IEEE 802.3 Interim Meeting, September 2018, Spokane, WA

Preliminary Studies on DFE Error Propagation, Precoding, and their Impact on KP4 FEC Performance for PAM4 Signaling Systems

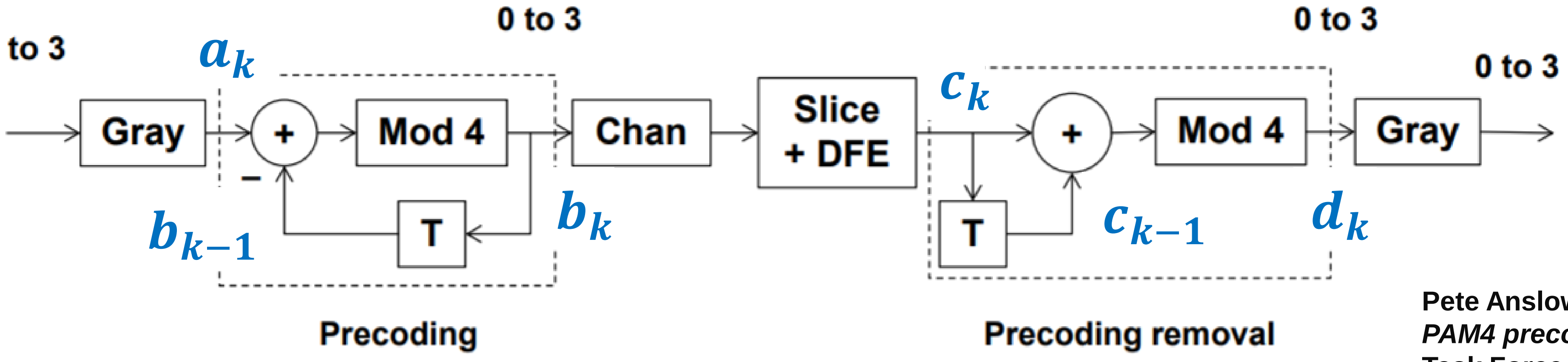
XILINX

Geoff Zhang
September, 2018

Outline

- 1/(1+D) precoding for PAM4 link systems
  - 1/(1+D) precoding implementations and working mechanism
  - Burst error removal conditions for PAM4 and examples
- DFE error propagation models
  - Error propagation modelling for a 1-tap and for *N*-tap DFE
  - Average burst error length discussions
  - Burst error length definition discussions
- Error propagation and precoding impact simulations
  - Link simulation setup and conditions
  - Simulation examples and discussions on precoding and FEC
    - DFE-less receiver channels
    - Various constructed DFE configurations
    - COM-computed DFE results
    - Silicon measured DFE data
- Summary and conclusions

1/(1+D) precoding for PAM4 links



Pete Anslow, "*FEC performance with PAM4 precoding*", IEEE P802.3bs Task Force, Waikoloa, July 2015

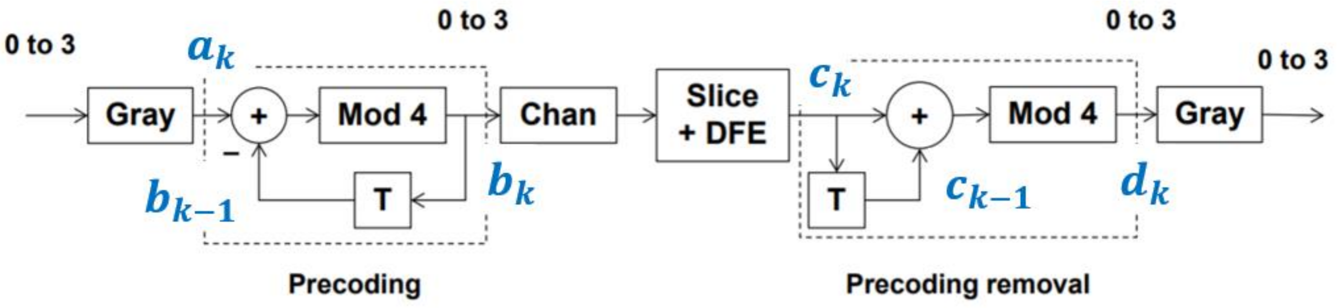
- When there are no symbol errors, $c_k = b_k$ and $c_{k-1} = b_{k-1}$, then,

$$\boldsymbol{b_k = (a_k - b_{k-1}) + 4 \cdot n} \text{ and } \boldsymbol{d_k = (c_k + c_{k-1}) + 4 \cdot m}, \quad \textit{where n, m = 0,1}$$

- Thus, the precoding decoder output, d_k, is equal to the precoding encoder input, a_k

$$\boldsymbol{d_k = (c_k + c_{k-1}) + 4 \cdot m = (b_k + b_{k-1}) + 4 \cdot m}$$
$$\boldsymbol{= (a_k - b_{k-1} + b_{k-1}) + 4 \cdot (n + m) = a_k}$$

1/(1+D) precoding – how errors are affected



- For two continuous symbol errors, $c_k = b_k + \alpha\ (\alpha \neq 0)$, and $c_{k-1} = b_{k-1} + \beta\ (\beta \neq 0)$ (Typically, $\alpha, \beta = \pm 1,$ but with *skip-level errors*, they could be $\pm 1, \pm 2, \pm 3$)

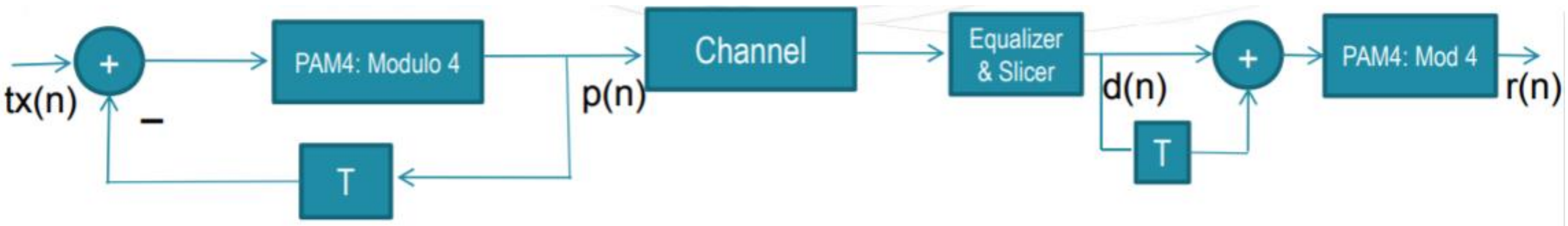
$$d_k = (c_k + c_{k-1}) + 4 \cdot m = \big((b_k + \alpha) + (b_{k-1} + \beta)\big) + 4 \cdot m = a_k + (\alpha + \beta) + 4 \cdot m$$

- Thus, for precoding to work, $\alpha + \beta = 0\ (\mathrm{mod})$ must be satisfied
  - For NRZ, this is always true, since 1+(-1) = 1+1 = (-1)+1 = (-1)+(-1) = 0 (mod)
  - However, for PAM4 this *zero-sum error pattern* does not always hold
- Now, if $c_k = \mathrm{b}_k$, but $c_{k-1} = \mathrm{b}_{k-1} + \alpha\ (\alpha \neq 0)$, i.e., errors is terminated

$$d_k = a_k + \alpha\ (\alpha \neq 0) + 4 \cdot m \neq a_k + 4 \cdot m$$

  - A correct symbol following an incorrect one becomes incorrect, regardless of the error pattern

PAM4 burst error removal example



Sudeep Bhoja, et al, "Precoding proposal for PAM4 modulation", 100 Gb/s Backplane and Cable Task Force IEEE 802 3 IEEE 802.3 Chicago September 2011

- **Precoder Input : tx(n)**
  - 2 2 2 2 0 3 2 0 1 3 3 0 0 0 0 2 3 0 3
- **Precoder Output : p(n)**
  - 0 2 0 2 2 1 1 3 2 1 2 2 2 2 2 0 3 1 2
- **DFE, Slicer Output : d(n)**
  - 0 1 1 1 3 0 2 2 3 0 3 1 3 1 3 0 3 1 2
- **Error Event : p(n) – d(n)**
  - 0 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 0 0 0 0
- **Decoder Output after 1+D at Rx : r(n)**
  - 2 1 2 2 0 3 2 0 1 3 3 0 0 0 0 3 3 0 3

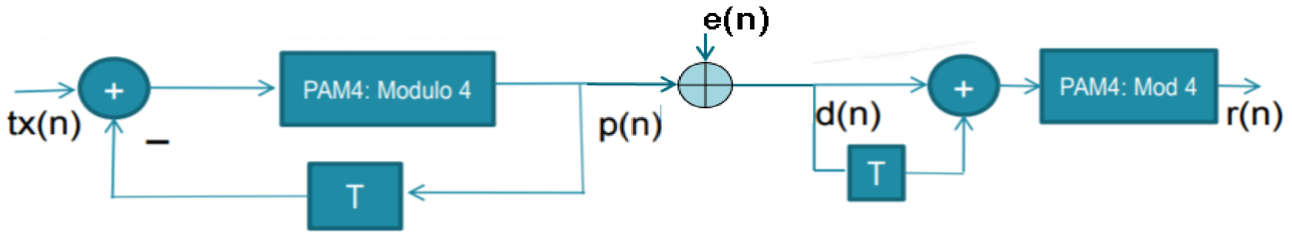
Entry Error　　　　Exit Error

- In this example it is seen that a burst of 14 PAM4 symbol errors are corrected by precoding, except
  - The very first symbol error and the correct symbol error at the end of the burst block
- There are no holes in the burst errors block in this example

XILINX

Burst error removal explained

- Slicer output: $d(n) = p(n) + e(n)$
- Precoding decoder output



$$r(n) = d(n) + d(n-1) + 4 \cdot m = p(n) + e(n) + p(n-1) + e(n-1) + 4 \cdot m$$

$$= tx(n) + e(n) + e(n-1) + 4 \cdot m = tx(n) \quad \text{if } e(n) + e(n-1) = 0 \ (\text{mod } 4)$$

- In the example
  - **Error Event : p(n) – d(n)**
    - 0 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 0 0 0 0
  - **Decoder Output after 1+D at Rx : r(n)**
    - 2 1 2 2 0 3 2 0 1 3 3 0 0 0 0 3 3 0 3

Error   Error

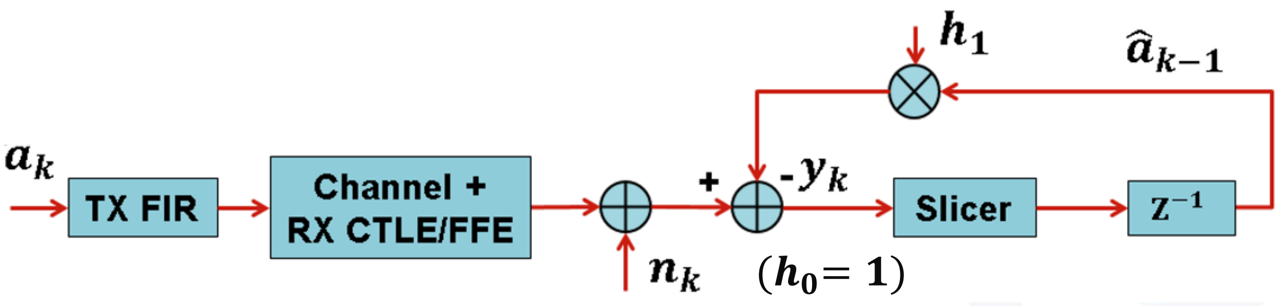
$$e = [0\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1\ 1\ -1\ 0\ 0]$$

