Introduction to DSP Based Serial Links

eSilicon®

Collaborate. Differentiate. Win.

Fernando De Bernardinis
fdebernardinis@silicon.com

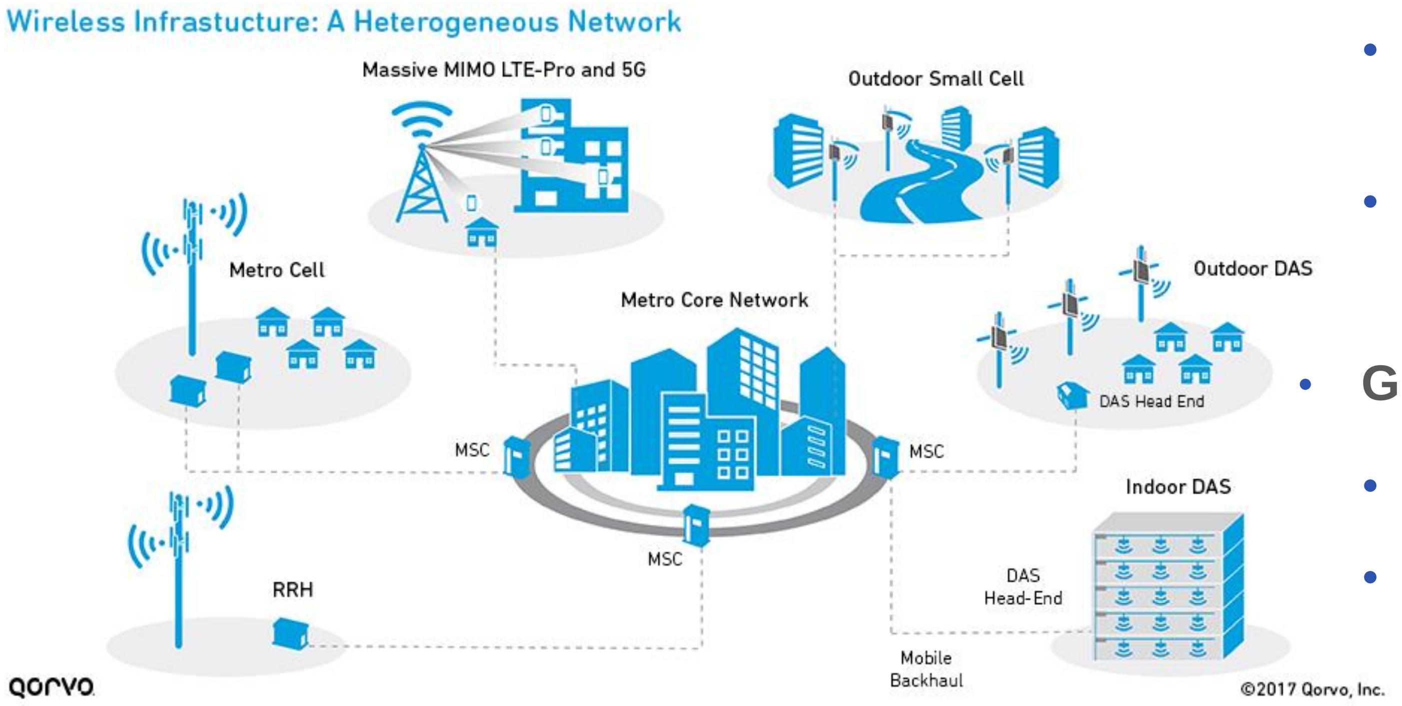
Presentation Outline

- What we do, and who we are
- Introduction to DSP based Serial Links
- ISSCC 2019 talk “A Sub-250mW 1-to-56Gb/s Continuous-Range PAM-4 42.5dB IL ADC/DAC-Based Transceiver in 7nm FinFET”

eSilicon®

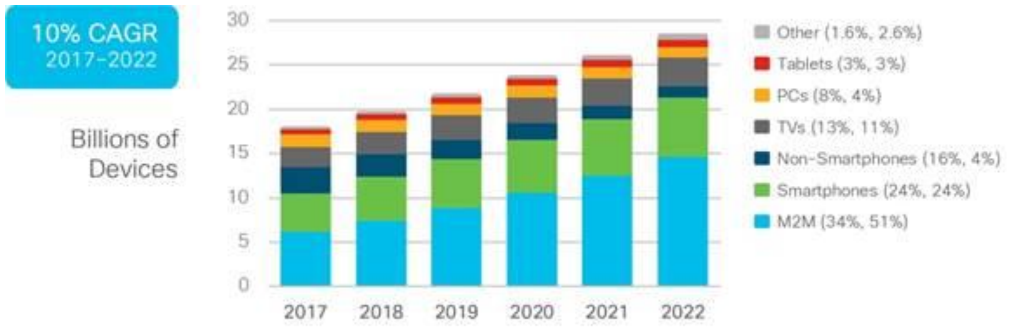
What we do, and who we are

A glimpse of what we are talking about



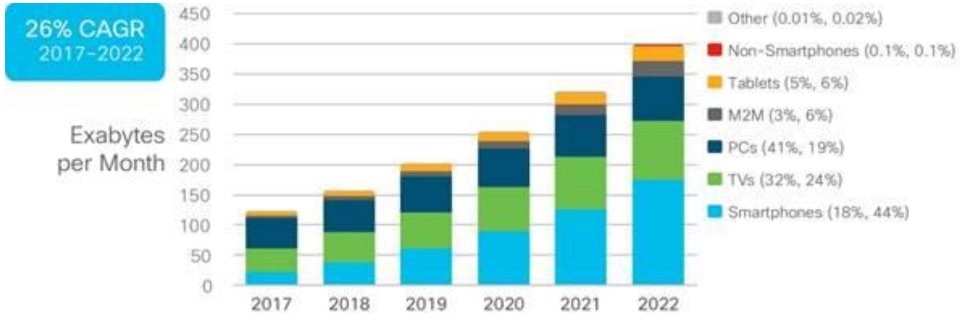
- **Internet users and devices/connections**
  - Number connected devices > 3X global population by 2022
  - Mobile data traffic will increase 7X between 2017 and 2022
- **Global application trends**
  - IP video traffic 82% all IP by 2022
  - Virtual Reality (VR) and Augmented Reality (AR) traffic will increase 12X
  - Internet gaming traffic up 9X
- **Traffic Projections**
  - Annual IP traffic will reach 4.8 ZB per year by 2022

A glimpse of what we are talking about



* Figures (n) refer to 2017, 2022 device share

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022

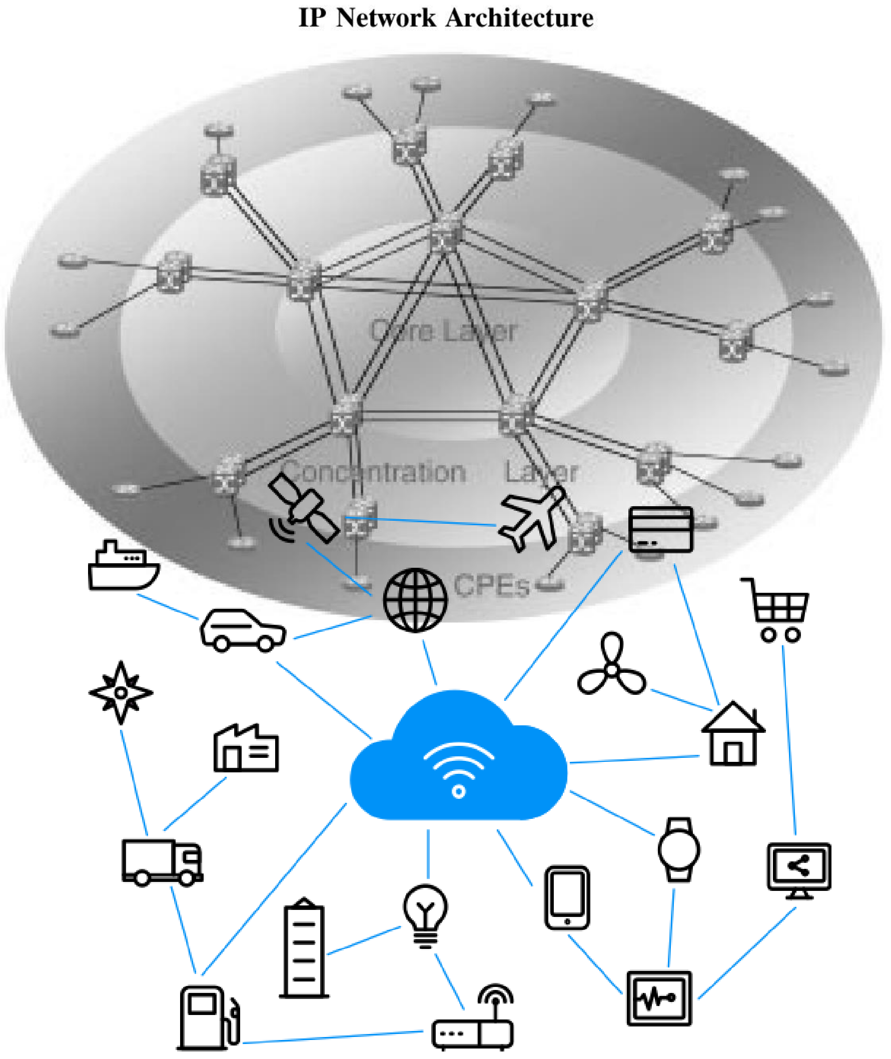


* Figures (n) refer to 2017, 2022 traffic share

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022

- **Internet users and devices/connections**
  - Number connected devices > 3X global population by 2022
  - Mobile data traffic will increase 7X between 2017 and 2022
- **Global application trends**
  - IP video traffic 82% all IP by 2022
  - Virtual Reality (VR) and Augmented Reality (AR) traffic will increase 12X
  - Internet gaming traffic up 9X
- **Traffic Projections**
  - Annual IP traffic will reach 4.8 ZB per year by 2022

A glimpse of what we are talking about



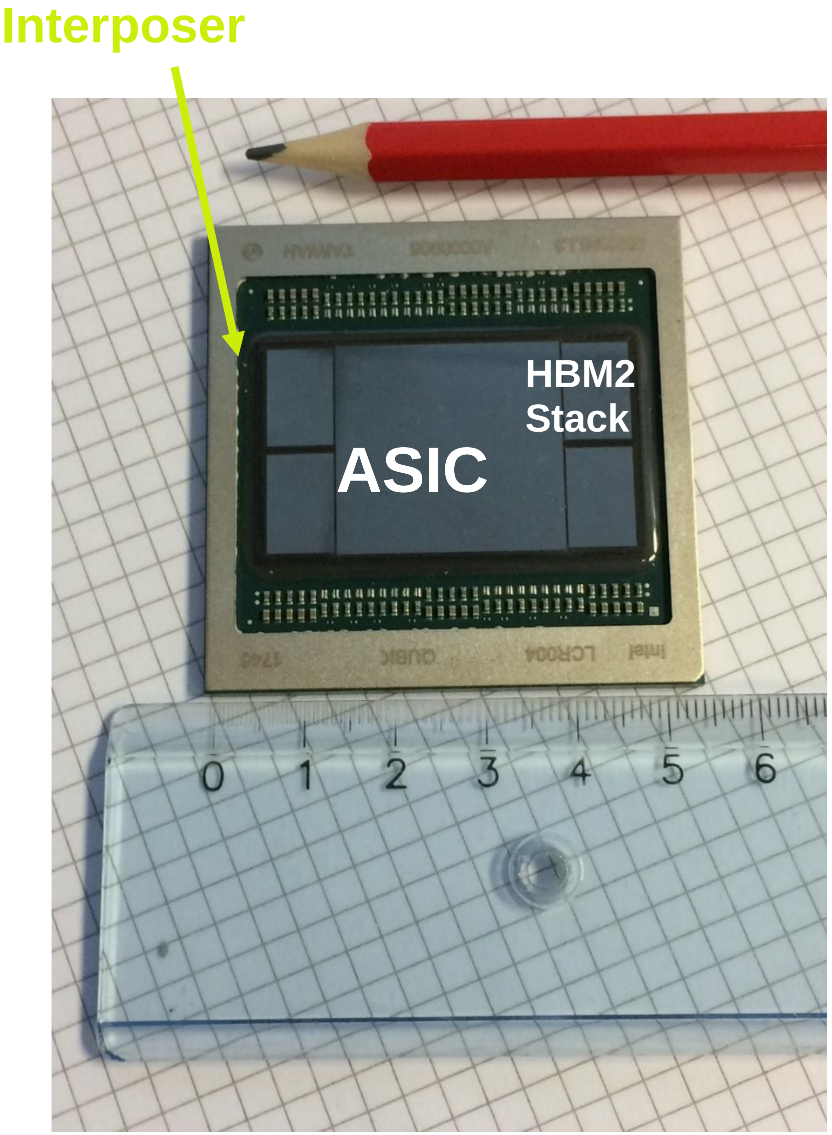
- **Global 5G mobile highlights**
  - 5G devices and connections will be over 3% of global mobile devices and connections by 2022.
  - Nearly 12% mobile traffic will be on 5G cellular connectivity by 2022
- **Networks and Data Centers are requiring a systematic upgrade of equipment**
  - Next Generation Servers, Switches & Routers
  - Next Generation (5G enabled) wireless infrastructure
  - Next Generation Silicon Technologies and Chips/ICs
- ***Key technologies enabling Next Generation Chips/ICs***
  - **Deep Submicron Technologies 7nm and 5nm FinFet**
  - **High Speed Transceivers/SerDes**
  - **Memories Management Interfaces**
  - **AI Building blocks**

A glimpse of what we are talking about

Power Efficiency is a MUST

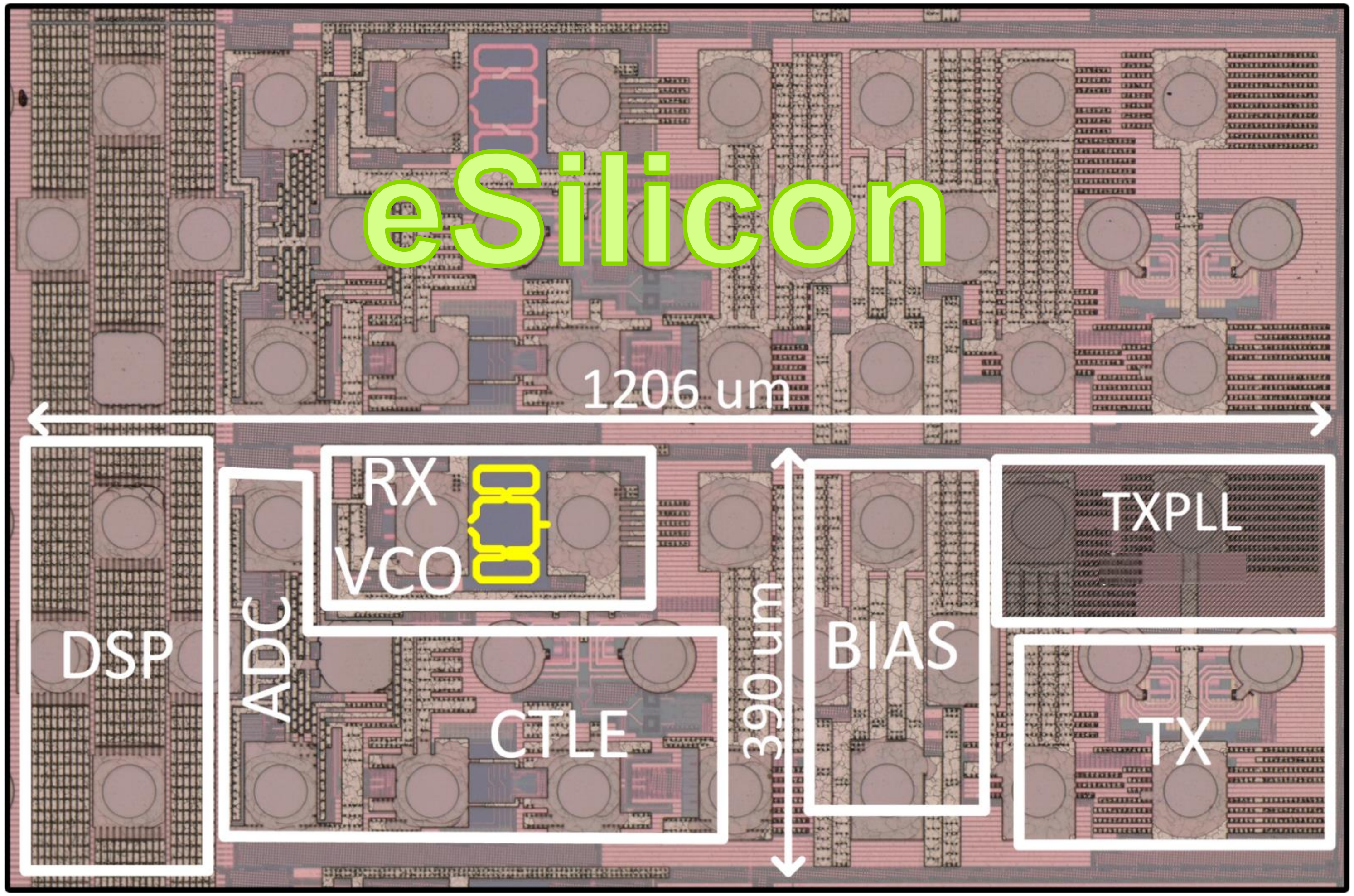
With a per-rack power consumption around **20 kW**, we can save about **1 kW** per rack with a *high efficiency SerDes*

