Investigation on Technical Feasibility of Stronger RS FEC for 400GbE

Mark Gustlin-Xilinx, Xinyuan Wang, Tongtong Wang-Huawei,

Martin Langhammer-Altera, Gary Nicholl-Cisco,

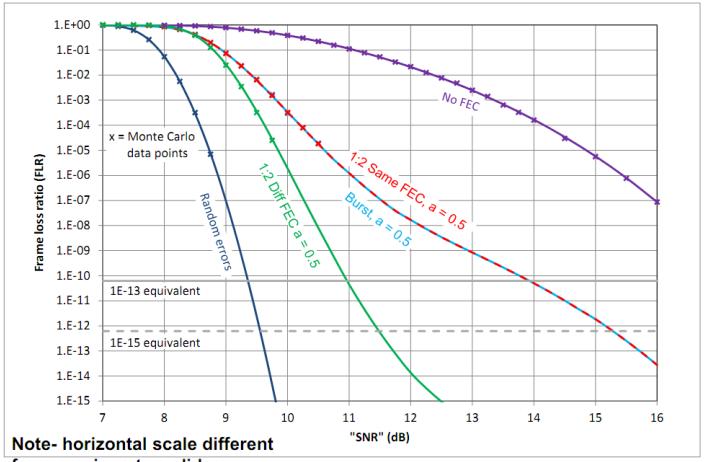
Dave Ofelt-Juniper, Bill Wilkie-Xilinx, Jeff Slavick-Avago,

Zhongfeng Wang-Broadcom

Introduction and Background

- This presentation investigates the technical feasibility of stronger RS FEC options
- BCH FEC options have different FEC performance for random and burst errors, with poor burst performance
- BCH FEC implementations require greater (when compared to RS FECs) logic resources even without KES duplication

BCH(2858,2570) all curves



Note- horizontal scale different from previous two slides

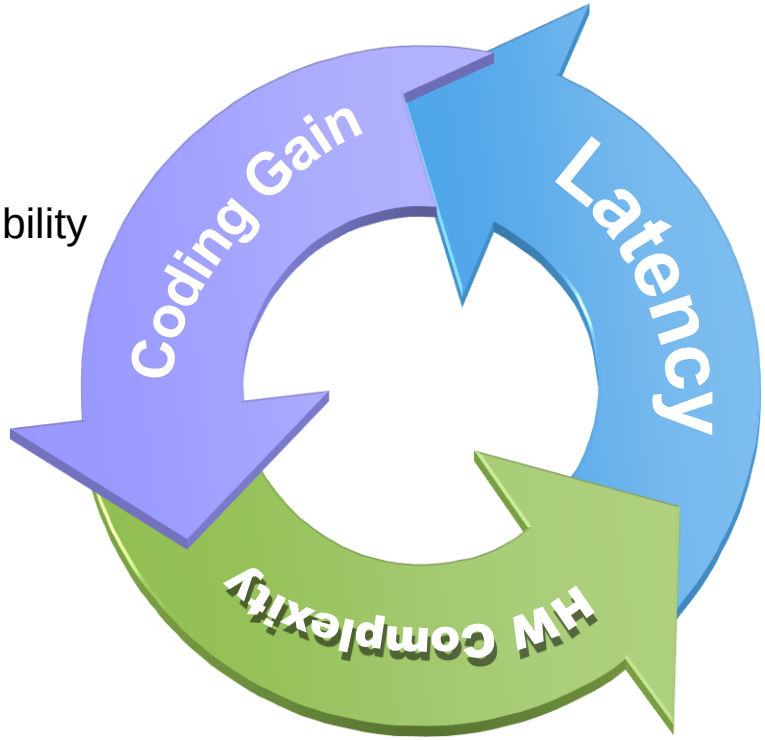
Type	Codeword	Area (6LUT)	Relative Area
RS KR4	(528,514,7)	10654	1
RS KP4	(544,514,15)	26554	2.5
BCH[1]	(2858,2570,24)	106806	10
BCH[2]	(9193,8192,71)	425000	40

anslow_3bs_02_1114

langhammer_3bs_01_1114

400GbE Stronger FEC Tradeoff

- Overhead vs. SerDes rate & technology feasibility



- Latency in sensitive applications, such as Finance, DC,……. Especially for short reach solutions,100/500m

