400GbE PCS Direct Coding Analysis

Haoyu Song, Wenbin Yang, Tongtong Wang

www.huawei.com

HUAWEI TECHNOLOGIES CO., LTD.

Introduction

- **In September York Interim, we presented "Reconsider PCS Coding for 400GbE"** [1]
  - Review the history of Ethernet PCS/line coding schemes
  - 400GbE raises new requirements for PCS coding schemes
  - For a PCS architecture with embedded RS-FEC, direct coding (DC) is possible and proper
  - We show it is feasible to design simple and efficient DC schemes
- **In this contribution, we will cover**
  - DC extension scheme that improves MTTFPA when the baseline RS-FEC is disabled
  - Comparisons between the direct coding (DC) and the transcoding (TC) schemes from the implementation perspective
    - Encoder itself
    - Effect in System

[1] http://www.ieee802.org/3/400GSG/public/13_09/song_400_01_0913.pdf

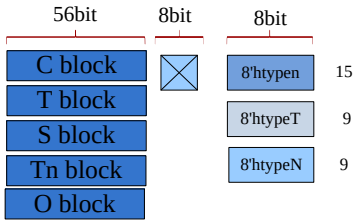
Recap the Rationality of Direct Coding

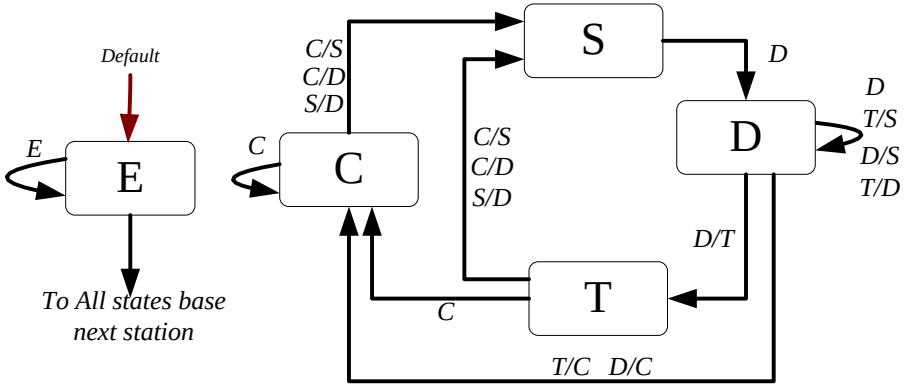
- **FEC is likely to be an integral part of PCS to protect the line, chip-to-chip, and chip-to-module interface** [1]
  - Higher gain FEC, if needed, can be added between PCS and PMD
- **RS-FEC can correct both burst errors and random errors, so it is likely to be chosen as the baseline FEC**
- **Study shows RS(544/528,514/516/520) are good RS-FEC algorithm candidates** [2]
  - RS(544/528,514) is adopted in 802.3bj which is good for design reuse [3]
- **RS-FEC requires better coding efficiency than 64b/66b to accommodate FEC checksum**
  - 802.3bj 256b/257b transcoding from 64b/66b is an afterthought
- **Is direct coding actually better and more straightforward?**

[1] http://www.ieee802.org/3/400GSG/public/13_07/gustlin_400_02_0713.pdf
[2] http://www.ieee802.org/3/bj/public/jan12/gustlin_01_0112.pdf
[3] http://www.ieee802.org/3/bm/public/sep12/gustlin_01_0912_optx.pdf

256b/257b Direct Coding (DC) Scheme

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



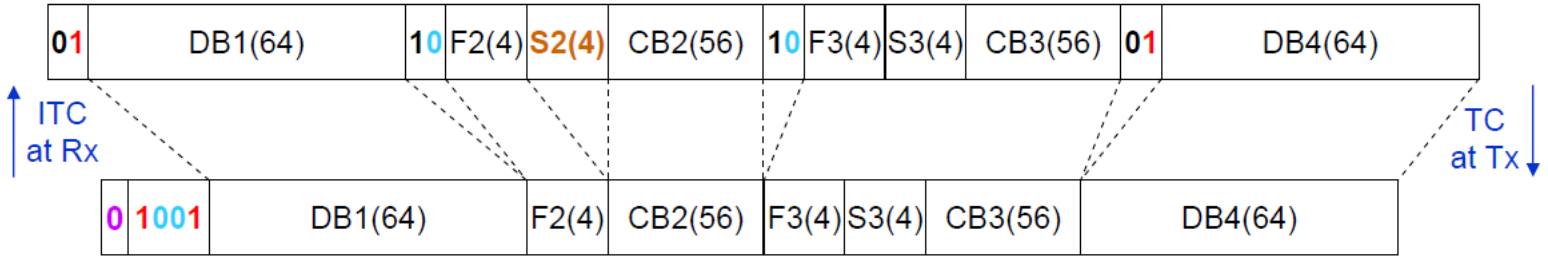


- **RS(544/528, 514) requires 256b/257b DC. We show a possible coding scheme and the coding FSM**
- **256b/260b DC can be realized by simply extending the block header bit to four bits to keep minimum 4bits Hamming distance**