How a typical Networking ASIC looks like



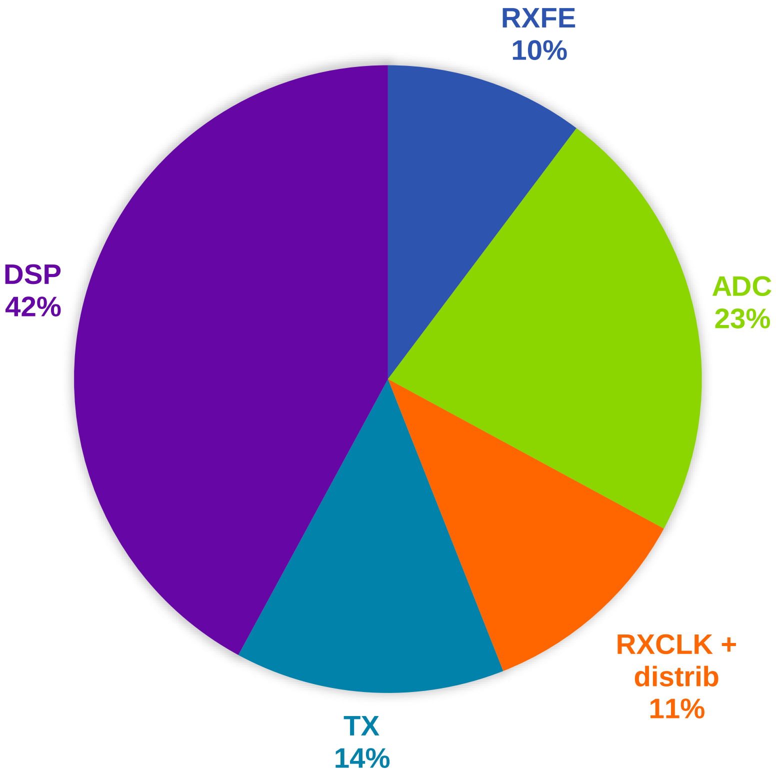
- 150M gates
- 900Mb embedded SRAM
- FCBGA with silicon interposer (2.5d)
- 4 instances of 4-high HBM2 stack (total 1Tbit)
- 128/256 lanes multi-protocol SerDes, 1 to 112Gb
- eSilicon differentiating IP:
  - **HBM Gen2 PHY**
  - Memory compilers
  - Specialty I/O
  - Custom memory instances optimized for area/power/performance
  - TCAM (supports 1.2G search/second)
  - **56G/112G High Speed SerDes**

[ISSCC2019] Our 56Gb/s Die Photo and Power Breakdown



data_lane area = 0.47mm^2

Total power = 234mW @56Gb/s on LR channel



Introducing eSilicon

What we do as a worldwide Company

eSilicon provides advanced Application Specific Integrated Circuits (ASIC)
to global high tech companies in the field of
High-Performance Computing, Networking,
Artificial Intelligence (AI) & 5G infrastructure markets

What we do in our Pavia Design Center

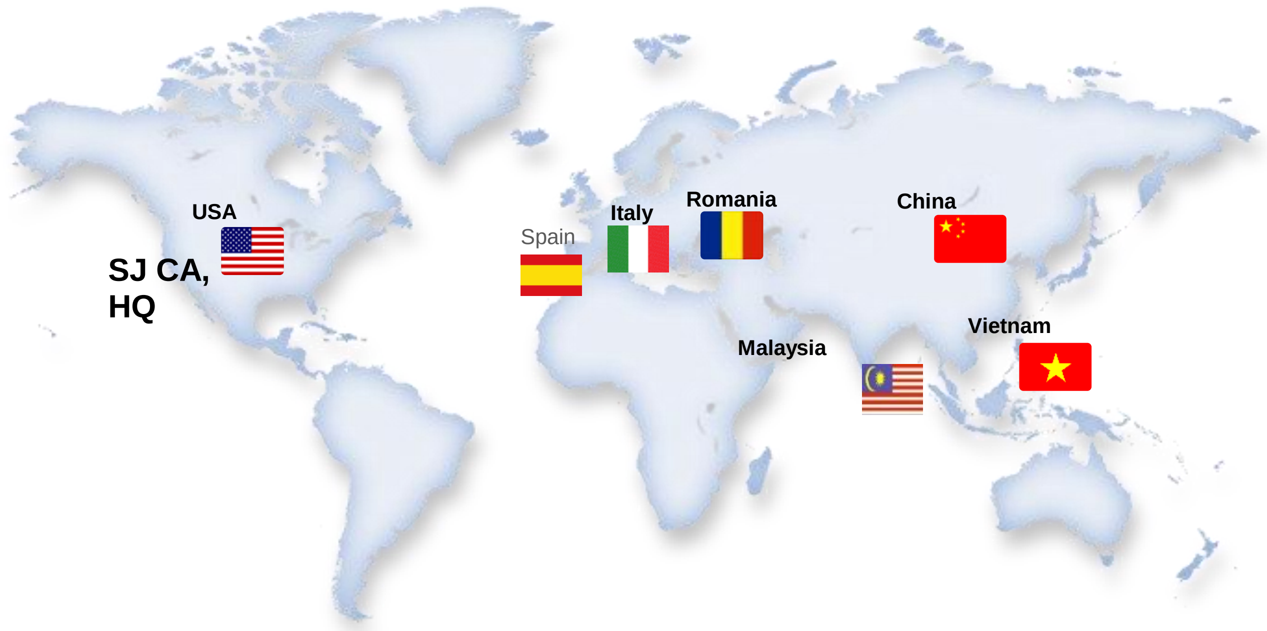
eSilicon Italy develops state of the art transceivers for

56G/112G Serial links and HBM/I interfaces in FinFet technology

from architecture definition and circuit design to lab validation

World-Class Engineering Talent

Global footprint



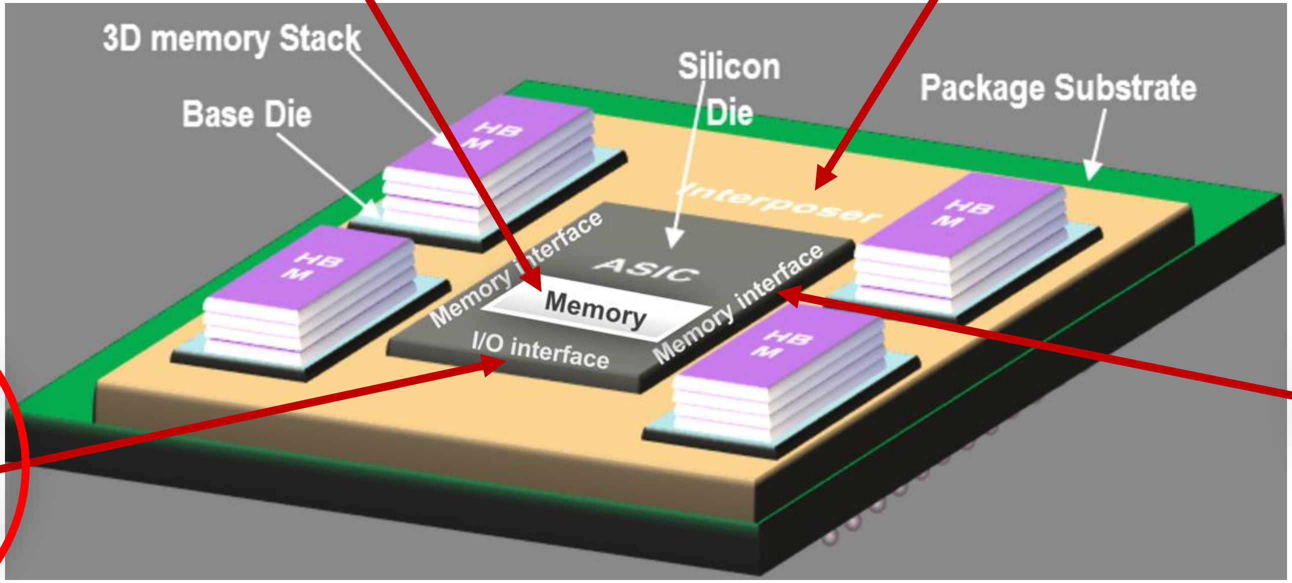
Founded in 2000
600 employee world wide, 77% R&D
300 Tape out done
Half a billion unit shipped
Pavia team was acquired in June 2017

Differentiating & Proprietary circuits for FinFET ASICs

Platform of differentiating circuits that enables system performance

Comprehensive power optimized memories, including TCAMs, compiled or custom, and unique memory synthesis

Interposer design, 2.5D integration & advanced packaging design



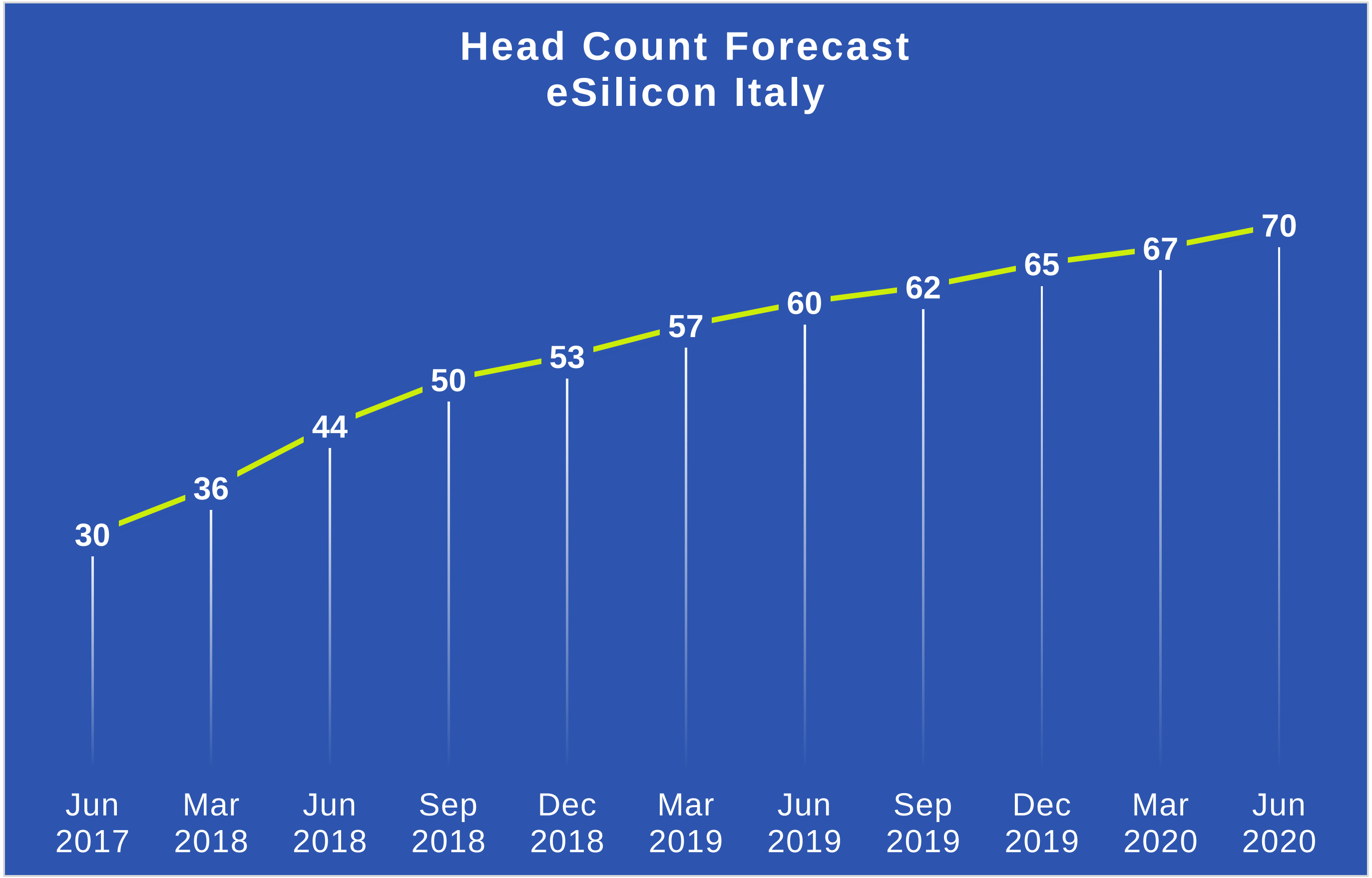
56G long-reach SerDes; 112G in development

HBM/HBI PHY

Typical advanced data center ASIC

Who we are and how many we want to be

- 30+ engineers in Mid 2017
  - Acquisition of a Marvell Team
- We are now about 55
  - 55% has a Ph.D. in Electronics
- **Technical Quality is our Focus**
  - 33 Patents issued
  - 10 scientific publications
  - Most recent, ISSCC 2019
- We are looking for:
  - **Analog and Digital Designers**
  - **System Architects**
  - **Firmware and Software Engineers**
  - **Lab and Field Application Engineers**



Opportunities [or, what it is like in Pavia]

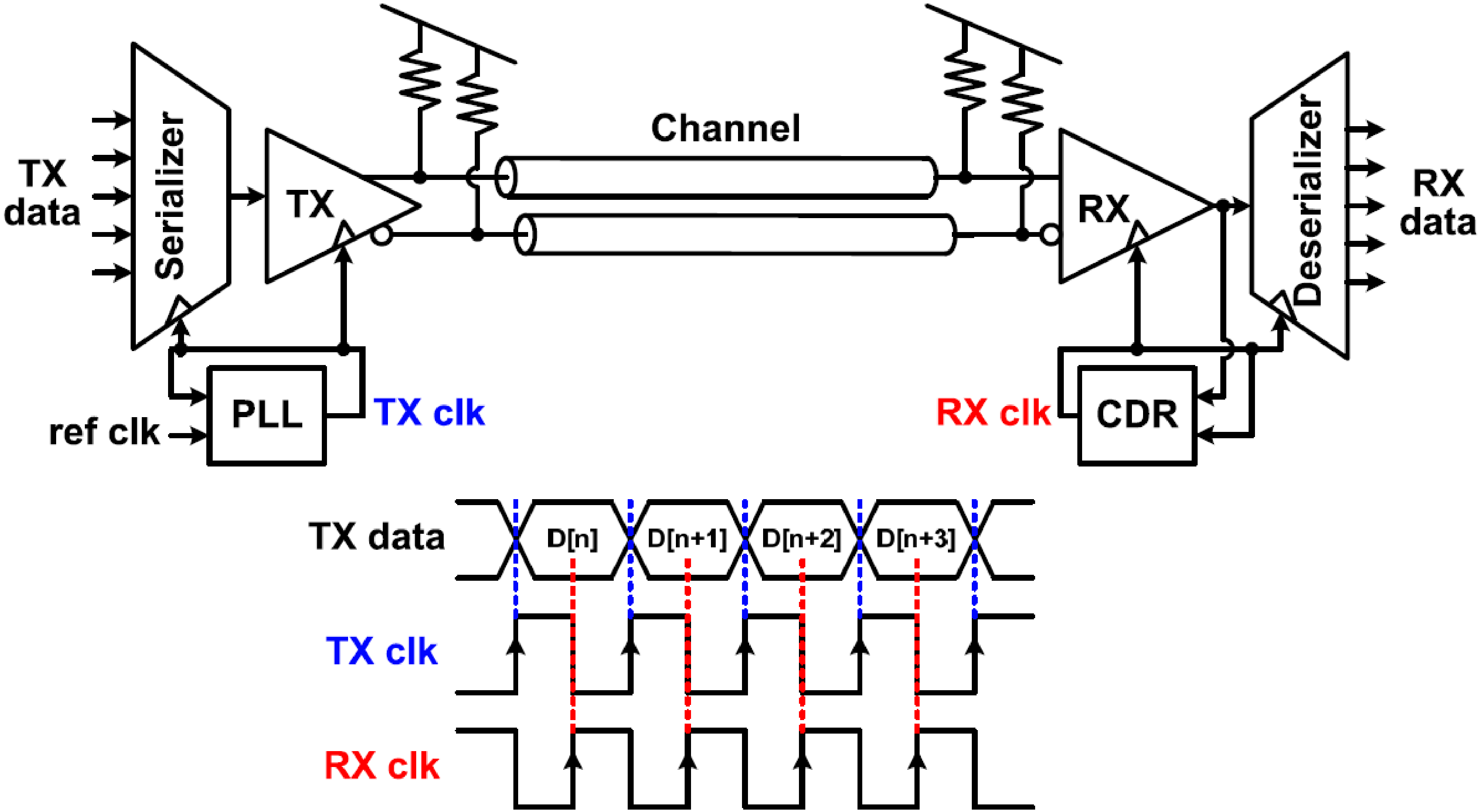
- We have a good degree of freedom from the US headquarters
  - we can run advanced research local programs:
- We value and invest on new employees
  - **Embed** young engineers into experienced and dynamic teams
  - Organize **Training** programs, e.g. FAST Academy (Finfet Analog Skill Training)
  - Work within an **International** and **Multicultural** Environment
- Cooperate with **Universities**:
  - Several programs with Italian Universities and Research Institutes, and growing
- We offer many **opportunities**:
  - https://www.esilicon.com/company/careers/
- **Internship** programs

Introduction to DSP based Serial Links

Contents

- Introduction to Serial Links
- The Channel
- The Transmitter
- Clock Recovery
- The Receiver
- Equalization with DSP
- Microcontroller and FirmWare
- The Overall System

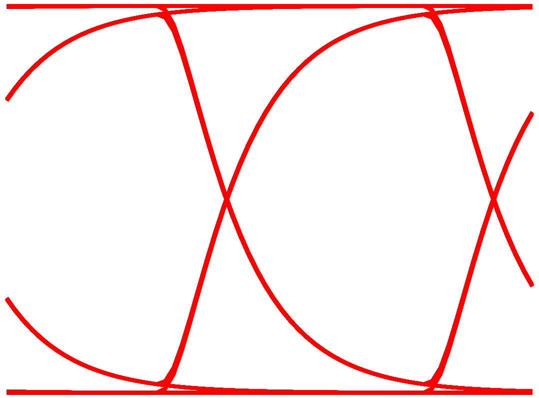
What is a Serial Link?