- Higher HW complexity will lead to higher power and difficulty in integrating into a host ASIC or FPGA
- Higher complexity/power can impact optical modules if the FEC is integrated into the module

History of Ethernet Latency

- In existing low latency Ethernet switches, you see latencies as low as 250-350ns (for 10GE). These switches use cut-through switching, and this is the total latency including switching time
- High frequency trading (HFT) in financial applications, high performance computing(HPC) in DC are especially sensitive to latency
- Latency in DC is incurred by upper layer protocol (TCP windows, flow control, etc) and much cost on server implementation, especially memory
- Our proposed FEC latency 400GbE target is <250ns. It was100ns for 802.3bj KR4/KP4 FEC

Coding Gain Calculation of RS(n,k,t,m) FEC

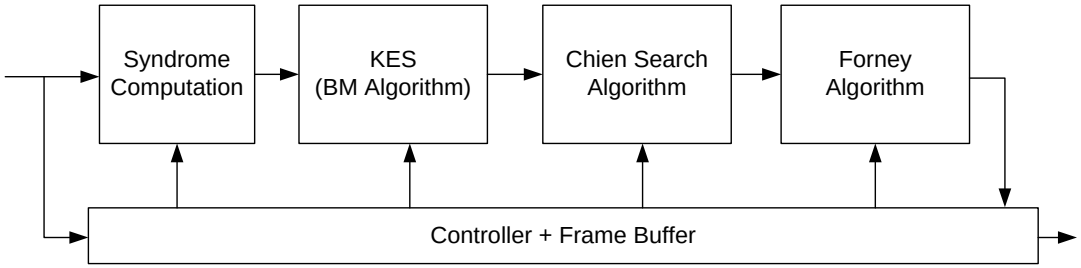
- CG/NCG is based on 802.3bs BER Objective: 1E-13
- Assuming white Gaussian noise random error only for easy analyst in this slides. Burst error just some additional penalty to CG/NCG
- Coding Gain is the reduction of raw BERin to a required BERpost value within the information signal
- Net Coding Gain is corrected to CG by the increased noise due to bandwidth expansion needed for FEC bits
- Code rate R is the ratio of bit rate without FEC to bit rate with FEC
- Transcoding to lower over-clock and improve Net Coding Gain

$$Coding\ Gain = 20\log_{10}[erfc^{-1}(2 * BERpost)] - 20\log_{10}[erfc^{-1}(2 * BERin)]$$

$$Net\ Coding\ Gain = 20\log_{10}[erfc^{-1}(2 * BERpost)] - 20\log_{10}[erfc^{-1}(2 * BERin)] + 10\log_{10} R$$

Latency Estimation of RS(n,k,t,m) FEC

- Use 100Gbps KR4 FEC@644MHz for ASIC as baseline in this slides
- Latency estimation based on (RS FEC correction ability) t and parallelism(p1/p2) on each sub blocks in the following diagram;
- FEC Decoder performs error detection with error correction, same as in CL91.5.3.3, aka Mode A in 802.3bj



