Proposal of BCH/Hamming Inner Code of Type 2 PHY/FEC Scheme (Concatenated FEC)

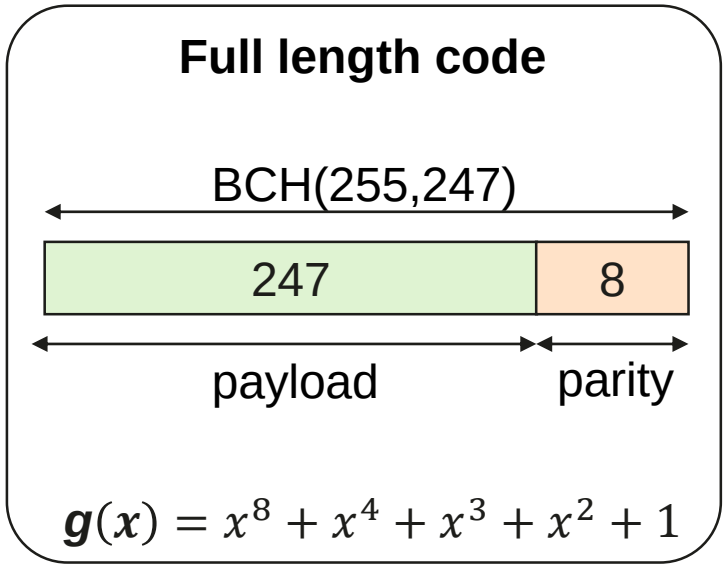
Xinyuan Wang, Xiang He, Hao Ren
Huawei Technologies

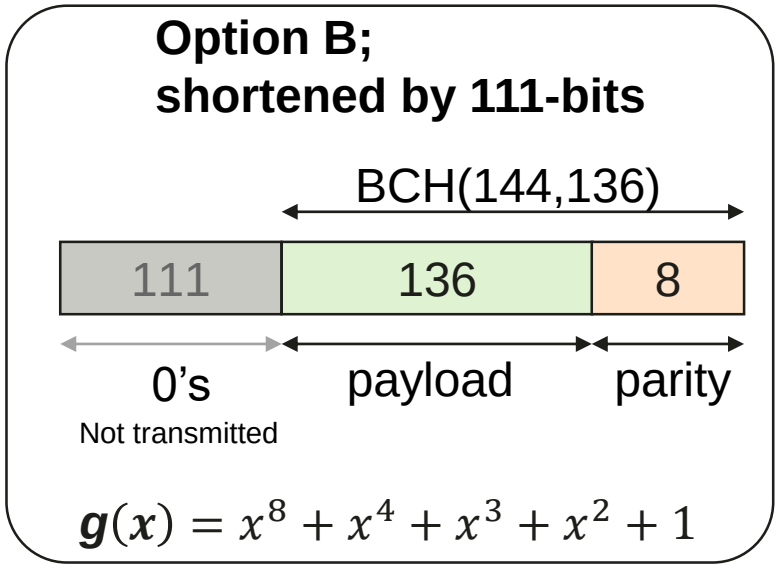
Background:

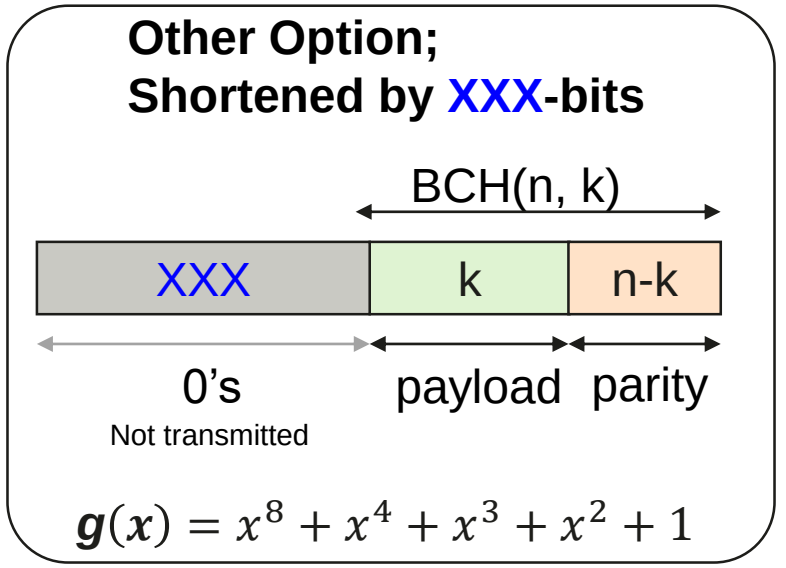
- Soft-decoded, BCH/Hamming inner codes with adopted RS(544,514) as outer code for 200 Gb/s per lambda IM-DD optical PMD are interested in P802.3dj with high net coding gain, and low area, power, and latency.
- BCH(144,136) in bliss_3df_01b_2211 and Hamming(128,120) with padding in farhood_3dj_01a_230206 were discussed in previous P802.3df/dj meeting.
- In this presentation, we will compare these two candidates from code constructing, implementation and integration in to PCS/PMA perspective to propose a suitable code.