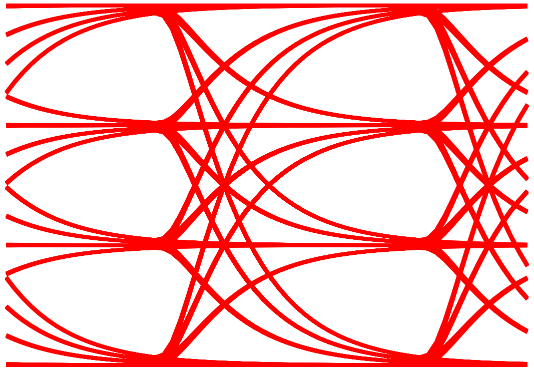


* from S. Palermo
ECEN 720 course

Modulation

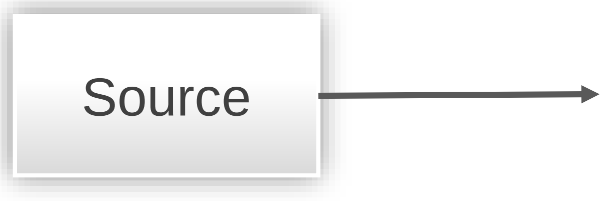
- Objective:
  - Transmit the highest possible amount of data on the physical medium
    - Symbol and Baudrate
  - Minimize power and area
- Modulation of choice is very simple
  - Pulse Amplitude Modulation, either
    - 2 levels – PAM2, or NRZ
    - 4 levels – PAM4
  - Timing is extracted from data
- What are the limitations?





Modulation and SNR

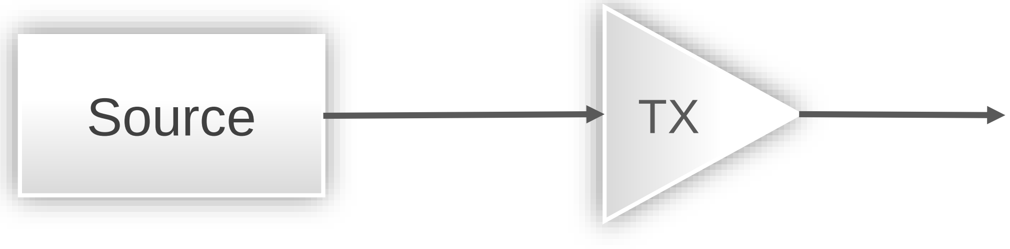
$$x_d(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k] \cdot \delta\,(t - kT)$$



- We can define an SNR
  - Suppose we sample at impulse peak
  - Data and noise mutually uncorrelated

Modulation and SNR

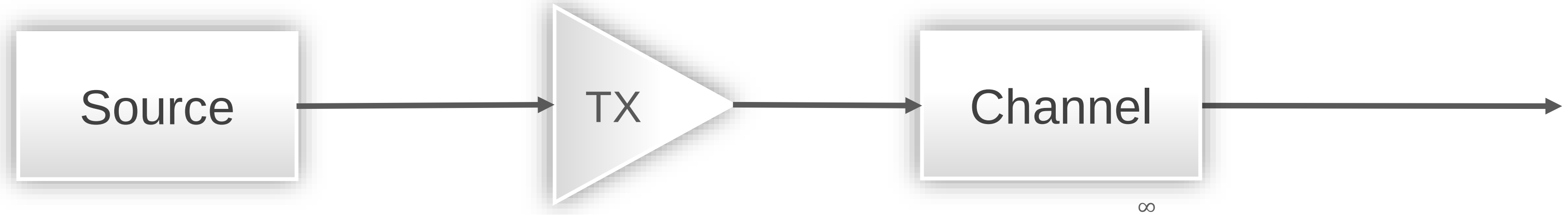
$$x_d(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k] \cdot \delta\,(t - kT) \qquad x_{TX}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k] \cdot h_{TX}\,(t - kT)$$



- We can define an SNR
  - Suppose we sample at impulse peak
  - Data and noise mutually uncorrelated

Modulation and SNR

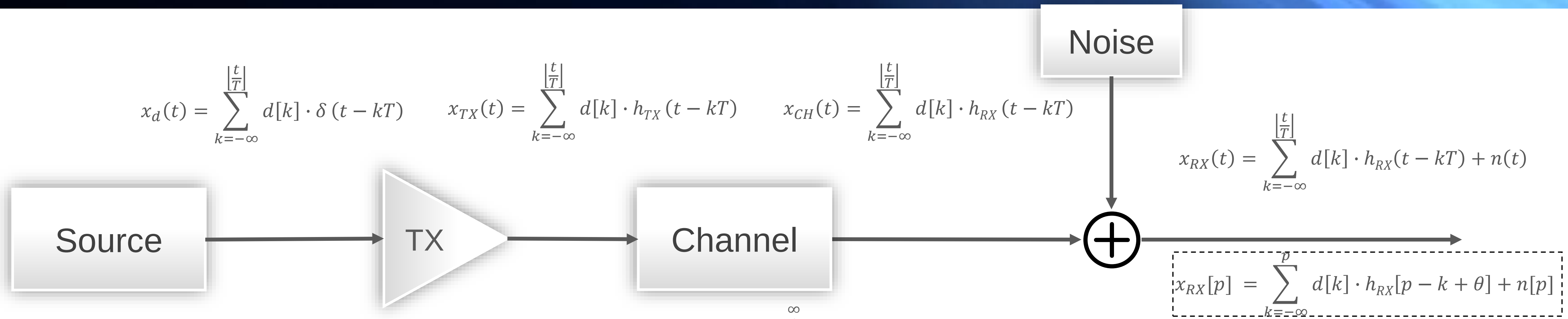
$$x_d(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k] \cdot \delta\,(t - kT) \qquad x_{TX}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k] \cdot h_{TX}\,(t - kT) \qquad x_{CH}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k] \cdot h_{RX}\,(t - kT)$$



$$h_{RX}(t) = (h_{TX} \circ h_{CHNL})(t) = \int_{-\infty}^{\infty} h_{TX}(\tau) h_{CHNL}(t - \tau) d\tau$$

- We can define an SNR
  - Suppose we sample at impulse peak
  - Data and noise mutually uncorrelated

Modulation and SNR



$$x_d(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot\delta\,(t-kT)$$

$$x_{TX}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot h_{TX}\,(t-kT)$$

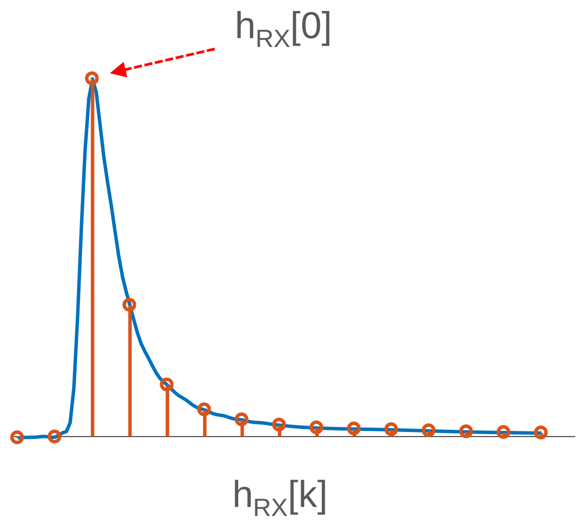
$$x_{CH}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot h_{RX}\,(t-kT)$$

$$x_{RX}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot h_{RX}(t-kT)+n(t)$$

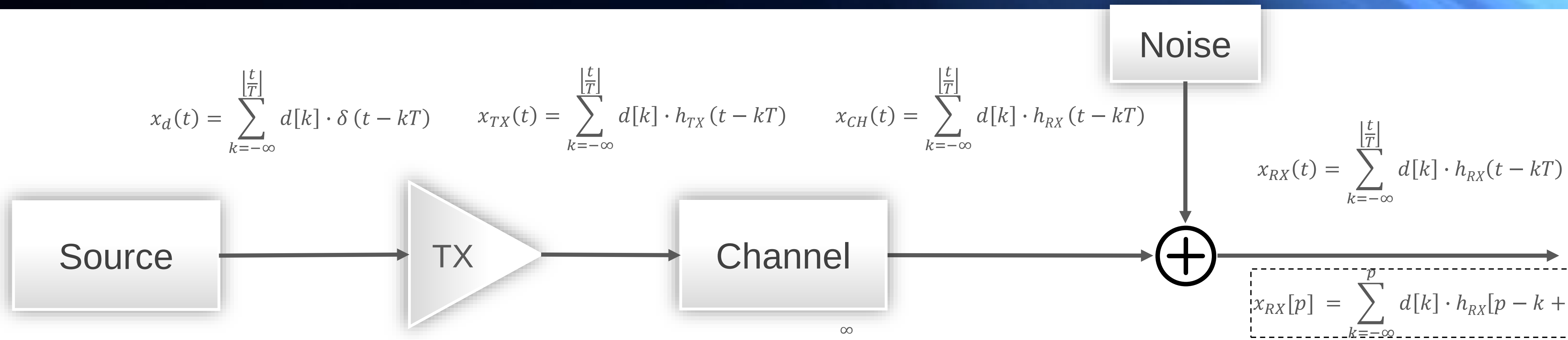
$$x_{RX}[p] = \sum_{k=-\infty}^{p} d[k]\cdot h_{RX}[p-k+\theta]+n[p]$$

$$h_{RX}(t) = (h_{TX}\circ h_{CHNL})(t) = \int_{-\infty}^{\infty} h_{TX}(\tau)h_{CHNL}(t-\tau)d\tau$$

- We can define an SNR
  - Suppose we sample at impulse peak
  - Data and noise mutually uncorrelated



Modulation and SNR



$$x_d(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot\delta\,(t-kT)$$

$$x_{TX}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot h_{TX}\,(t-kT)$$

$$x_{CH}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot h_{RX}\,(t-kT)$$

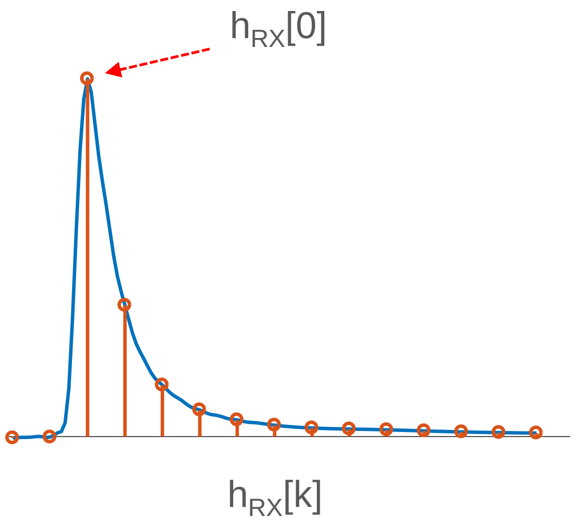
$$x_{RX}(t) = \sum_{k=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot h_{RX}(t-kT)+n(t)$$

$$\boxed{x_{RX}[p] = \sum_{k=-\infty}^{p} d[k]\cdot h_{RX}[p-k+\theta]+n[p]}$$

$$h_{RX}(t) = (h_{TX}\circ h_{CHNL})(t) = \int_{-\infty}^{\infty} h_{TX}(\tau)h_{CHNL}(t-\tau)d\tau$$

- We can define an SNR
  - Suppose we sample at impulse peak
  - Data and noise mutually uncorrelated

$$x_{RX,n}[p] = \underbrace{d[p]\cdot h_{RX}[0]}_{\text{Signal}} + \underbrace{\sum_{k=-\infty}^{p-1} d[k]\cdot h_{RX}[p-k]}_{\text{ISI}} + \underbrace{n[p]}_{\text{noise}}$$



Probability of Error

- Slicer operation

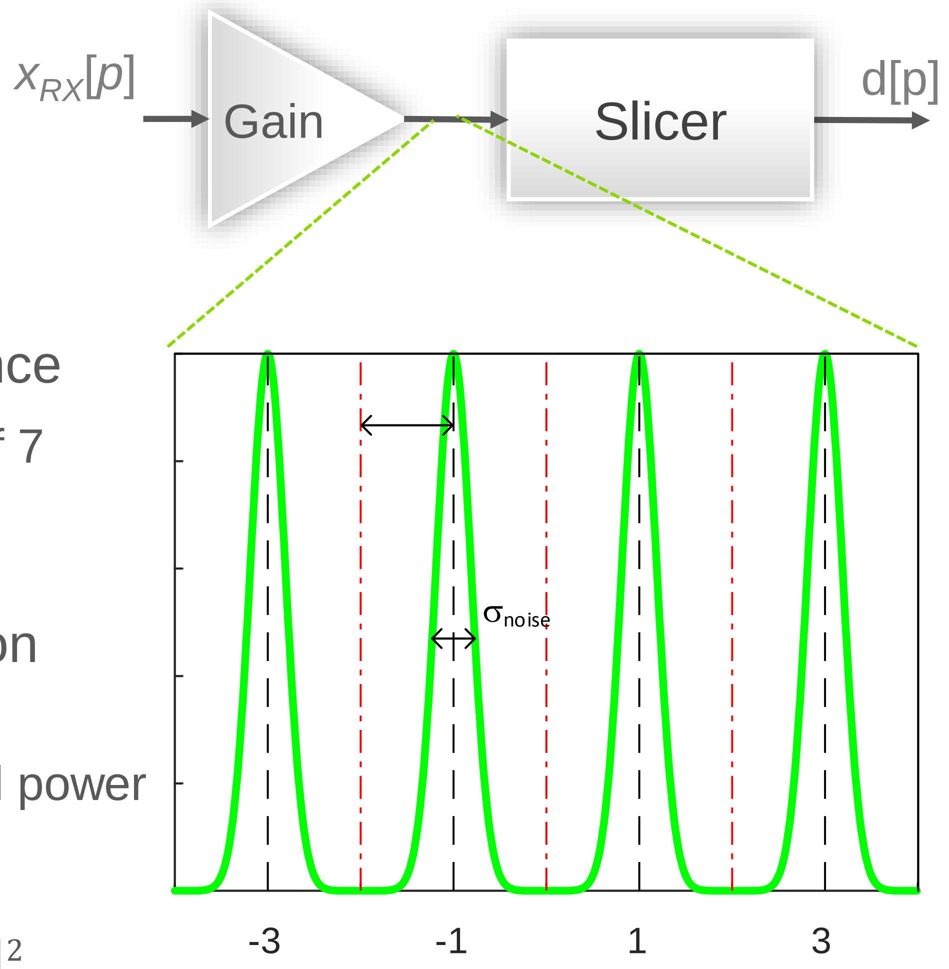
  $x_{RX}[p]$ → $d[p]$ if ($n[p]$+ISI[p]) < ½ symbol distance

- Probability of error = P($n[p]$+ISI[p]) > ½ symbol distance

- In case of PAM, we scale gain to keep the same symbol distance
  - a PAM4 signal x_{RX} requires a gain of 3 wrt to NRZ, PAM8 a gain of 7
    - Assume the same swing on x_{RX}
    - Symbols get mapped onto {…, -3, -1, 1, 3, …}

- We want to understand how ISI and noise scale with modulation
  - Noise power simply scales by gain^2
  - ISI scales by gain^2 too, but is also depends on the average symbol power

$$ISI = E\left\{\left(\sum_{k=-\infty}^{p-1} d[k]\cdot h_{RX}[p-k]\right)^2\right\} = E\{d[k]^2\}\sum_{k=-\infty}^{p-1} h_{RX}[p-k]^2$$



