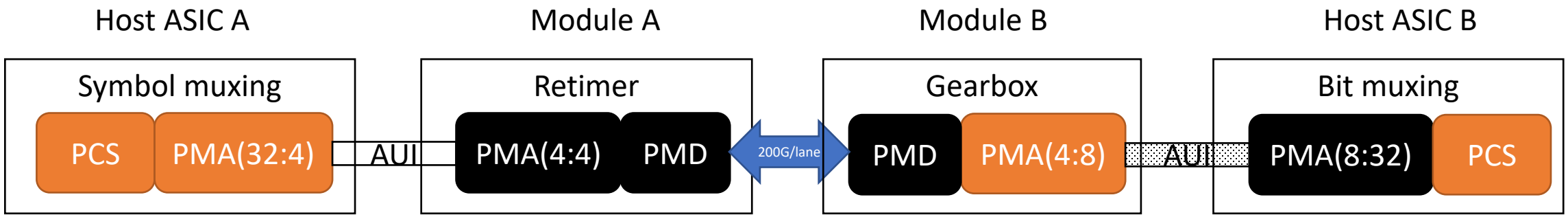
Bit/PAM4 only function

Bit muxed

Symbol muxed

Modules have to deskew and convert between bit-muxing and symbol muxing – but do not need full RS-FEC capability

Transition: mixed electrical, 200G/lane optical



- Host A uses symbol muxing (mandatory for 200G/lane)
- Module A is not symbol aware
- Host B uses bit muxing
- Module B is a 8:4 gearbox (with additional logic for muxing conversion)
- Symbol muxing is carried over the optical segment and electrical segment A

FEC Symbol-aware function

Bit muxed

Symbol muxed

Bit/PAM4 only function

Compatibility between hosts with different electrical rates is maintained

Summary

- A path for 200G/lane signaling using symbol-muxing PMA was presented
  - Uses the existing PCS specifications
  - Enables good performance of the RS FEC with correlated errors.
- Implementations with co-located PCS and with non-gearbox modules are straightforward.
- Gearbox modules (100G electrical, 200G optical) need additional mux-conversion logic
  - But these modules are expected to exist mainly in a transition phase.
- Interoperability of 200G/lane optical PHYs is possible:
  - With existing PCS implementations
  - With different modules (AUI rates) in each host.