How to construct a BCH/Hamming code

- In he_3df_01_221005, constructing a narrow-sense binary primitive BCH code with t = 1.
  - Shorten the m = 8 primitive BCH(255,247), by prefixing to the message bits a sequence of 0s.
    - E.g., we can use primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$ (“implicit + 1” notation 0x8E) to construct the code.
    - There are many other primitive polynomials with degree of 8: 0x95, 0xAF, 0xB1, 0xB2, 0xB4, 0xE1, 0xF3, ...
    - The zero prefix sequence is not transmitted and is only used to calculate the parity of the primitive code.



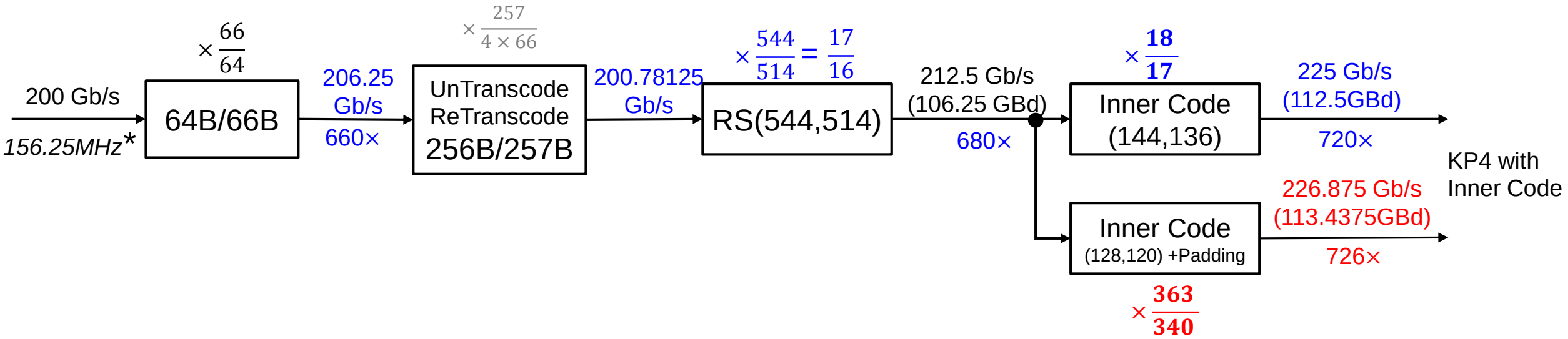




Generator polynomial unchanged.

