Logic baseline proposal for 800GBASE-LR1

Eric Maniloff, Ian Betty, Sebastien Gareau, James Harley -- Ciena

Bo Zhang, Kishore Kota, Lenin Patra -- Marvell

IEEE P802.3dj Interim

May, 2023 - San Antonio

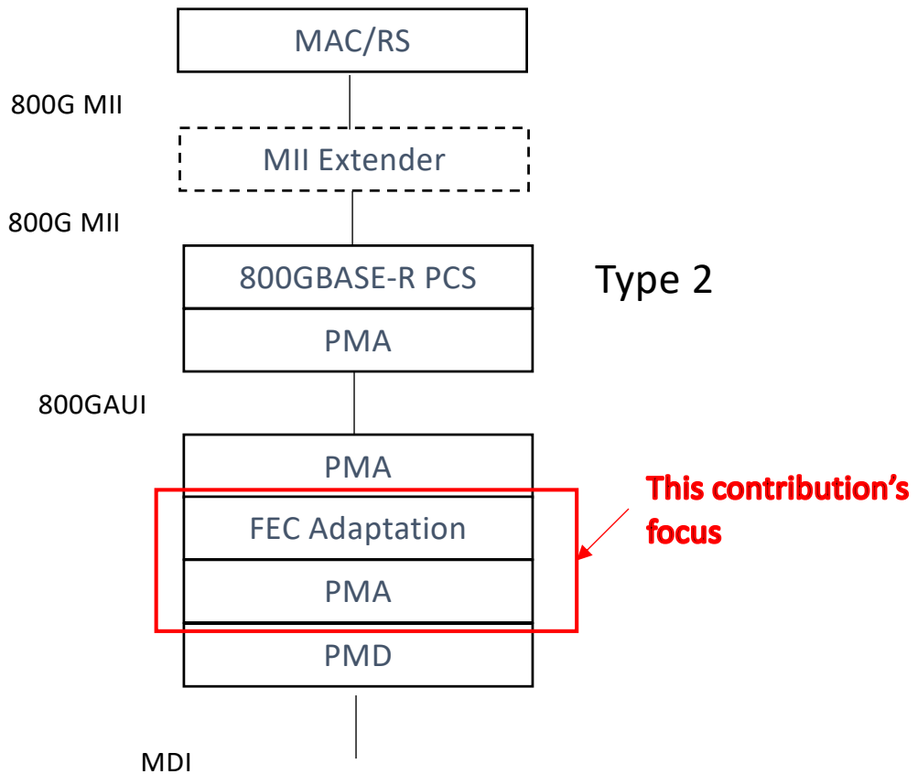
Supporters

Chongjin Xie - Alibaba
Tony Chan Carusone - Alphawave
Or Vidal – Alphawave
Mike Wingrove - Ciena
Hideki Isono - Fujitsu Optical Components
Ali Ghiasi - Ghiasi Quantum
Cedric Lam - Google
Xiang Zhou - Google
Xiang Liu - Huawei
Kechao Huang - Huawei
Peter Stassar – Huawei
Piers Dawe - Nvidia
Di Zhang - Kuaishou
Ernest Muhigana – Lumentum
Madhu Krishnaswamy - Lumentum
Arash Farhoodfar - Marvell
Samuel Liu, Marvell
Pranay Aiya, Marvell
Zhigang Gong - O Net
Gurinder Parhar - Source Photonics
Frank Chang - Source Photonics

