Reconsider PCS Coding for 400GbE

Haoyu Song, Wenbin Yang, Tongtong Wang

Why PCS Encoding

- **DC Current Balance**
  - Equal number of positive and negative pulses
- **Clock Recovery**
  - Enough transitions and short run length
- **Improve MTTFPA**
  - Large code hamming distance
- **In-band Non-data Control Information**
  - Maintain efficiency for large code word
- **Adapt to particular line rate**
- **As technology evolves, some are less important and some new requirements emerge.**

New Factors to Consider

- **Line speed**
  - Higher line speed requires lower coding overhead
  - Higher line speed requires low coding complexity and latency
- **Scrambler**
  - DC balance and run length are determined by scrambling
- **FEC**
  - Code hamming distance is no longer critical
  - Code overhead affects the over-clock ratio
  - Code size affects the FEC block alignment
- **PMD Line Modulation**
  - Reach specifications influence PMD solutions and FEC algorithms
    - SMF: OOK, PAMn, or DMT
    - MMF: requirements depend on reach
    - Backplane/Cable: NRZ or PAM4

History of Ethernet Coding

Ethernet Standard		Line/PCS Code	Pro	Con
10M		Manchester	Simple, DC balance, self-clocking	Up to 50% overhead
100BASE-X/T4		4B/5B, 8B/6T	Simple	20% overhead
1000BASE-X		8B/10B	Relatively simple	20% overhead
10GBASE-LX4/CX4				
10GBASE-SR/LR/ER		64B/66B	3% overhead	Relatively complex
40G				
100G	802.3ba			
	802.3bj	256B/257B Trans.	0.4% overhead, suitable for FEC, allow logic reuse	Based on 64B/66B, extra step with longer latency
400G		TBD		

- **802.3bj inherits the 802.3ba architecture and is made adaptive to the additional RS-FEC, so a transcoding from 64B/66B is used**

400GbE PCS Coding Considerations

- **What are not so important any more – Just do the best**
  - Code DC balance and run length
    - Scrambler
  - Code hamming distance for MTTFPA
    - FEC: UCR becomes the most important factor to affect MTTFPA
- **What are important – Design in a holistic way**
  - Low complexity and low latency
  - Low coding overhead
  - Suitable for the base-line FEC algorithm
  - Suitable for the PCS lanes and PMA interface
  - Suitable for possible higher gain FEC algorithms for different PMDs

What’ s different for 400GbE

- **FEC is needed in many PHY application scenarios, so it is possible to be included as an integral part of the PCS**
  - Higher gain FEC, if needed, can be added between PCS and PMD
  - Guarantee BER performance at MAC/PLS service interface
- **RS-FEC can correct both burst errors and random errors, and has enough coding gain for many physical interfaces, so it is likely to be chosen as the base-line FEC algorithm**
- **RS-FEC puts some constraints on the PCS coding word size**
  - RS(n, k, t, m) must satisfy 0 < k < n < 2^m
    - k is at most 2^m-1-2t (e.g. 239B data per codeword for m=8 and t=8)
  - PCS coding word should align with the RS codeword
    - PCS coding word size is typically 64i+j (i = 1, 2, 4, 8 … and j is a small integer)
- **There are some feasible coding word sizes but not a lot**