XILINX

What if errors assume a different signature

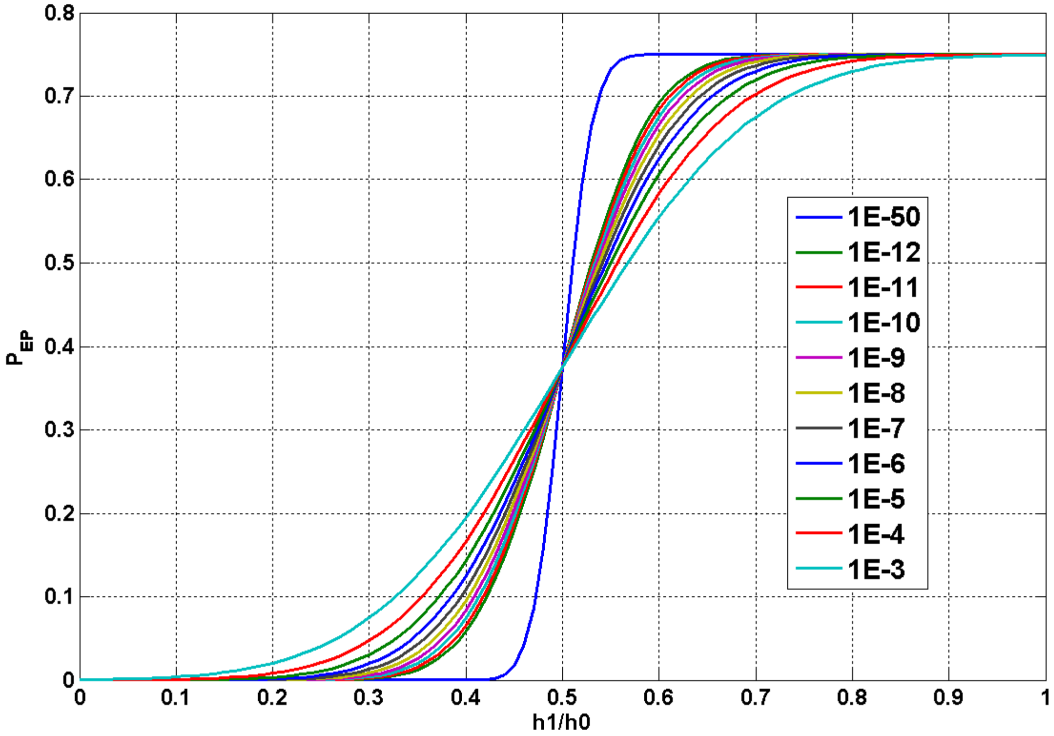
- Still assuming no skip-level errors, but the error pattern is changed slightly
  - **Precoder Input : tx(n)**
    - 2 2 2 2 0 3 2 0 1 3 3 0 0 0 0 2 3 0 3
  - **Precoder Output : p(n)**
    - 0 2 0 2 2 1 1 3 2 1 2 2 2 2 2 0 3 1 2
  - **DFE, Slicer Output : d(n)**
    - 0 1 1 ~~1~~ 3 3 0 2 2 ~~3~~ 1 ~~0~~ 2 3 1 ~~3~~ 1 ~~1~~ 3 ~~3~~ 1 0 3 1 2
  - **Error Event : p(n) – d(n)**
    - 0 1 -1 ~~1~~ -1 -1 1 -1 1 ~~-1~~ 1 ~~1~~ -1 -1 1 ~~-1~~ 1 ~~1~~ -1 ~~-1~~ 1 0 0 0 0
  - **Decoder Output after 1+D at Rx : r(n)**
    - 2 1 2 ~~2~~ 0 ~~0~~ 2 3 2 0 ~~1~~ 3 3 ~~3~~ 1 0 ~~0~~ 2 0 0 ~~3~~ 1 3 0 3
- The burst errors are only conditionally removed, while the block length is not reduced
- The question is, can errors ever behave like this?

Error propagation model for 1-tap DFE



$$y_k = a_k + h_1 \cdot (a_{k-1} - \hat{a}_{k-1}) + n_k + ISI_{res}$$

- P_{EP} (a.k.a. "a"), the error propagation probability for the next symbol if the current symbol is incorrect, can be derived explicitly. This is plotted to the right
- Assuming no skip-level errors due to noise, for a 1-tap DFE (at h1), the error event should satisfy the *zero-sum error pattern*, since
  - if $(a_{k-1} - \hat{a}_{k-1})$ = +1, $(a_k - \hat{a}_k)$ = -1 or 0 (terminated)
  - if $(a_{k-1} - \hat{a}_{k-1})$ = -1, $(a_k - \hat{a}_k)$ = +1 or 0 (terminated)
- However, for a multiple-tap DFE architecture, the *zero-sum error pattern* cannot be guaranteed



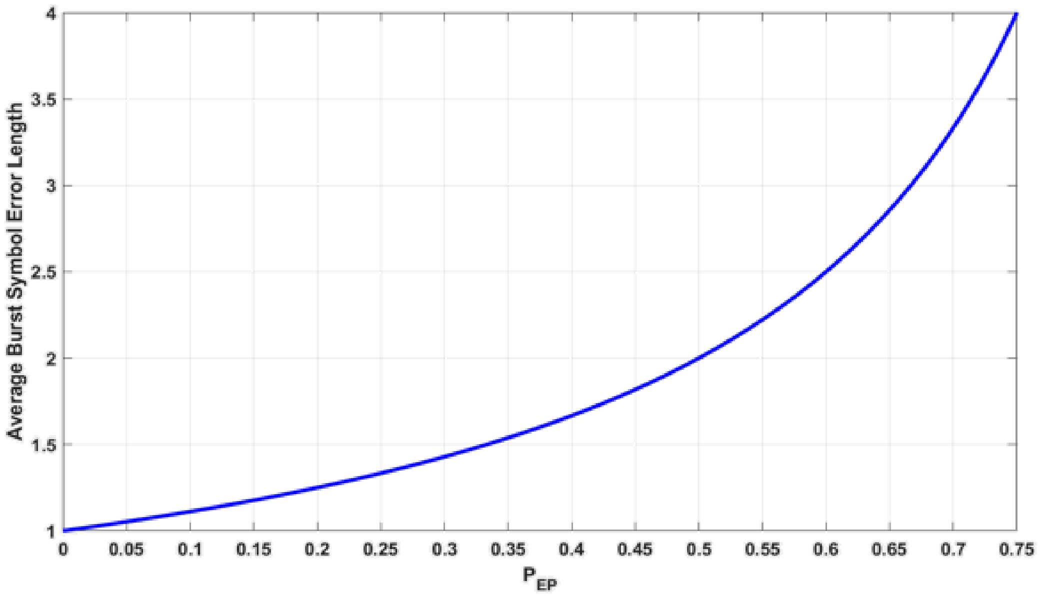
Average length of burst errors

- The average burst symbol error length (K) can be derived as

$$K = (1 - P_{EP}) \cdot \sum_{n=1}^{\infty} n \cdot {P_{EP}}^{n-1} = 1/(1 - P_{EP})$$

  - Thus, for h1/h0=1, P_{EP} = 0.75, K = 1/(1-0.75) = 4

- This equation does not apply with multi-tap DFE
  - P_{EP} is no longer a constant; it becomes a strong function of error signature in the previous N_b (DFE tap number) symbols and the data pattern in that period as well
  - Burst error length definition will be discussed later; the average burst error length does not reflect the true picture of error distribution



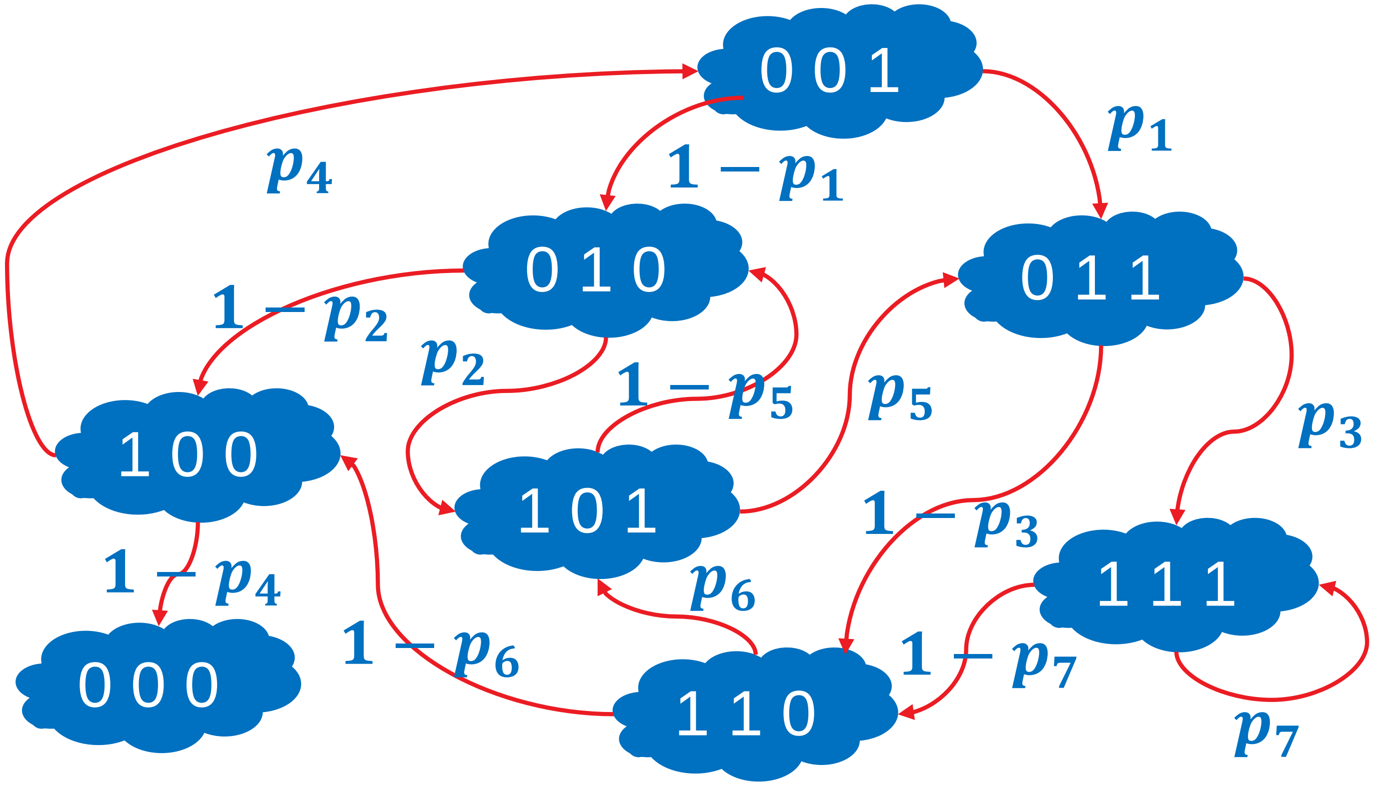
XILINX

Error propagation for an *N*-tap DFE receiver

- For an *N*-tap DFE, the signal at the slicer can be expressed as

$$y_k = a_k + \sum_{m=1}^{N} h_m \cdot (a_{k-m} - \hat{a}_{k-m}) + n_k + ISI_{res}$$

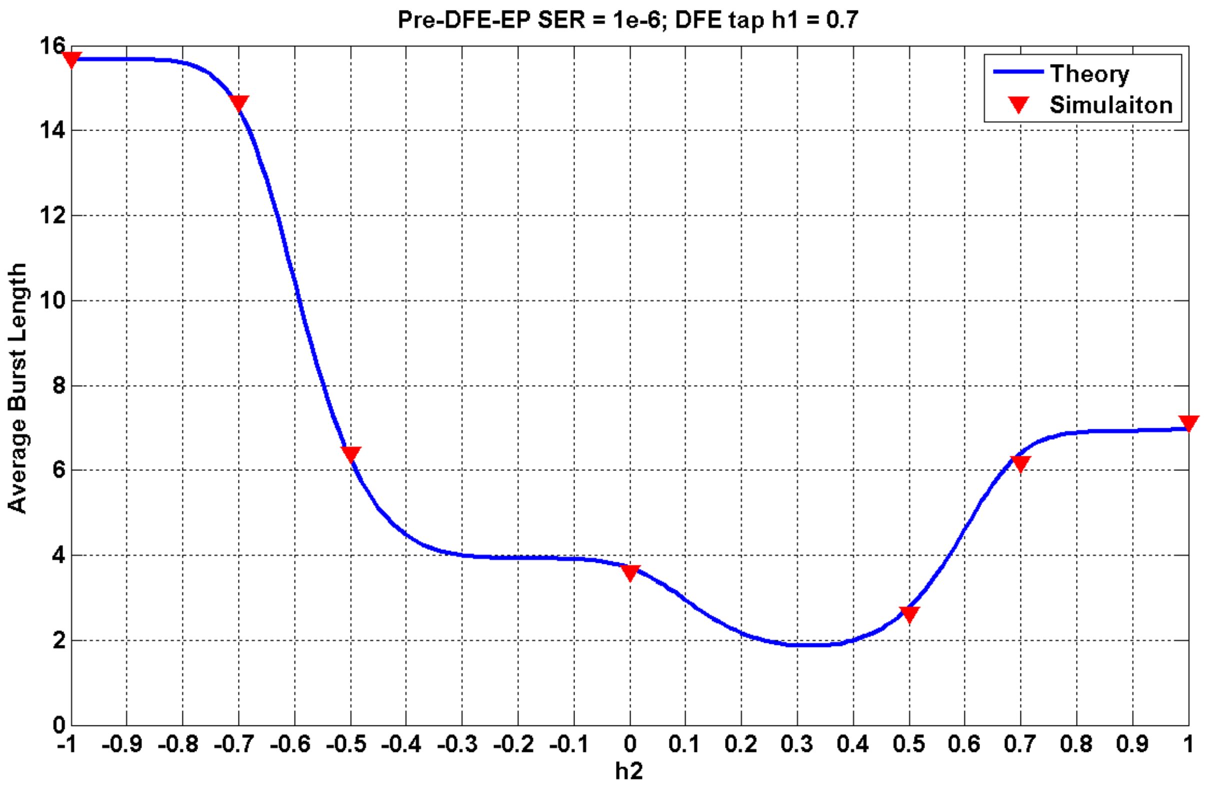
- An example on how symbol errors propagate for a 3-tap DFE receiver is shown. The notation follows



- If p_n are known, then we can compute burst error length
- It is obvious that not every symbol has to be incorrect for the errors to continually propagate