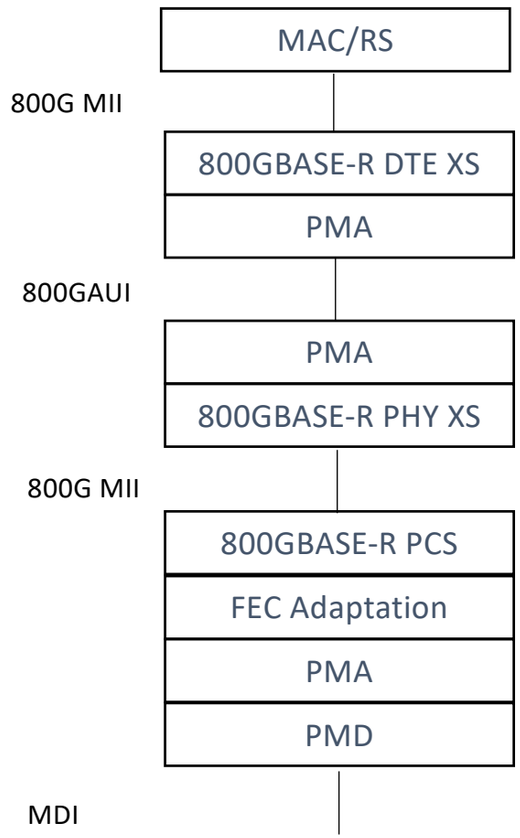
Overview

- This contribution provides a logic baseline for 800GBASE-LR1 based on a BCH inner code
  - This approach provides a power and latency optimized solution for the 802.3dj 800G 10km single wavelength objective
- The building blocks are presented, to allow a logical baseline to be adopted
- Previously, optical budgets have been presented for 800GBASE-LR1 based on this FEC scheme
  - https://www.ieee802.org/3/dj/public/23_03/maniloff_3dj_01a_2303.pdf
- The approach defined is compatible with the current OIF 800LR IA, in order to allow re-use of the logical architecture
  - Details follow oif2022.340.02 “800LR Baseline Proposal Update“ and oif2022.341.00 “800LR Proposal updates and discussion”

Data Path Detail



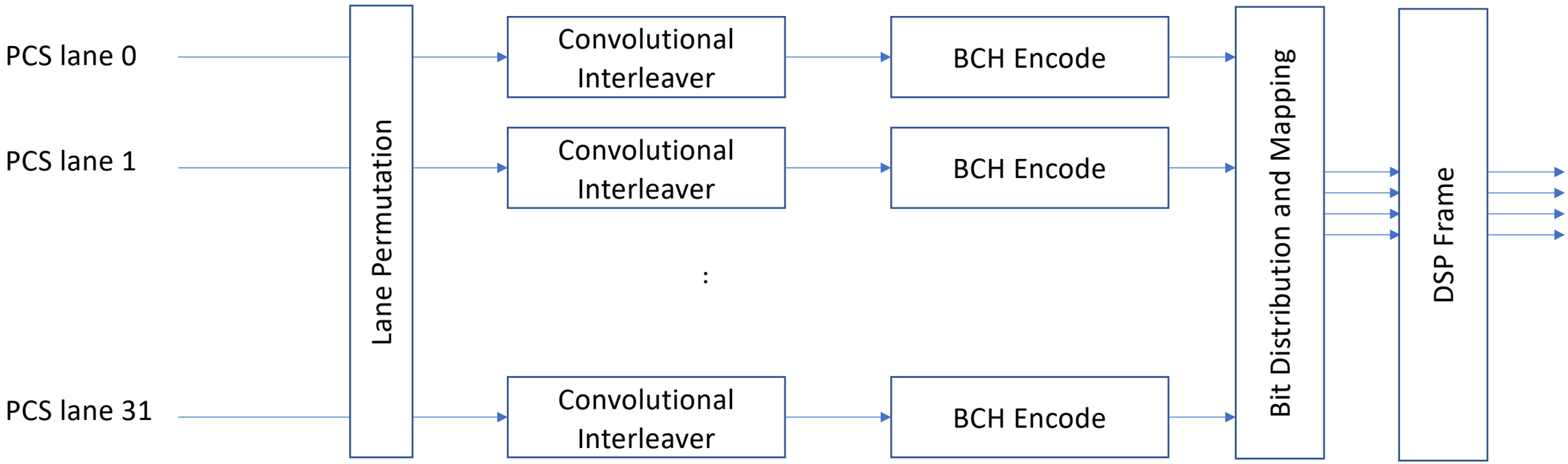
Type 2



Type 3

- https://www.ieee802.org/3/dj/public/adhoc/optics/0223_OPTX/brown_3dj_optx_adhoc_01a_230222.pdf
- Either Type 2 or Type 3 can be supported
- Focus of this contribution is on Type 2