$t_{syndrome}$ = *n/p1, p1=16 for KR4/KP4 FEC implementation in this slides*

t_{KES} = *x2t, (if* $t_{KES} > t_{syndrome}$ *, duplicate KES in this slides)*

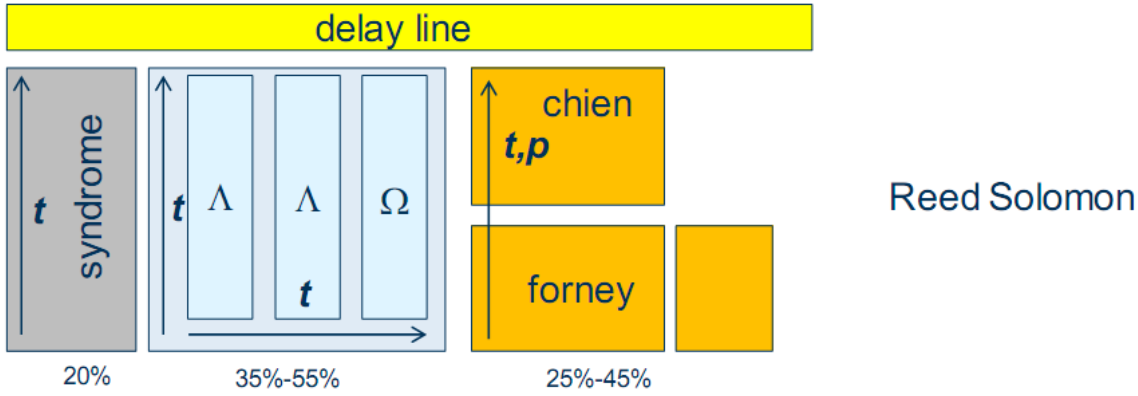
x=1 for t<=15, x=2 for t>15; For longer RS FEC, level of pipelining in the iterative calculation may increase due to longer critical path

$t_{chien} + t_{forney}$ = *n/p2+1, p2=66/68 for KR4/KP4 FEC implementation in this slides, p2≥p1*

FEC Decode Latency = ~($t_{syndrome} + t_{KES} + t_{chien} + t_{forney}$)

Area Estimation of RS(n,k,t,m) FEC

- For area estimation refer to langhammer_3bs_01_1114



- With modification for low latency target and larger area permitted, KR4 FEC ASIC area ratio is:

  Syndrome: KES: (Chien+Forney)=20%:40%:40%

- if $t_{KES} > t_{syndrome}$, duplicate KES block to match the throughput of syndrome. This will increase area cost significantly for longer block RS FEC

Summary of RS FEC Options Considered

RS FEC(n,k,t,m)	CG	NCG*	BERin	Overhead	SerDes Rate	Block Time	Latency**	Area Ratio
Group 1 : Similar RS FEC as KR4 FEC								
RS(528,514,7,10)	5.39	5.28	3.92E-05	0%	25.78125	51.2ns	~87ns	1X
RS(544,514,15,10)	6.64	6.39	3.09E-04	3.03%	26.5625	51.2ns	~112ns	2.9X
RS(560,514,23,10)	7.3	6.93	7.60E-04	6.06%	27.34375	51.2ns	~208ns	14.5X
RS(576,514,31,10)	7.76	7.26	1.30E-03	9.09%	28.125	51.2ns	~258ns	33.4X
Group 2 : Large Block RS FEC								
RS(1056,1028,14,11)	6.07	5.95	1.29E-04	0%	25.78125	102.4ns	~172ns	2.6X
RS(1088,1028,30,11)	7.12	6.88	6.06E-04	3.03%	26.5625	102.4ns	~315ns	16.7X
RS(1120,1028,46,11)	7.7	7.33	1.20E-03	6.06%	27.34375	102.4ns	~414ns	54.8X
RS(1152,1028,62,11)	8.11	7.61	1.90E-03	9.09%	28.125	102.4ns	~514ns	129.5X
Group 3 : RS(255,239) Like RS FEC								
RS(255,239,8,8)	6.12	5.83	1.39E-04	6.7%	27.5	18.9ns	~49ns	1.1X
RS(510,478,16,9)	6.85	6.57	4.21E-04	6.7%	27.5	42.5ns	~162ns	5.3X
RS(1020,956,32,10)	7.34	7.06	7.95E-04	6.7%	27.5	93.1ns	~304ns	27.2X
Group 4 : 256/257b coding friendly RS FEC***								
RS(800,771,14,10)	6.29	6.13	1.83E-04	1.01%	26.04	76.8ns	~140ns	2.6X
RS(816,771,22,10)	6.95	6.71	4.84E-04	3.03%	26.5625	76.8ns	~232ns	9.4X
RS(840,771,34,10)	7.58	7.22	1.10E-03	6.06%	27.34375	76.8ns	~306ns	30.6X
RS(864,771,46,10)	8.02	7.53	1.80E-03	9.09%	28.125	76.8ns	~379ns	72.1X

