Configurations and Analyses of FEC and Alignment Markers for 400GE

Zhongfeng Wang
Broadcom Corp., USA

IEEE P802.3bs, Task Force meeting, May 2015

SUPPORTERS

- **To be added later.**

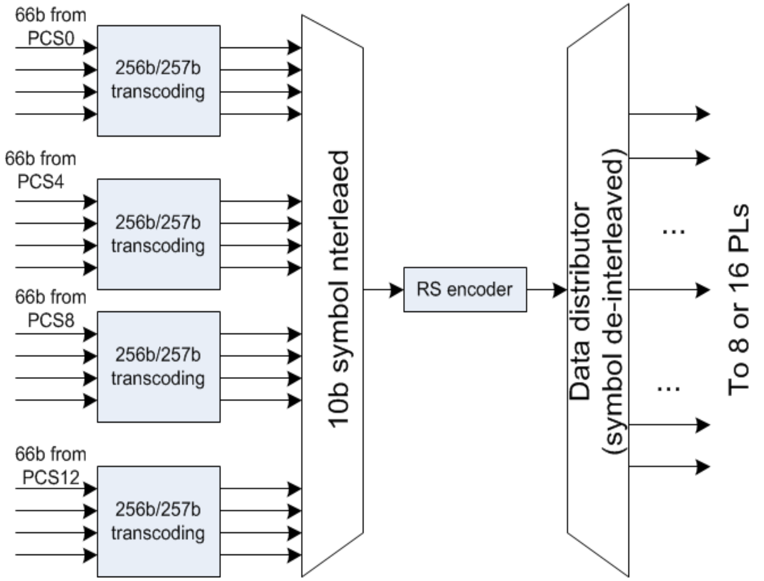
INTRODUCTION

- **Given KP4 FEC, RS(544, 514, t=15, m=10) is adopted for 400GE in March 2015 meeting, this presentation discusses pros and cons for possible FEC configurations: 1X400G, 2X200G, 4x100G, 8x50G, and 16x25G.**

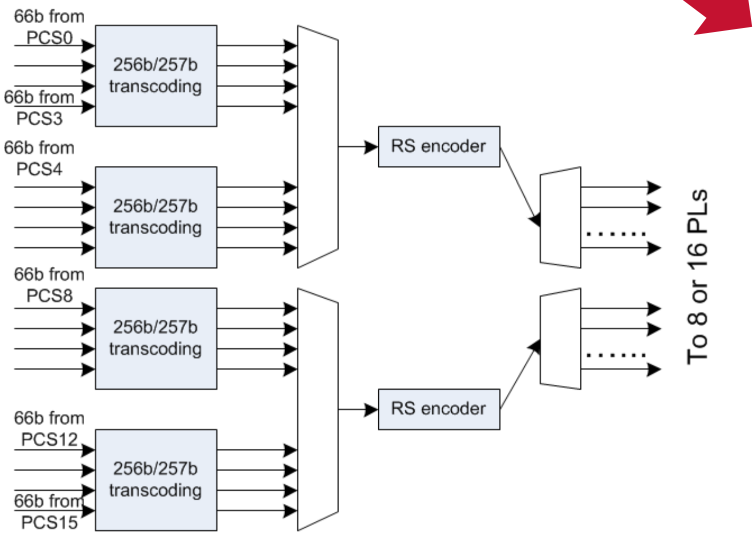
- **In addition, we discuss a new way of data alignment in order to reduce hardware complexity and power consumption in practical implementation.**