Backup

Recall: 400GBASE-R 16-lane AM block mapping

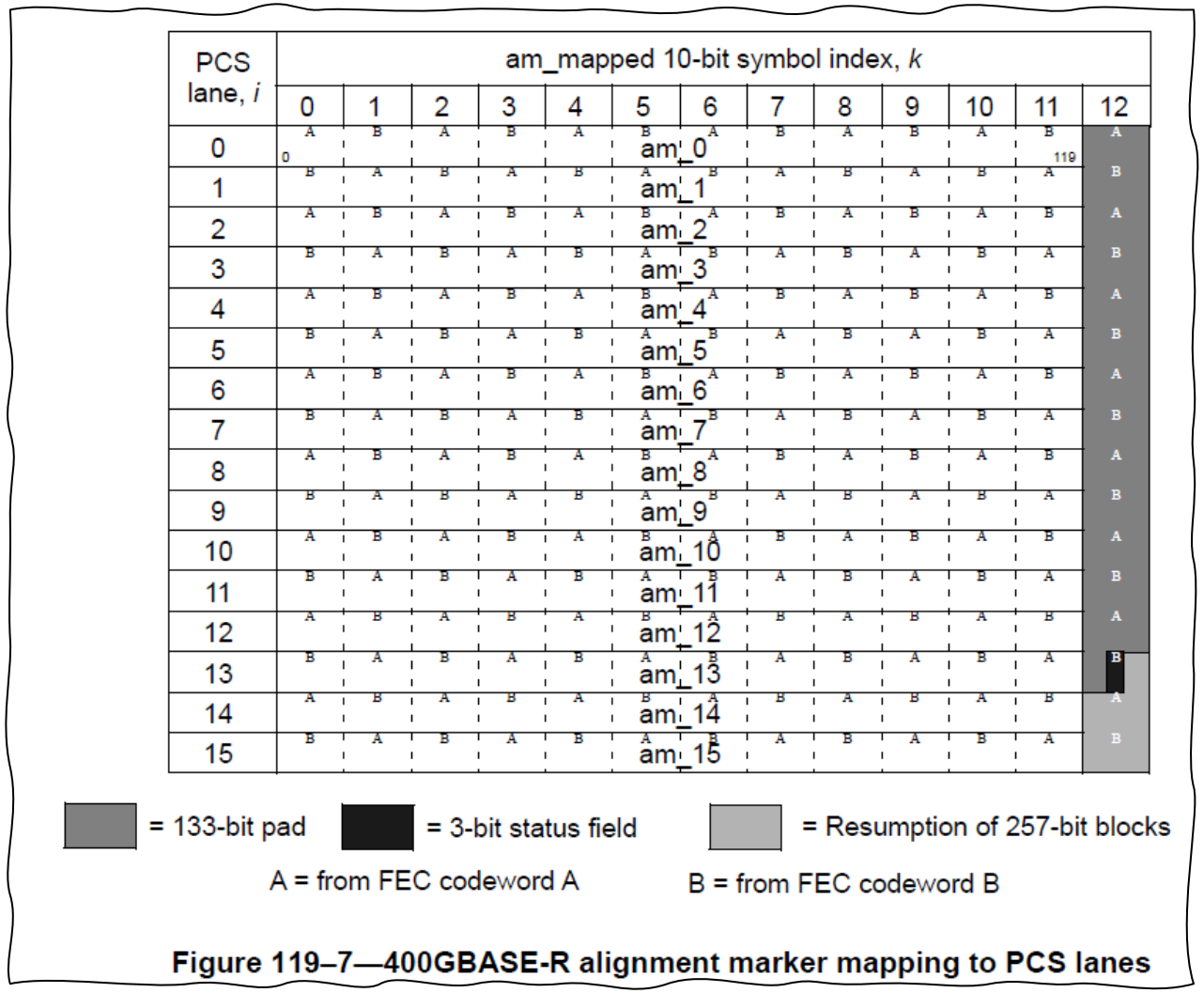


Figure 119–7—400GBASE-R alignment marker mapping to PCS lanes

Source: shrikhande_3df_01a_221004

800GBASE-R PCS

AM Marker Encoding

- CM0-CM5 and UP0-UP2 are unchanged from 400GbE CL119
- UM0/UM3 for Flow lanes 0-15 are inverted from 400GbE
- UM1/UM2/UM4/UM5 for Flow lanes 16-31 are inverted from 400GbE
- Prevents lock with 400GbE ports
- Maintains DC balance