A 2-tap DFE error propagation example

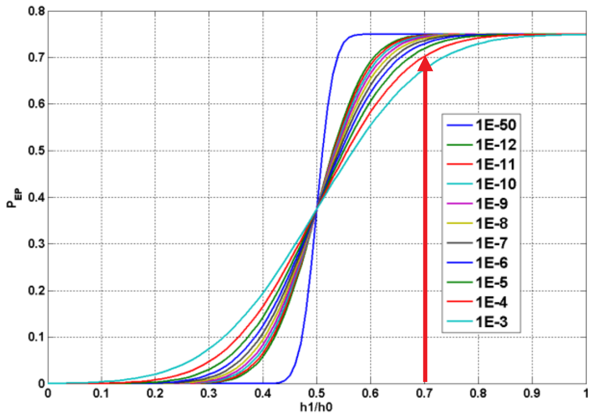
- To get a closed form solution for *N*-tap DFE is unlikely
  - Some assumptions applying to 1-tap DFE are not true any more in general
- For a 2-tap DFE, the assumptions still more or less hold
  - It is seen that, for a given h1, whether h2 is positive or negative, although the same magnitude, the impact on burst error length is completely different
    - This can be interpreted physically. But with more taps it gets harder and harder
  - Later we will see that the average error burst length (the "a" concept) is not a good indicator on how bad the error propagation is. It is not a good indicator on error propagation impact on system performance with FEC, either

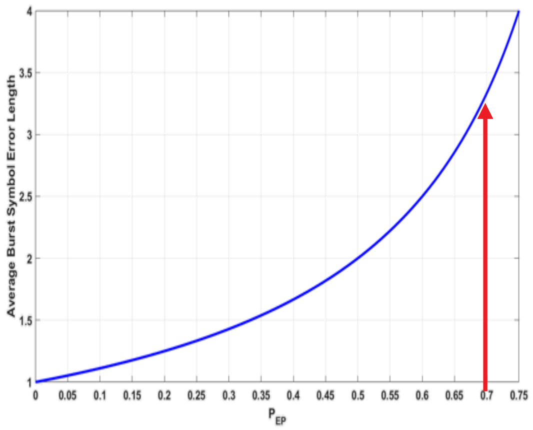


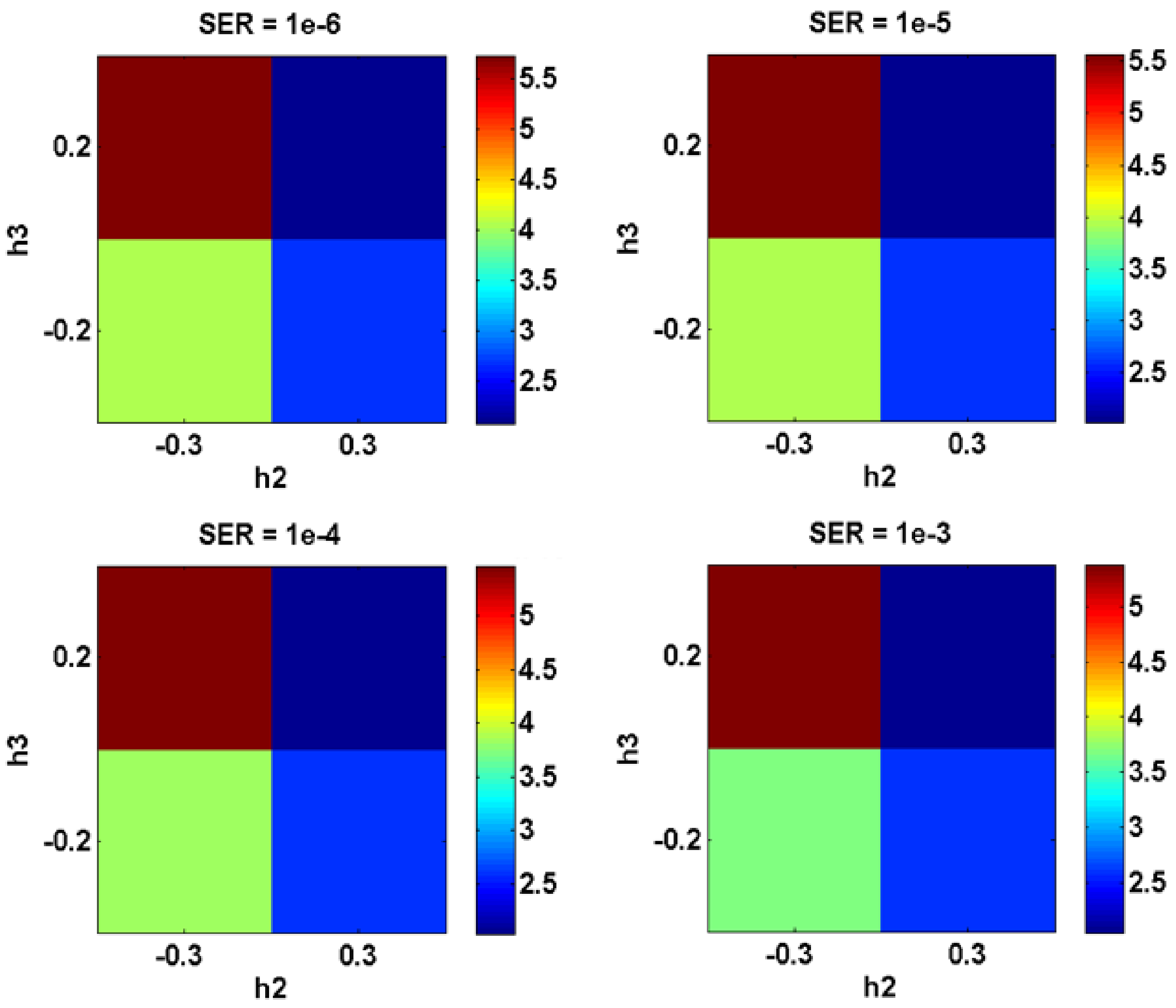
XILINX

A 3-tap DFE error propagation example

- For h1=0.7, SER ratios are computed
  - The min{SER ratio} is ~2; this is much smaller than the value for a single tap at 0.7, which is >3
  - The max{SER ratio} is >5.7; this is much larger than the value for a single tap at 0.7, or even equal to 1.0
- Thus, using the average burst error length to reversely estimate error propagation probability, “a”, will not lead to correct conclusions



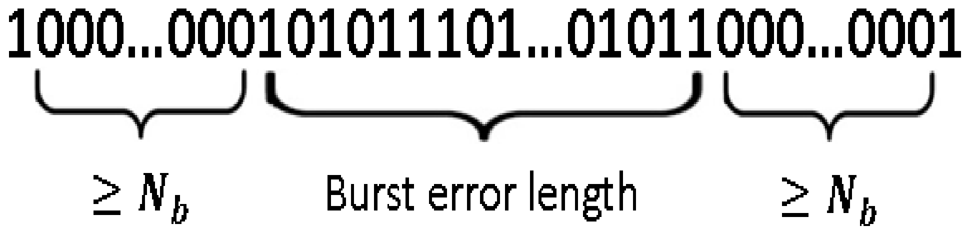




XILINX

Burst error length definition discussions

- From the above the analysis we conclude that precoding can remove all the errors, when the zero-sum error pattern is met, except
  1. The very first error that remains uncorrected
  2. The very first correct symbol following a wrong symbol
- However, a burst error length is not necessarily a continuous block of wrong symbols. More appropriately, a burst error length is defined as a contiguous sequence of symbols such that the first and last symbols are in error and there exists no contiguous subsequence of N_b correctly received symbols within the error burst. Let's call it *BEL*
  - N_b is typically set to the number of DFE taps
  - When N_b =1, there are no holes in the burst



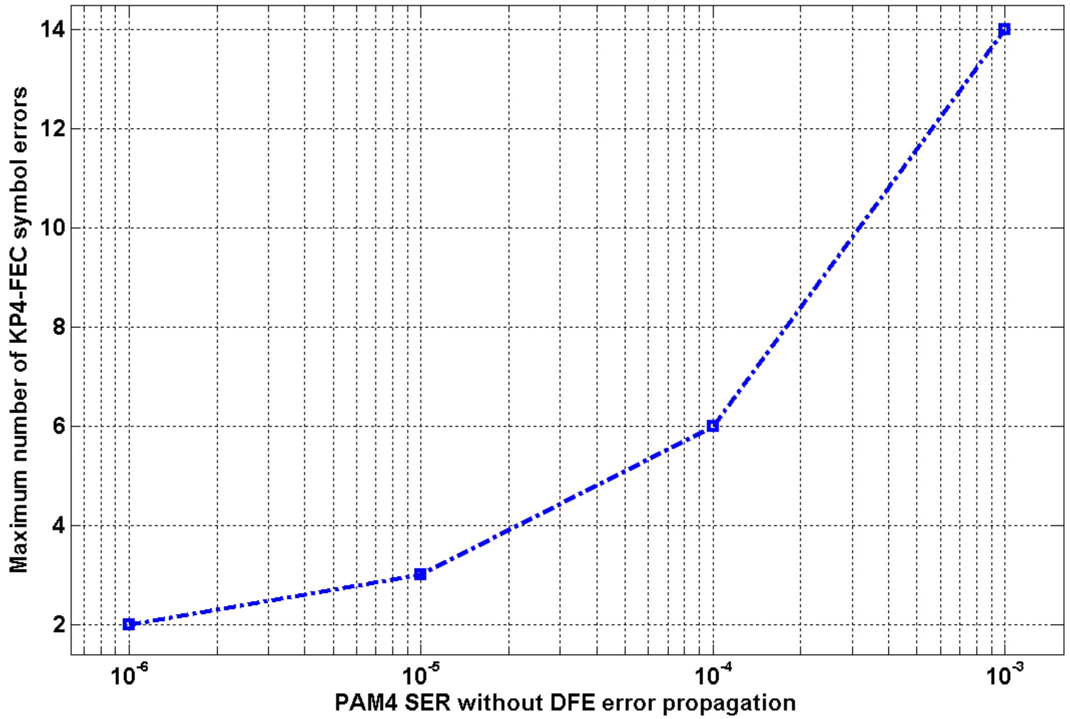
- Thus, a burst error can contain multiple continuous errors, interleaved with correct symbols
  - The contribution of precoding is very difficult to assess without simulations

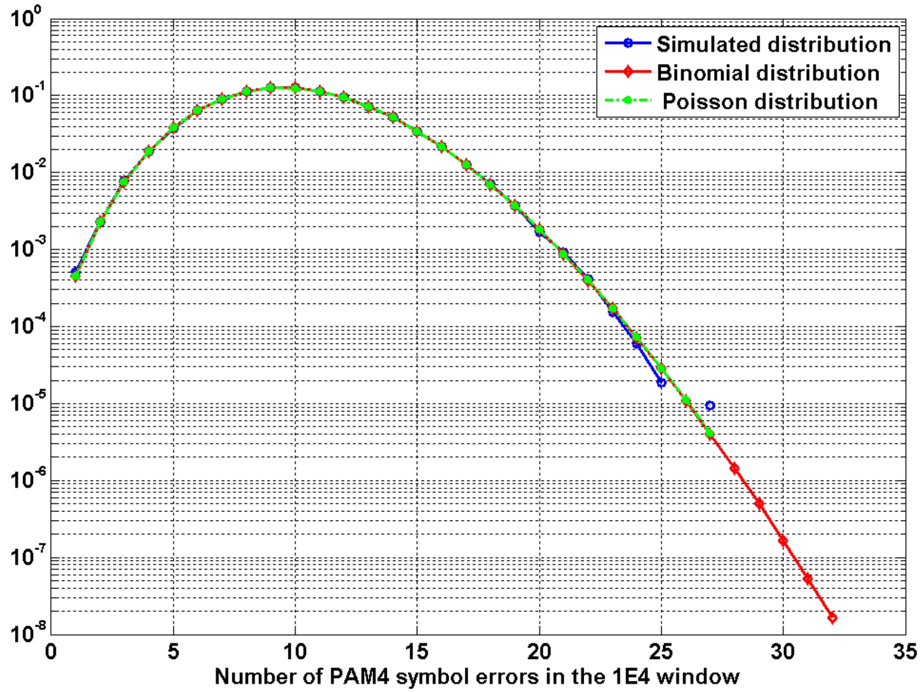
Link simulations for error propagation and precoding

- Simulation is in the time domain, performed in a symbol-by-symbol manner
  - The simulation is done in Matlab
- A total of 10 cycles of PRBS31Q, Gray-coded and mapped to PAM4, are simulated
  - Over 10 billion PAM4 symbols
- Random noise and minor residual ISI are adjusted
  - Such that the base SER (PAM4 symbol error ratio) without DFE error propagation is around a desired pre-set target
- For each setup there are 3 sets of simulations, done in parallel across the 3 setups:
  1. Base link simulation without DFE error propagation
  2. DFE is enabled, but precoding is off
  3. DFE is enabled, and precoding is on
- The simulated errors are further post-processed to evaluate KP4 FEC performance