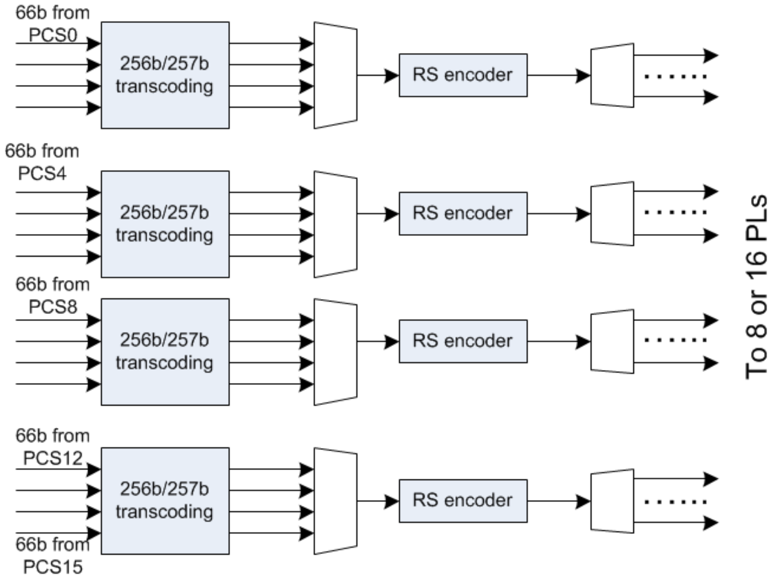
THREE DIFFERENT FEC CONFIGURATIONS

- **1x400G**

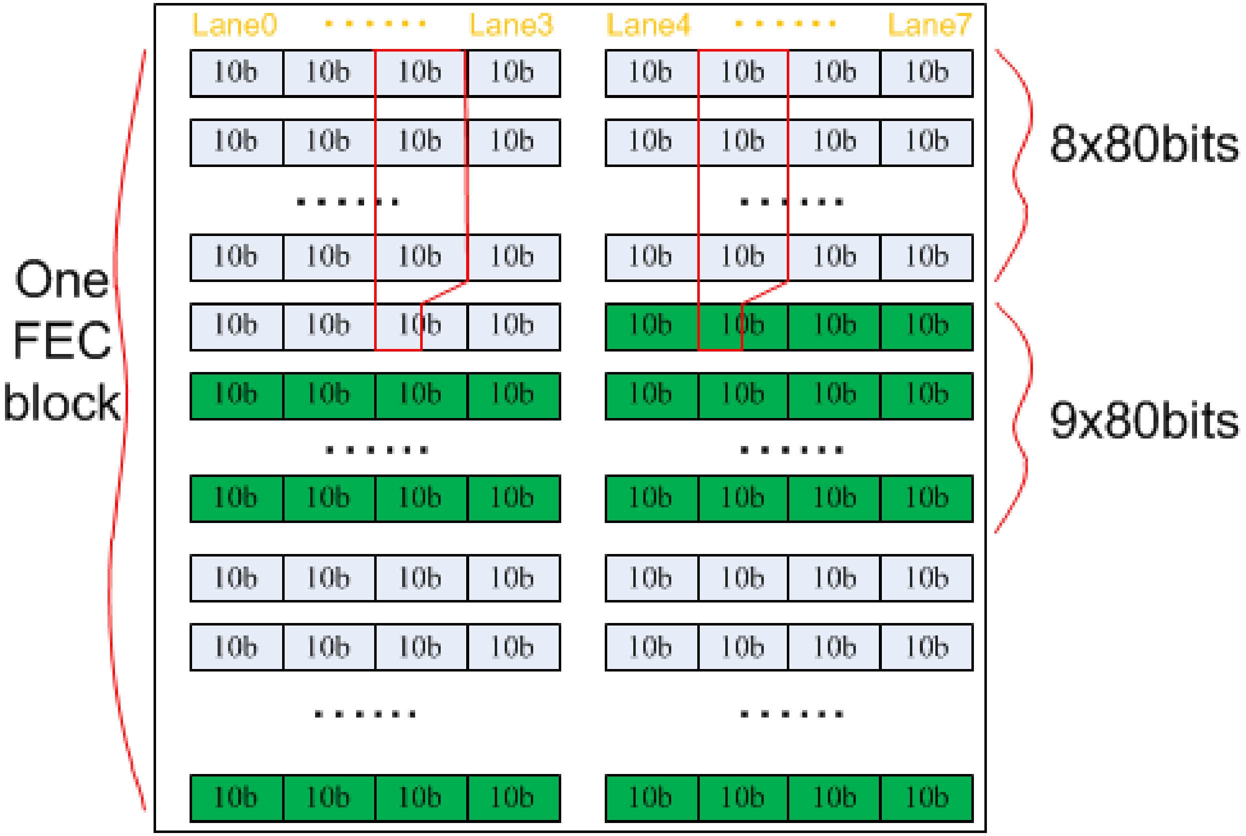


- **2x200G and 4x100G**



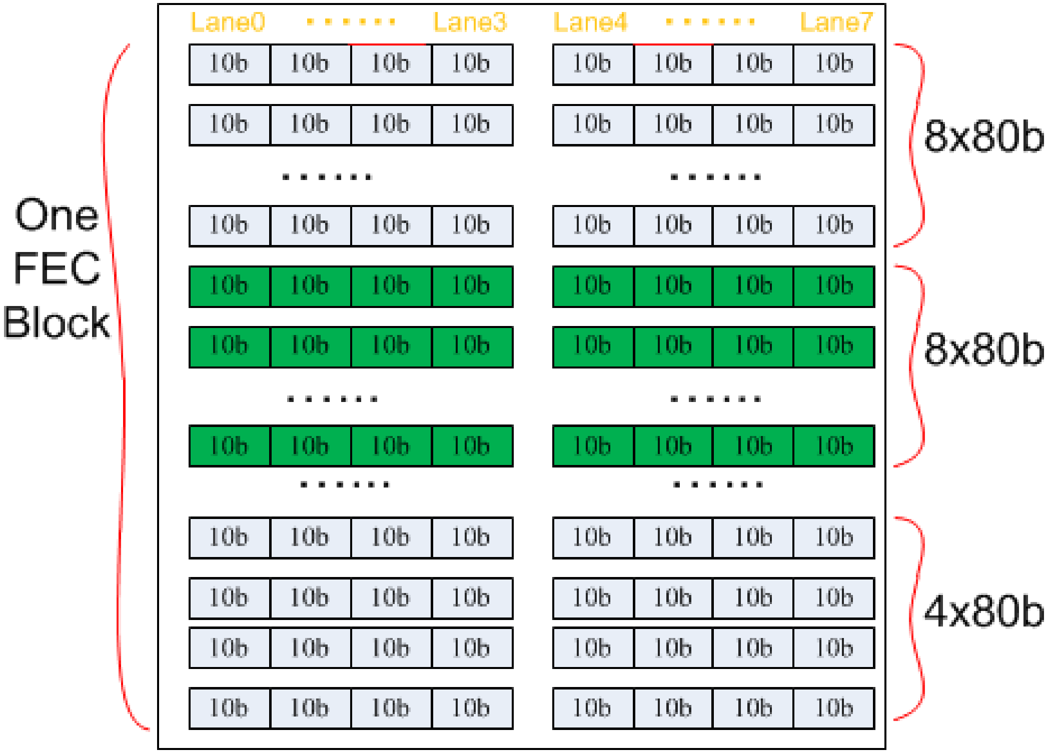


DATA DISTRIBUTION OF 1X400G FEC (IA)