Probability of Error

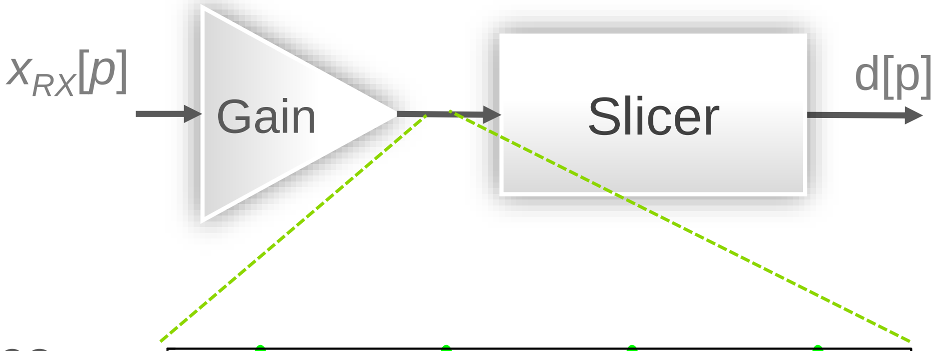
- Slicer operation

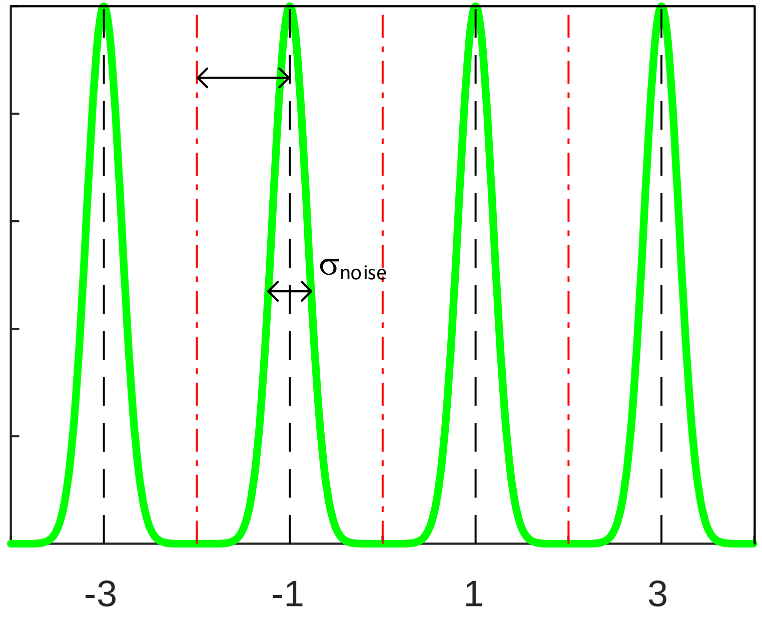
  $x_{RX}[p]$ → $d[p]$ if ($n[p]$+ISI[p]) < ½ symbol distance

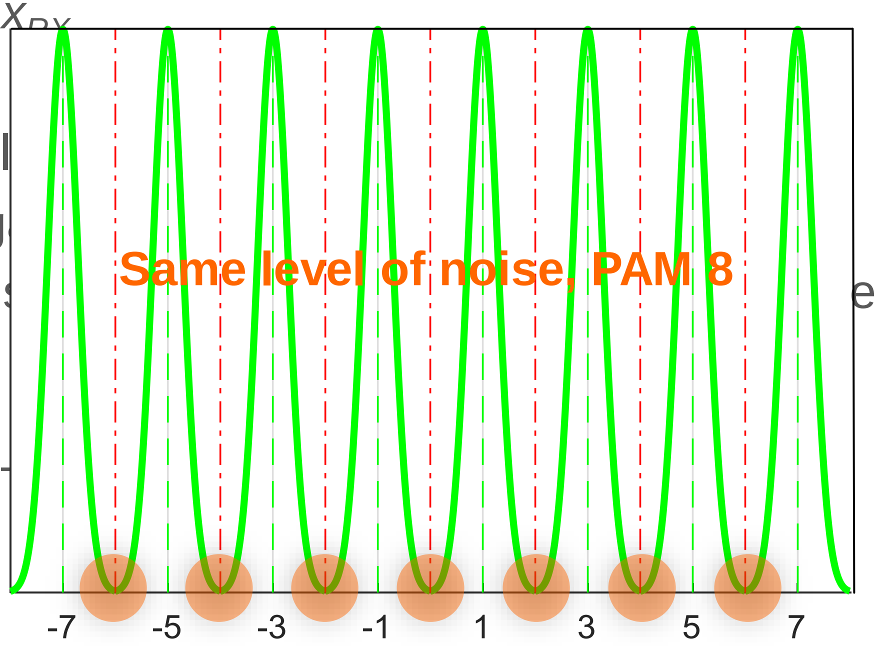
- Probability of error = P($n[p]$+ISI[p]) > ½ symbol distance
- In case of PAM, we scale gain to keep the same symbol distance
  - a PAM4 signal x_{RX} requires a gain of 3 wrt to NRZ, PAM8 a gain of 7
    - Assume the same swing on x_{RX}
    - Symbols get mapped onto {.
- We want to understand how ISI
  - Noise power simply scales by g
  - ISI scales by gain^2 too, but is al

er

$$ISI = E\left\{\left(\sum_{k=-\infty}^{p-1} d[k]\cdot h_{RX}[p\right.\right.$$






Same level of noise, PAM 8

Modulation Comparison Table

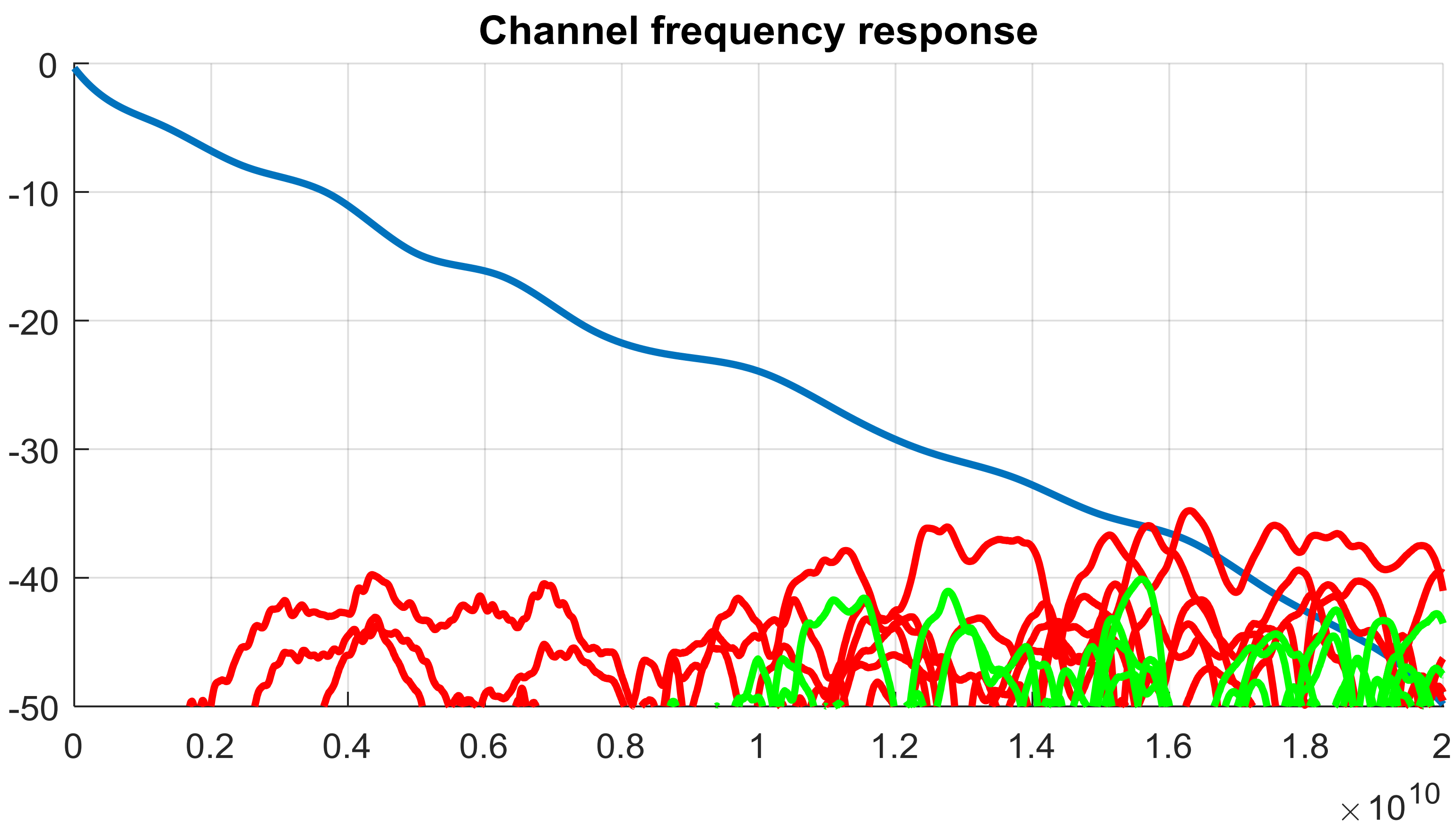
Modulation	Gain	Noise	ISI	Speed
NRZ	1	1	1	1x
PAM4	3	9	5	2x
PAM8	7	49	21	3x

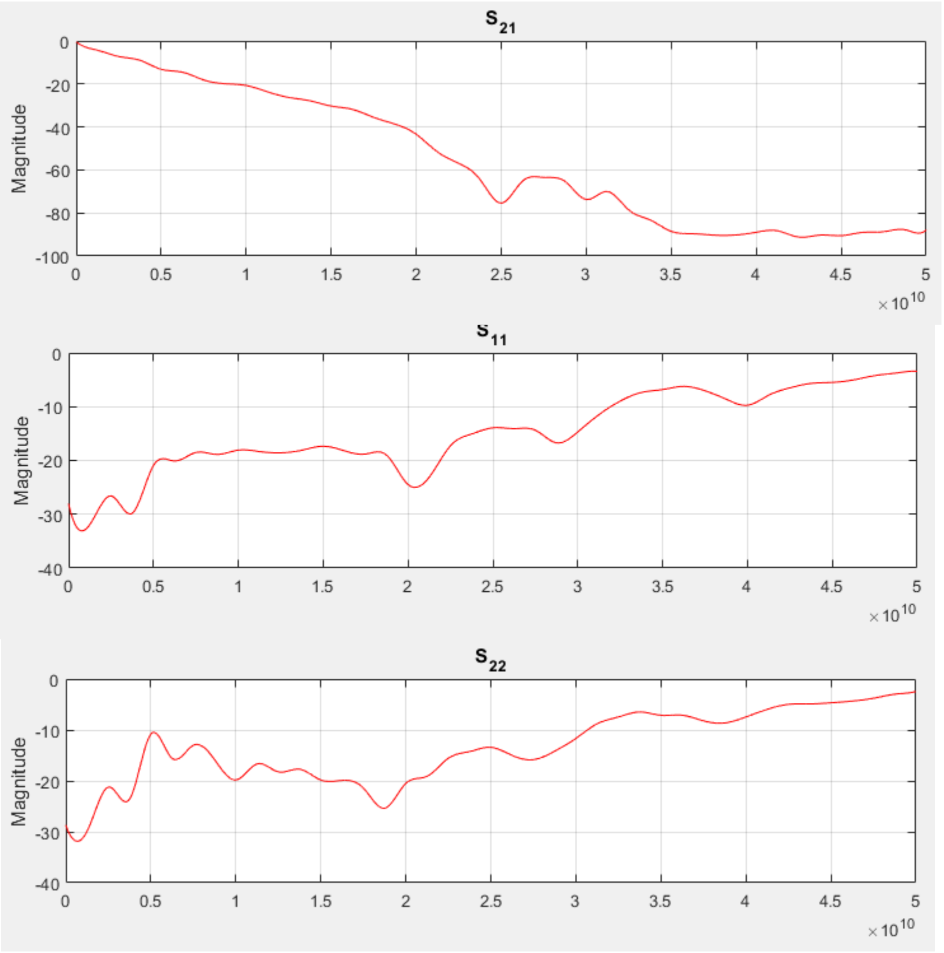
- Relative to NRZ, there is a big penalty involved with higher modulations
  - Considering noise alone, noise power increases by 9.5 dB in PAM4, and by 16.9 dB in PAM8
  - ISI is somewhat better
- Because of this, there is no current effort in increasing modulation alphabet
  - Why did we go to PAM4 though?

The Channel

- With increasing speeds, the whole design has to cope with **insertion loss**
- Channels are indexed with attenuation at the **Nyquist** frequency
  - Current standards address more than 30 dB of channel loss
    - **Package** loss has to be added both on RX and TX to obtain bump-to-bump budget
- Other than raw attenuation, many features affect performance
  - Insertion Loss Deviation – measure of the “roughness” of the channel profile
  - Reflections that can arise at its interface (S11 and S22 parameters)
  - Regular behavior after Nyquist
- Channels are characterized and models with S-parameters
  - A microwave tool
  - Can be composed and transformed to other representations