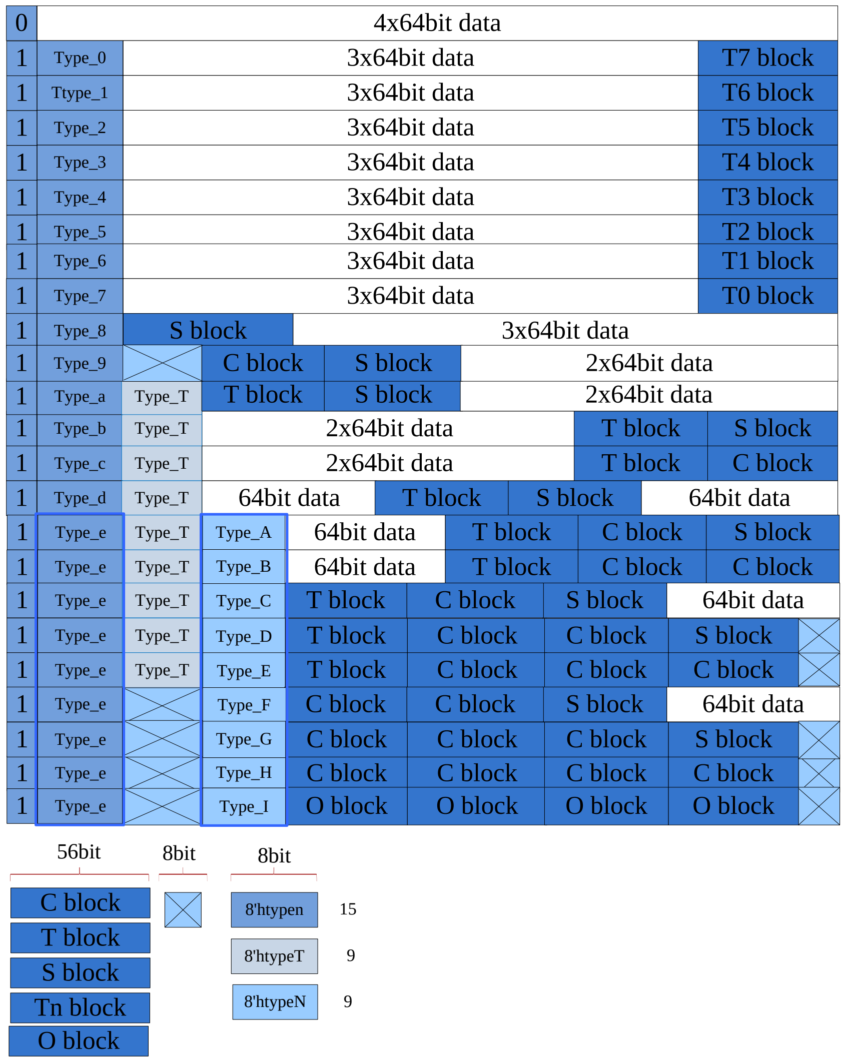
Possible Coding Block Sizes & Approach

Option	FEC Code RS(n, k, t, m)	Trans-coding	Effective Gain BER= 10^{-15}	Overall Latency	Total Area (40nm gates)	Total Power	Input BER for 10^{-15} BER	Input BER for 10^{-12} BER
1	RS(528, 514, 7, 10)	512b/514b	4.87 dB	99.4 ns	275k	101 mW	4.68×10^{-6}	2.34×10^{-5}
2	RS(528, 513, 7, 10)	512b/513b	4.87 dB	99.4 ns	285k	105 mW	4.68×10^{-6}	2.34×10^{-5}
3	RS(528, 516, 6, 10)	512b/516b	4.52 dB	96.8 ns	243k	88 mW	1.86×10^{-6}	1.12×10^{-5}
4a	RS(468, 456, 6, 9)	512b/513b	4.51 dB	96.3 ns	197k	72 mW	1.82×10^{-6}	1.23×10^{-5}
4b	RS(234, 228, 3, 9)	512b/513b	2.06 dB	52.9 ns	108k	40 mW	2.39×10^{-10}	9.77×10^{-8}
5a	RS(528,516,6,10)	256b/258b	4.52 dB	90 ns	212k	77mW	1.86×10^{-6}	1.12×10^{-5}
5b	RS(264,258,3,10)	256b/258b	2.35dB	49 ns	113k	41mW	9.12×10^{-10}	1.12×10^{-7}

*** Refer to *gustlin_01_0112.pdf***

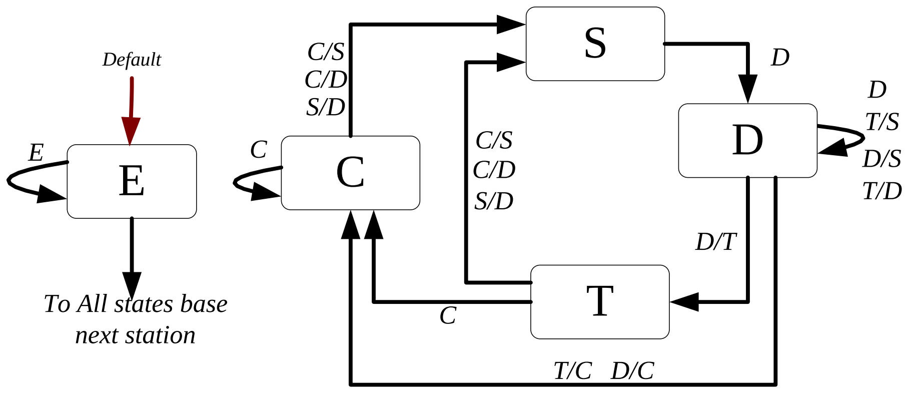
- 256b/257b(512b/514b) and 256b/258b (512b/516b) are some proper PCS coding block sizes
- If a base-line FEC is mandatory and standardized, it's better to apply direct coding rather than transcoding for efficiency
  - Refer to *gustlin_400_02_0713.pdf*