- The latency and area ratio is based on current RS FEC in ASIC and possible to decrease by optimized FEC algorithm or implementation

* : NCG doesn't include gain from 256/257 Transcoding at 0.12dB
**: Added latency for FEC only
***: Needs dummy bits to support FEC lane distribution

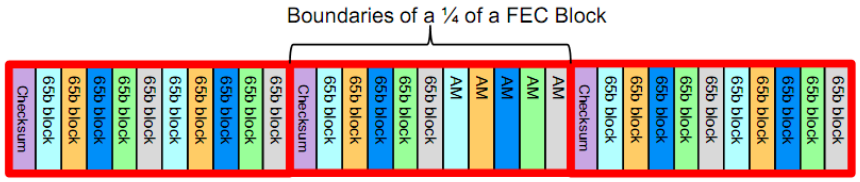
Generic Rules for RS(n,k,t,m) FEC in Logic Layer with *i* FEC Lanes

- Rule 1: Prefer to keep 16384*66bit*20 AM spacing

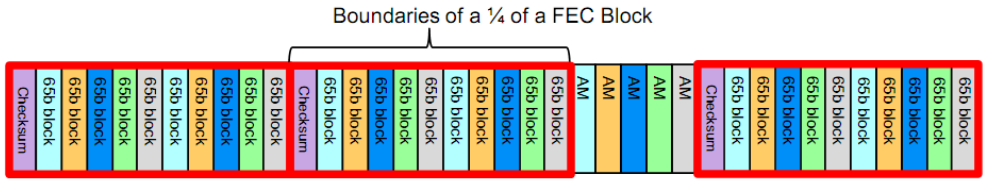
FEC Across Multiple Lanes

At least two implementations are possible:

1. Alignment Markers are included in the FEC blocks



2. Alignment Markers are not included in the FEC blocks



For either case, the Alignment Markers must repetitively be in the same location relative to FEC block starts:

1. (AM Spacing * # PCS Lanes * block size) must be divisible by (FEC block size)
2. ((AM Spacing-1) * # PCS Lanes * block size) must be divisible by (FEC block size)

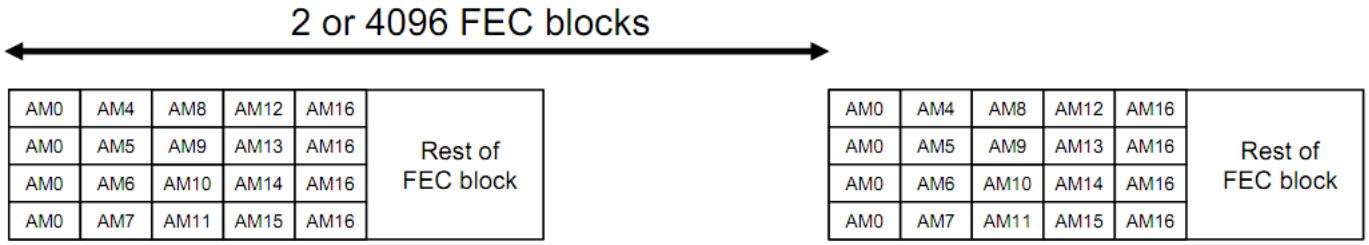
gustlin_02a_0511

Generic Rules for RS(n,k,t,m) FEC in Logic Layer with *i* FEC Lanes (cont'd)