Case 1: DFE-less, AWGN dominated channel

- AWGN dominated channel, with minor contributions from residual ISI
  - AWGN is adjusted to achieve SER ~ 1e-6, 1e-5, 1e-4, 1e-3
- The maximum KP4 symbol error count is plotted for different SER
  - SER ~ 1e-3 seems to be a good rule of thumb for KP4 FEC correction capability due to AWGN
- SER statistics for 1e-3 is computed, as an example
  - The computed *Variance/mean* = 0.9982 ~ 1.0, a good approximation to Poisson distribution



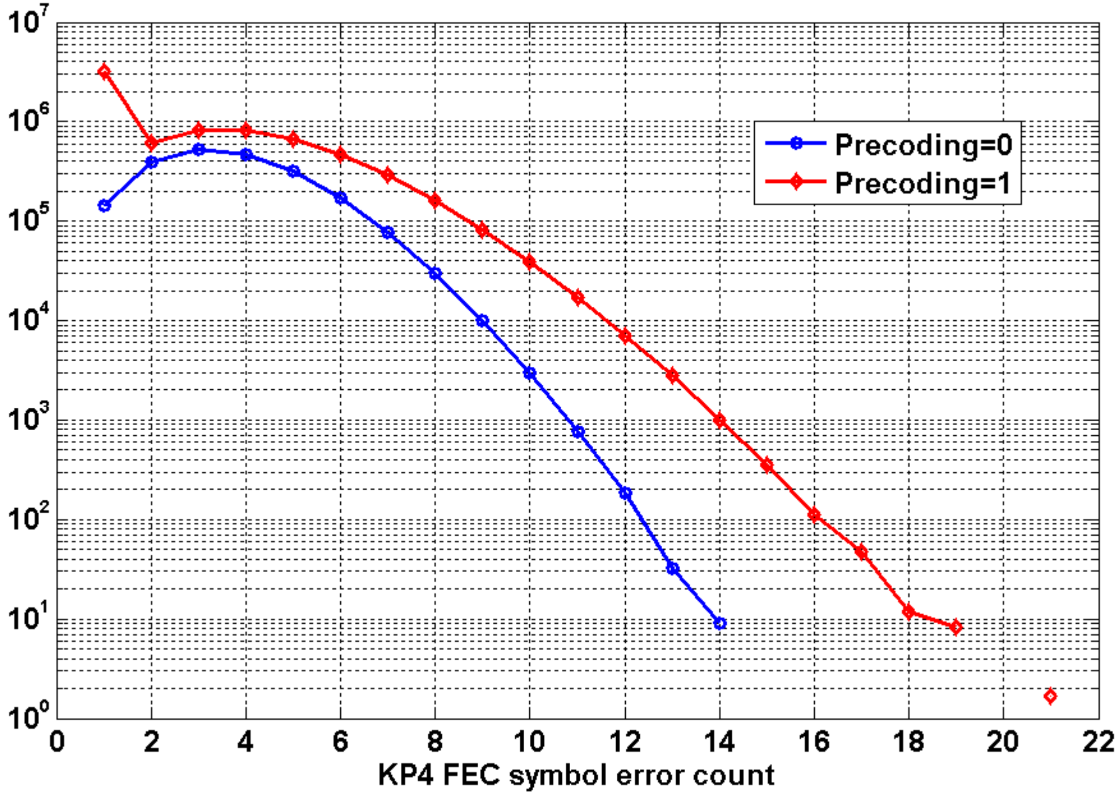


XILINX

Case 1: DFE-less, AWGN dominated channel (Con't)

- Precoding impact on KP4 FEC performance is studied for SER = 1e-3
  - Precoding roughly doubled the SER, as expected from theory

    (SER with precoding = 1.9957e-3)/ (SER without precoding = 9.9899e-4) ≈ 2

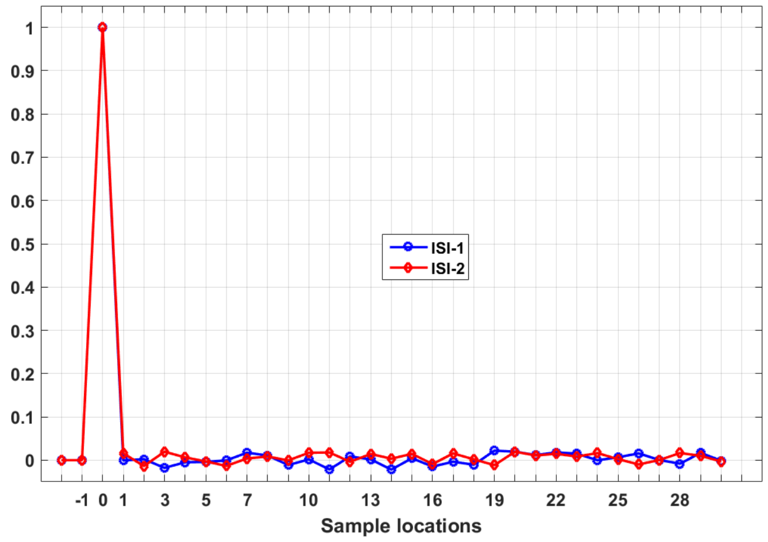


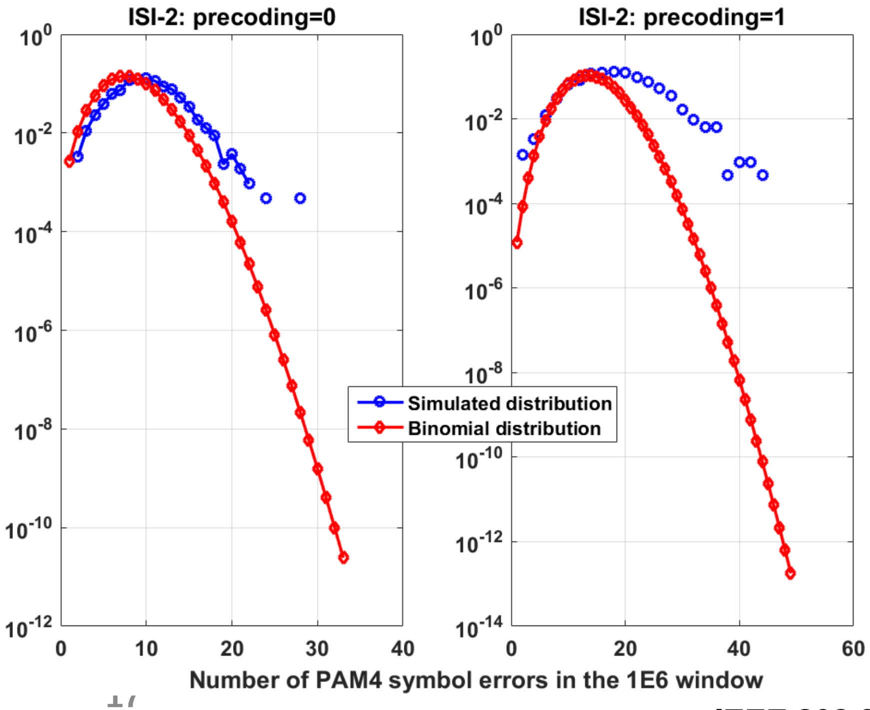
- Precoding for FEC performance
  - Precoding weakened FEC performance
    - Without precoding the maximum KP4 FEC symbol error count is 14
    - With precoding the maximum KP4 FEC symbol error count is 21
      - There are a lot more beyond 14
- Thus, for AWGN dominated channels, 1/(1+D) precoding should be avoided

Case 2: DFE-less, residual ISI dominated channel

- Two residual ISI distributions are simulated
  - AWGN impact now is secondary to residual ISI impact

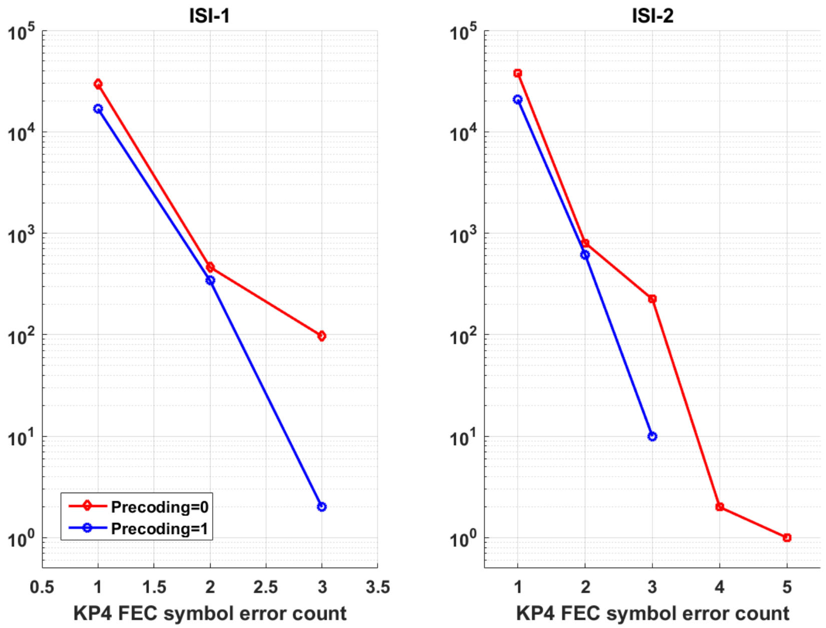
Residual ISI	ISI-1	ISI-2
Precoding = 0	9.9968E-6	7.9991e-6
Precoding = 1	1.8214e-5	1.3961e-5





Unlike an AWGN-dominated channel in which symbol error statistics follows Binomial distribution, the error statistics from a residual-ISI-dominated channel takes a different profile

For both residual-ISI dominated channel, precoding actually made KP4 FEC performance worse

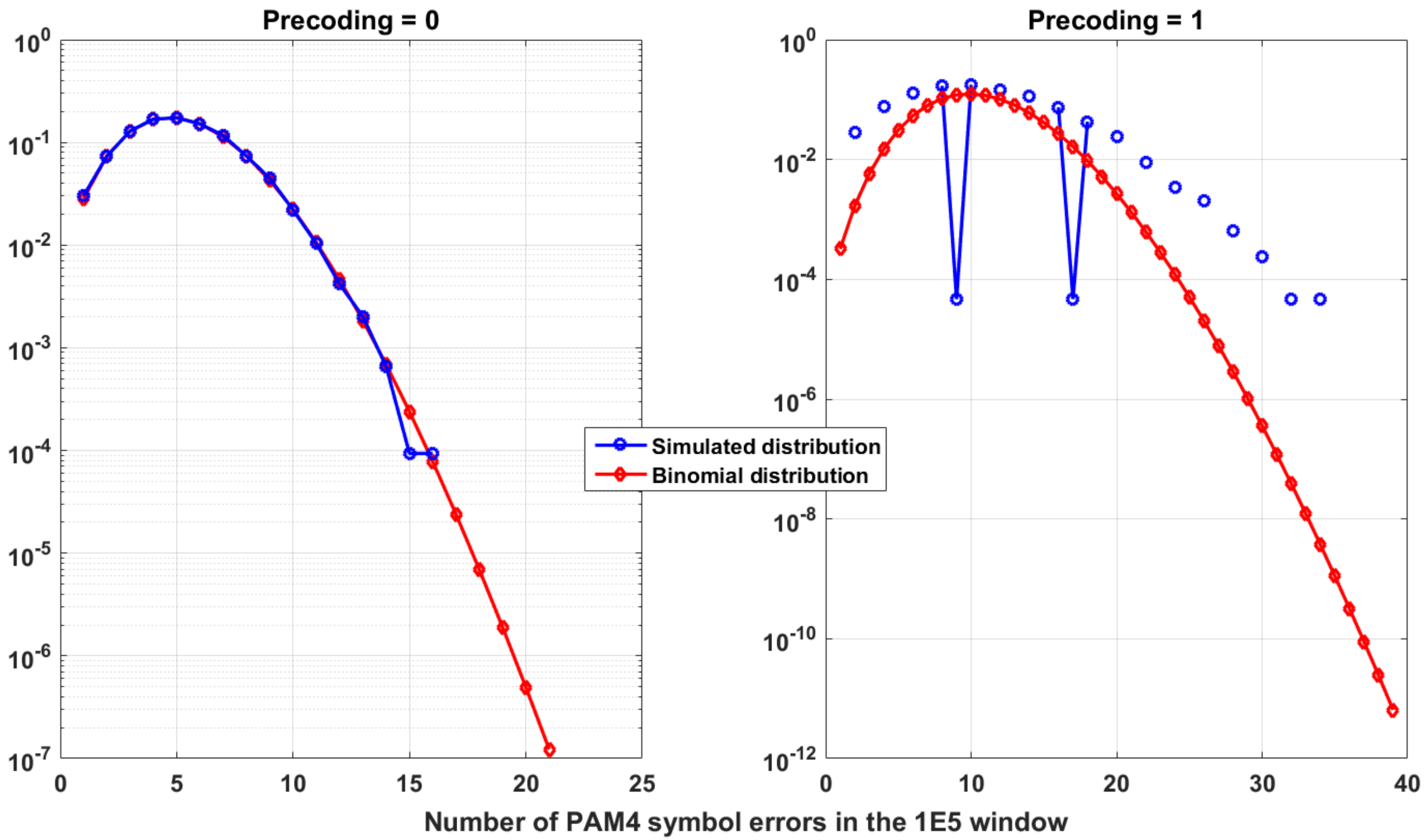


Case 3: DFE-less, single residual ISI dominated channel

- Consider only a signal cursor residual ISI dominating the channel errors
  - Precoding in does not help SER or maximum FEC symbol errors, link in Cases 1 and 2
  - PAM4 symbol error distributions without precoding is less a problem than with precoding