256b/260b DC Extends Hamming Distance

- **The Hamming distance of 256b/257b DC code is only one**
  - If FEC is enabled, the MTTFPA derived from UCR shows acceptable performance just as 802.3bj
  - If FEC is disabled, this lead to poor MTTFPA (~10^3 years at a 10^-12 BER)
- **When FEC is disabled, the saved FEC checksum overhead allows 8 extra bits per 256b data block**
  - Use 4 bits as the block header (i.e. 256b/260b) to achieve the same 4-bit code Hamming distance as 64b/66b
    - e.g. data block sync header = 1010, and data/control block sync header = 0101
  - The remaining 4 bits per 256b block can be used in other ways (e.g. some lightweight checksum)
- **Depending on the FEC configuration, we can switch between 256b/257b DC and 256b/260b DC with small cost**
  - The PCS encoding process is exactly the same

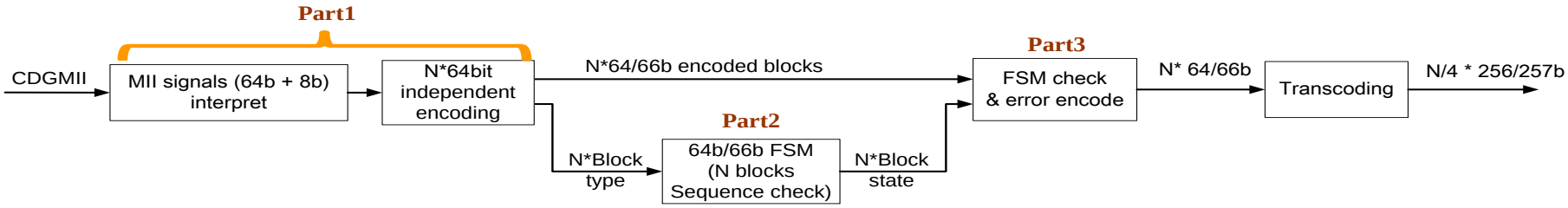
64b/66b+TC Hamming Distance



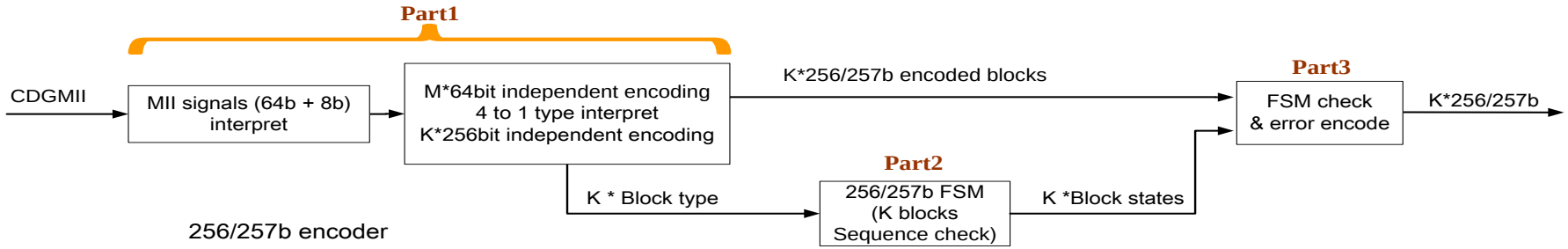
- **64b/66b to 256b/257b Transcoding process compresses the 2-bit 64b/66b sync header to 1-bit**
- **The TC code Hamming distance is just one**
  - When 256b/257b TC code's header bit "0" becomes "1", this error may cause MTTFPA problem
  - Any error in the 4-bit bitmap may also be undetected to cause MTTFPA problem

64b/66b+TC & 256b/257(260)b DC Implementation

- **Block diagrams in FPGA-based implementation**



64/66b encoder + 256/257b transcoder



256/257b encoder

* Parameters ***N, M,*** & ***K*** vary in different implementations, e.g. ***N=20*** when working frequency is 312.5MHz;