- Rule 2: Alignment marker is uniquely identify for each FEC lanes and friendly Idle delete(64bit) for IPG adjustment. Generally AM length should at least LCM(Least Common Multiple) of "m, i and 64”
- Rule 3: FEC information block: k*m should be divisible by encoder length if no dummy bit added, e.g. 257bit of 256/257 TC/DC, 65bit of 64/65 TC or 513bit of 512/513 TC
- Rule 4: FEC block: n*m should be divisible by i*m. for example, i=4 in KR4/KP4 FEC
- Rule 5: Feasible RCM(integer Reference Clock Multiplier) with 156.25MHz. For example, KP4 FEC with 3% over-clocking, RCM=170 for 26.5625Gbps

RS(576/560/544/528,514,31/23/15/7,10)

- 4096 FEC blocks in AM period with 0%/3.03%/6.06%/9. 09% over-clocking;

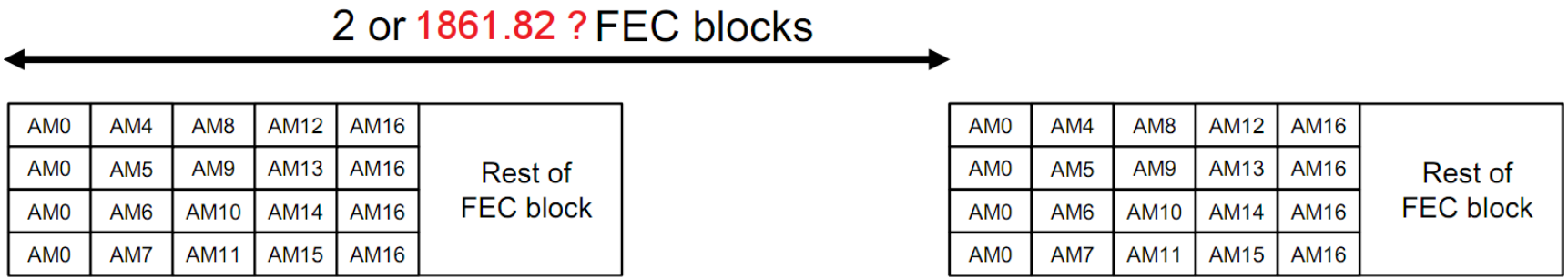


gustlin_400_02a_1113

- AM=320bit;
- FEC Information Block=5140bit=257*20 with 256/257 TC/DC;
- FEC Block=(576/566/544/528)*10=(144/140/136/132)*4*10;
- RCM=180/175/170/165@156.25MHz.

RS(1152/1120/1088/1056,1028,62/46/30/14,11)

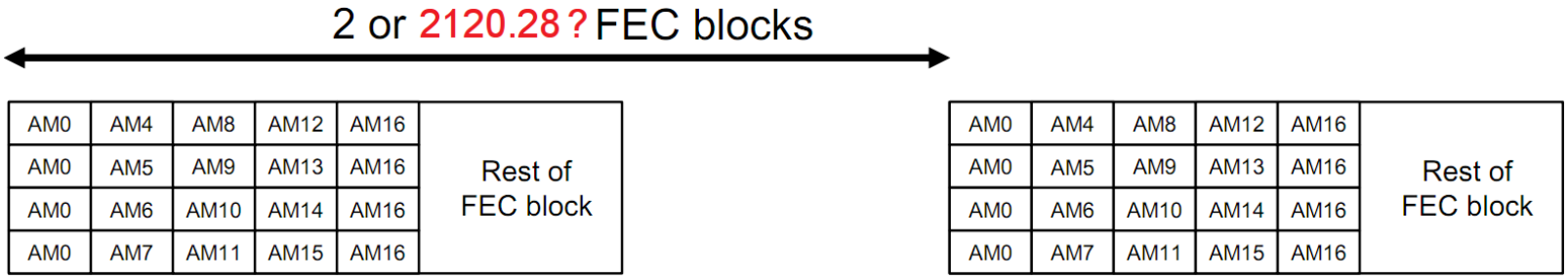
- Not an integer number of FEC blocks in AM spacing!
  - Change AM distance? Or
  - Overlap 1st FEC Block with part of AM area? Not a good option for coupling AM with FEC blocks.