An Example of 256b/257b Direct Coding



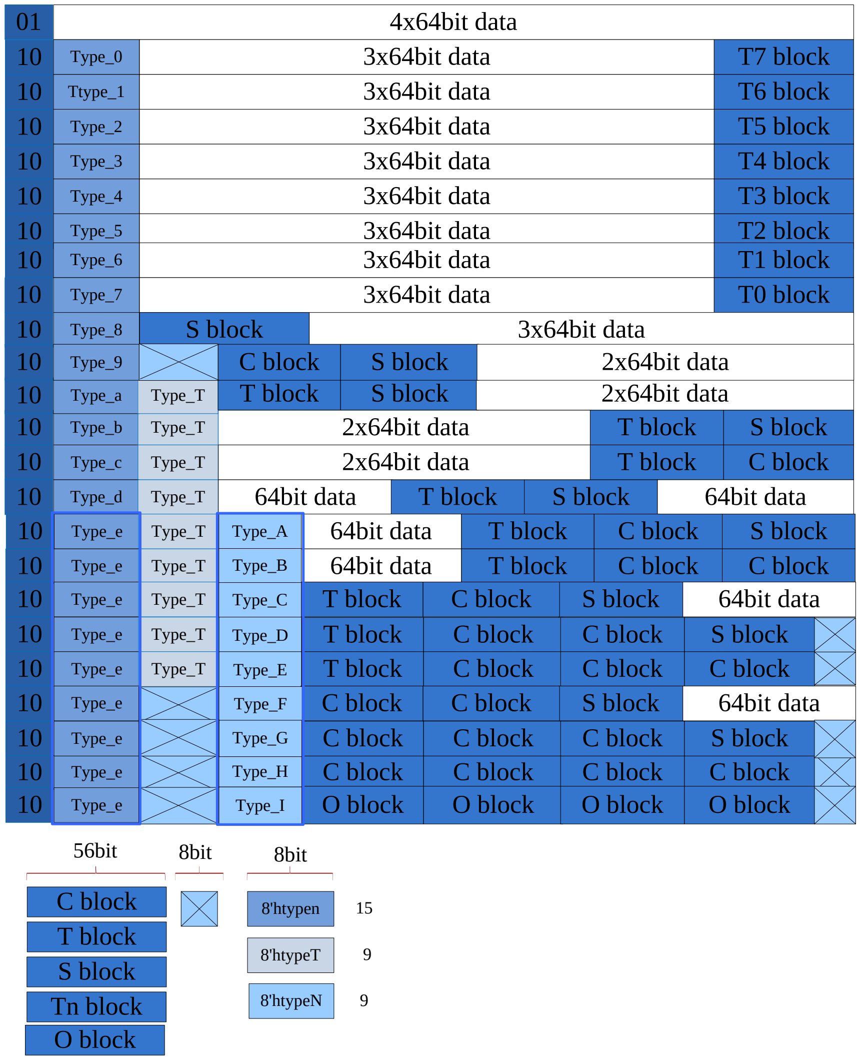
0	4x64bit data							
1	Type_0			3x64bit data			T7 block	
1	Ttype_1			3x64bit data			T6 block	
1	Type_2			3x64bit data			T5 block	
1	Type_3			3x64bit data			T4 block	
1	Type_4			3x64bit data			T3 block	
1	Type_5			3x64bit data			T2 block	
1	Type_6			3x64bit data			T1 block	
1	Type_7			3x64bit data			T0 block	
1	Type_8	S block		3x64bit data				
1	Type_9	☒		C block	S block	2x64bit data		
1	Type_a	Type_T		T block	S block	2x64bit data		
1	Type_b	Type_T		2x64bit data		T block	S block	
1	Type_c	Type_T		2x64bit data		T block	C block	
1	Type_d	Type_T		64bit data	T block	S block	64bit data	
1	Type_e	Type_T	Type_A	64bit data	T block	C block	S block	
1	Type_e	Type_T	Type_B	64bit data	T block	C block	C block	
1	Type_e	Type_T	Type_C	T block	C block	S block	64bit data	
1	Type_e	Type_T	Type_D	T block	C block	C block	S block	☒
1	Type_e	Type_T	Type_E	T block	C block	C block	C block	☒
1	Type_e	☒	Type_F	C block	C block	S block	64bit data	
1	Type_e	☒	Type_G	C block	C block	C block	S block	☒
1	Type_e	☒	Type_H	C block	C block	C block	C block	☒
1	Type_e	☒	Type_I	O block	O block	O block	O block	☒