- Consider 680b bus, 5440/8=680 = (8x80+40) bits/cyc. Clock= 625Mhz
- 680/8=85bits/lane/cyc. It requires one extra cycle in either Tx or Rx side.
- Both encoding and decoding can be done efficiently.
- With 16x25G configuration, 85/2=42.5bits/cyc. Non-trivial implementation.

DATA DISTRIBUTION OF 1X400G FEC (IB)



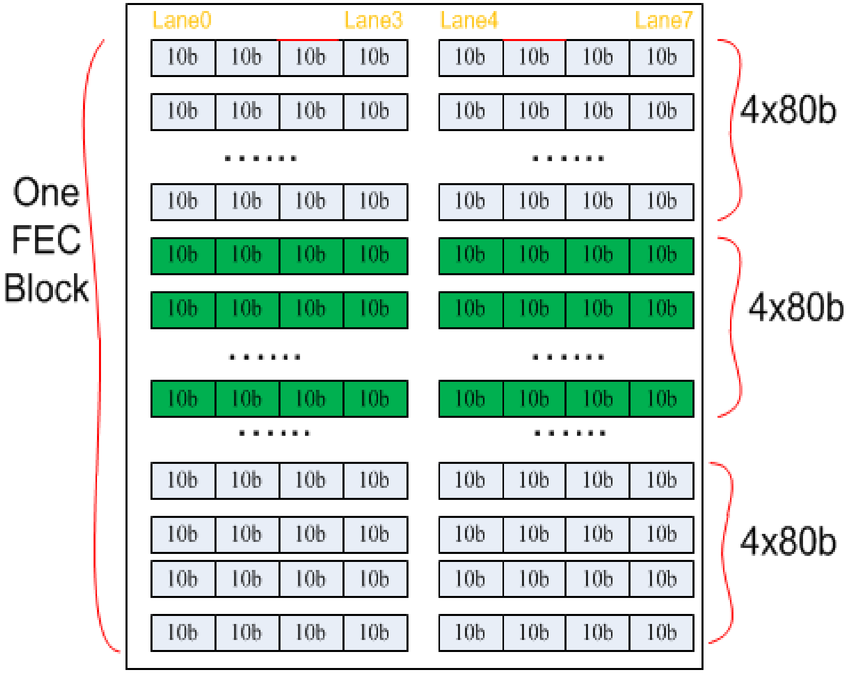
- Using 640b bus, 5440/640= 8.5 cycles . 5140=640x**8** + 20
- Buffer is needed at both Tx (need wait for full cycle data) and Rx sides. A total of two extra cycles are required.
- Both FEC encoder and decoder require extra logic since a FEC block is not transmitted or received in an integer number of cycles.

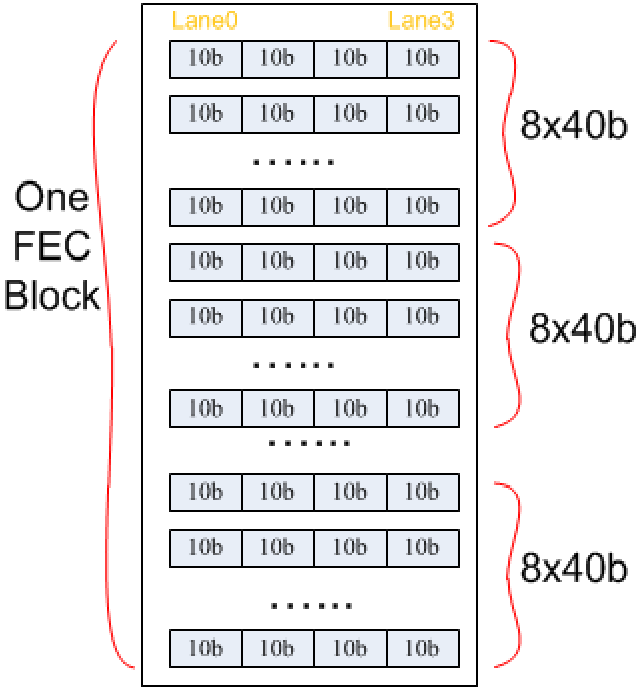
ANALYSES

- With 8x50G aggregation and 1x400G FEC configuration, using 680b bus-width may be better than using 640b for bus-width.

- With 16x25G aggregation and 1x400G FEC configuration, using 640b bus-width may be a better tradeoff than 680b bus-width.