Flow Lane #	Encoding															
	CM0	CM1	CM2	UP0	CM3	CM4	CM5	UP1	UM0	UM1	UM2	UP2	UM3	UM4	UM5
0	0x9A	0x4A	0x26	0xB6	0x65	0xB5	0xD9	0xD9	**0xFE**	0x71	0xF3	0x26	**0x01**	0x8E	0x0C
1	0x9A	0x4A	0x26	0x04	0x65	0xB5	0xD9	0x67	**0xA5**	0xDE	0x7E	0x98	**0x5A**	0x21	0x81
2	0x9A	0x4A	0x26	0x46	0x65	0xB5	0xD9	0xFE	**0xC1**	0xF3	0x56	0x01	**0x3E**	0x0C	0xA9
3	0x9A	0x4A	0x26	0x5A	0x65	0xB5	0xD9	0x84	**0x79**	0x80	0xD0	0x7B	**0x86**	0x7F	0x2F
4	0x9A	0x4A	0x26	0xE1	0x65	0xB5	0xD9	0x19	**0xD5**	0x51	0xF2	0xE6	**0x2A**	0xAE	0x0D
5	0x9A	0x4A	0x26	0xF2	0x65	0xB5	0xD9	0x4E	**0xED**	0x4F	0xD1	0xB1	**0x12**	0xB0	0x2E
6	0x9A	0x4A	0x26	0x3D	0x65	0xB5	0xD9	0xEE	**0xBD**	0x9C	0xA1	0x11	**0x42**	0x63	0x5E
7	0x9A	0x4A	0x26	0x22	0x65	0xB5	0xD9	0x32	**0x29**	0x76	0x5B	0xCD	**0xD6**	0x89	0xA4
8	0x9A	0x4A	0x26	0x60	0x65	0xB5	0xD9	0x9F	**0x1E**	0x73	0x75	0x60	**0xE1**	0x8C	0x8A
9	0x9A	0x4A	0x26	0x6B	0x65	0xB5	0xD9	0xA2	**0x8E**	0xC4	0x3C	0x5D	**0x71**	0x3B	0xC3
10	0x9A	0x4A	0x26	0xFA	0x65	0xB5	0xD9	0x04	**0x6A**	0xEB	0xD8	0xFB	**0x95**	0x14	0x27
11	0x9A	0x4A	0x26	0x6C	0x65	0xB5	0xD9	0x71	**0xDD**	0x66	0x38	0x8E	**0x22**	0x99	0xC7
12	0x9A	0x4A	0x26	0x18	0x65	0xB5	0xD9	0x5B	**0x5D**	0xF6	0x95	0xA4	**0xA2**	0x09	0x6A
13	0x9A	0x4A	0x26	0x14	0x65	0xB5	0xD9	0xCC	**0xCE**	0x97	0xC3	0x33	**0x31**	0x68	0x3C
14	0x9A	0x4A	0x26	0xD0	0x65	0xB5	0xD9	0xB1	**0x35**	0xFB	0xA6	0x4E	**0xCA**	0x04	0x59
15	0x9A	0x4A	0x26	0xB4	0x65	0xB5	0xD9	0x56	**0x59**	0xBA	0x79	0xA9	**0xA6**	0x45	0x86
16	0x9A	0x4A	0x26	0xB6	0x65	0xB5	0xD9	0xD9	0x01	**0x8E**	**0x0C**	0x26	0xFE	**0x71**	**0xF3**
17	0x9A	0x4A	0x26	0x04	0x65	0xB5	0xD9	0x67	0x5A	**0x21**	**0x81**	0x98	0xA5	**0xDE**	**0x7E**
18	0x9A	0x4A	0x26	0x46	0x65	0xB5	0xD9	0xFE	0x3E	**0x0C**	**0xA9**	0x01	0xC1	**0xF3**	**0x56**
19	0x9A	0x4A	0x26	0x5A	0x65	0xB5	0xD9	0x84	0x86	**0x7F**	**0x2F**	0x7B	0x79	**0x80**	**0xD0**
20	0x9A	0x4A	0x26	0xE1	0x65	0xB5	0xD9	0x19	0x2A	**0xAE**	**0x0D**	0xE6	0xD5	**0x51**	**0xF2**
21	0x9A	0x4A	0x26	0xF2	0x65	0xB5	0xD9	0x4E	0x12	**0xB0**	**0x2E**	0xB1	0xED	**0x4F**	**0xD1**
22	0x9A	0x4A	0x26	0x3D	0x65	0xB5	0xD9	0xEE	0x42	**0x63**	**0x5E**	0x11	0xBD	**0x9C**	**0xA1**
23	0x9A	0x4A	0x26	0x22	0x65	0xB5	0xD9	0x32	0xD6	**0x89**	**0xA4**	0xCD	0x29	**0x76**	**0x5B**
24	0x9A	0x4A	0x26	0x60	0x65	0xB5	0xD9	0x9F	0xE1	**0x8C**	**0x8A**	0x60	0x1E	**0x73**	**0x75**
25	0x9A	0x4A	0x26	0x6B	0x65	0xB5	0xD9	0xA2	0x71	**0x3B**	**0xC3**	0x5D	0x8E	**0xC4**	**0x3C**
26	0x9A	0x4A	0x26	0xFA	0x65	0xB5	0xD9	0x04	0x95	**0x14**	**0x27**	0xFB	0x6A	**0xEB**	**0xD8**
27	0x9A	0x4A	0x26	0x6C	0x65	0xB5	0xD9	0x71	0x22	**0x99**	**0xC7**	0x8E	0xDD	**0x66**	**0x38**
28	0x9A	0x4A	0x26	0x18	0x65	0xB5	0xD9	0x5B	0xA2	**0x09**	**0x6A**	0xA4	0x5D	**0xF6**	**0x95**
29	0x9A	0x4A	0x26	0x14	0x65	0xB5	0xD9	0xCC	0x31	**0x68**	**0x3C**	0x33	0xCE	**0x97**	**0xC3**
30	0x9A	0x4A	0x26	0xD0	0x65	0xB5	0xD9	0xB1	0xCA	**0x04**	**0x59**	0x4E	0x35	**0xFB**	**0xA6**
31	0x9A	0x4A	0x26	0xB4	0x65	0xB5	0xD9	0x56	0xA6	**0x45**	**0x86**	0xA9	0x59	**0xBA**	**0x79**