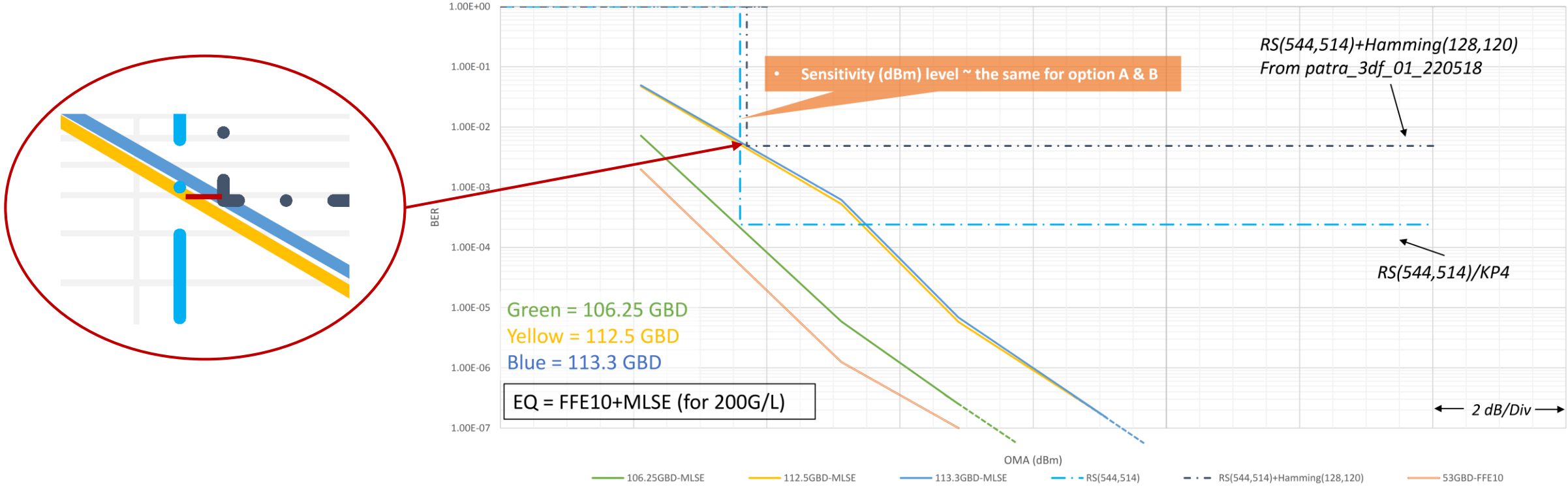
Requirement for inner code rate: enabling integer PLL

- BCH(144,136): inner code rate 17/18 to enable integer PLL, clear rate number at 720×156.25M = 112.5 GBd.
- Hamming(128,120) with padding: inner code rate 363/340 to enable integer PLL, rate number at 726×156.25M = 113.4375 GBd.



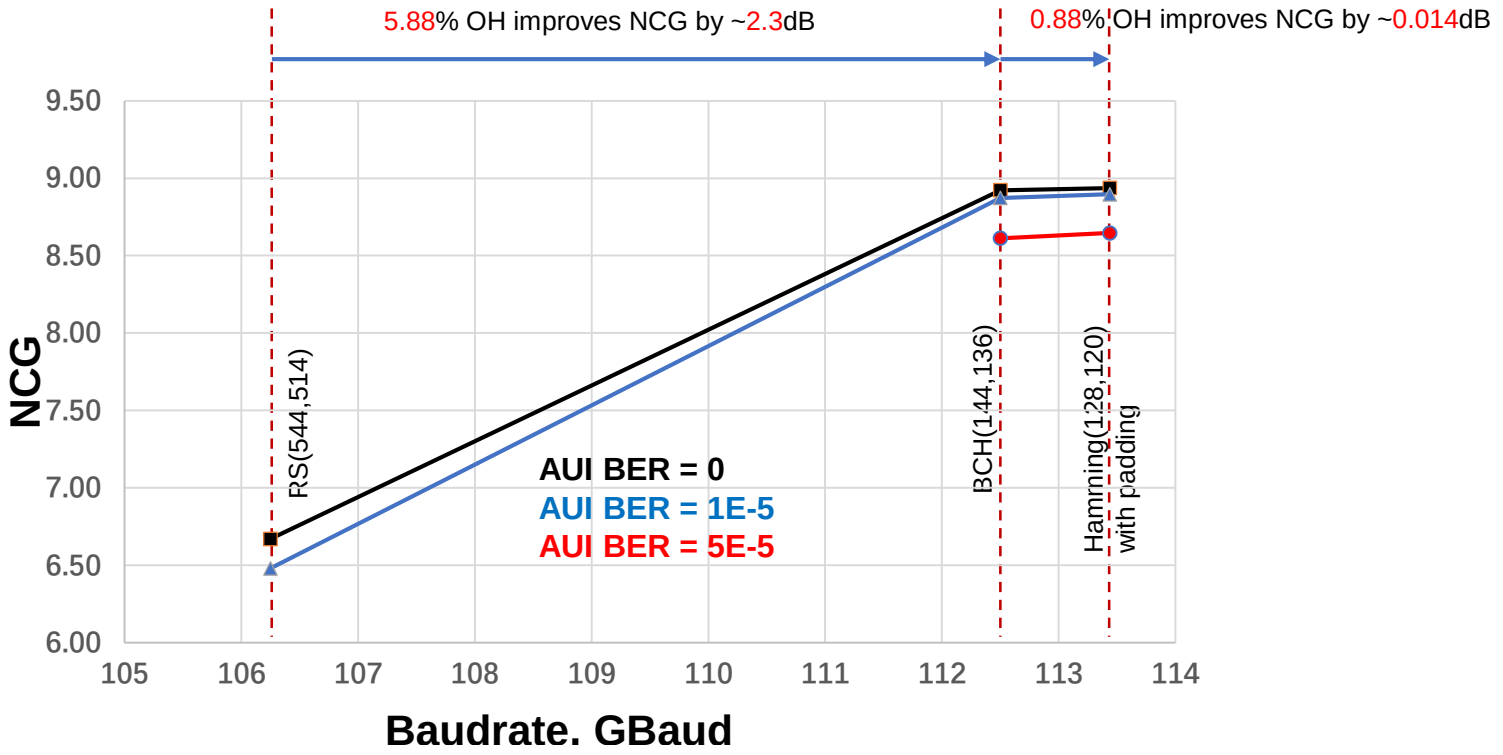
Tradeoff needed as overhead of inner code will impact PHY capability