- AM=319bit with 1 dummy bit;
- FEC Information Block=1028*11bit=257*44 with 256/257 TC/DC;
- FEC Block=(1152/1120/1088/1056)*11=(288/280/272/264)*4*11;
- RCM=180/175/170/165@156.25MHz.

RS(1020,956,32,10)

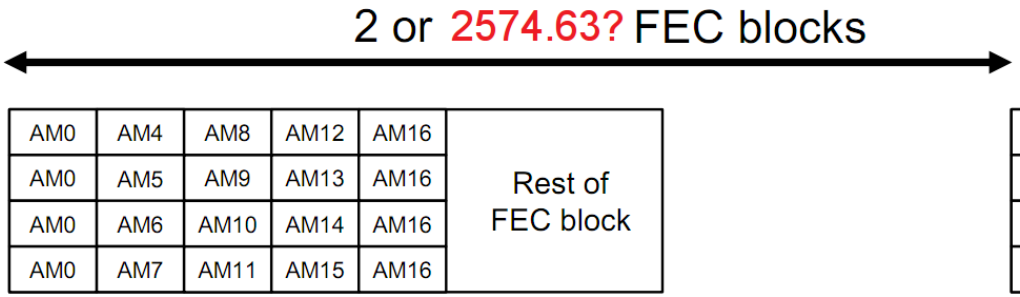
- Not an integer number of FEC blocks in AM spacing!



- AM=320bit;
- FEC Information Block=9560bit, not an integer number of 65,66,257,513bit; Change to 9570bit for adapting to 66bit block;
- FEC Block=(1020)*10=255*4*10;
- RCM, Not an integer number @156.25MHz.

RS(840,771,34,10)

- Extend FEC block to 840*m for easy implementation with 10bit dummy bit;
- Not an integer number of FEC blocks in AM spacing!



- AM=320bit;
- FEC Information Block=771*10bit=257*30 with 256/257 TC/DC;
- FEC Block=(840)*10=(210)*4*10;
- RCM=175@156.25MHz. Same over-clock as RS(560,514,23,10).

Compare of Possible Stronger RS FEC for 400GbE

- We can pick some candidate stronger RS FECs with latency < ~250ns and Area < ~30X KR4 FEC.

RS FEC(n,k,t,m)	CG	NCG	BERin	Overhead	SerDes Rate	Block Time	Latency	Area Ratio	Hardware complexity
RS(528,514,7,10)	5.39	5.28	3.92E-05	0%	25.78125	51.2ns	~87ns	1X	802.3bj
RS(544,514,15,10)	6.64	6.39	3.09E-04	3.03%	26.5625	51.2ns	~112ns	2.9X	802.3bj
RS(560,514,23,10)	7.3	6.93	7.60E-04	6.06%	27.34375	51.2ns	~208ns	14.5X	Implementation compatible with 802.3bj; costs more logic resource
RS(576,514,31,10)	7.76	7.26	1.30E-03	9.09%	28.125	51.2ns	~258ns	33.4X	Implementation compatible with 802.3bj;costs significant logic resource
RS(1088,1028,30,11)	7.12	6.88	6.06E-04	3.03%	26.5625	102.4ns	~315ns	16.7X	costs more logic resource and requires to change AM spacing of 16384; Rule 1 not satisfied
RS(1020,956,32,10)	7.34	7.06	7.95E-04	6.7%	27.5	93.1ns	~304ns	27.2X	cost too more logic resource and require to change AM spacing of 16384; Rule 1,2,5 not satisfied
RS(840,771,34,10)	7.58	7.22	1.10E-03	6.06%	27.34375	76.8ns	~306ns	30.6X	cost too more logic resource and require to change AM spacing of 16384; Rule 1 not satisfied