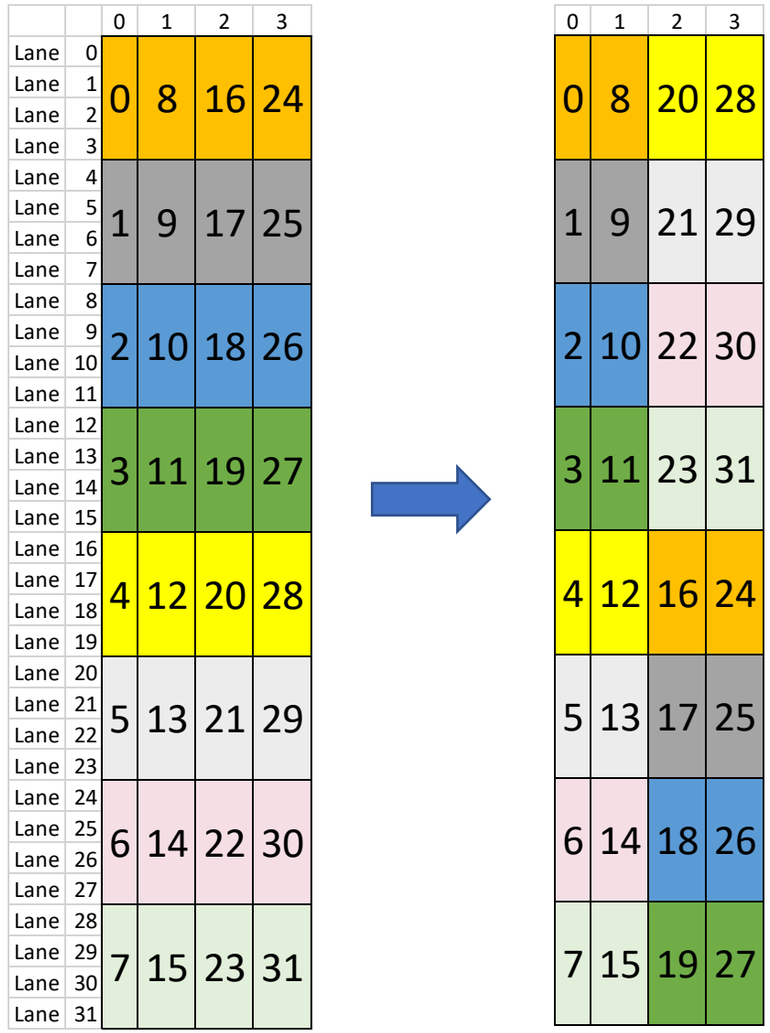
800LR FEC Adaptation



Synchronous data path maintains Ethernet ± 50 ppm timing

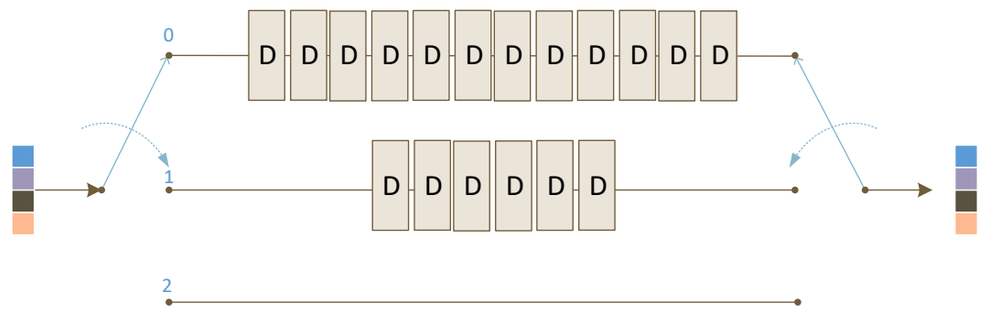
Lane Permutation

- Purpose: Provides 10bit symbols from 4 interleaved RS(544,514) codewords on each lane
- Prerequisites: 10bit symbol alignment across lanes. Lane reorder and FEC codeword deskew across lanes is optional
- Operates on 4 RS-symbol boundaries across 32 PCS lanes