56bit	8bit	8bit	
C block	☒	8'htypen	15
T block		8'htypeT	9
S block		8'htypeN	9
Tn block			
O block			



- **256b/257b direct coding is possible and straightforward**
- **256b/258b direct coding can simply extend the block header bit to two bits**

256b/257b and 256b/258b Comparison



01	4x64bit data							
10	Type_0	3x64bit data	T7 block					
10	Ttype_1	3x64bit data	T6 block					
10	Type_2	3x64bit data	T5 block					
10	Type_3	3x64bit data	T4 block					
10	Type_4	3x64bit data	T3 block					
10	Type_5	3x64bit data	T2 block					
10	Type_6	3x64bit data	T1 block					
10	Type_7	3x64bit data	T0 block					
10	Type_8	S block	3x64bit data					
10	Type_9	☒	C block	S block	2x64bit data			
10	Type_a	Type_T	T block	S block	2x64bit data			
10	Type_b	Type_T	2x64bit data	T block	S block			
10	Type_c	Type_T	2x64bit data	T block	C block			
10	Type_d	Type_T	64bit data	T block	S block	64bit data		
10	Type_e	Type_T	Type_A	64bit data	T block	C block	S block	
10	Type_e	Type_T	Type_B	64bit data	T block	C block	C block	
10	Type_e	Type_T	Type_C	T block	C block	S block	64bit data	
10	Type_e	Type_T	Type_D	T block	C block	C block	S block	☒
10	Type_e	Type_T	Type_E	T block	C block	C block	C block	☒
10	Type_e	☒	Type_F	C block	C block	S block	64bit data	
10	Type_e	☒	Type_G	C block	C block	C block	S block	☒
10	Type_e	☒	Type_H	C block	C block	C block	C block	☒
10	Type_e	☒	Type_I	O block	O block	O block	O block	☒

56bit	8bit	8bit	
C block	☒	8'htypen	15
T block		8'htypeT	9
S block		8'htypeN	9
Tn block			
O block			

- 256b/257b direct coding has good MTTFPA when FEC is used
- 256b/257b encoding has better efficiency
- 256b/258b direct coding has better MTTFPA than 256b/257b when FEC is bypassed
- 256b/257b and 256b/258b direct encoding can almost reuse the same implementation
- We can choose one direct coding scheme based on the FEC bypass status

Alternative Architecture and Issues

- **Moving FEC out of PCS is another possible architecture**
  - Then PCS encoding is better to remain 64B/66B
  - Almost certainly need transcoding for FEC
  - This will complicate the PHY system design

- **PCS is usually integrated with MAC in a same ASIC**
- **A complex PCS with a simple PMD is preferred more than a simple PCS and a complex PMD.**
- **If a base-line and bypass-able FEC can solve the most of the problem, why don't directly embedded it in MAC/PCS ASIC?**
- **Additional FEC can still be implemented out of PCS to support various PMDs if needed**

Summary

- **400GbE should define a unified logic architecture suitable for most PMDs.**
- **FEC may be mandatory for Backplane/Cable, MMF, and some SMF solutions.**
- **It is desired to maintain the same PCS coding scheme for architectures with PCS FEC enabled or disabled for efficiency**
- **256b/257b or 256b/258b direct coding is feasible and a good choice for 400GE PCS**
- **Future work**
  - Explore other coding schemes for 400GbE
  - Analyze direct coding logic resource, performance, and latency
  - Detail the 256b/257b direct coding scheme
  - Analyze MTTFPA for FEC with direct coding
  - Study the unified 400GbE PHY architecture