Note: in table above, bolded text indicates changes from CL 119 AM values

13

800GBASE-R 32-lane AM block mapping

PCS lane, i	am_mapped 10-bit symbol index, k													
	0	1	2	3	4	5	6	7	8	9	10	11	12	
0	am_0												A	Pad and status field
1	am_1												B	
2	am_2												A	
3	am_3												B	
4	am_4												A	
5	am_5												B	
6	am_6												A	
7	am_7												B	
8	am_8												A	
9	am_9												B	
10	am_10												A	
11	am_11												B	
12	am_12												A	
13	am_13												B	(6 bits) (4 bits)
14	am_14												A	Resumption of 257-bit blocks
15	am_15												B	
16	am_16												C	Pad and status field (replicated?)
17	am_17												D	
18	am_18												C	
19	am_19												D	
20	am_20												C	
21	am_21												D	
22	am_22												C	
23	am_23												D	
24	am_24												C	
25	am_25												D	
26	am_26												C	
27	am_27												D	
28	am_28												C	
29	am_29												D	(6 bits) (4 bits)
30	am_30												C	Resumption of 257-bit blocks
31	am_31												D	

Per 172.2.4.4:
Each flow is identical to the function specified in 119.2.4.4

400GBASE-R flow 0

400GBASE-R flow 1

Mapping of 32-lane AM block to 4 physical lanes with the proposed muxing order

PMA lane, j	am_muxed 10-bit symbol index, m		
	0-95	**96-103**	**104**
0	am_0, am_1, am_16, am_17, am_8, am_9, am_24, am_25 - symbol interleaved (checkerboard removed)	Result of symbol muxing of k=12	
1	am_2, am_3, am_18, am_19, am_10, am_11, am_26, am_27 - symbol interleaved (checkerboard removed)		
2	am_4, am_5, am_20, am_21, am_12, am_13, am_28, am_29 - symbol interleaved (checkerboard removed)		
3	am_6, am_7, am_22, am_23, am_14, am_15, am_30, am_31 - symbol interleaved (checkerboard removed)		

A B C D   A

On each PMA lane, an AM group has 120*8=960 bits, spanning 96 RS symbols and 480 UI