	PAM4 SER
Precoding = 0	5.2155e-5
Precoding = 1	1.0380e-4

	Max FEC SE
Precoding = 0	5
Precoding = 1	7

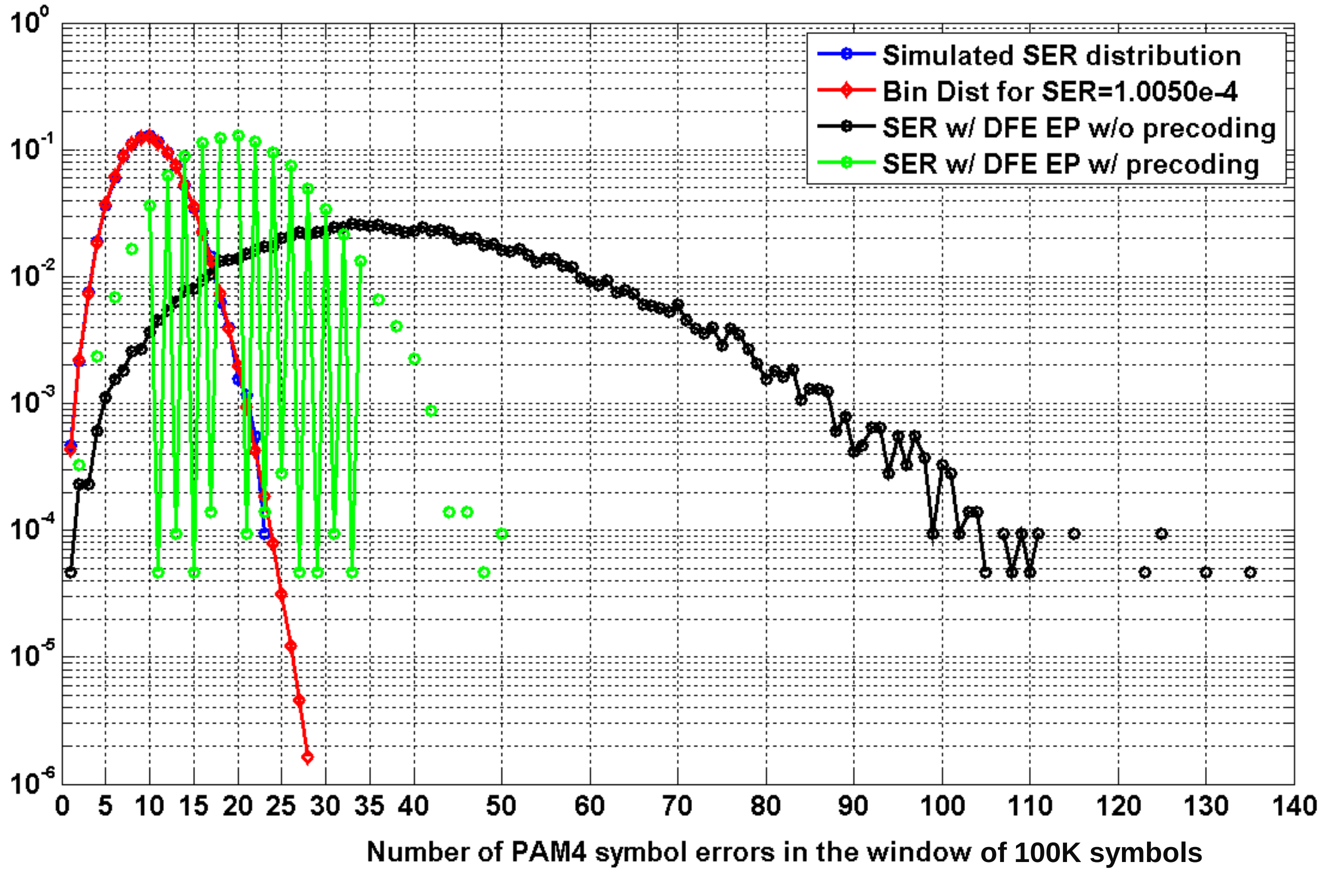


XILINX

Case 4a: 1-tap DFE at the first post-cursor

- First, the limit case, h1/h0 = 1, is studied for base SER = 1.0050e-4
  - The SER after precoding is reduced by roughly half from that without precoding
  - Theoretically, the SER with precoding is always twice of the base SER

Conditions	SER
Precoding = 0	4.0227E-4
Precoding = 1	2.0047E-4



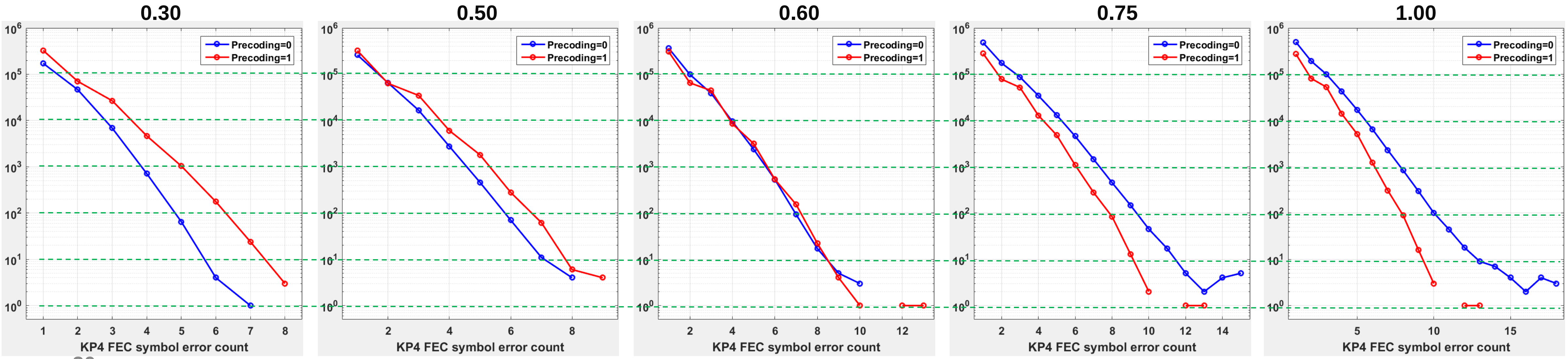
- It is observed that
  - Without error propagation the error distribution (blue) essentially follows binominal distribution (red)
  - With error propagation the error distribution is very different (black)
  - Precoding effectively reduced the error occurrence (with the given setting) for this setup (green)

Case 4a: 1-tap DFE at the first post-cursor (Con't)

- Different h1/h0 values are simulated
  - It is seen that after precoding the SER is essentially the same, as stated on the previous slide

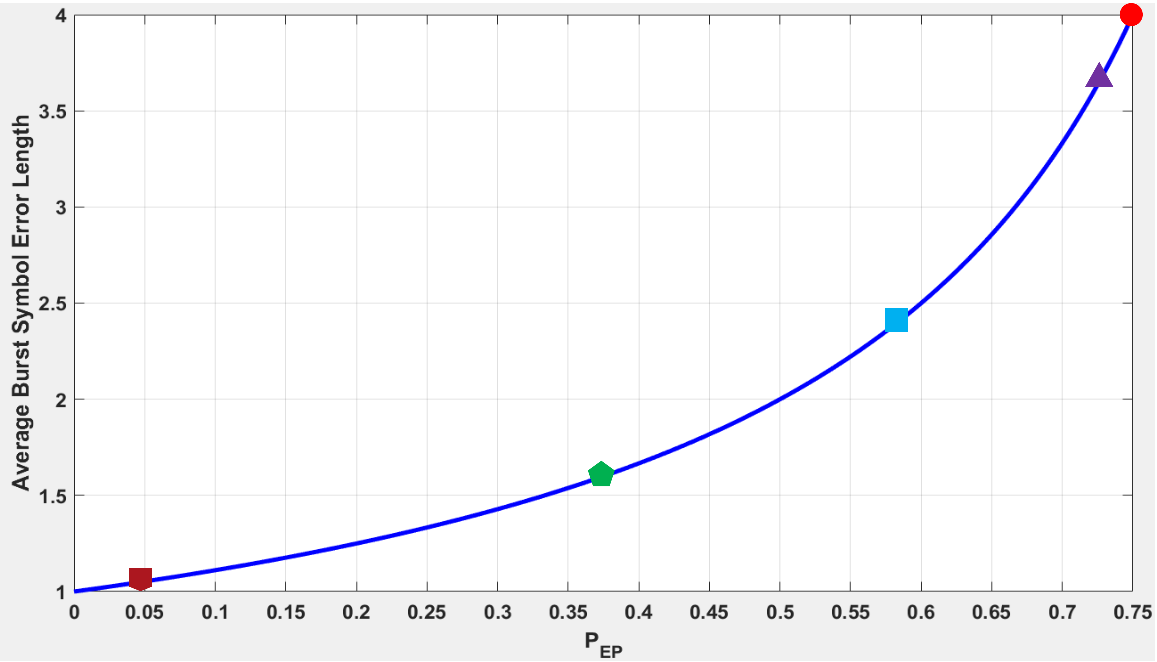
h1/h0	0.30	0.50	0.60	0.75	1.00
Precoding = 0	1.0556e-4	1.6125e-4	2.4126e-4	3.7143e-4	4.0227E-4
Precoding = 1	2.0051e-4	2.0050e-4	2.0049e-4	2.0046e-4	2.0047E-4

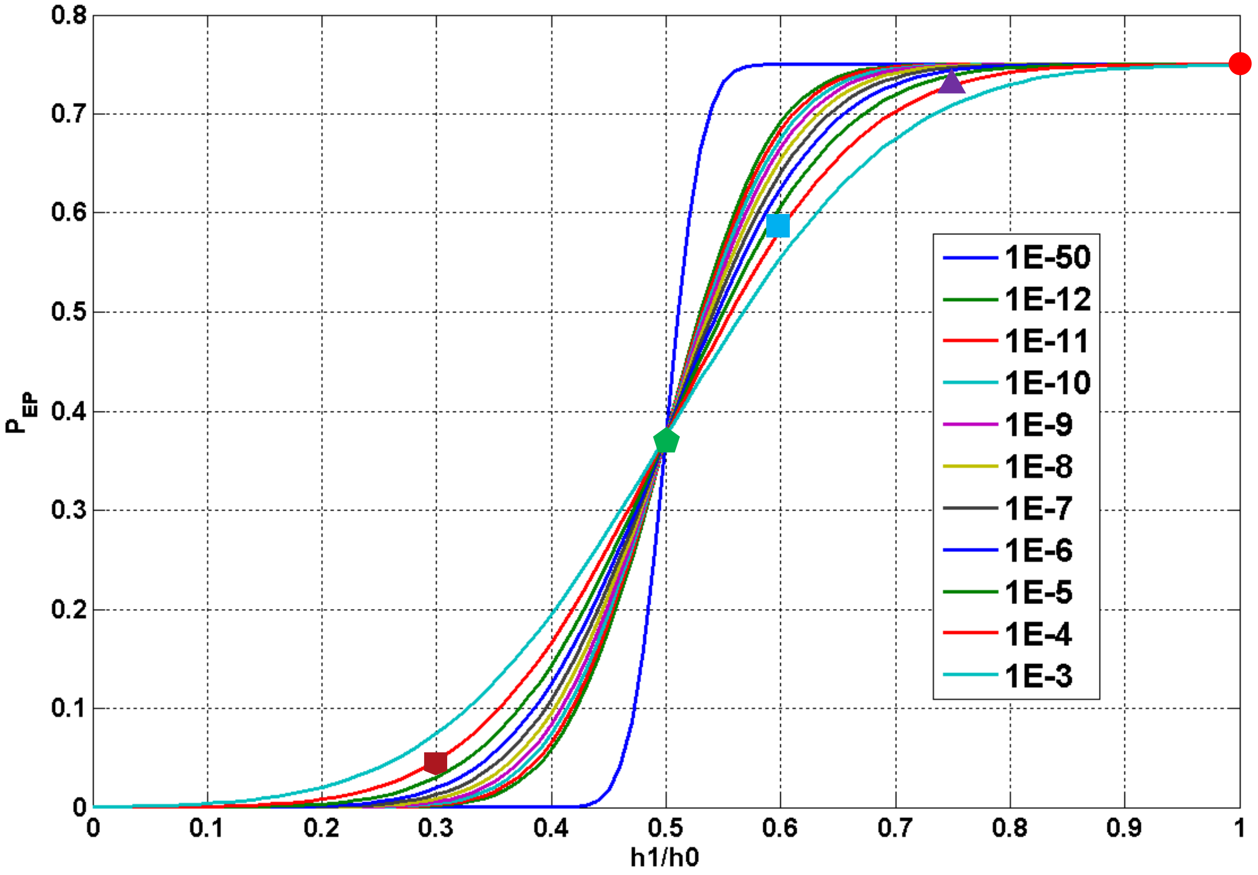
- The impact on KP4 FEC performance: red curve moved from above to below the blue curve
  - Precoding works well for h1/h0 > ~0.6 (but the exact value is a function of many conditions)
  - Precoding makes the overall FEC performance worse for h1/h0 < ~0.6