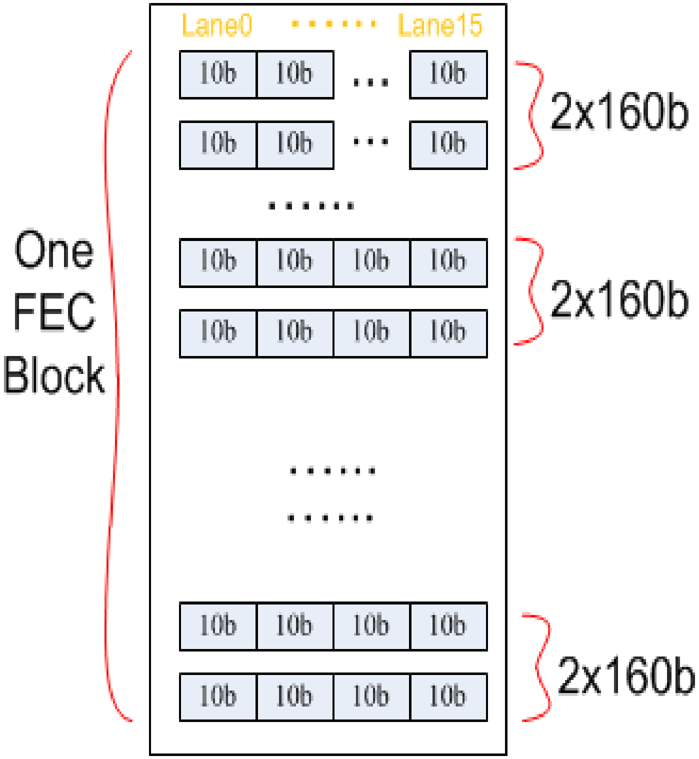
DATA DISTRIBUTION OF 2X200G FEC (II)

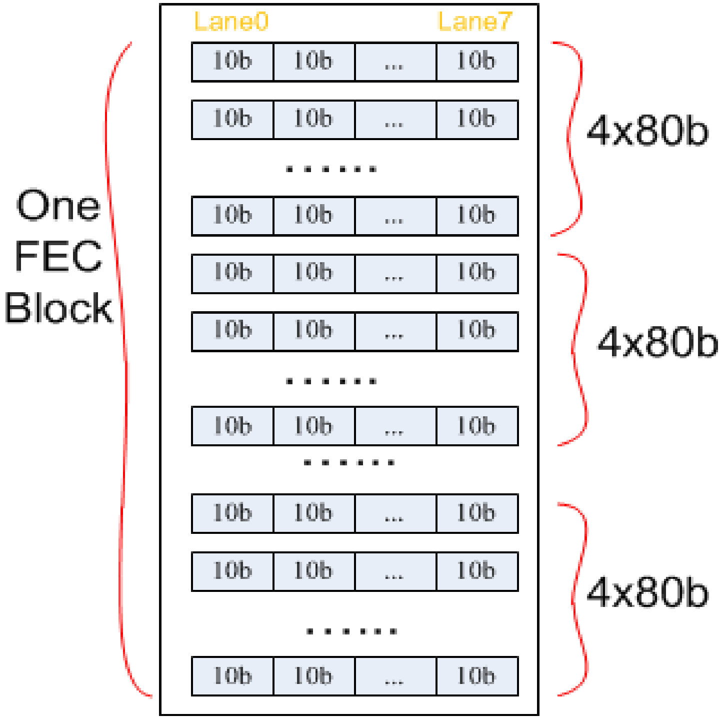




- Option-IIA: 2 FEC codes are symbol-interleaved across 8 PLs. Can reduce performance loss due to burst errors.
- Option-IIB: 200G FEC is coded across 4 Pls (Preferred for implementation).
- May need 2 independent parallel Chien Search engines to reduce latency.
- Total latency change over 1x400G:
  - +12.8 - 2x1.6 = +9.6ns (625Mz clock), or
  - +12.8 - 2x3.2 = +6.4ns (312.5Mhz clock)

DATA DISTRIBUTION OF 2X200G FEC (CONT’D)





- Assume16x25G is considered for 400GE
- Option-A: 2 FEC codes are symbol-interleaved across 16 PLs.
  2x160=320bits/cyc
- Option-B: 200G FEC is coded across 8 Pls.
  4x80=320bits/cyc

DATA DISTRIBUTION OF 4X100G FEC (III)

- This case is easy to handle.
- Consider 8x50G aggregation:
  - Buswidth for each physical lane may be selected as 80bits.

- Assume16x25G is considered for 400GE
  - Buswidth for each physical lane can be selected as 40bits

- To reduce latency, this scheme may need 4 independent highly parallel Chien search engines. Tradeoff between “parallelism level” and “peak power consumption” should be well studied.

DATA DISTRIBUTION OF 8X50G FEC (IV)

- This case is easy to handle.
- Consider 8x50G aggregation:
  - Buswidth for each physical lane may be selected as 80bits.
  - Each FEC block is transmitted/received in 68 cycles.

- Assume16x25G is considered for 400GE
  - Buswidth for each physical lane can be selected as 40bits
  - Each FEC block is transmitted/received in 68 cycles.

- To reduce latency, this scheme may need 8 independent highly parallel Chien search engines. Peak power could be a concern.