AM group content of PMA(32:4) output
(PCS lane grouping: [0, 1, 16, 17, 8, 9, 24, 25]+2n)

n	AM block content (96 10-bit symbols) - transmission order left to right, LSB first
0	**0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192**, 0x362, 0x042, 0x362, 0x042, 0x202, 0x2B2, 0x202, 0x2B2, 0x194, 0x196, 0x194, 0x196, 0x195, 0x195, 0x195, 0x195, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1F6, 0x276, 0x1F6, 0x276, 0x0B6, 0x3F6, 0x0B6, 0x3F6, 0x3ED, 0x256, 0x01D, 0x1A6, 0x1E9, 0x0EA, 0x219, 0x31A, 0x37A, 0x1C7, 0x085, 0x238, 0x312, 0x1CC, 0x0ED, 0x233, 0x2F3, 0x07E, 0x20C, 0x081, 0x075, 0x13C, 0x08A, 0x1C3, 0x2A6, 0x049, 0x166, 0x389, 0x057, 0x058, 0x397, 0x398, 0x0E0, 0x215, 0x31F, 0x1EA, 0x0CE, 0x3B7, 0x331, 0x048, 0x204, 0x032, 0x1FB, 0x3CD, 0x30C, 0x22A, 0x0F3, 0x1D5
1	**0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192**, 0x062, 0x1A2, 0x062, 0x1A2, 0x3A2, 0x2C2, 0x3A2, 0x2C2, 0x195, 0x195, 0x195, 0x195, 0x195, 0x197, 0x195, 0x197, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x136, 0x3B6, 0x136, 0x3B6, 0x076, 0x136, 0x076, 0x136, 0x01F, 0x398, 0x3EF, 0x068, 0x2A0, 0x1D7, 0x150, 0x227, 0x201, 0x3CF, 0x1FE, 0x030, 0x19B, 0x3AD, 0x264, 0x052, 0x156, 0x3D0, 0x1A9, 0x32F, 0x3D8, 0x238, 0x327, 0x2C7, 0x19E, 0x380, 0x25E, 0x040, 0x0A3, 0x17E, 0x363, 0x2BE, 0x0C3, 0x3F8, 0x33C, 0x007, 0x149, 0x192, 0x2B6, 0x26D, 0x0BD, 0x2A4, 0x342, 0x15B, 0x31E, 0x09C, 0x0E1, 0x363
2	**0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192**, 0x212, 0x322, 0x212, 0x322, 0x182, 0x142, 0x182, 0x142, 0x197, 0x197, 0x197, 0x197, 0x194, 0x194, 0x194, 0x194, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x3B6, 0x276, 0x3B6, 0x276, 0x336, 0x2F6, 0x336, 0x2F6, 0x151, 0x2D4, 0x2A1, 0x124, 0x1D5, 0x0EC, 0x225, 0x31C, 0x13F, 0x147, 0x2C0, 0x2B8, 0x25F, 0x3D9, 0x1A0, 0x026, 0x2F2, 0x1D1, 0x20D, 0x12E, 0x095, 0x3C3, 0x06A, 0x33C, 0x0AC, 0x2B9, 0x36C, 0x179, 0x04C, 0x0A9, 0x38C, 0x369, 0x2E2, 0x301, 0x11D, 0x0FE, 0x09A, 0x283, 0x365, 0x17C, 0x0BA, 0x036, 0x345, 0x3C9, 0x0F1, 0x1A8, 0x30E, 0x257
3	**0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x29A, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192, 0x192**, 0x3D2, 0x222, 0x3D2, 0x222, 0x102, 0x342, 0x102, 0x342, 0x194, 0x194, 0x194, 0x194, 0x196, 0x197, 0x196, 0x197, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x1B5, 0x0B6, 0x3B6, 0x0B6, 0x3B6, 0x1B6, 0x076, 0x1B6, 0x076, 0x3DE, 0x293, 0x02E, 0x163, 0x35B, 0x195, 0x0AB, 0x265, 0x1D8, 0x272, 0x227, 0x18D, 0x2E9, 0x3EC, 0x116, 0x013, 0x1A1, 0x15B, 0x15E, 0x1A4, 0x2A6, 0x179, 0x259, 0x186, 0x1B3, 0x084, 0x273, 0x344, 0x1AA, 0x293, 0x26A, 0x153, 0x234, 0x09D, 0x1CB, 0x362, 0x04C, 0x05A, 0x3B3, 0x3A5, 0x292, 0x179, 0x16D, 0x286, 0x219, 0x164, 0x1E6, 0x29B