Channel Example

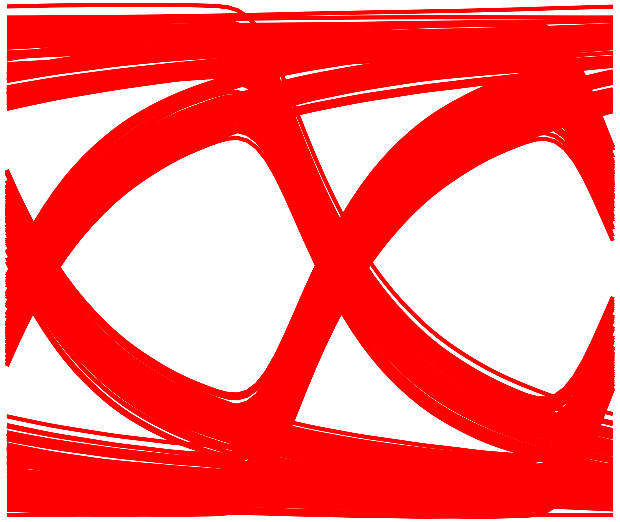




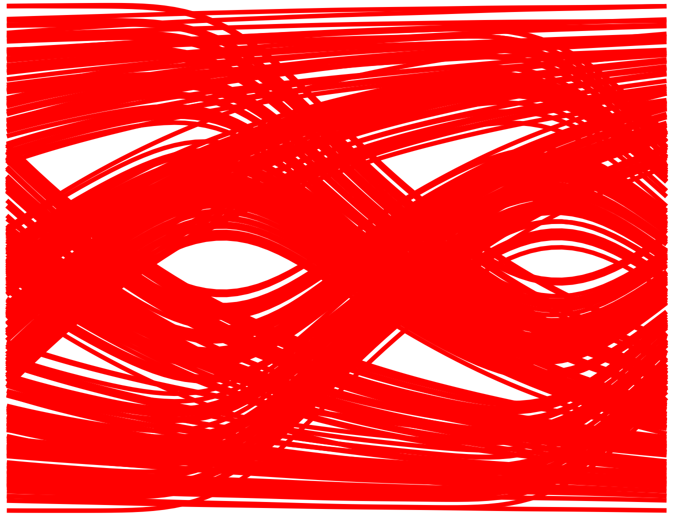
The Channel

- Let’s look at the effect of the attenuation on the eye diagram

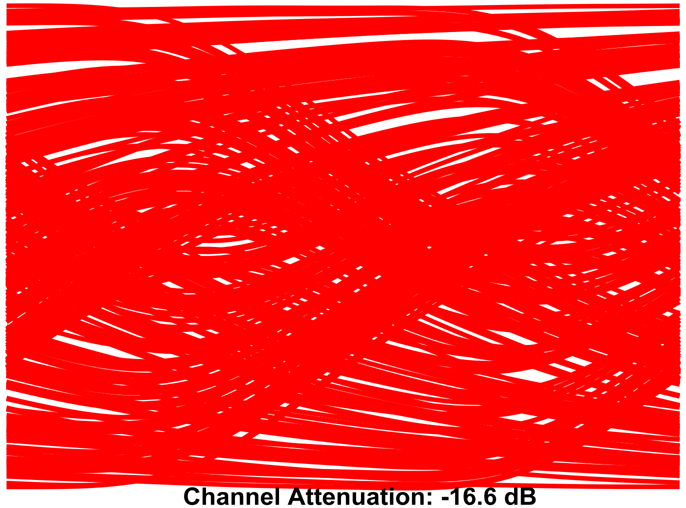
Channel Attenuation: -6.6 dB



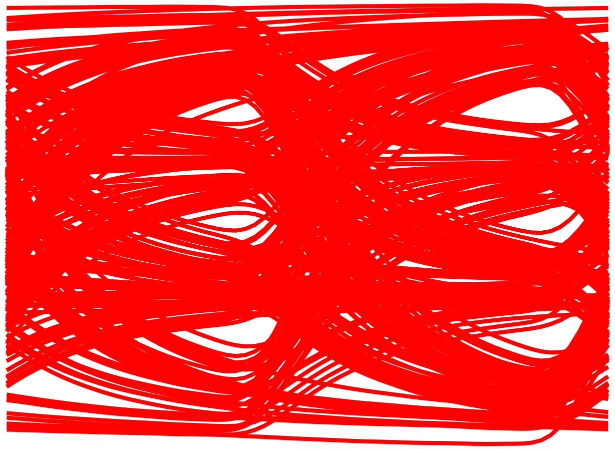
Channel Attenuation: -12.8 dB



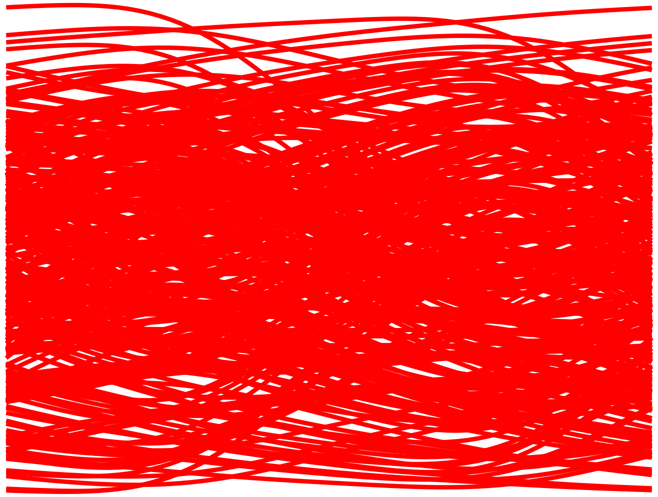
Channel Attenuation: -16.6 dB



Channel Attenuation: -6.6 dB

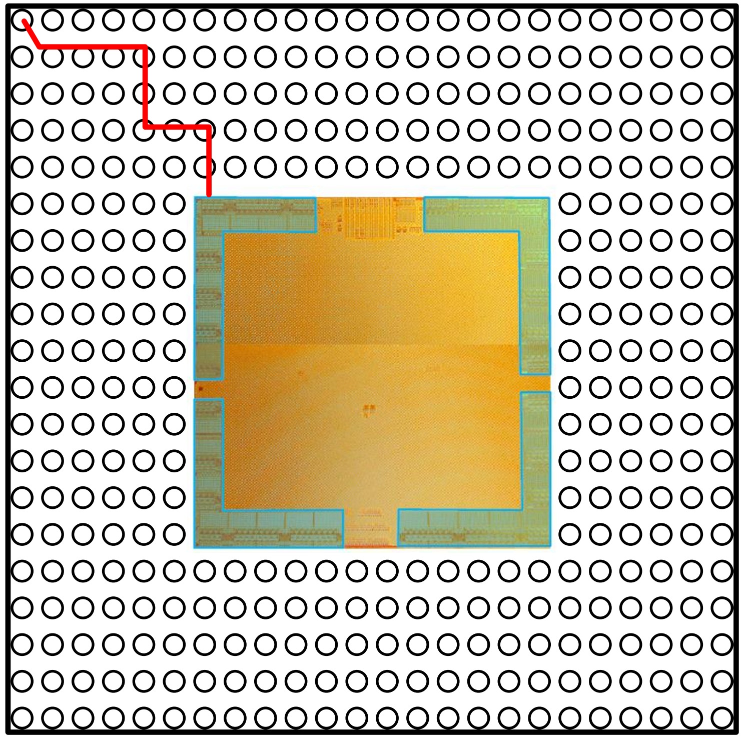


Channel Attenuation: -16.6 dB



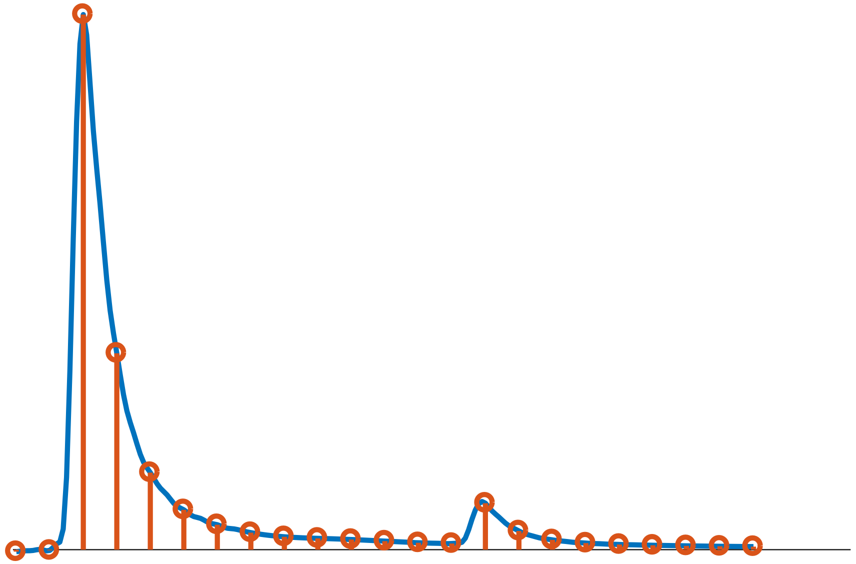
Package and Reflections

- Packages play a non-negligible role in the system budget
  - The impact becomes comparable to the channel @100G
- ASICs for switches are as big as feasible with current manufacturing constraints
- Consider a commercial example
  - 128x25G serdes lanes (3.2 Tb/s I/O)
  - ~30% is I/O (blue portion)
  - The die may be around 400-500 mm^2
  - The package can be 60 or 70 mm per side!
    - The longest serdes escape is ~30mm
    - might be longer in future…



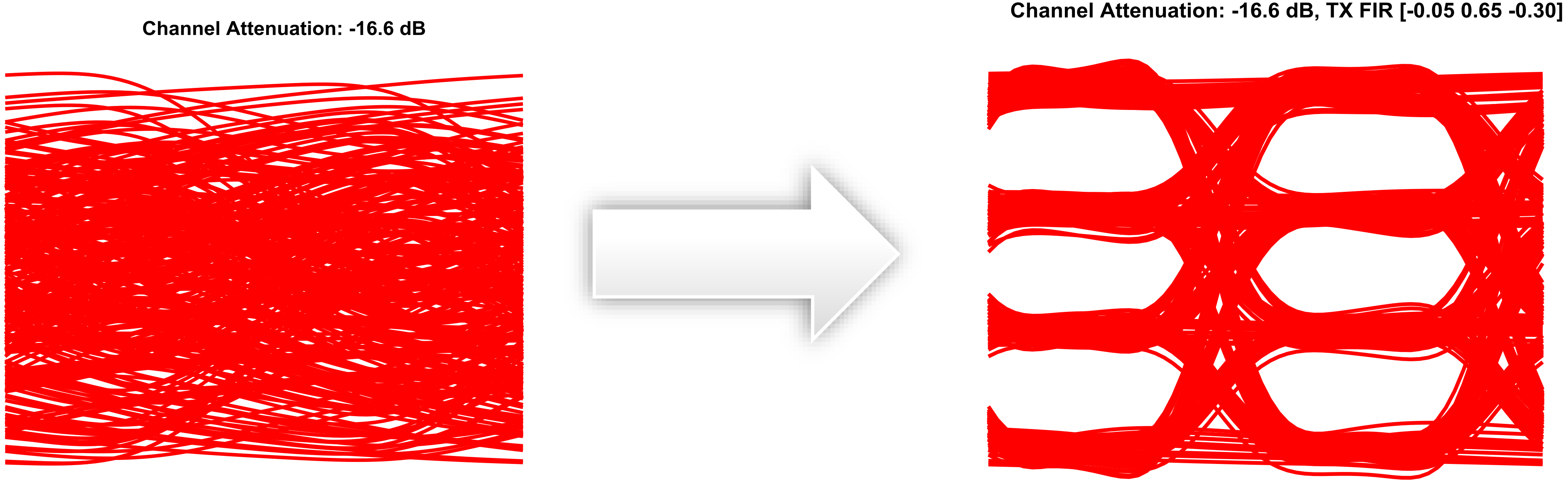
Package and Reflections

- Let’s consider reflections from the package/channel interface
- After how many symbols do we get an effect at the receiver?
  - 30 mm package, considered twice
  - Approximate propagation speed with 0.15 m/ns
  - The time of flight through 60mm → 0.4 ns
- What does this mean?
  - Current standard is 56 Gb/s PAM4
  - Symbol time (UI) is ~36ps
  - Reflection will occur after ~11 symbols
- We may need to design the equalizer to handle this
  - … as well as design the overall system to exploit it! → Firmware



Transmitter Equalization

- The transmitter can be used to shape the transmitted impulse
- Current standards require at least a 3 tap FIR filter
  - Usually indicated as {precursor, cursor, postcursor}
- As the TX amplitude is constrained, the sum of the taps must be 1 (or less)
  - The filter actually de-emphasizes low frequency components

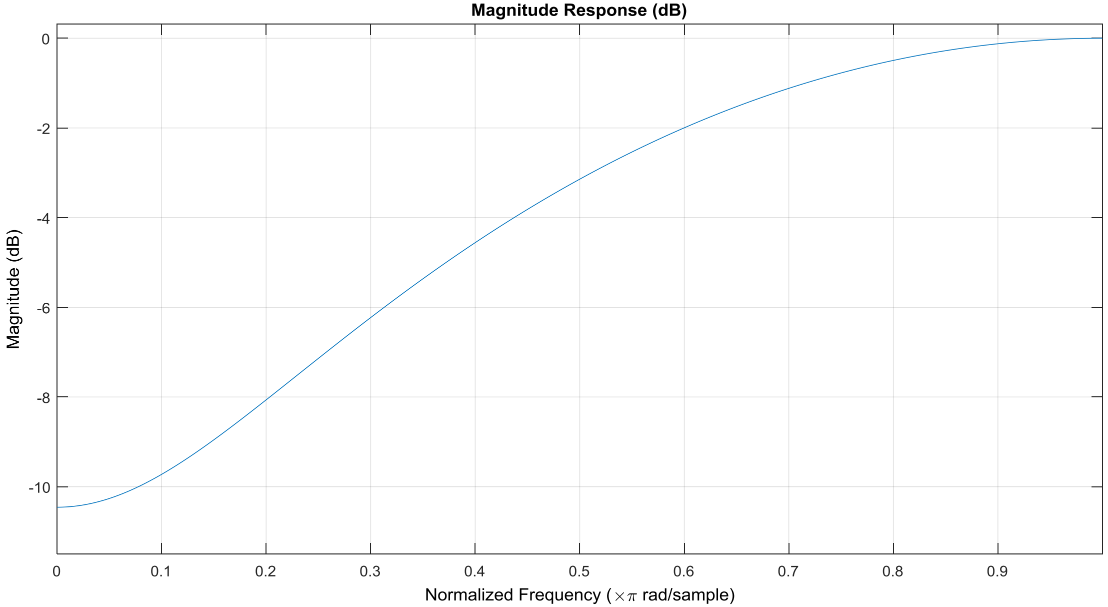


Transmitter Equalization

- In the z-domain, we have

$$\text{TXFIR} = c_{-1} + c_0\, z^{-1} + c_1\, z^{-2}$$

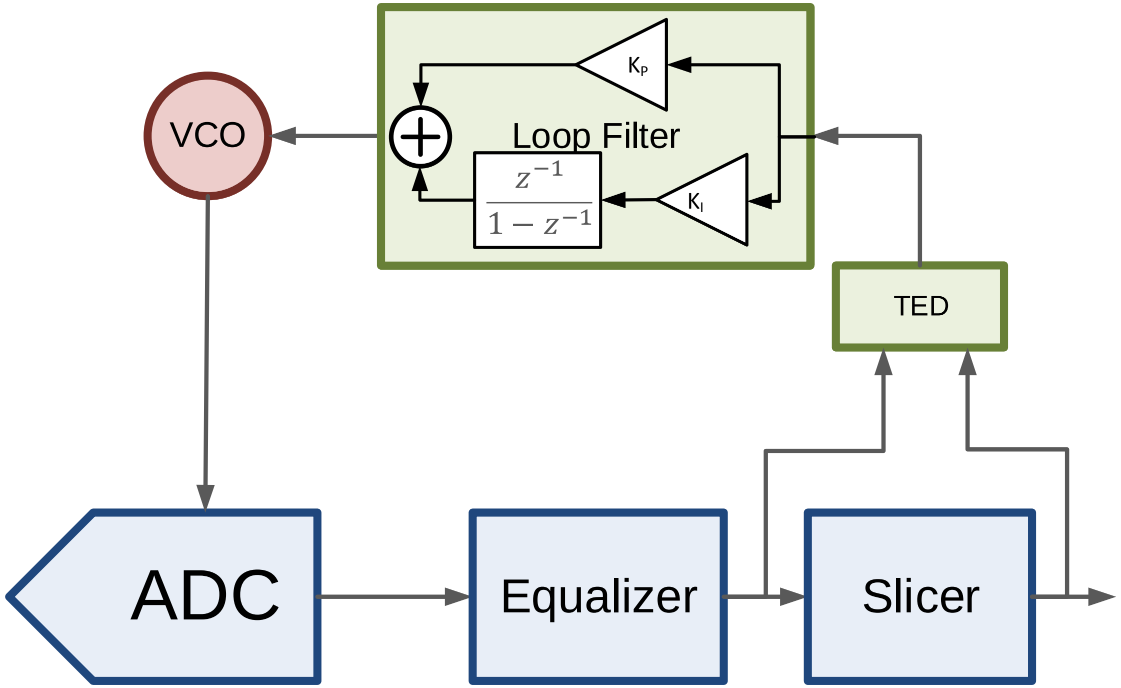
- In the frequency domain, the previous filter looks like



- There is no backchannel provided by the standard to adapt the TX FIR
  - At startup, a training protocol is used to determine the initial values
  - It cannot be changed afterwards…
- At the system level, training is handled jointly with receiver equalization (DSP optimization) and involves FSMs or, better, firmware.
  - Mathematically speaking, we need to solve a non linearly constrained optimization problem

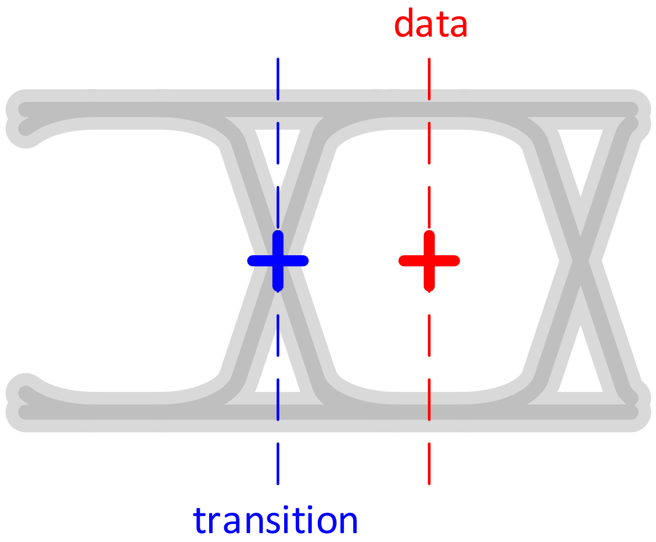
Digital Timing Loop

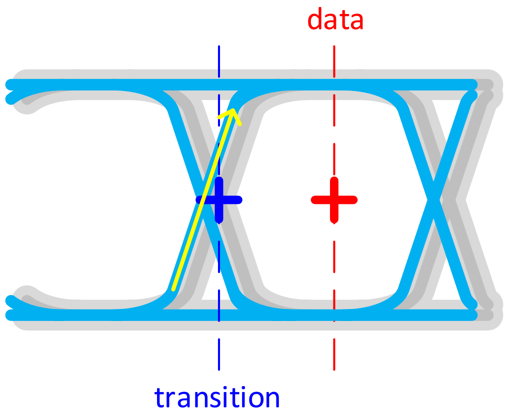
- The TED provides indication of how early/late the sampling time is
  - This is analogous to a PFD in a PLL…
- The Timing Loop is then built in an analogous manner to a normal PLL
- We can build a linear model for the DTL
  - Set its bandwidth
- Standards set constraints on the tracking bandwidth
  - Phase noise on the transmit side will affect the incoming data

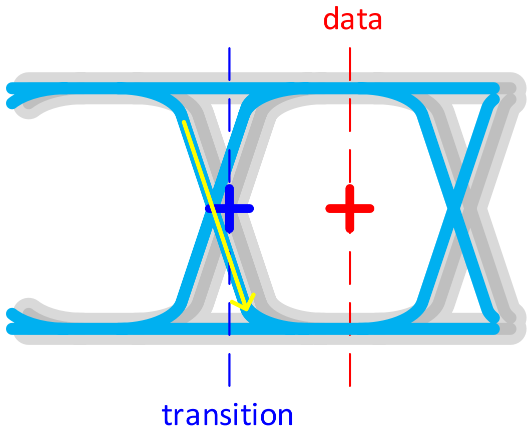


Time Error Detector

- On the receiver, clock has to be extracted from data
- In an analog implementation, a dedicated sampler is provided to sample at the transition
- Its operation can be intuitively grasped
  - If during a -1 → 1 data transition the sample at crossing is positive, we are sampling late
  - If during a 1 → -1 data transition the sample at crossing is positive, we are sampling early