- In welch_3dj_01_230206, the degradation of increased baud rate from 112.5 GBd of BCH(144,136) to 113.4375 GBd of Hamming(128,120) with padding seems to be about -0.1dB optical.
- Refer to he_3dj_01a_230206, due to ~0.88% higher overhead, the NCG of “(128,120)+padding” is only 0.014 dB higher than (144,136), so **negative** FEC performance improved.



Limited upper bound overall FEC performance by higher overhead inner code

- The ~0.88% more overhead of Hamming (128,120)+padding gives 0.014dB NCG only, comparing to ~5.88% overhead of BCH(144,136) with ~2.3dB NCG improvement from RS(544,154) .



- The additional power saved by using a lower baud rate code in overall E2E optical PHY/link, can be better used to improve soft-decoding algorithm for more coding gain. Some basic ideas are: using more LRPs or using more test patterns, or using more sophisticated decoding algorithms than Chase-II to get **positive** FEC performance improvement by BCH(144,136).

Padding to Hamming(128,120) introduce additional complexity/power and latency

- wangz_3dj_01a_230206 revisited why KP4 was selected in 802.3bj, and compared current inner code proposals:

Inner code	NCG	Over clocking	Codec complexity	Design & verification	Total Rx power increase	Gearbox Latency
(144,136)	very close	9.09 %	Lower	Easier	0% (ref. design)	None
(128,120) +padding	very close	10.0 %	Higher	Complex	15~20% x power of (ref-codec)	+ 2 cycles Tx+Rx at least

- In maniloff_3df_01b_2207 with supporters, a concatenated code with BCH(126,110) was proposed for 800 Gb/s coherent solution aimed to eliminate gearbox:

Concatenated RS(544,514)+BCH(126,110) with full 11- way 10-bit Symbol interleaving for no correlation of BCH decoder errors for each RS(544,514) codeword

1/64th Pilot DSP frame with no Training Symbols to eliminate gearbox and enable coherent phase detection with no cycle slips

Low latency encoder architecture with no gearbox and no PCS lane de-skew

An alternative is RS(544,514)+BCH(176,160) with 800G ZR DSP frame

Are the padding bits required in P802.3dj?