FEC Lane Convolutional Interleaver

- Purpose: Ensure each BCH codeword contributes no more than one 10-bit RS symbol to each RS codeword.
- One Convolutional Interleaver per PCS lane.
- Interleaver has three parallel delay lines
- D represents storage of 40b
- Word width is 4 symbols or 40 bits at I/O



Interleaver Latency of 55 nanoseconds

BCH Encoder

- Purpose: Inner code works in conjunction with outer KP4 FEC to provide a high-performance FEC for 800LR
- BCH (126,110) code operates on 11 RS symbols at output of convolutional interleaver
- Simple Chase decoders can achieve a pre-FEC BER threshold of ~1.1e-2 for a post-KP4 BER of 1e-15
- Code definition:
  - Define c as a binary vector of length 126, and $c(x)$ a polynomial of degree 125

with the coefficients defined by c (where the bit 0 of c represents the coefficient of power 125).

  - Then c is a codeword of the BCH(126,110) code if $c(x)$ is divisible by the binary polynomial

$$g(x) = x^{16} + x^{14} + x^{11} + x^{10} + x^9 + x^7 + x^5 + x^3 + x + 1$$

Symbol Mapping and Polarization Distribution

Additional details in oif2022.341.00, "800LR Proposal updates and discussion"

		10bits 0	10bits 1	10bits 2	10bits 3	10bits 4	10bits 5	10bits 6	10bits 7	10bits 8	10bits 9	10bits 10	16bits Syndrome
Lane	0	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	S0
Lane	1	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	S1
Lane	2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	S2
Lane	3	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	S3
Lane	4	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	S4
Lane	5	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	S5
Lane	6	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	S6
Lane	7	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	S7
Lane	8	J0	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	S8
Lane	9	K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	S9
Lane	10	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	S10
Lane	11	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	S11
Lane	12	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	S12
Lane	13	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	S13
Lane	14	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	S14
Lane	15	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	S15
Lane	16	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	S16
Lane	17	U0	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	S17
Lane	18	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	S18
Lane	19	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	S19
Lane	20	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	S20
Lane	21	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	S21
Lane	22	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	S22
Lane	23	AA0	AA1	AA2	AA3	AA4	AA5	AA6	AA7	AA8	AA9	AA10	S23
Lane	24	AB0	AB1	AB2	AB3	AB4	AB5	AB6	AB7	AB8	AB9	AB10	S24
Lane	25	AC0	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	AC10	S25
Lane	26	AD0	AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8	AD9	AD10	S26
Lane	27	AE0	AE1	AE2	AE3	AE4	AE5	AE6	AE7	AE8	AE9	AE10	S27
Lane	28	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	S28
Lane	29	AG0	AG1	AG2	AG3	AG4	AG5	AG6	AG7	AG8	AG9	AG10	S29
Lane	30	AH0	AH1	AH2	AH3	AH4	AH5	AH6	AH7	AH8	AH9	AH10	S30
Lane	31	AJ0	AJ1	AJ2	AJ3	AJ4	AJ5	AJ6	AJ7	AJ8	AJ9	AJ10	S31