Clock Recovery

- At equilibrium, we are forcing $E\{(d[k]-d[k-1])t[k]\}=0$
- In a digital implementation, we only have access to data points
- A commonly used criterion is the Mueller&Müller Time Error Detector
- $TED[p]=\ \varepsilon[p]\cdot(d[p+1]-d[p-1])$
  - ε[*p*] is the difference between the output and the input of the slicer

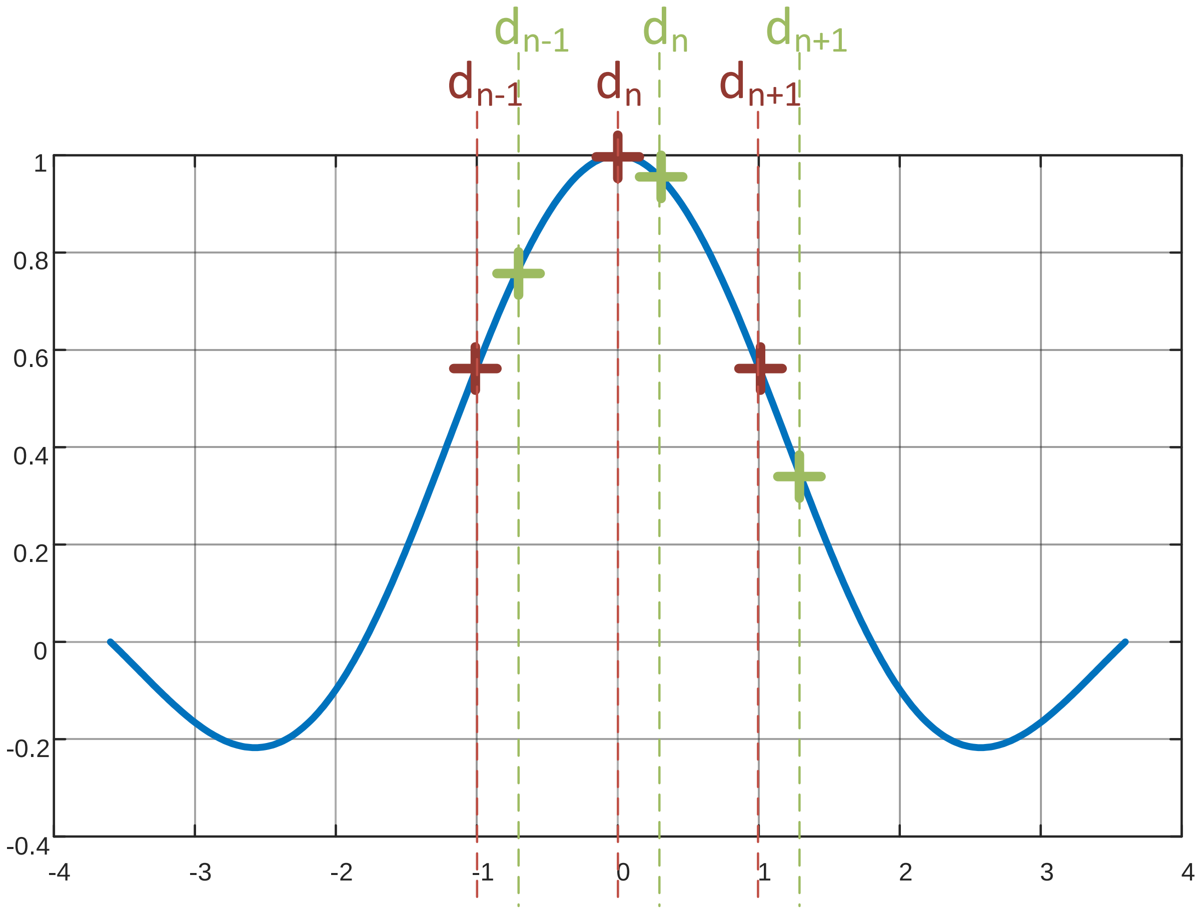
$$\varepsilon[p]=\sum d[k]\cdot h_{RX}[p-k+\theta]-d[p]\cdot h_{RX[0]}=\sum_{k\neq p} d[k]\cdot h_{RX}[p-k+\theta]$$

- It can be shown that, given the impulse response at the slicer h(t), we have:
  - $E\{d[p+1]\cdot\varepsilon[p]\}=E\left\{d[p+1]\cdot\sum_{k\neq p} d[k]\cdot h_{RX}[p-k+\theta]\right\}=$

    $=\sum_{k\neq p} E\{d[p+1]\cdot d[k]\}h_{RX}[p-k+\theta]=E\{d^2\}\cdot h_{RX}[p-1+\theta]$
  - $E\{\varepsilon[p]\cdot d[p-1]\}=E\{d^2\}\cdot h_{RX}[p+1+\theta]$
- So that the equilibrium point corresponds to *h*[*p*+1] = *h*[*p*-1]

Baud-rate Sampling

- Sampling time is inferred from ISI content of sampled data
- CDR is locked once the ISI content of sample (n-1) is equal to the ISI content of symbol (n+1)
  - If we sample late, or early, the difference between the ISI samples (n-1) and (n+1) will be proportional to the sampling time error
- This is possible because of quantized data d_n
  - An ADC is required

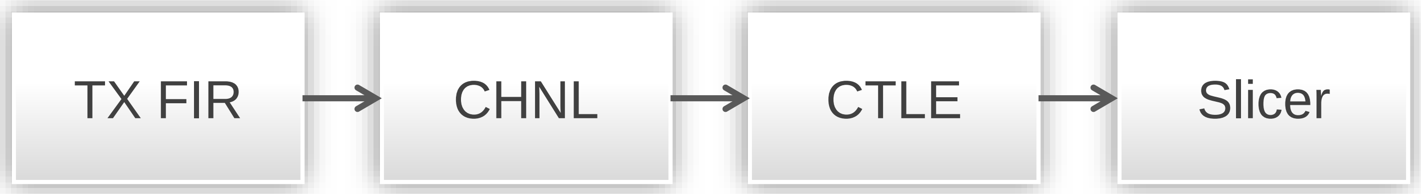


The Receiver

- Target: minimize probability of error in recovered symbols
- Why do errors happen?
  - Misequalization → ISI
  - Noise
  - Timing
- From communication theory, we have a criterion for “zero” ISI at receiver
  - Zero forcing equalizer (really optimal?)
  - The frequency response of the cascade of the channel and the equalizer must be 1
- We can use this as an intuitive criterion to setup the equalization chain

Continuous Time Linear Equalizer

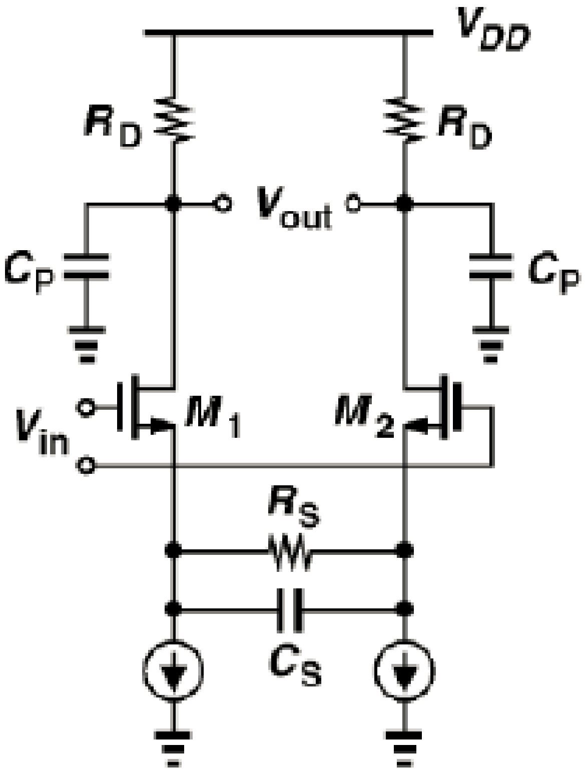
- In a short reach receiver, we can sample right after the CTLE and recover data



- In practice, performance is limited
  - Only a finite set of equalization curves are available
  - Adaptation is a problem
    - Temperature changes
    - Transmitter drifts
    - Aging

Continuous Time Linear Equalizer

- Usually, this is the first block in the Analog Front End (AFE)
- Example: differential pair with RC degeneration
  - At DC, the amplifier is degenerated
  - At high frequency, the capacitor C_S shunts R_S
- We can even tune this circuit, changing
  - R_S – moves the zero, changing peaking
  - C_S – moves both pole and zero
- Maybe limited in bandwidth…

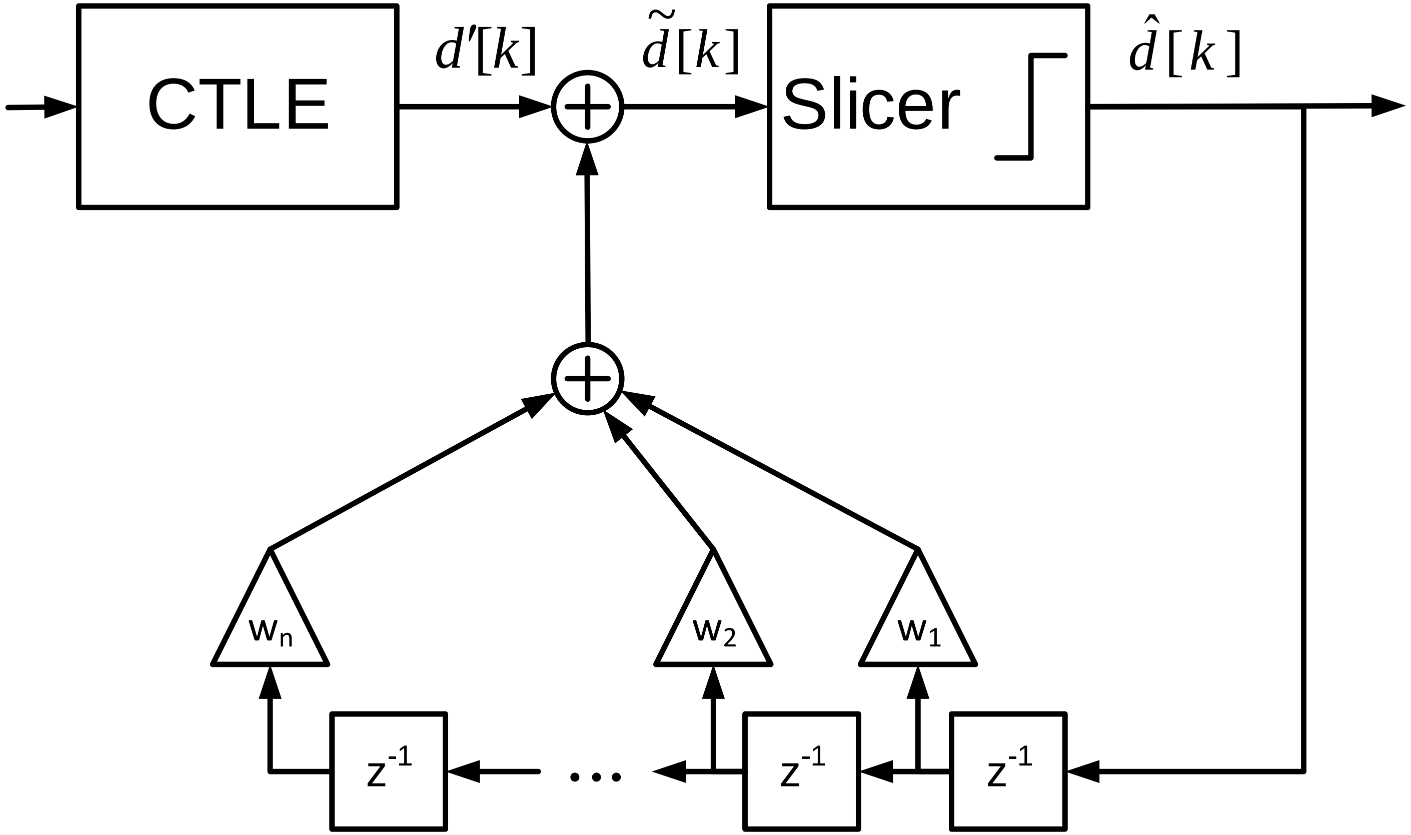


Decision Feedback Equalizer

- The ISI due to past symbols can be removed from incoming data
  - Assuming no errors…
- We have

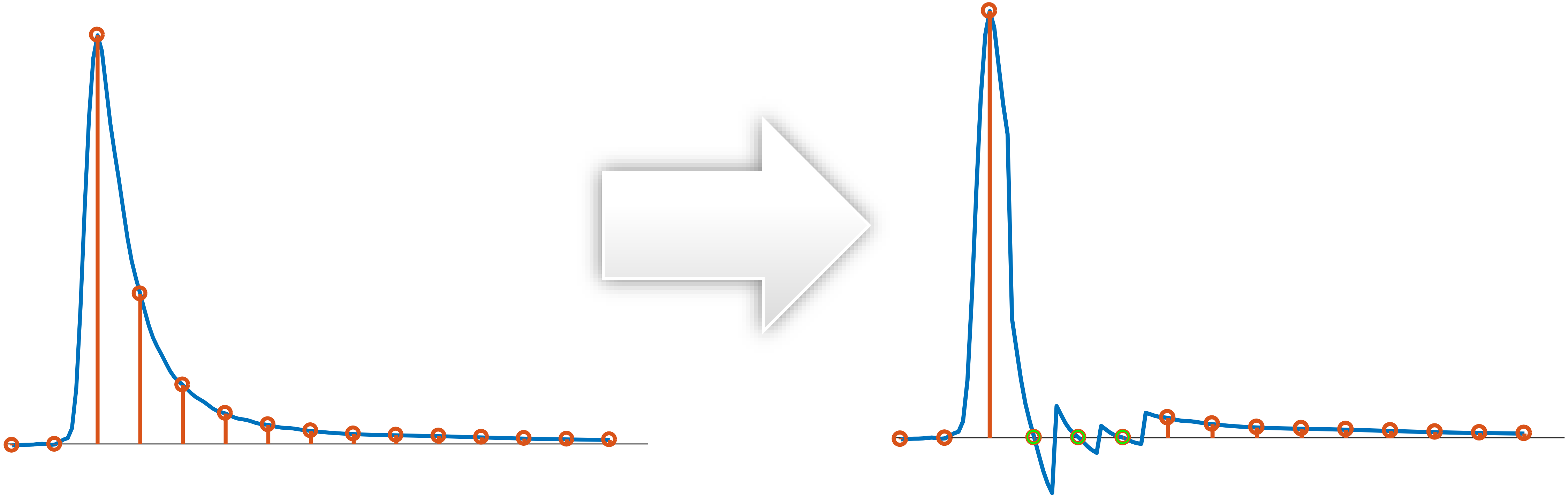
$$d'[k] = \sum_{p=-\infty}^{k} d[p] \cdot h'[k-p]$$

- Optimal weights are exactly the impulse taps
  - If h'[0] is the cursor, then
  - w_1 = h'[1], w_2 = h'[2], …, w_i = h'[i]



Decision Feedback Equalizer

- What happens to the equalized impulse

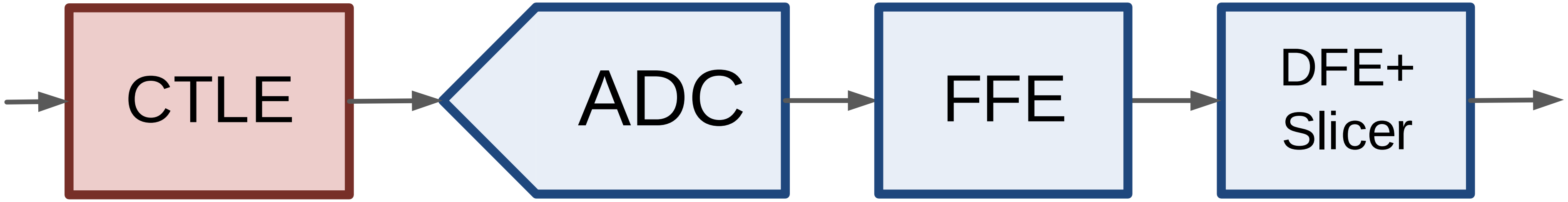


Decision Feedback Equalizer

- We can cancel many taps of the impulse response with long DFEs
  - Boost high frequency without noise emphasis
- However, we have some fundamental limitations:
  - **Causality** – we cannot equalize pre-cursor taps
  - **Speed** – DFE is intrinsically a 1UI loop
- DFE can be implemented in analog receivers
  - It has been the workhorse equalizer for generations of serdes
  - In an analog implementation, the DFE implements a DAC summing its contribution to the input of the slicer
  - Some complication may arise when the timing loop is considered