- No explicit proposals on how to use the padding bits as feedback channel between TX and RX.
  - Is it a patch up solution for Hamming(128,120) to satisfy integer PLL as the main reason?
  - Should all FEC code proposals in P802.3dj support this feedback channel to guarantee normal operating? Padding bit can be added to any code if necessary.
- In P802.3dj, a FEC code should enable reliable communications to meet BER/FLR and MTTFPA.
  - The full concatenated code coding gain is required for the padding bits to meet these requirement.
  - Padding protected by (128,120) can achieve ~1E-3/4 BER at ~4E-3 pre-FEC BER with HD/SD decoder respectively.
  - Assuming padding bits are protected by the same (128,120) code with soft-decision decoding, and a CRC-8 as in farhood_3dj_01a_230206, the MTTFPA is < 35 ms even if we assume inner FEC can always correct 3 bit errors.
  - Repeated transmissions of messages and using majority voting may improve MTTFPA but efficiency gets lower.
  - The equivalent bandwidth of padding is $226\ Gbps * \frac{38*8}{384} * \frac{3}{3264} = 164\ Mbps$. Retransmission of 10 times will lower it to 16.4 Mb/s.

384b padding protection	MTTFPA
None	4 us
(128,120) w/ hard-decision	15 us
(128,120) w/ soft-decision	< 530 us
(128,120) w/ soft-decision + CRC8 on 38 bytes	< 35 ms

List of some basic integer PLL based BCH/Hamming inner codes

- Table below listed some basic inner codes with rate from 710×156.25M = 110.9375 GBd to 730×156.25M = 114.0635 GBd.

m (Galois field index)	Inner code (n,k)	Code rate	Baud Rate (GBd)	Bit Rate (Gb/s)	Multiple of 156.25MHz
10	BCH(210,200)	20/21	111.5625	223.125	714
	BCH(180,170)	**17/18**	**112.5**	**225**	**720**
	BCH(146,136)	68/73	114.0625	228.125	730
	eBCH(231,220)	20/21	111.5625	223.125	714
	eBCH(198,187)	17/18	112.5	225	720
	eBCH(181,170)	170/181	113.125	226.25	724
	eBCH(284,272)	68/71	110.9375	221.875	710
	eBCH(267,255)	85/89	111.25	222.5	712
	eBCH(252,240)	20/21	111.5625	223.125	714
	eBCH(216,204)	17/18	112.5	225	720
	eBCH(182,170)	85/91	113.75	227.5	728
9	BCH(213,204)	68/71	110.9375	221.875	710
	BCH(189,180)	20/21	111.5625	223.125	714
	BCH(179,170)	170/179	111.875	223.75	716
	BCH(162,153)	17/18	112.5	225	720
	BCH(145,136)	136/145	113.28125	226.5625	725
	eBCH(210,200)	20/21	111.5625	223.125	714
	eBCH(180,170)	**17/18**	**112.5**	**225**	**720**
	eBCH(146,136)	68/73	114.0625	228.125	730
	eBCH(231,210)	20/21	111.5625	223.125	714
	eBCH(198,187)	17/18	112.5	225	720
	eBCH(181,170)	170/181	113.125	226.25	724
8	BCH(178,170)	85/89	111.25	222.5	712
	BCH(168,160)	**20/21**	**111.5625**	**223.125**	**714**
	BCH(144,136)	**17/18**	**112.5**	**225**	**720**

Self-sync of basic integer PLL codes

- Inner codes does not rely on padding bits to achieve frame sync.
  - Self-sync as defined in Clause 74 is a code-independent codeword sync method.
- Self-sync requires minimal logic (if any) since decoders are already implemented.
  - Gate count is estimated to be <10% of AM lock.
- Self-sync performs better than AM lock in every aspect, especially at ~10^{-3} BER level.