shuffled

	10bits 0	10bits 1	10bits 2	10bits 3	10bits 4	10bits 5	10bits 6	10bits 7	10bits 8	10bits 9	10bits 10	16bits Syndrome
0	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	S0
1	B9	B10	B0	B1	B2	B3	B4	B5	B6	B7	B8	S1
2	C7	C8	C9	C10	C0	C1	C2	C3	C4	C5	C6	S2
3	D5	D6	D7	D8	D9	D10	D0	D1	D2	D3	D4	S3
4	E3	E4	E5	E6	E7	E8	E9	E10	E0	E1	E2	S4
5	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F0	S5
6	G10	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	S6
7	H8	H9	H10	H0	H1	H2	H3	H4	H5	H6	H7	S7
8	J6	J7	J8	J9	J10	J0	J1	J2	J3	J4	J5	S8
9	K4	K5	K6	K7	K8	K9	K10	K0	K1	K2	K3	S9
10	L2	L3	L4	L5	L6	L7	L8	L9	L10	L0	L1	S10
11	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	S11
12	N9	N10	N0	N1	N2	N3	N4	N5	N6	N7	N8	S12
13	P7	P8	P9	P10	P0	P1	P2	P3	P4	P5	P6	S13
14	Q5	Q6	Q7	Q8	Q9	Q10	Q0	Q1	Q2	Q3	Q4	S14
15	R3	R4	R5	R6	R7	R8	R9	R10	R0	R1	R2	S15
16	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T0	S16
17	U10	U0	U1	U2	U3	U4	U5	U6	U7	U8	U9	S17
18	V8	V9	V10	V0	V1	V2	V3	V4	V5	V6	V7	S18
19	W6	W7	W8	W9	W10	W0	W1	W2	W3	W4	W5	S19
20	X4	X5	X6	X7	X8	X9	X10	X0	X1	X2	X3	S20
21	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y0	Y1	S21
22	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	S22
23	AA9	AA10	AA0	AA1	AA2	AA3	AA4	AA5	AA6	AA7	AA8	S23
24	AB7	AB8	AB9	AB10	AB0	AB1	AB2	AB3	AB4	AB5	AB6	S24
25	AC5	AC6	AC7	AC8	AC9	AC10	AC0	AC1	AC2	AC3	AC4	S25
26	AD3	AD4	AD5	AD6	AD7	AD8	AD9	AD10	AD0	AD1	AD2	S26
27	AE1	AE2	AE3	AE4	AE5	AE6	AE7	AE8	AE9	AE10	AE0	S27
28	AF10	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	S28
29	AG8	AG9	AG10	AG0	AG1	AG2	AG3	AG4	AG5	AG6	AG7	S29
30	AH6	AH7	AH8	AH9	AH10	AH0	AH1	AH2	AH3	AH4	AH5	S30
31	AJ4	AJ5	AJ6	AJ7	AJ8	AJ9	AJ10	AJ0	AJ1	AJ2	AJ3	S31

To DP-QAM16 mapper

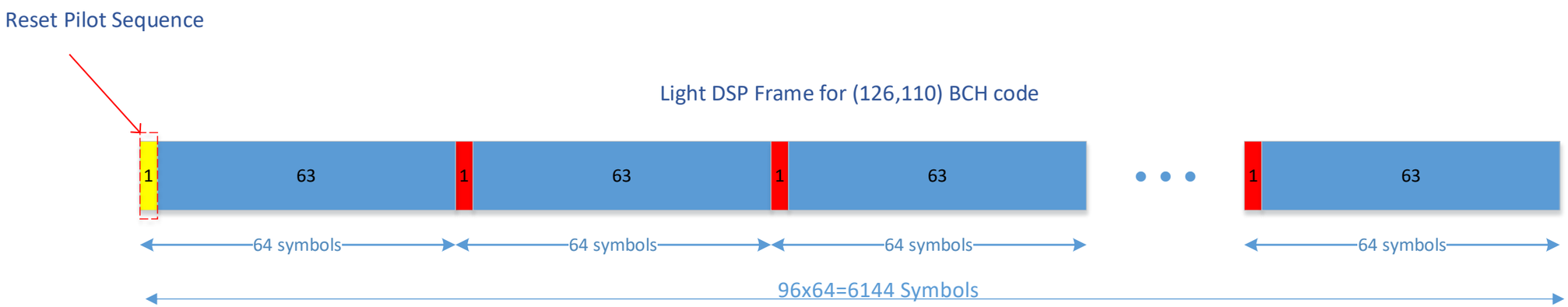
- The RS symbols encoded by each BCH encoder are shuffled to improve burst tolerance
- 4 contiguous lanes are mapped to XI/XQ/YI/YQ of 63 DP-QAM16 symbols (shown as red box in the accompanying figure)
- Mapping ensures bits from each BCH codeword are distributed among polarization and constellation points