DATA DISTRIBUTION OF 16X25G FEC (V)

- This case is easy to handle.

- Assume16x25G is considered for 400GE
  - Buswidth for each physical lane can be selected as 40bits
  - Each FEC block is transmitted/received in 136 cycles.

- To reduce latency, this scheme may need 16 independent highly parallel Chien search engines. Peak power can be a concern.

COMPLEXITY AND LATENCY COMAPRISON

- **Latency**
  - 1x400G: 78ns (+ TC ~ 1.3x2 ns + ENC ~ 1.6x1 ns) (680b bus)
  - 2x200G: 84ns (+ TC ~ 1.3x2 ns + ENC ~ 1.5x2 ns) (320b bus)
  - 4x100G: 110ns (+ TC ~ 1.3x3 ns + ENC ~ 1.5x3 ns) (160b bus)
  - 8x50G : 173ns (+ TC ~ 1.3x5 ns + ENC ~ 1.5x5 ns) (80b bus)
  - 16x25G: 299ns (+ TC ~ 1.3x9 ns + ENC ~ 1.5x9ns) (40b Bus)

- **HW Complexity**
  - 1x400G: 4KES
  - 2x200G: 4KES
  - 4x100G: 4KES
  - 8x50G: 8KES
  - 16x25G: 16KES

  - 1x400G: 1 x 68-P CS
  - 2x200G: 2 x 68-P CS
  - 4x100G: 4 x 68-P CS
  - 8x50G: 8 x 34-P CS (8x 68-P may be too expensive)
  - 16x25G: 16x 17-P CS

DISTRIBUTION OF ALIGNMENT MARKERS

- Assume 16 PCS lanes

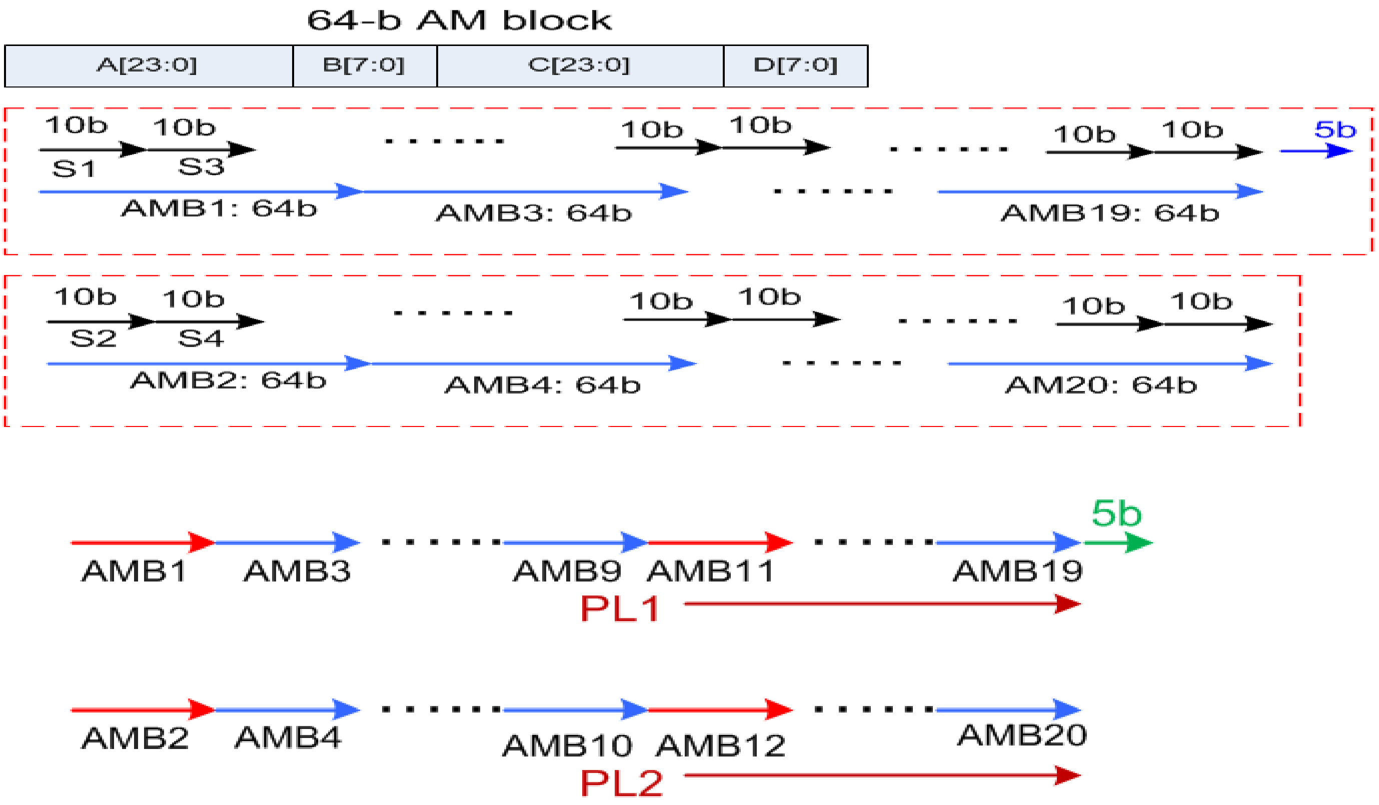
- The distance between two consecutive AM blocks can be [1]
  - 16400 66-b blocks, or
  - 16000 66-b blocks,
  - 16640 66-b blocks
- 16640x66x16/5280= 3328 FEC blocks
  - least redundancy
  - biggest GCM(16640, 16384)
- 16400x66x16/5280= 3280 FEC blocks.
- 16000x66x16/5280= 3200 FEC blocks.