DSP-based Equalization

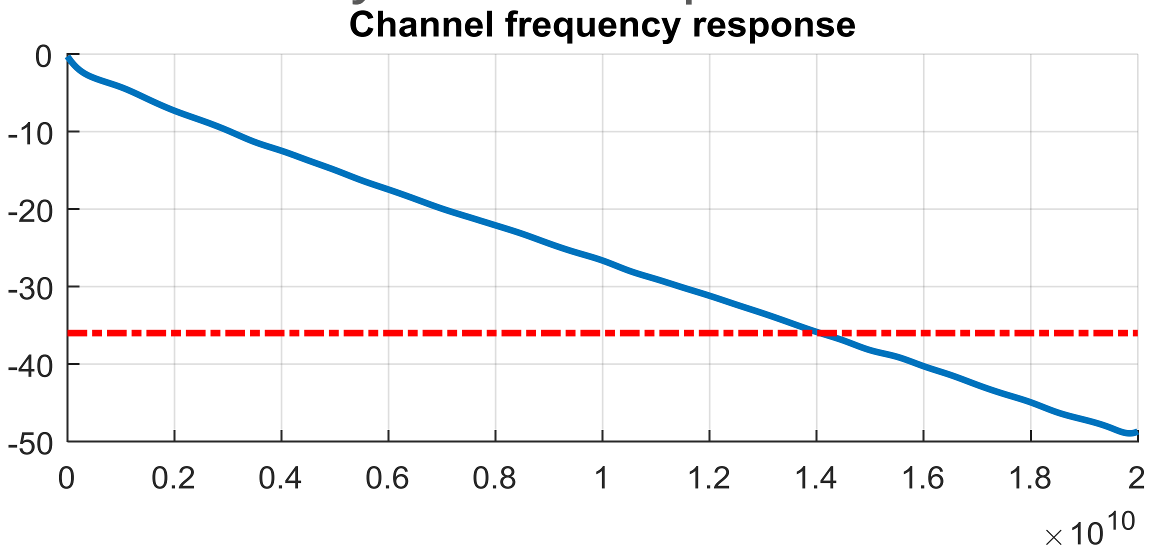
- The next step is a DSP based serdes



- The ADC samples the CTLE output at baudrate
  - No oversampling is conceivable at these speeds
  - 56Gb/s → 28 GHz sampling
  - Resolution is typically between 6 and 8 bits
- The rest of the equalization performs as in the analog case
  - FFE and DFE are already discrete time, now they operate on discretizes samples

CTLE and ADC

- The ADC becomes the single most complex component in the system
  - As it will be detailed later, we should talk about a sub-system, with many interactions with the DSP/firmware and the overall architecture of the serdes
- Is there still a need for a CTLE?
  - In practice, **YES**.
- Consider a high attenuation channel, and overlay the ADC quantization noise spectrum
- We need to boost the signal at high frequency to have some SNR to drive digital equalization

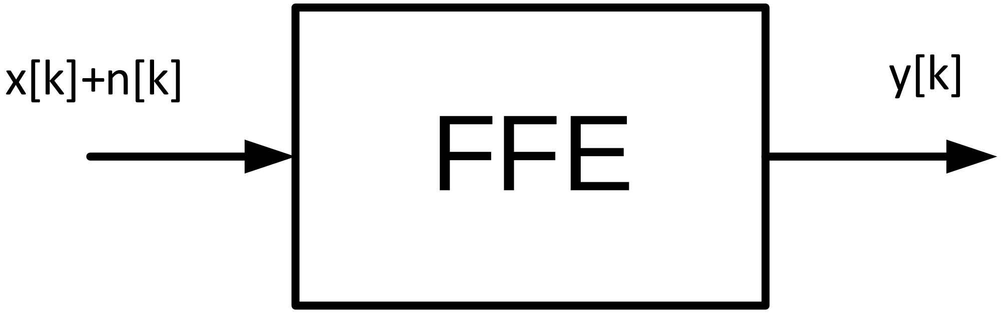


DSP-based Equalization

- DSP based implementations are intrinsically parallel
  - ADC data are already available with a given parallelism, say 32
  - Cannot implement serial processing at 28GHz…
- FFE can be easily parallelized to any degree
  - If latency is not an issue, we can even pipeline it
- DFE is intrinsically a 1UI loop
  - In a parallel implementation, we need to take 32 decisions in 32T
  - We can work on the problem (e.g. unrolling) but cannot change its nature!
  - PAM4 DFE is much more expensive than NRZ in DSP implementations
  - How many DFE taps are needed?

Feed-Forward Equalizers

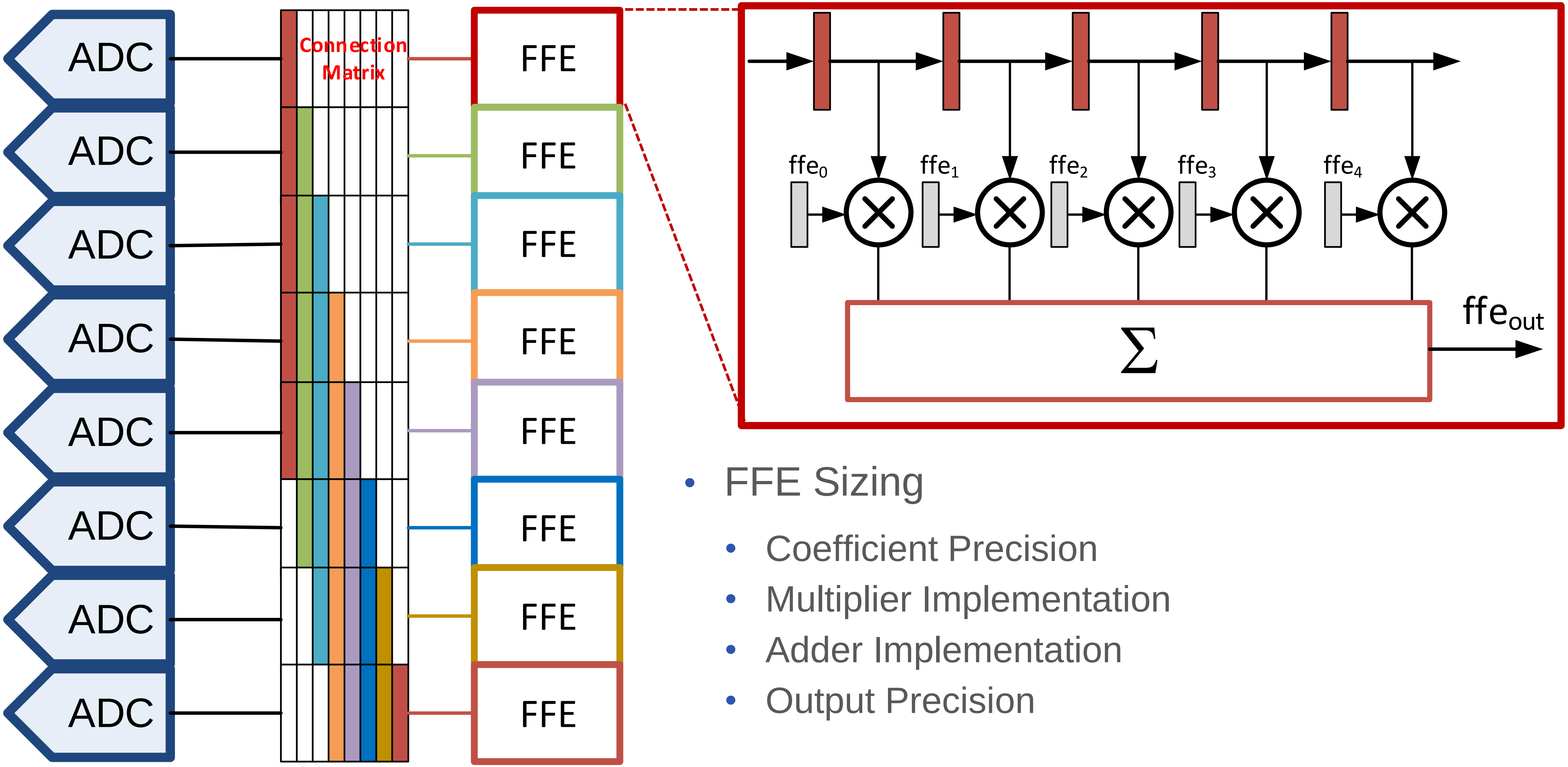
- Traditional FIRs can also be implemented in an analog receiver
  - However, they are limited by analog impairments
- The nice advantage of FFEs is that they do not exhibit causality loops and tight timing constraints
- There is no error propagation, but the input noise is emphasized by the filter



$$y[k] = \sum_{p=0}^{\#FFE-1} w[p]\,x[k-p] + \sum_{p=0}^{\#FFE-1} w[p]\,n[k-p]$$

$$noiseGain = \sum_{p=0}^{\#FFE-1} |w[p]|^2$$

Parallel FFE



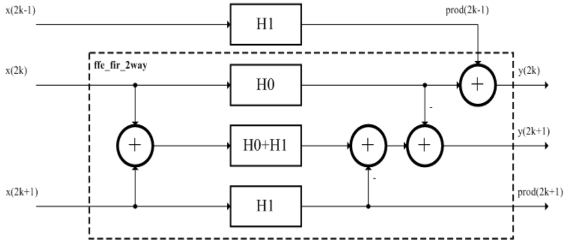
- FFE Sizing
  - Coefficient Precision
  - Multiplier Implementation
  - Adder Implementation
  - Output Precision

DSP Power

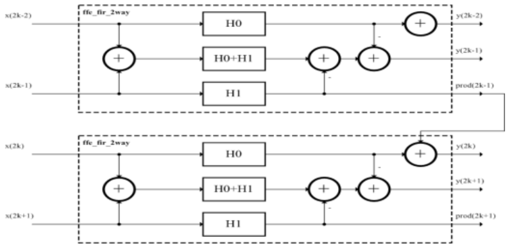
- It is now widely accepted that 1 tap DFE plus a “long” FFE are close to optimal
- But what about power?
  - DSP implementations were not competitive for technology nodes larger than 16nm
  - Yet, DSP architecture has to be carefully chosen to achieve maximum efficiency
- A figure of merit which is widely used to assess power efficiency is pJ/bit
  - Wide range of values
  - Measures the overall serdes efficiency (including AFE)

DSP Design Space Exploration

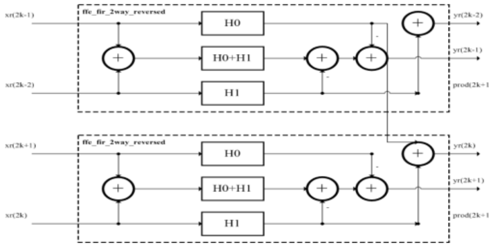
- FIR architecture can be explored to find the best implementations for a given context
  - E.G: FIR Transfer function H(z) can be rewritten to be implemented using parallel polyphase transformation $H(z)= E_0(z^2) + z^{-1} * E_1(z^2)$
  - Result can then be implemented and compared applying given technology constraints



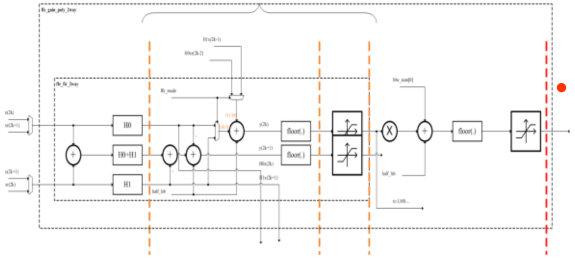
Polyphase FFE base block architecture



Polyphase FFE direct mode connection



Polyphase FFE reverse mode connection



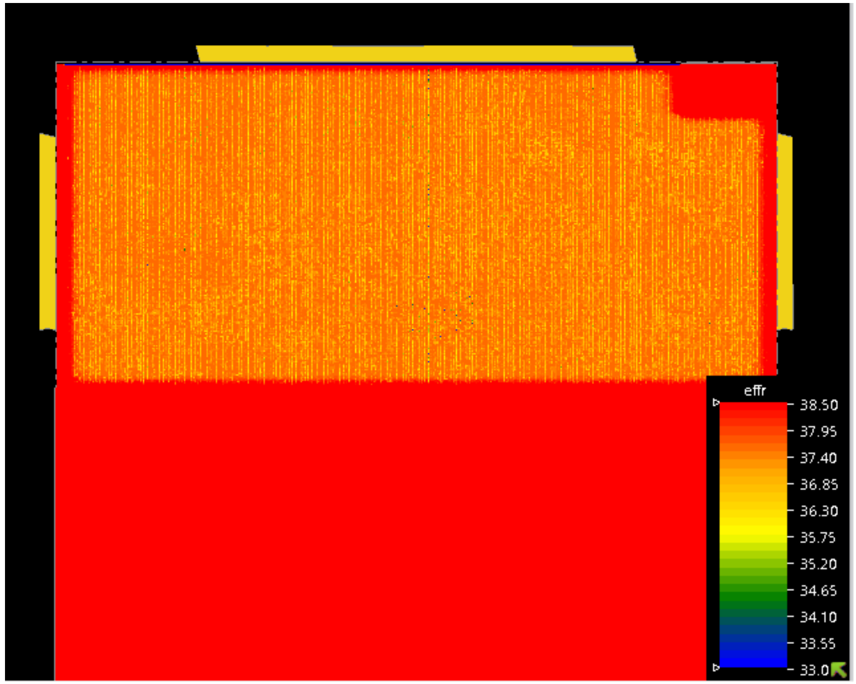
Polyphase FFE direct/reverse mode stage

FIR AREA/POWER COMPARISON (preCTS GATE LEVEL)				
implementation	area	total power (mW)	power increase %	area increase %
base	**8683**	**63.87**	0	0
4-2 compressor	**8518**	**54.06**	-15.36	-1.9
4-2 compressor, 2-way-polyphase	**10561**	**78.27**	22.55	21.63
4-2 compressor, brent-kung adder	**8675**	**55.6**	-12.95	-0.09

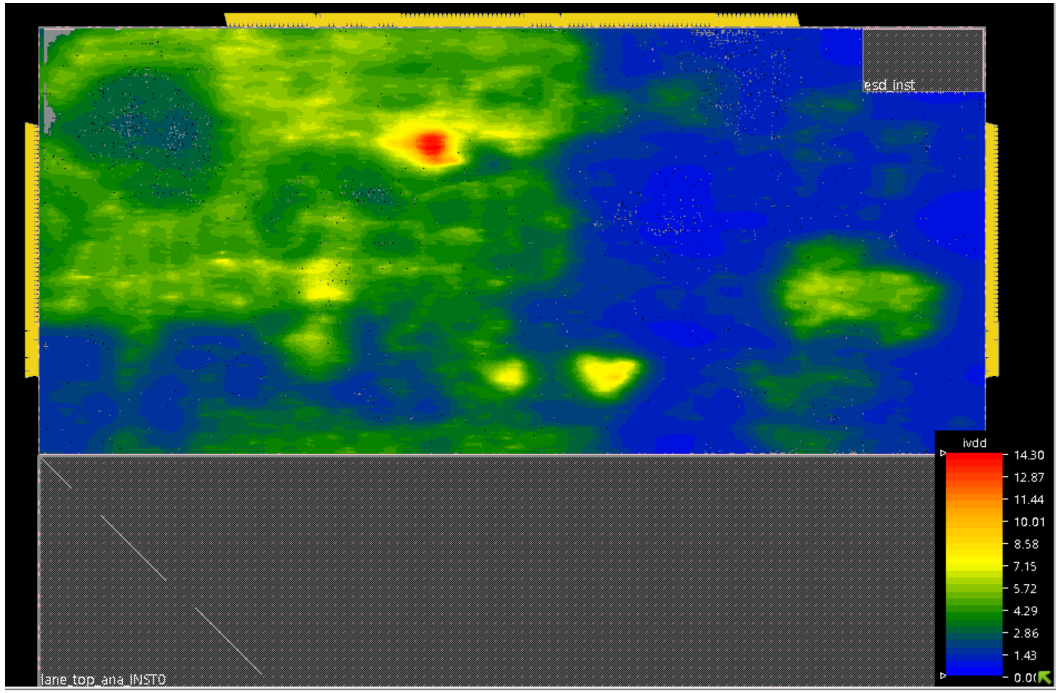
PG Grid

- **Classical PG grid resistance analysis and static IRdrop do not show critical issues**