Mapping to DP-QAM16 symbols

- Suppose bits of BCH encoder output for time k and lane p (for $p = 0,1,\cdots 31$) are denoted: $\left(b_0^{k,p}, b_1^{k,p}, \cdots, b_{109}^{k,p}, c_0^{k,p}, c_1^{k,p}, \cdots c_{15}^{k,p}\right)$ where the information bits are denoted $b_0^{k,p}, b_1^{k,p}, \cdots, b_{109}^{k,p}$ and the check/parity bits are denoted $c_0^{k,p}, c_1^{k,p}, \cdots c_{15}^{k,p}$
- Bit shuffling: Information bit q of lane p is rotated to position $(q + 20 * p)\%110$ prior to mapping
- Mapping: Suppose $S_l^k = \left(s_{XI}^{k,l}, s_{XQ}^{k,l}, s_{YI}^{k,l}, s_{YQ}^{k,l}\right)$ denote the DP-QAM16 symbols (prior to pilot insertion) for $l = 0,1,\cdots,503$
  - $s_{XI}^{k,l}$ is formed from bits $[2 * l\%63 + l\%2, 2 * l\%63 + (l+1)\%2]$ of lane $4 * \left\lfloor \frac{l}{63} \right\rfloor + (2l + \left\lfloor \frac{l}{2} \right\rfloor \%2)\%4$
  - $s_{XQ}^{k,l}$ is formed from bits $[2 * l\%63, +l\%2, 2 * l\%63 + (l+1)\%2]$ of lane $4 * \left\lfloor \frac{l}{63} \right\rfloor + \left(2l + \left\lfloor \frac{l}{2} \right\rfloor \%2 + 1\right)\%4$
  - $s_{XI}^{k,l}$ is formed from bits $[2 * l\%63 + l\%2, 2 * l\%63 + (l+1)\%2]$ of lane $4 * \left\lfloor \frac{l}{63} \right\rfloor + \left(2l + \left\lfloor \frac{l}{2} \right\rfloor \%2 + 2\right)\%4$
  - $s_{YQ}^{k,l}$ is formed from bits $[2 * l\%63, +l\%2\ 2 * l\%63 + (l+1)\%2]$ of lane $4 * \left\lfloor \frac{l}{63} \right\rfloor + \left(2l + \left\lfloor \frac{l}{2} \right\rfloor \%2 + 3\right)\%4$