- Can put 5 consecutive AM blocs together per PCS lane. In this case we have 5 AM blocks per AM group per PCS lane for every 5xN 66-b blocks, where N=16000, 16400, or 16640 depending on the choice of distance.

[1] http://www.ieee802.org/3/400GSG/public/13_09/wang_z_400_01_0913.pdf

DISTRIBUTION OF ALIGNMENT MARKERS

- Assume 16 PCS lanes
- 20 AM blocks are distributed over two PLs as follows:



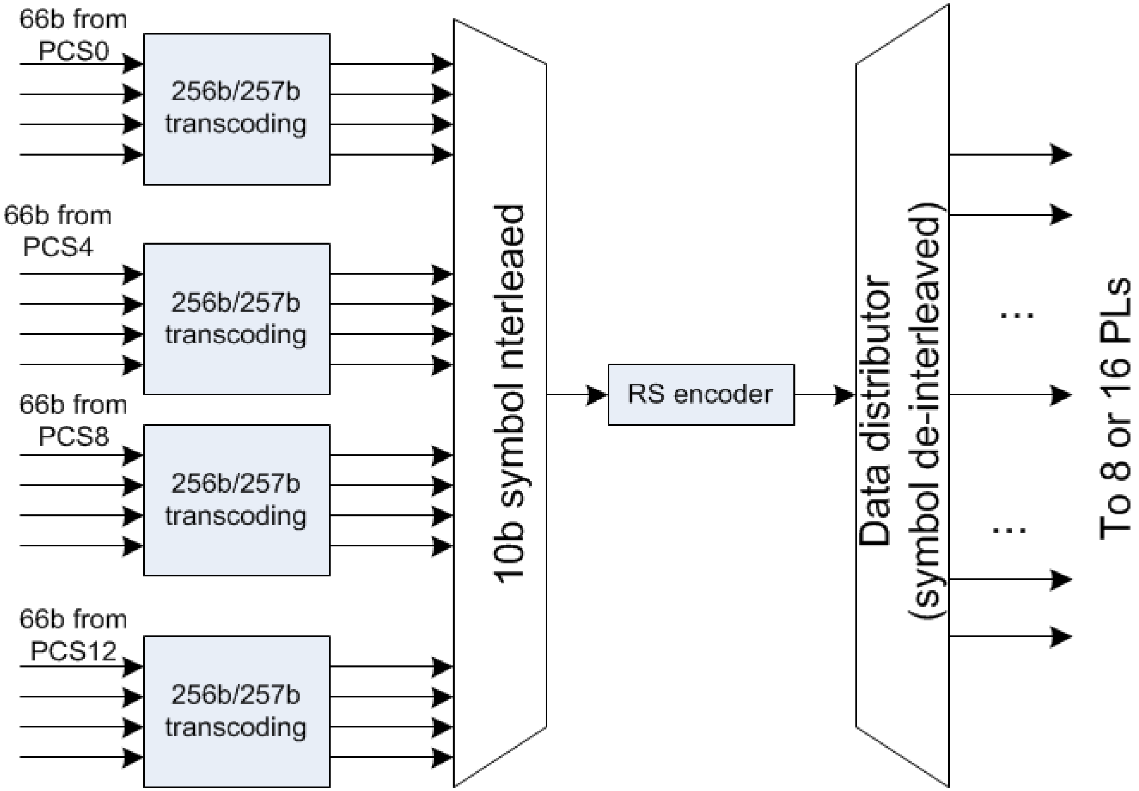
64-b AM block

A[23:0]	B[7:0]	C[23:0]	D[7:0]

- 20 AM blocks can be distributed over 4 PLs as we did in 100GBase-KR4.