Average burst error length discussions

- h1=1.00: 4.0227e-4/1.0050e-4 = **4.0002**
- h1=0.75: 3.7143e-4/1.0050e-4 = **3.6958**
- h1=0.60: 2.4126e-4/1.0050e-4 = **2.4006**
- h1=0.50: 1.6125e-4/1.0050e-4 = **1.6045**
- h1=0.30: 1.0556e-4/1.0050e-4 = **1.0503**





- For a 1-tap DFE at h1, the average burst error length and the “a” concept work perfectly

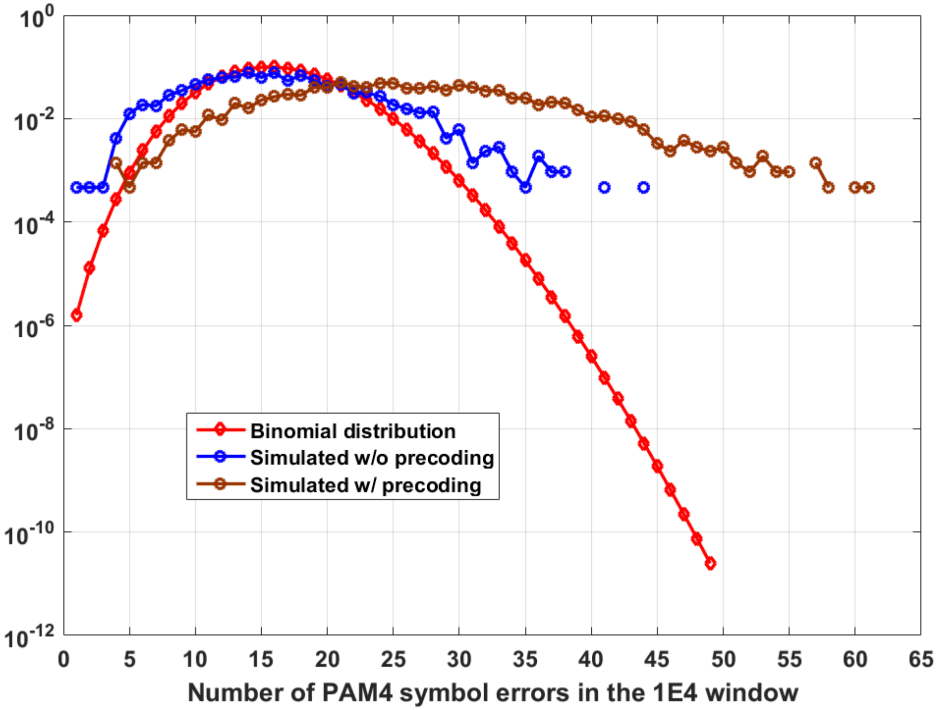
XILINX

Case 4b: 1-tap DFE at the first post-cursor (Con't)

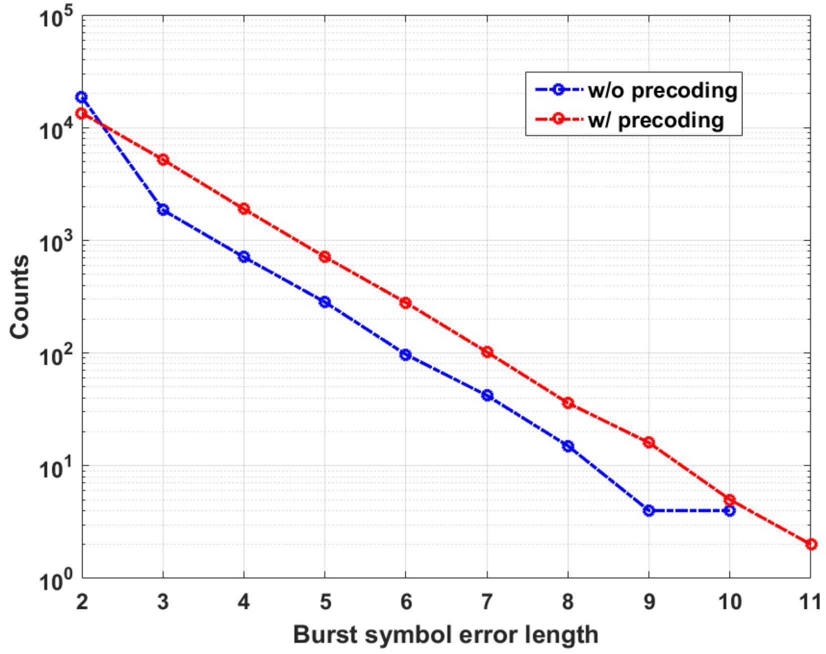
- Set h1=-0.5 for base SER=1e-5
  - The SER ratio after and before precoding is 1.6319 (for h1=+0.5, this ratio is 1.2434). Thus, h1 tap magnitude and polarity both matter

Conditions	SER
Precoding = 0	1.6142e-5
Precoding = 1	2.6342e-5

- PAM4 symbol error statistics
  - Precoding makes the number of errors in a window larger; precoding increases the probability of errors



- Maximum burst symbol error length

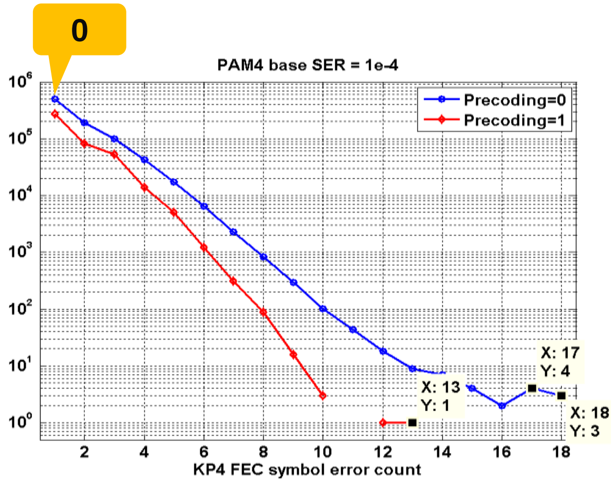


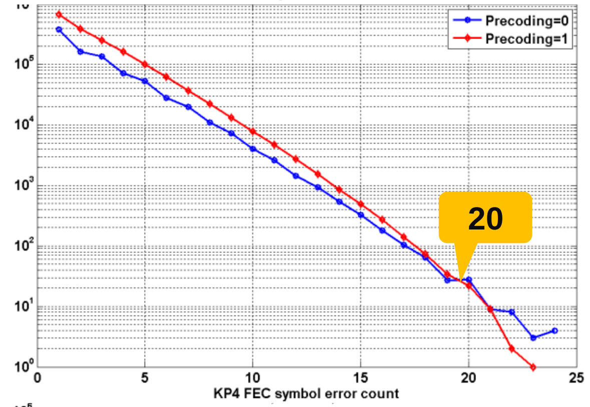
XILINX

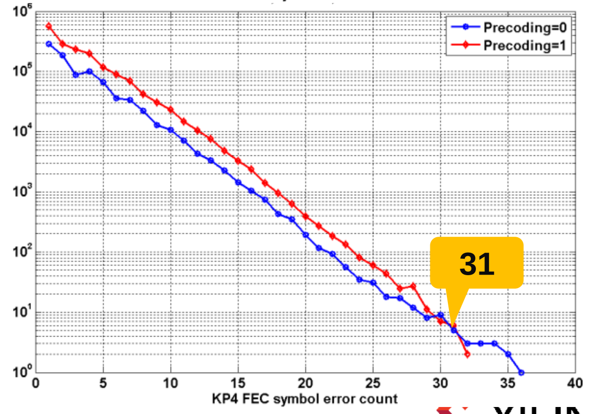
Case 4c: DFE with only one non-zero tap

- For $h_k/h0 = 1$ (k=1, 2, 3), and no other non-zero taps
  - When precoding is not applied, the overall SER is about 4x of the base SER (1.0050e-4) for all 3 examples, as expected
  - When precoding is enabled, it is seen that
    - For k=1, the new SER halved the base SER; precoding helps
    - For k=2,3, the new SER doubled the SER without precoding which can be proven theoretically; precoding hurts
- DFE tap locations
  - The farther away the tap, the more negative impact on error propagation
  - The error signature is not obviously reflected from the post-DFE SER
  - The concept of "a" only applies for a 1-tap DFE at the first post-cursor

Non-zero DFE tap	h1/h0 = 1	h2/h0 = 1	h3/h0 = 1
Precoding = 0	4.0227E-4	4.0942E-4	4.0351e-4
Precoding = 1	2.0047E-4	8.0124E-4	8.0235e-4





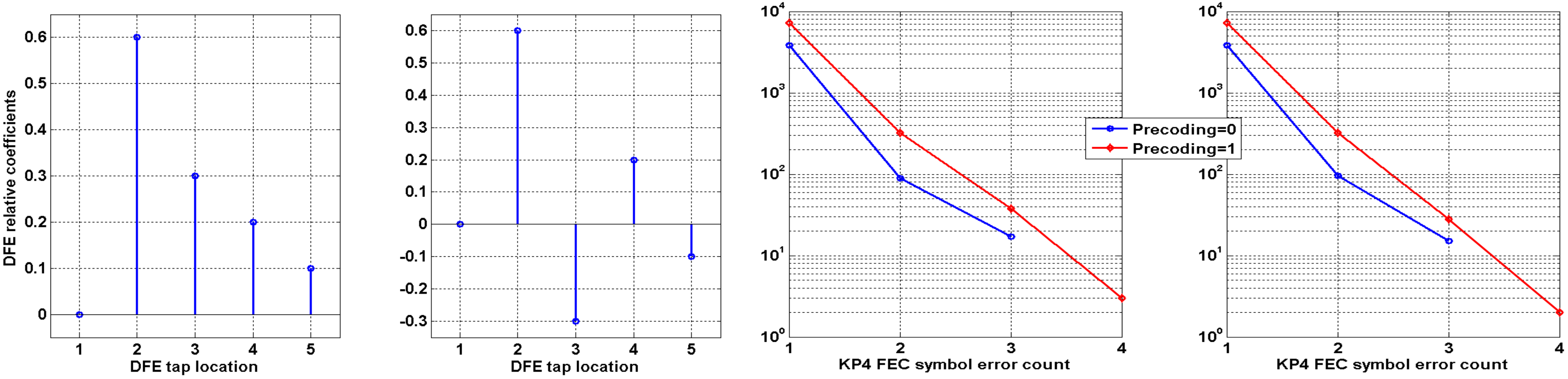


XILINX

Case 5: 5-tap DFE, with h1=0

- Two 5-tap DFE's are constructed as shown
  - Tap h1 is set to 0, emulating some designs
  - The resulting SERs are always higher when precoding is turned on
- For both configurations the link shall work better when precoding is not used with or without FEC

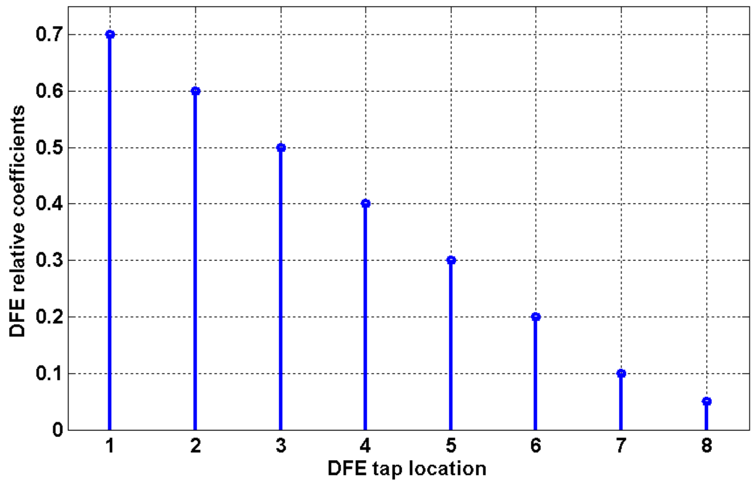
Base = 1.0324e-6	DFE-1	DFE-2
Precoding = 0	1.8212e-6	1.8268e-6
Precoding = 1	3.5600e-6	3.5465e-6