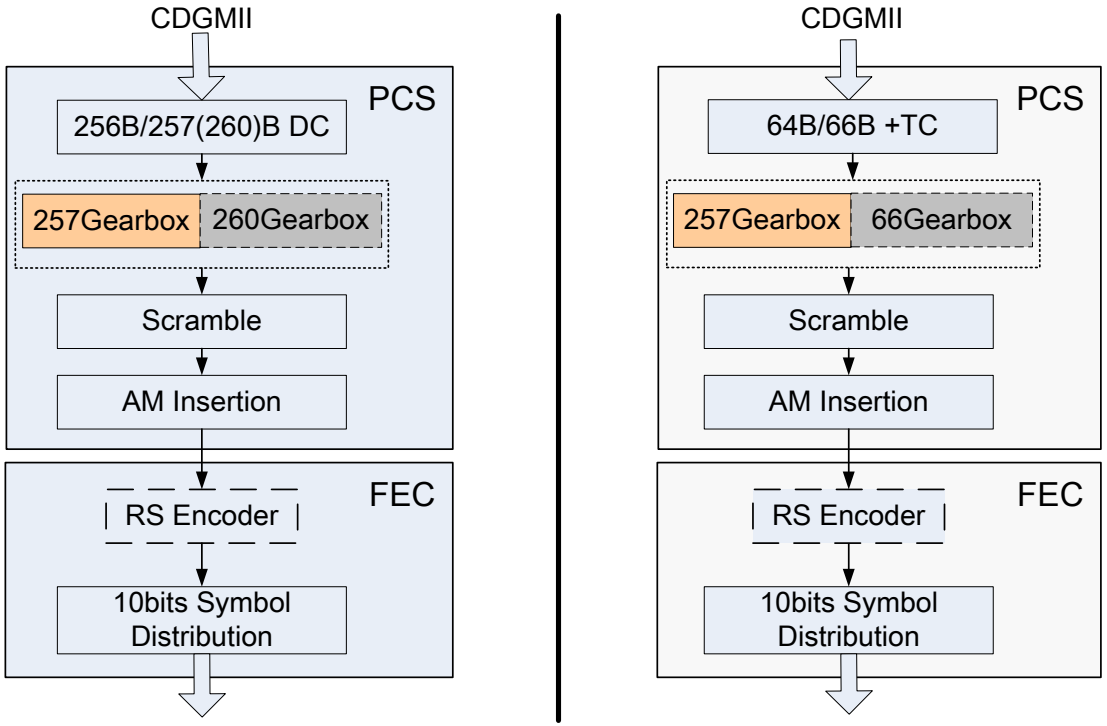
• Our comparison is based on the same data width and working frequency (i.e. 312.5MHz) → ***N=M=20, K=M/4=5;***

Encoding Cost Comparison

	64b/66b+TC			256b/257(260)b DC			Note
	LUT	Register	Latency (cycles)	LUT	Register	Latency (cycles)	
Part1 (MII interface interpret & block encode)	5328	11295	9	7128	9095	6	256/257b uses more LUTs due to more possible data offsets in encoding
Part2 (Encoder FSM)	253	264	0	~100	~100	0	256/257b has a smaller FSM cost due to less iterations in processing
Part3 (Valid check and Error encode)	1319	1320	1	1319	1320	2	Common process for both methods
Transcoding	216*5	1280	1	N/A	N/A	N/A	
Total	7980	14159	11	8547	10515	8	

- DC consumes 26% register resource and only 7% more LUTs compared with TC
- DC coding latency is 25% better than TC

Architecture Comparison – PCS Gearbox



- **257Gearbox in yellow is used when FEC is enabled**
- **Different Gearboxes are used when FEC is disabled**
  - Logic consumption ratio between 260Gearbox and 66Gearbox is about 0.8k:2.08k(Luts) = 1:2.6
- **If RS(544, 520) is adopted, only 260Grearbox is needed**

256b/257b DC vs. 64b/66b+TC: Performance

	64b/66b+TC	256/257b DC
Good Code Hamming Distance w/o FEC	No	Yes, by 256b/260b DC extension
Good MTTFPA w/ FEC	Yes	Yes
Support FEC Bypass	Yes	Yes
Leverage redundant bits when FEC is bypassed	No	4bits per 256b could be used for standard or proprietary extension
Gearbox cost	Support both 256/257 and 64/66 gearbox with higher cost	Support both 256/257 and 256/260 gearbox with lower cost

- **When FEC is disabled, DC can still have the same code Hamming distance as the 64b/66b scheme, so MTTFPA is also the same.**
- **256/257 DC is a straightforward method for 400GE, while legacy TC method is less efficient and less flexible**

Summary

- 400GbE should define an unified logic architecture suitable for most PMDs
- It is desirable to maintain the same PCS coding scheme for architectures with the baseline PCS FEC enabled or disabled
- 256b/257(260)b DC has lower implementation cost and better latency performance than the equivalent TC
- 256b/257(260)b DC is a qualified candidate for unified 400GbE PCS coding scheme and can replace 64b/66b without any conceivable drawback