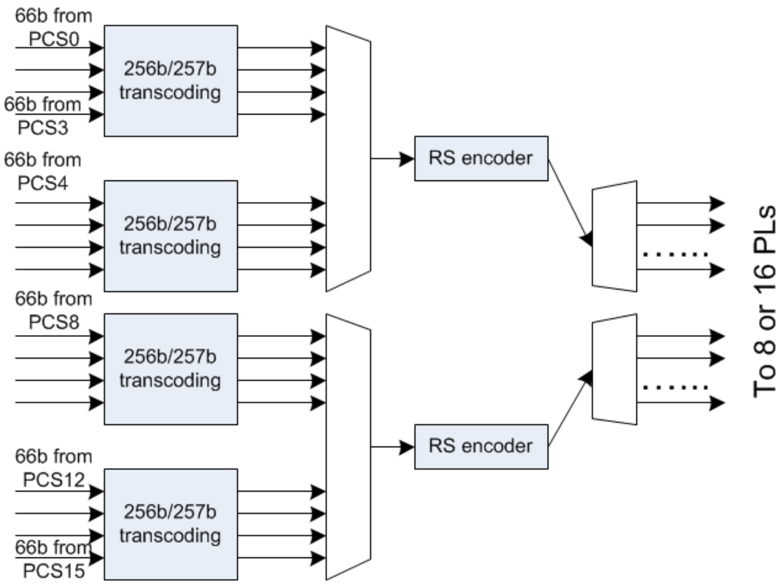
TRANSCODING OPTIONS FOR 1X400G FEC

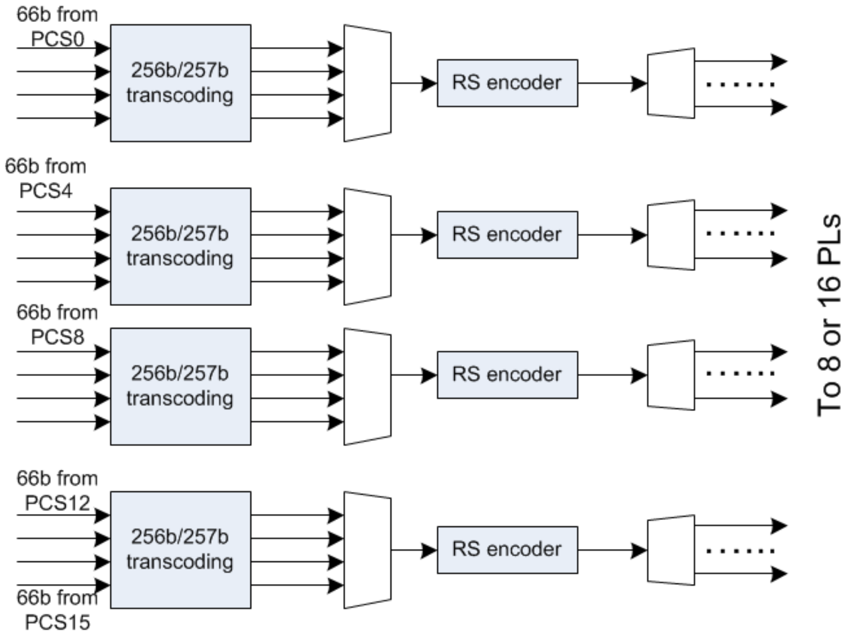
- Data from different PCS lanes should be interleaved at 66-b.
- Option-I: take input from 4 PCS lanes, perform transcoding (preferred)
- Option-II: take input from 8 PCS lanes, perform transcoding
- Option-III: take input from 16 PCS lanes, perform transcoding



TRANSCODING OPTIONS FOR 2X200G AND 4X100G FEC CONFIGURATIONS

- Take input from 4 PCS lanes, perform transcoding
- 2x200G case: the processing flow is shown in the left.
- 4x100G case: the processing flow is shown in the right.