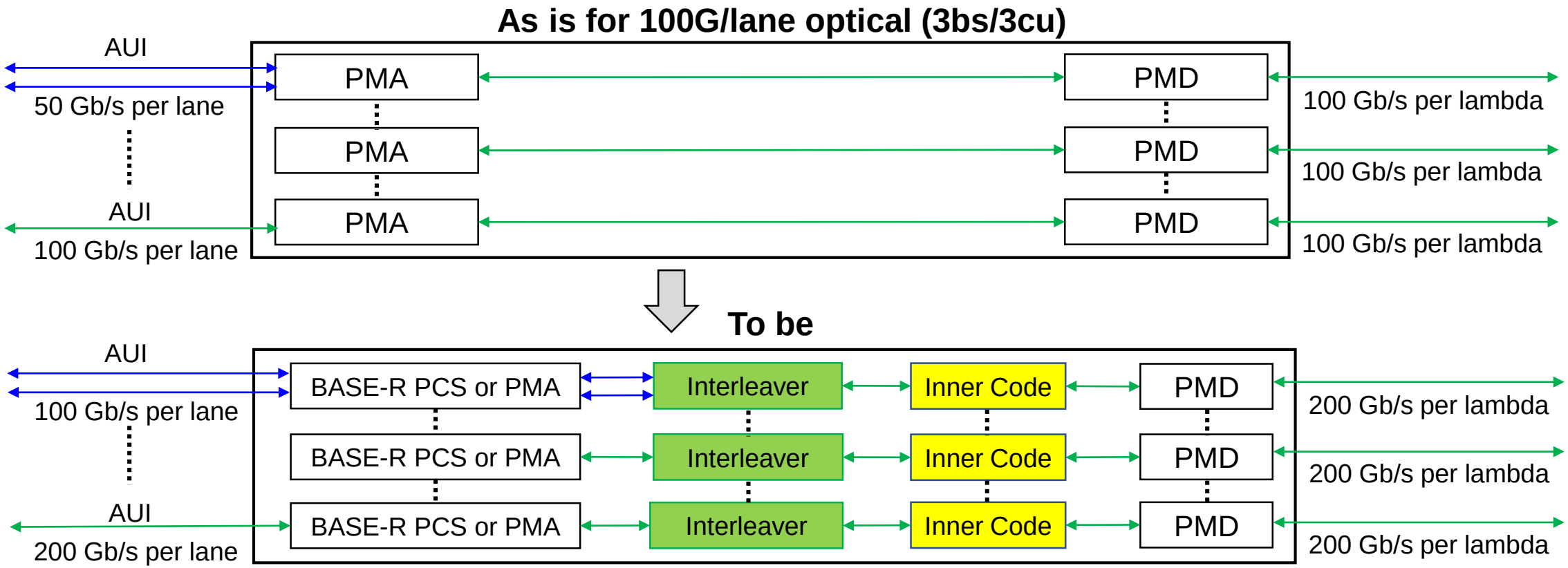
	CL119 AM Lock	**BCH(144,136) Self-sync**	**Notes**
Mean time to lock, **µs**	~150	**<1**	Mean time to find the codeword boundary on a bit stream. Lower is better.
Mean time to false-lock, **years**	7.5 x10^{21}	**1.4x10^{23}**	Mean time that it locks to a wrong position. Should never happen.
Mean time to false-unlock, **years**	1.7x10^{15}	**1.9x10^{18}**	Mean time that the lock breaks during normal operation. Should never happen.
Mean time to unlock, **µs**	~500	**0.26**	Mean time to drop sync when needs to. Lower is better.

The above table is based on 2.4E-3 pre-FEC BER, 200 Gb/s per lambda and the following assumptions:

- Finding the sync position: Check 30 codewords for each position, pick the position with most correct codewords.
- Validate the position found: See if at least 110 codewords in the following 200 codewords are also good. If so, sync established.
- Monitor sync status: See if there are less than 70 codewords are correct in the following 200 codewords. If so, drop sync.

25G/lane vs 100G/lane design of inner codes

- With 1.6TE PCS baseline adopted, 100G/lane design of inner code process flow is more reasonable than 25G/lane based design.
  - Especially with symbol-pair muxing PMA, going all the way back to 25G/lane PCS lanes is unnecessary.



Conclusions:

- Propose to adopt BASIC integer PLL based BCH/Hamming inner code WITHOUT padding, such as BCH(144,136), for P802.3dj “200 Gb/s per lambda” optical PMDs.
- BCH(144,136) results relaxed requirements for optical TRX and link, comparing to Hamming(128,120) with padding, with lower complexity, lower power, and lower latency advantages.
- Using 100G/lane rather than 25G/lane design for the inner code flow is more reasonable considering 1.6TE.

Thanks!