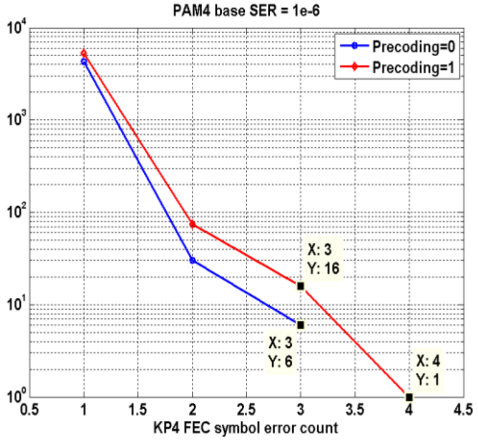
XILINX

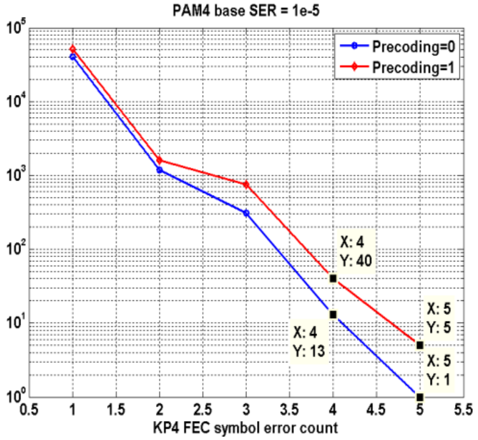
Case 6: 8-tap DFE

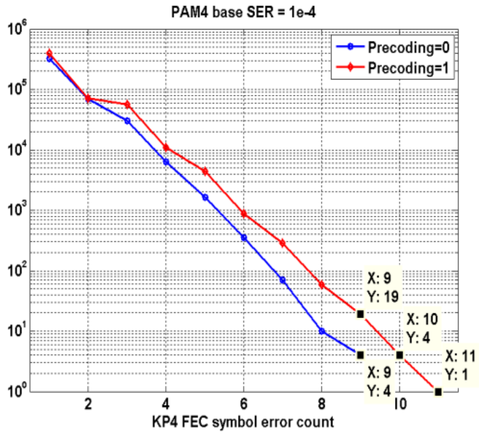
- Precoding only made the link worse
  - The error propagation in terms of the average SER increase is around 2x without precoding; and around 2.5x with precoding
  - With only the first DFE the average SER increase is around 3.3. Now, with some extra DFE tap with given setting, the average error count increase is smaller



Base	DFE enabled		KP4 Max Symbol Errors	
Noise and ISI	precoding = 0	precoding = 1	precoding = 0	precoding = 1
1.0324e-6	2.0233e-6	2.4987e-6	3	4
1.0090e-5	1.9809e-5	2.5024e-5	5	5
1.0050E-4	1.9952e-4	2.5380e-4	9	11



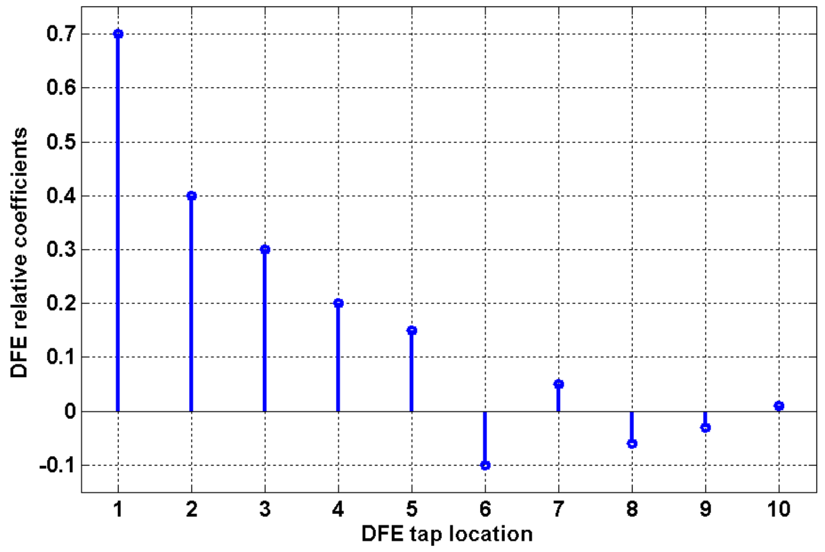




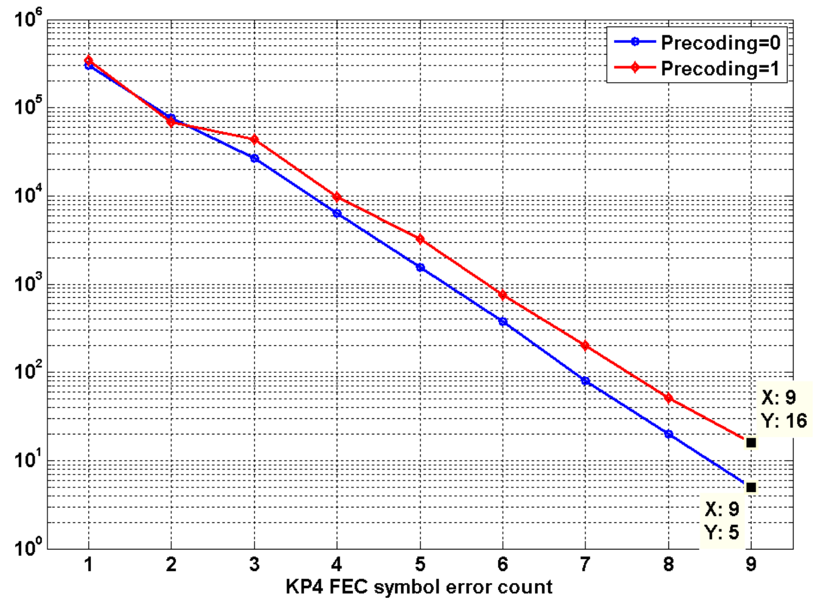
XILINX

Case 7: 10-tap DFE

- The 10-tap DFE is constructed as shown
- Two base SER's are simulated, 1e-6 and 1e-4
  - With precoding the SER is always worse
- Statistically, precoding hurts KP4 FEC performance for the example of SER = 1e-4
  - The impact on KP4 symbol error count between precoding on and off is the same from the limited simulations



	DFE enabled	
Noise and ISI	**precoding = 0**	**precoding = 1**
1.0324e-6	1.8831e-6	2.1355e-6
1.0050E-4	1.9247e-4	2.1980e-4



	KP4 Max Symbol Errors	
Noise and ISI	**precoding = 0**	**precoding = 1**
1.0324e-6	4	4
1.0050E-4	9	9

Case 8: Alternating DFE tap configurations

- To see how bad error propagation can be, for a base SER at 1e-5, four DFE lengths are tried
  - 5-tap = [0.7, -0.1, 0.1, -0.1, 0.1]
  - 7-tap = [0.7, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1]
  - 9-tap = [0.7, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1]
  - 11-tap = [0.7, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1]
- It is observed that, for *this set of* alternating DFE tap coefficients, precoding always improves performance in SER, BEL, and KP4 FEC max symbol error count
  - However, the performance is so bad that the FEC completely fails even for the 7-tap case
  - Does precoding always help in the alternating DFE tap configurations?

DFE tap configurations	SER		BEL		Max FEC SE	
	PreC=0	PreC=1	PreC=0	PreC=1	PreC=0	PreC=1
5-tap	4.2673e-5	2.2012e-5	39	20	8	7
7-tap	6.2625e-5	3.2712e-5	104	58	21	17
9-tap	1.5739e-4	1.0047e-4	607	333	122	98
11-tap	8.8070e-4	7.2101e-4	3071	2378	544	508

Case 8: Alternating DFE tap configurations (Con't)

- Reducing h1 to 0.4 from 0.7, and repeat the simulation for two cases
  - 5-tap = [0.4, -0.1, 0.1, -0.1, 0.1]
  - 11-tap = [0.4, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1, -0.1, 0.1]
- It is observed that if h1 values is reduced or the rest DFE taps become larger relative in magnitude, precoding effect diminishes
  - Precoding could degrade overall link performance; its effect should be analyzed case by case

DFE tap configurations	SER		BEL		Max FEC SE	
	PreC=0	PreC=1	PreC=0	PreC=1	PreC=0	PreC=1
5-tap: h1=0.7	4.2673e-5	2.2012e-5	39	20	8	7
5-tap: h1=0.4	1.3674e-5	2.0415e-5	27	26	7	7
11-tap: h1=0.7	8.8070e-4	7.2101e-4	3071	2378	544	508
11-tap: h1=0.4	5.0022e-5	3.8338e-5	887	492	178	167

Case 9: DFE from COM settings

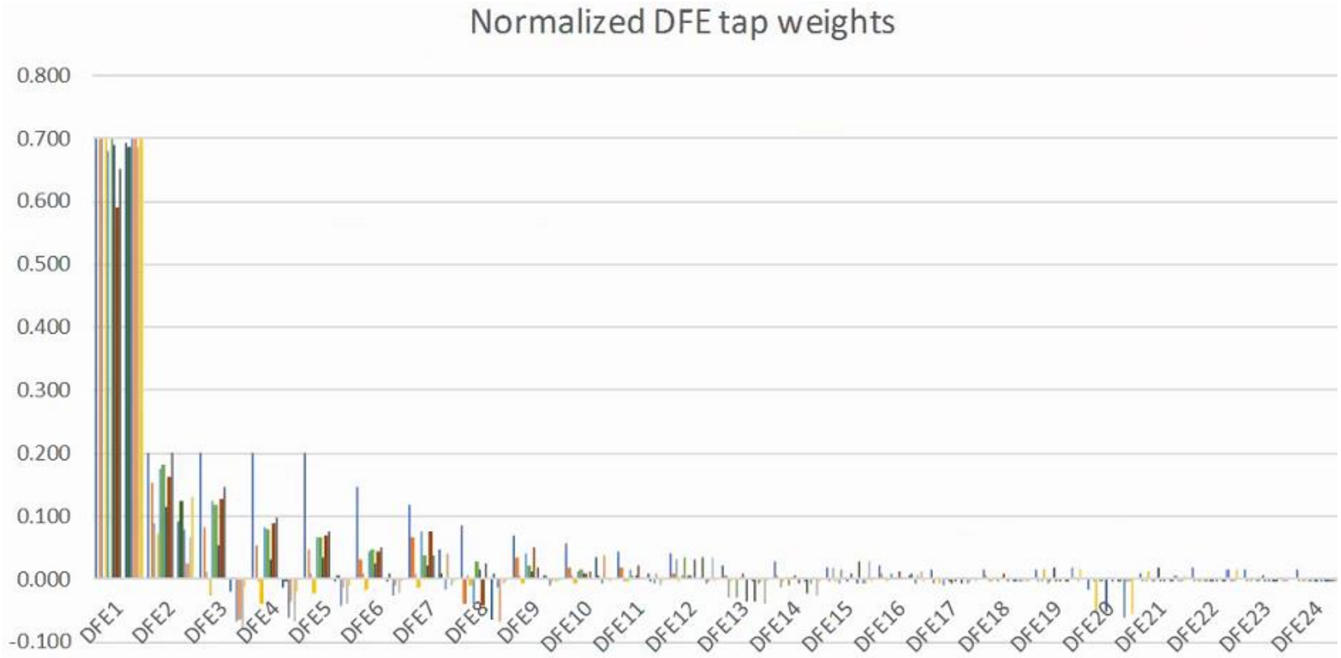
- DFE values are from COM based analysis
  - The spreadsheet, "*some_DFE_tap_results.xlsx*", was simulated and prepared by Richard Mellitz, which I received from Pete Anslow
  - Four of cases are singled out for the simulation in the presentation

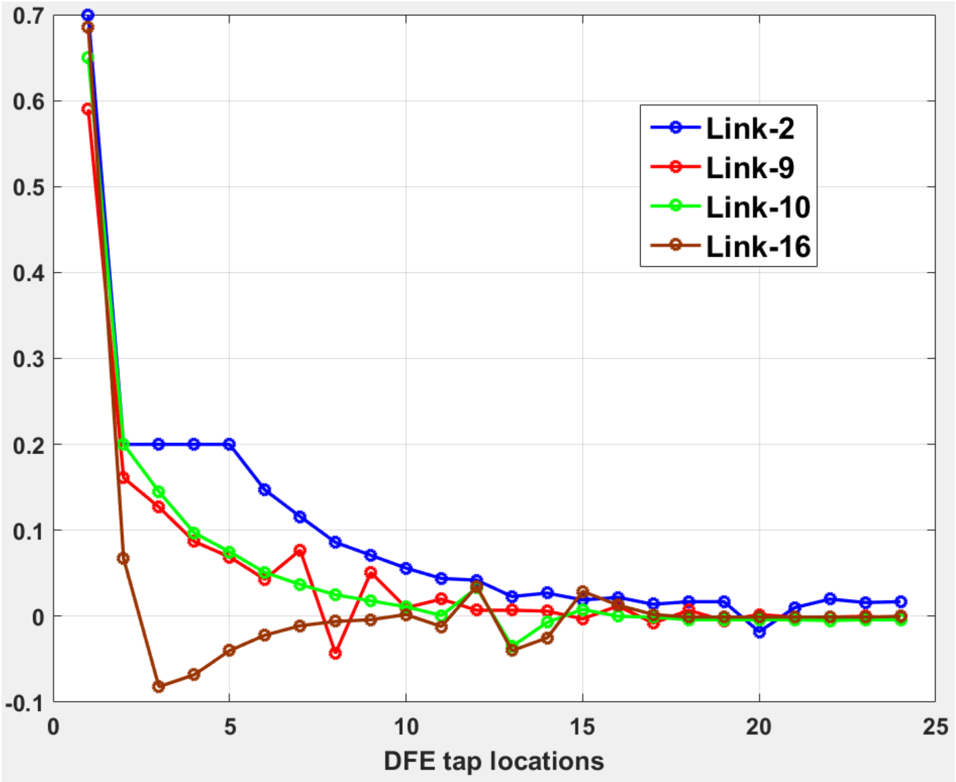
Link 2: "tracy_100GEL_05_0118\B56_Thru_CblBP", 30mm