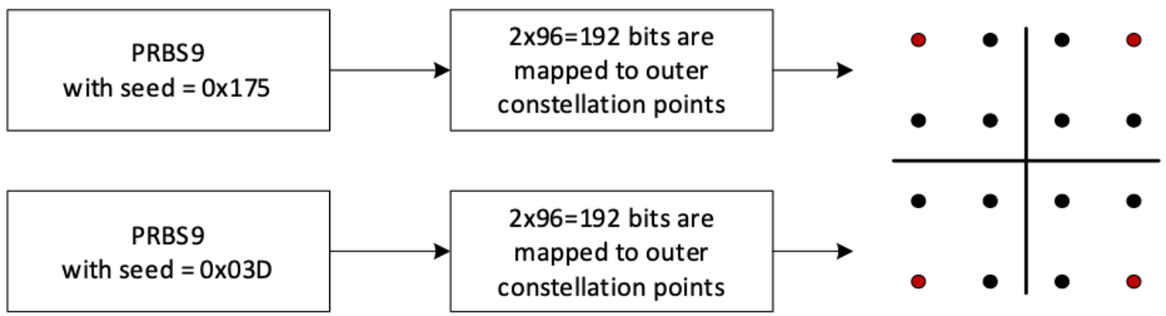
800LR DSP Frame

- Pilots are included in 1 of 64 symbols (one pilot symbol, 63 payload symbols.)
  - 63 payload symbols is an easy multiple for BCH(126,110), no RES (padding) needed
- DSP Frame is 96*64= 6144 symbols including 96 pilots.
- The pilot sequence is a PRBS9 pattern initialized at the beginning of the DSP Frame.
  - Seed value for pilot reset is chosen to ensure DC balance.
- Each DSP Frame is aligned to 32 (126,110) BCH codewords and aligned to the 32 FEC lanes.
- This simplified frame relative to 800ZR, reduces latency by allowing clock domain crossings to be avoided through bus width adjustments.

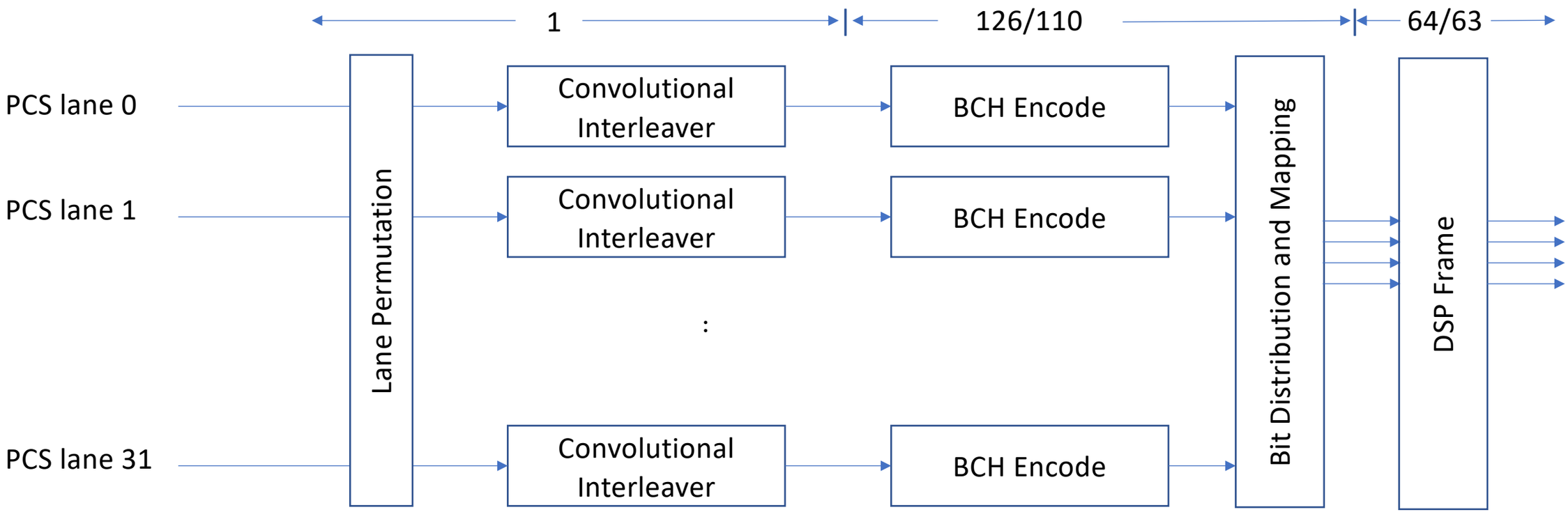


Pilot Sequence

- Pilot symbols are implemented on the outer symbols of the dual pol 16QAM constellation, allowing robust framing to a DP QPSK constellation
- The pilot sequence is a fixed 96 symbol sequence from a PRBS9



Timing Overview



Summary

- A baseline proposal for the 800GBASE-LR1 logical implementation is presented
- This baseline provides a low power and latency solution for an optimized coherent interface for 800GBASE-LR1
- This logical approach can also be applied to 800GBASE-ER1

Thanks!