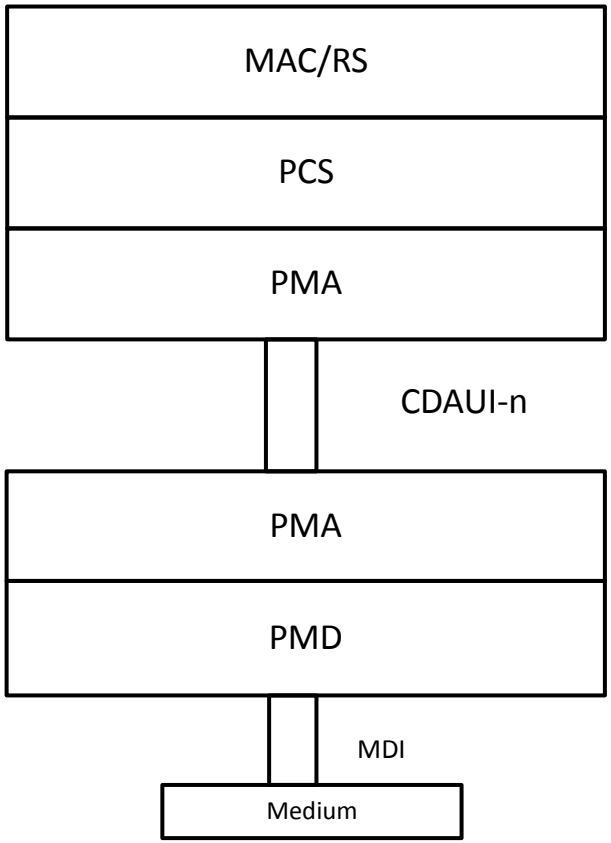
Comparison of 4X100G/1X400Gbps RS(528,514) FEC in 400GbE Logic Layer

RS(528,514,7,10)(100Gbps) 160bit@644MHz(ASIC)	Area	Latency (Cycle)
1. Syndrome(16 parallel)	0.2a	33
2. KES(BM)	0.4a	14
3. Chien(66 parallel)	0.15a	8
4. Forney	0.25a	1
TOTAL	a	56 Cycle(~87ns)

RS(528,514,7,10)(400Gbps) 660bit@624MHz(ASIC)	Area	Latency (Cycle)
1. Syndrome(66 parallel)	0.825a	8
2. KES(BM) (X2 duplication)	0.8a	14
3. Chien(66 parallel)	0.15a	8
4. Forney	0.25a	1
TOTAL	2.025a	31Cycle(~49ns)

- Exact comparison is affected by process node or combinational logic, etc.
- To meet our low latency criteria, 1x400G RS FEC@~49ns is around 2x size of 1x100G RS FEC@~87ns
- For real implementation of high parallelism in 400G RS FEC, the reasonable area of 1x400G RS FEC is larger than 2.5x size of 1x100G RS FEC

Proposal of 400GbE Logic Layer with RS FEC



RS FEC in the PCS to provide a single FEC in the system

gustlin_3bs_02a_1114

Summary

- RS FEC seems like a good fit for this project: less complex to implement and better gain in the face of burst errors when compared to a BCH code
- There are several good RS FEC candidates in this presentation, we need to make the right tradeoff between latency, complexity and gain for the PMDs in order to select the best FEC code
- Further work on gain/latency/area of stronger RS FEC by deeper analysis of FEC model and algorithm. Provide RS FEC candidates for PMD discussion

Thank you