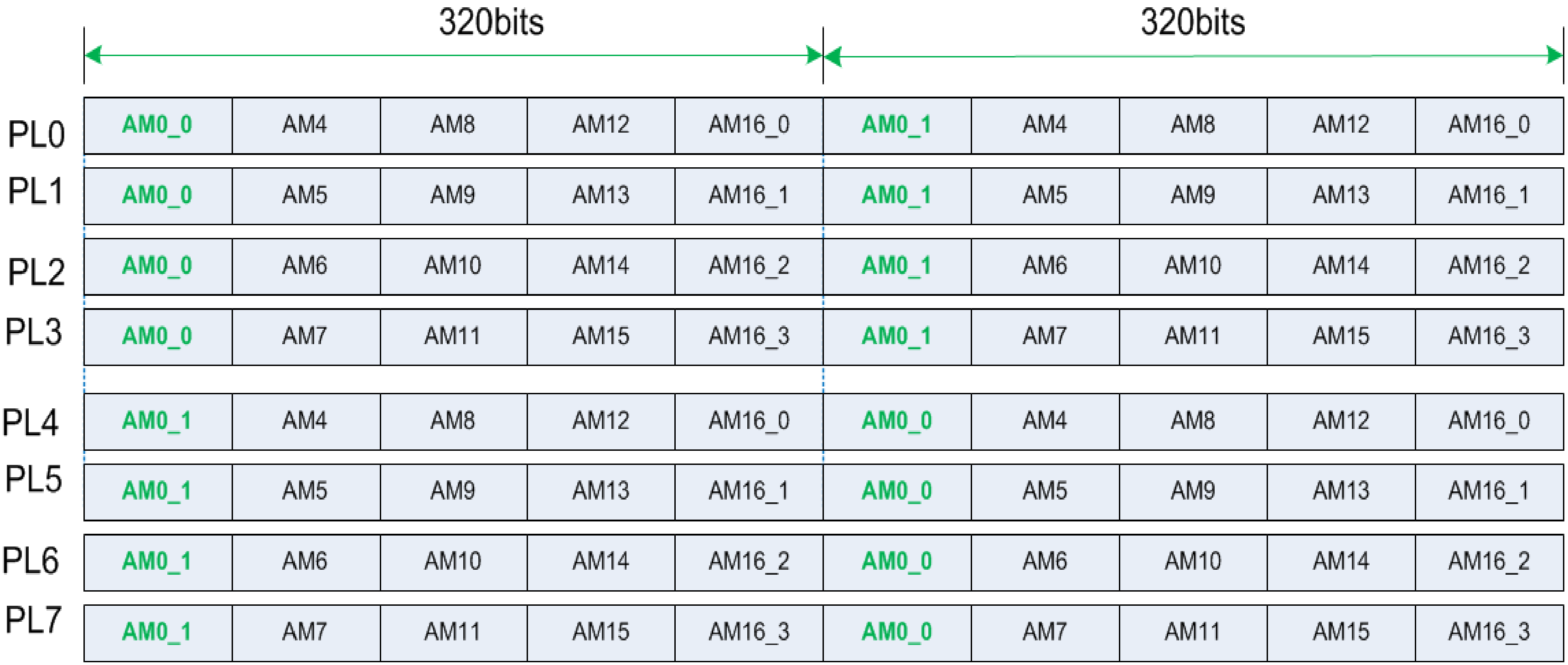
NEW DATA PATTERN MATCHING SCHEME

- Assume using a data bus of 64bits per PL: x[k][63:0].
- To check whether the input data match the AM block, we need a check a total of 64 cases per cycle, which involves significant hardware overhead.
- Consider 8x50G configuration, 80bits or 96bits may be chosen as buswidth. As the buswidth increases, the pattern matching logic complexity will be linearly increased.

- Here we propose an area-efficient approach for data matching. We check if the current received data (with partial history) matches with another data segment received in the past with a fixed distance (e.g., 2 cycles). For buswidth of 64 bits, we may only need check one case: {x[k-1][62:0], x[k][63:0]) with {x[k-3][62:0], x[k-2][63:0]}, which leads to much reduced HW.

- To enable the above discussed efficient data matching algorithm, we need set two related data patterns on a physical lane to be the same or bit-reversed.

ALIGNMENT MARKERS FOR 8 PL'S



	320bits					320bits				
PL0	AM0_0	AM4	AM8	AM12	AM16_0	AM0_1	AM4	AM8	AM12	AM16_0
PL1	AM0_0	AM5	AM9	AM13	AM16_1	AM0_1	AM5	AM9	AM13	AM16_1
PL2	AM0_0	AM6	AM10	AM14	AM16_2	AM0_1	AM6	AM10	AM14	AM16_2
PL3	AM0_0	AM7	AM11	AM15	AM16_3	AM0_1	AM7	AM11	AM15	AM16_3
PL4	AM0_1	AM4	AM8	AM12	AM16_0	AM0_0	AM4	AM8	AM12	AM16_0
PL5	AM0_1	AM5	AM9	AM13	AM16_1	AM0_0	AM5	AM9	AM13	AM16_1
PL6	AM0_1	AM6	AM10	AM14	AM16_2	AM0_0	AM6	AM10	AM14	AM16_2
PL7	AM0_1	AM7	AM11	AM15	AM16_3	AM0_0	AM7	AM11	AM15	AM16_3

- Assume AM0_0 is bit-reversed version of AM0_1 for the portions of data matching.
- Assume AM16_0, AM16_1, AM16_2, and AM16_13 are partially bit-reversed version of each other, e.g., AM16_0={u, v}, AM16_1={~u, v}, AM16_2={~u, ~v}, AM16_3={u, ~v}.

ALIGNMENT MARKERS FOR 16 PL’S

PL0	AM0_0	AM4	AM0_2	AM12	AM16_0
PL1	AM0_0	AM5	AM0_2	AM13	AM16_1
PL2	AM0_0	AM6	AM0_2	AM14	AM16_2
PL3	AM0_0	AM7	AM0_2	AM15	AM16_3
PL4	AM0_1	AM4	AM0_3	AM12	AM16_0
PL5	AM0_1	AM5	AM0_3	AM13	AM16_1
PL6	AM0_1	AM6	AM0_3	AM14	AM16_2
PL7	AM0_1	AM7	AM0_3	AM15	AM16_3
PL8	AM0_2	AM4	AM0_0	AM15	AM16_3
			………		
PL14	AM0_3	AM6	AM0_1	AM14	AM16_2
PL15	AM0_3	AM7	AM0_1	AM15	AM16_3

DATA MATCHING ALGORITHM

- Assume buswidth is set as 80 bits. Input data at time instant k is denoted as x[k][79:0].
- Refer to page-16, rather than comparing 80 cases, we only need consider one case with following two extended vectors:
  - {x[k-1][78:0], x[k][79:0]}
  - {x[k-5][78:0], x[k-4][79:0]}

- In the above, we defined a match when there are M (e.g., M=47) bits matched between two vectors under the condition that each matched segment is sufficient longer (e.g., N>= 23 bits out of 24 consecutive bits stream).

- Once we identify a match, there are many ways to find the head of AM block within the vector. A simple way is to search for head bit-by-bit by matching corresponding 48-b pattern with the known AM block.

FINAL COMMENTS

- We have shown tradeoffs for different configurations of FEC for 400GE.

- We discussed transcoding operation and alignment marker distribution over multiple physical lanes. We proposed a new data matching algorithm that can drastically (>> 10x) reduce hardware implementation complexity.

- Details of data matching implementation may be provided in future meetings.