Link 9: "mellitz_3ck_02_081518_CBP\CaBP_BGAVia_Opt2_24dB", 12mm

Link 10: "mellitz_3ck_02_081518_CBP\CaBP_BGAVia_Opt2_24dB", 20mm

Link 16: "mellitz...72518_channels--Z0d_100_14p25in_2dBpi_meg6_rtf", 20mm

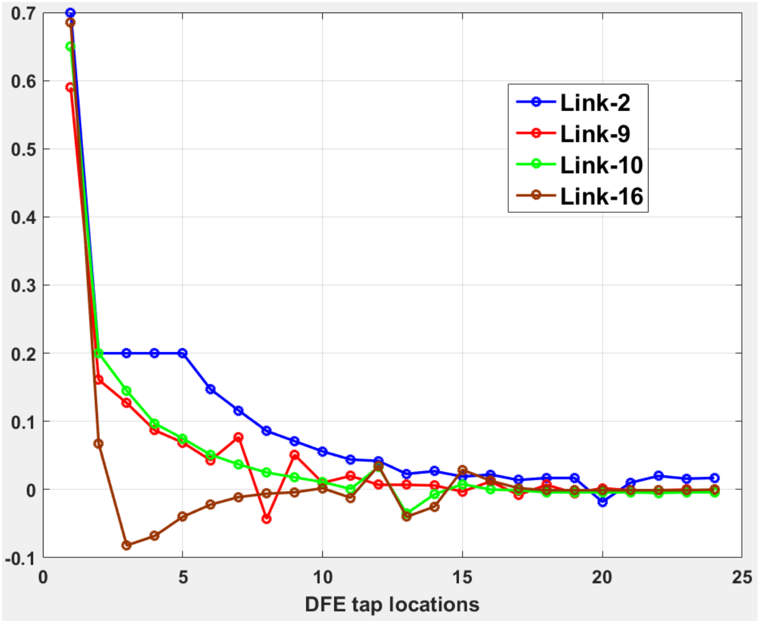


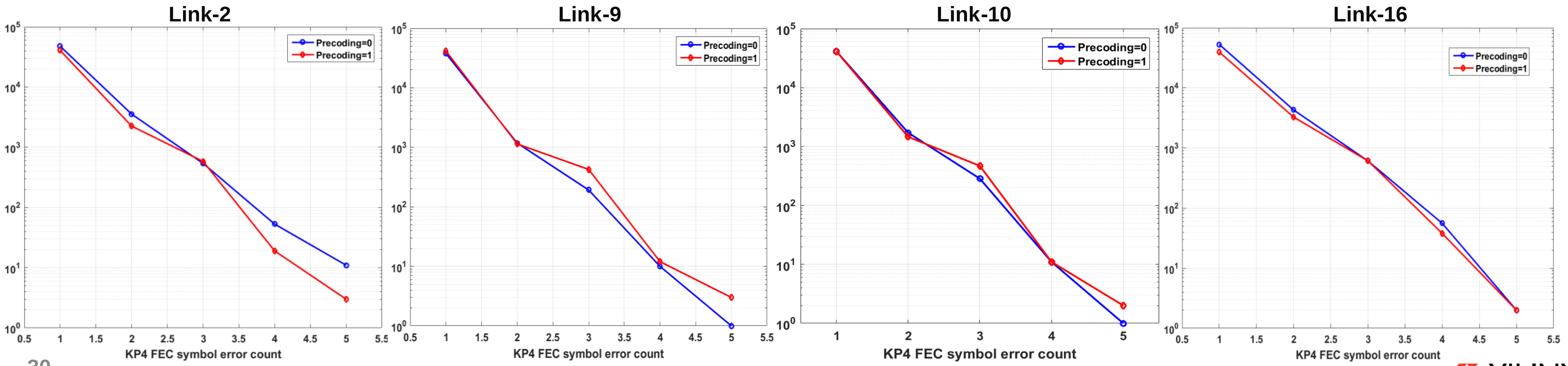


Case 9: DFE from COM settings (Con't)

- Observations from the simulation results
  - h1 is dominant, the outcome is somewhat 1-tap-DFE-like
  - Precoding improved in some and degraded in others
  - The impact of precoding is situation dependent

Links	2	9	10	16
Precoding = 0	2.4205e-5	1.8180e-5	2.0154e-5	2.6985e-5
Precoding = 1	2.0317e-5	2.0043e-5	2.0124e-5	2.0231e-5

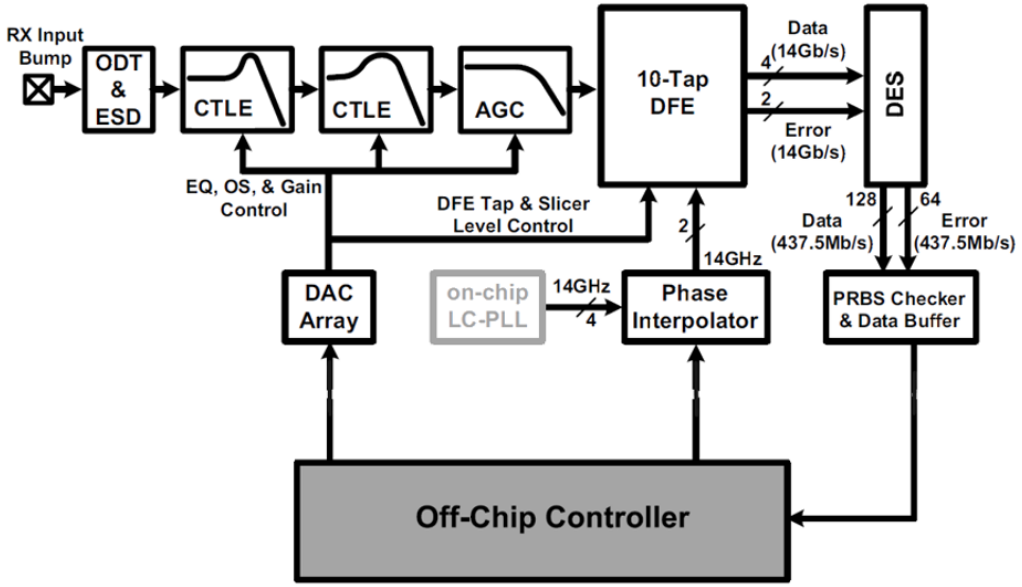




XILINX

Case 10: DFE from silicon measurement

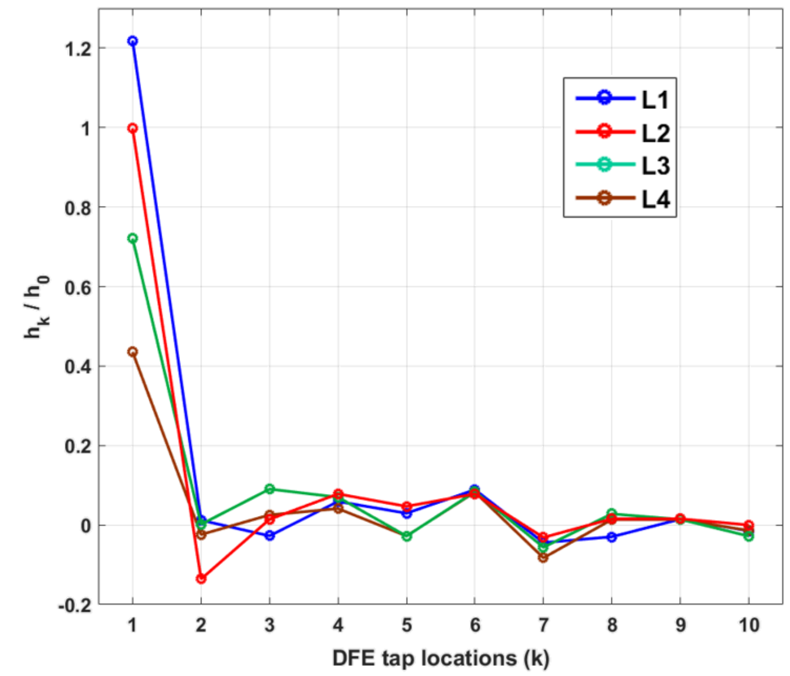
- An analog-based RX architecture test-chip designed to meet CEI-56G-MR-PAM4
  - It is seen from the RX architecture highlight, there is a 2-stage CTLE and an AGC, and 10-tap DFE
  - DFE h1 is intended to take care of the loss around the Nyquist frequency
  - The rest 9 DFE taps are mainly for fine tuning what is left over after TX FIR and CLTE
- Lab measurement
  - 13 channels, from backplanes to copper cables
  - Ball-to-ball loss ranges from 15.5 to 34.5dB, while BER ranged from 3.56e-10 to 7.41e-6 (not a direct function of losses), without crosstalk
  - Different amount of crosstalk was applied so that the system achieved BER roughly around 1e-4
  - Next, 4 channels are selected for simulations; the h1/h0 values cover the largest (1.2189) and the smallest (0.4352) and some middle ones

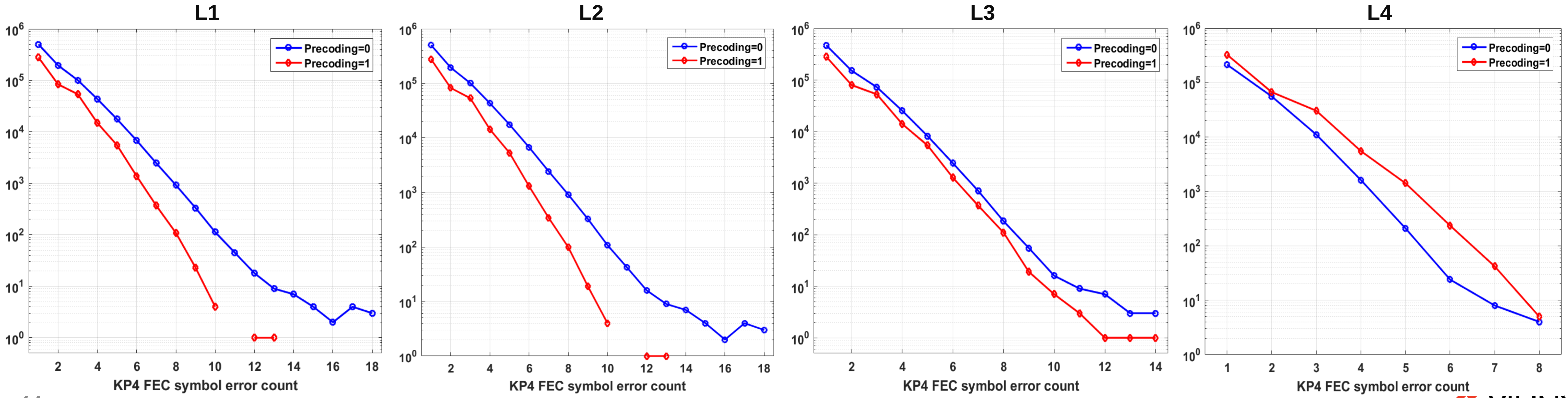


Jay Im, et al, "A 40-to-56Gb/s PAM-4 Receiver with 10-Tap Direct Decision-Feedback Equalization in 16nm FinFET", ISSCC 2017

Precoding impact for Case 10

- DFE tap convergence
  - With TX 3-tap FIR and RX 2-stage CTLE, the resultant DFE tap coefficients all showed h1 domination; 4 cases are selected
- It is seen that in 3 out of 4 cases precoding contributed positively to enhancing the KP4 FEC performance
  - The case in which precoding did not help FEC showed h1<0.5
- However, for a different design also with 10-tap DFE, precoding impact should be studied





XILINX

Summary and Conclusions

- DFE error propagation for PAM4 signaling is preliminarily studied
  - Beyond 1-tap DFE, error propagation effect is very complicated in general
  - The "a" approach for studying FEC performance only works for h1-tap DFE
- For a h1-tap DFE, precoding effect depends on the tap coefficient strength
  - The tap coefficient has to be large enough (roughly >0.6) for the precoding to help
  - With ADC based design, though DFE typically only has 1-tap, its value usually is small
- For designs with a multi-tap DFE, the front-end linear EQ plays a big role
  - TX FIR and RX CTLE should work jointly to help DFE tap coefficient distribution
- The 1/(1+D) precoding is helpful conditionally for PAM4 channel links
  - DFE coefficient signature directly impacts error signature, thus FEC performance
  - Multi-tap DFE coefficient polarity plays an equal role as the DFE tap strengths
  - Precoding effect for a multi-tap DFE architecture should be analyzed case by case
- Nevertheless, precoding is recommended in the standard for PAM4 links
  - Precoding implementation overhead is minimal
  - Use of precoding should be carefully accessed

Adaptable.
Intelligent.

XILINX®