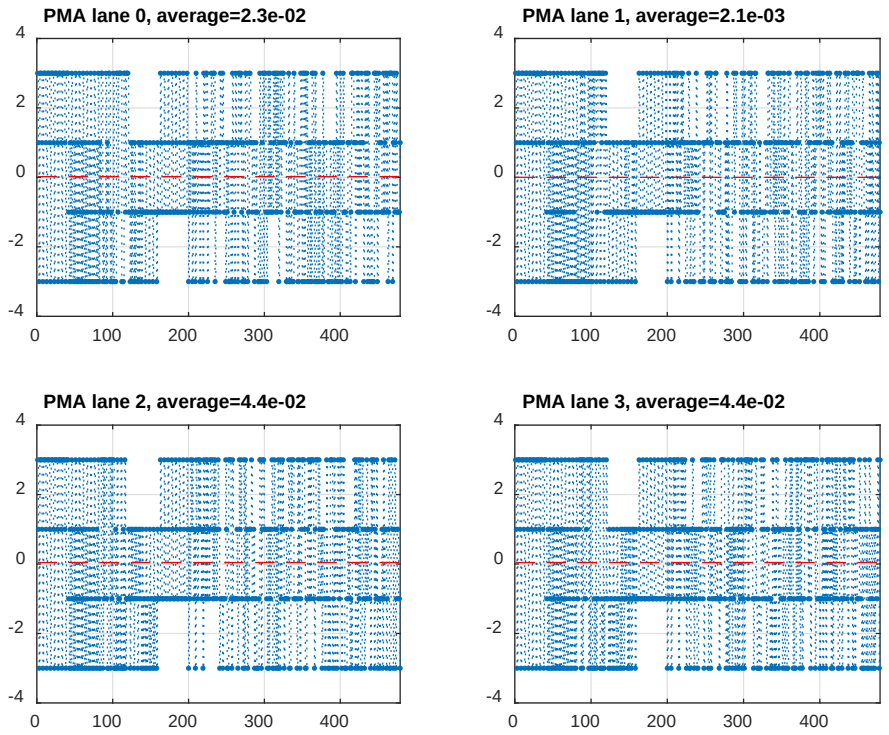
The PCS AMs are de-skewed, and then symbol-muxed (with checkerboard pattern removed) create AM groups. The first 16 symbols (160 bits, 80 PAM4 symbols) in the AM group are identical in all 4 lanes. The 200G/lane AM locking logic identifies the physical lanes and the alignment from the known content of the AM groups.

The resulting PAM4 pattern has very low DC content and reasonable transition densities (see next slides).

Properties of AM groups

(800GBASE-R)

PAM4 sequences of AM groups
(no precoding)



- The standard deviation of the PAM4 constellation {-3, -1, 1, 3} is $\sqrt{5}$
- The average of n=480 random PAM4 symbols has a standard deviation $\sigma_n = \frac{\sqrt{5}}{\sqrt{480}} \approx 0.1$
- The maximum DC content across groups (lanes 2 and 3) is about $\frac{1}{2}\sigma_n$
  - This is a very typical block in terms of DC content
- DC content is not an issue

Transition densities of AM groups
(no precoding)

$\sigma_n = \frac{\sigma}{\sqrt{480}}$ where σ is the standard deviation of the transition probability distribution

Transition type	Density in random data (mean ± σ_n for n=480)	AM group PMA lane 0	AM group PMA lane 1	AM group PMA lane 2	AM group PMA lane 3
All transitions	75% ± 4%	74%	75%	78%	80%
Zero crossings	50% ± 3%	55%	53%	54%	57%
Symmetric crossings	25% ± 2%	24%	24%	26%	25%

- Transition density values deviate from the expected values by up to $2.3\sigma_n$ (usually upward). This deviation occurs in ~1% of random data blocks.
- It is nowhere as pathologic as the "clock content" issue explored in 802.3bs (anslow_01_121916_elect)
  - Likely not an issue for CDRs, but may be implementation dependent