  - **Grid resistance**



  - **VDD-VSS drop**



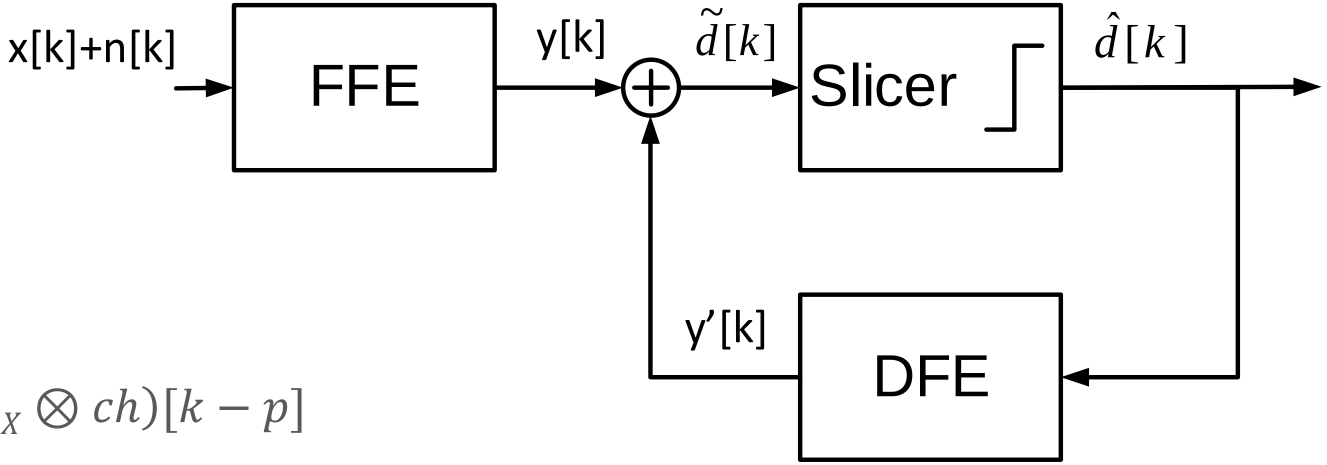
  - **Max Static IRDrop ~ 15mV**

Adaptation

- So far, we have neglected a fundamental problem:
  How can we determine the optimal coefficients for FFE/DFE?
- Also, we need to provide adaptativity to temperature/voltage/aging
  - The channel response changes
  - The link partner TX might change
  - The AFE transfer function might change
  - We need to pass temperature cycles without extra errors in the data stream
- We need to go back to Communication/DSP theory and revisit
  Mean Squared Error (MSE)/Least Mean Squares (LMS)

MSE

- MSE minimizes the expectation of the square of the error at the slicer



- The slicer error is defined as

$$\varepsilon = E\left\{\left|\hat{d}[k] - \tilde{d}[k]\right|^2\right\}$$

$$\hat{d}[k] = d[k-q] \qquad x[k] = \sum_{p=-\infty}^{\left\lfloor \frac{t}{T} \right\rfloor} d[k]\cdot(h_{TX} \otimes ch)[k-p]$$

$$\tilde{d}[k] = y[k] - y'[k] = \sum_{p=0}^{\#FFE-1} w_{FFE}[p]\cdot(x[k-p]+n[k-p]) - \sum_{p=1}^{\#DFE} w_{DFE}[p]\cdot\hat{d}[k-p]$$

- With the hypothesis that no error is made at the slicer
  - q represents the delay introduced by the processing path
- It can be easily seen that the cost function depends both ISI and noise
  - w_{FFE} taps determine noise gain (w_{DFE} taps do not!)

LMS

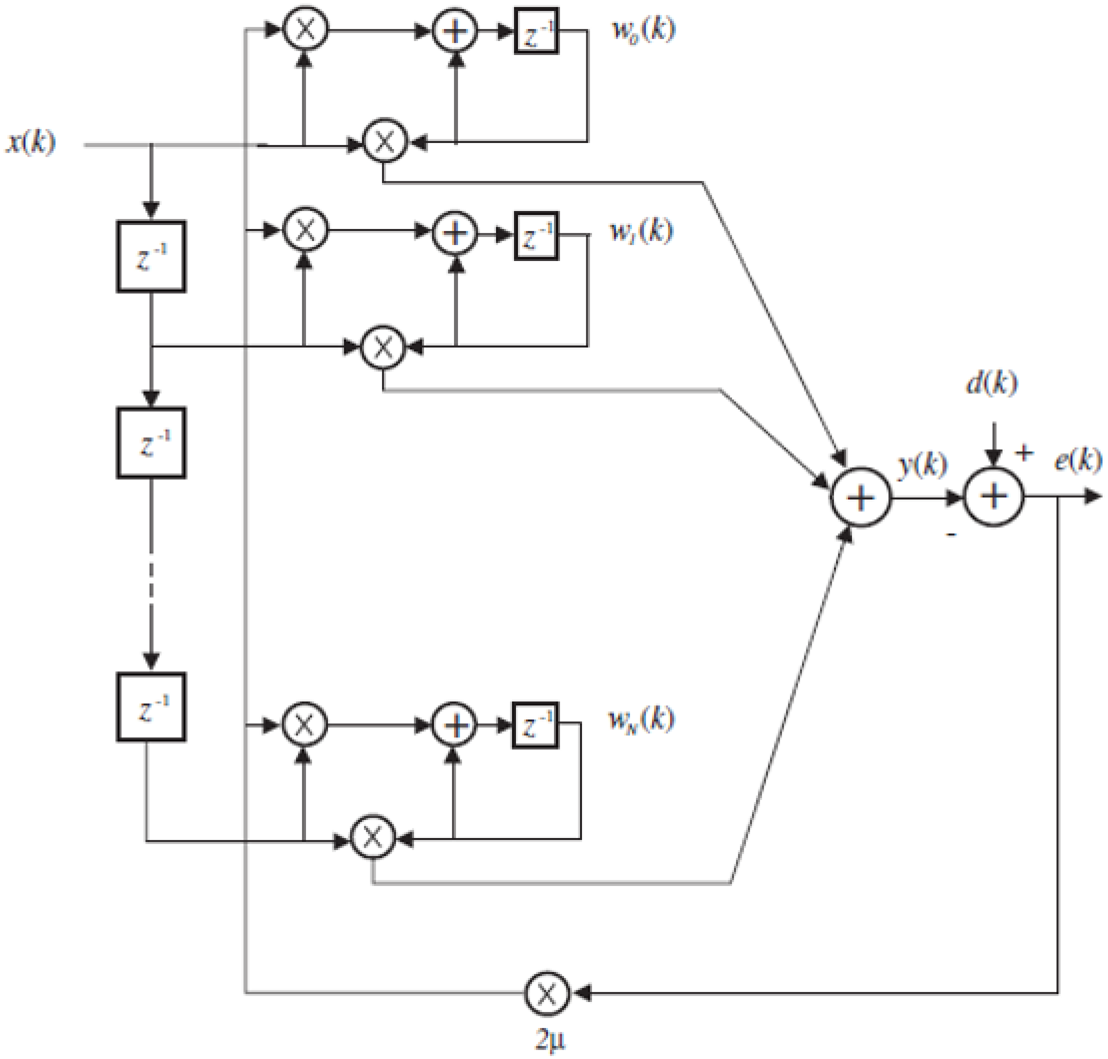
- We can workout the math and find a closed solution to the problem
- But we need a practical way to implement it
  - For example, noise correlation data and impulse response may not be known
- The LMS algorithm implements an iterative way to find the MSE solution
- We start with ergodicity, we substitute expectation with temporal average
- Then, we get a simple recursive formula for coefficient update

$$\mathrm{w}_{\mathrm{i}}(k+1) = \mathrm{w}_{\mathrm{i}}(k) + 2\mu\,\varepsilon(k)\cdot x(k-i)$$

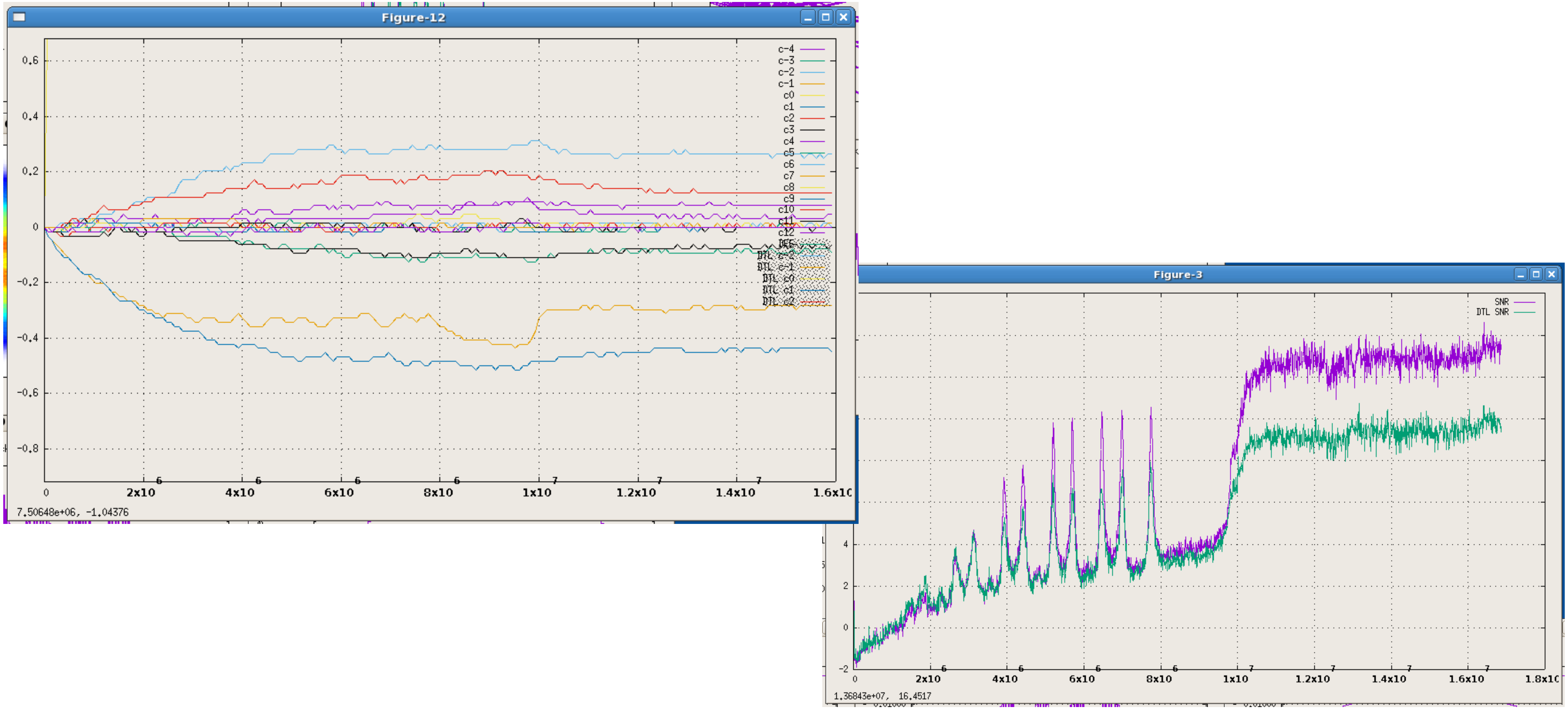
- At steady state, input data and error must be uncorrelated
- Convergence speed is controlled by the coefficient μ
  - Determining the bandwidth of convergence is, in general, non trivial

LMS Scheme

- In practice, we can save power and area with some approximations
- $\varepsilon(k)\cdot x(k-i)$ is approximated
  - $sign(\varepsilon(k))\cdot x(k-i)$ – sign error
  - $\varepsilon(k)\cdot sign(x(k-i))$ – sign input
  - sign-sign
- There are convergence issues to be checked with approximations
  - Non linear approximations…
- The update cycle does not need to be performed at speed

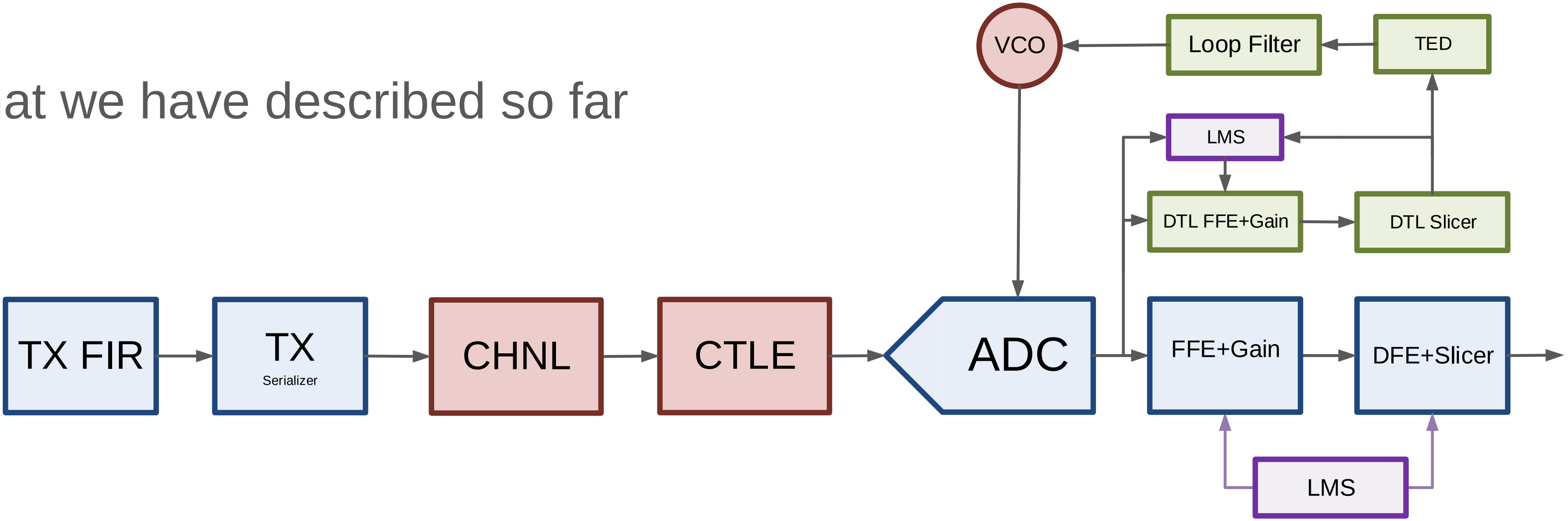


LMS on the field



The overall System

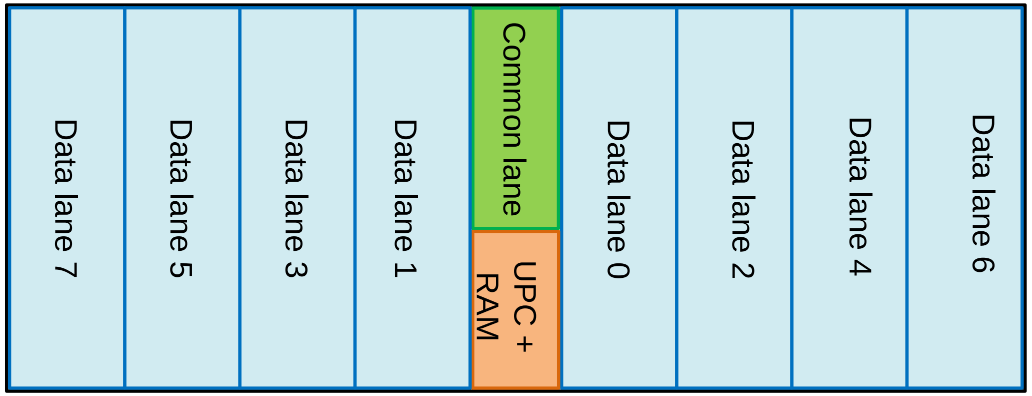
- What we have described so far

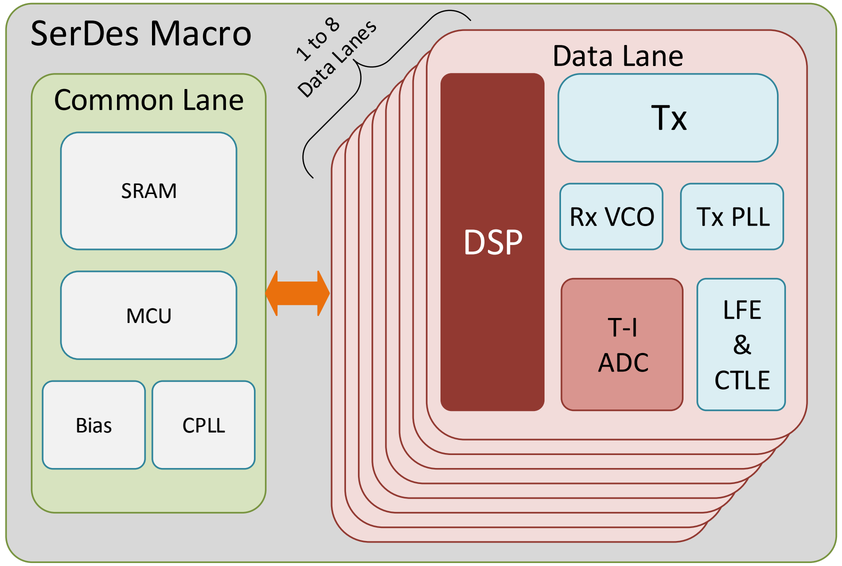


- The complexity of the system requires on board microcontroller
- The firmware manages the **whole system**
  - Calibration of analog blocks
  - System configuration
  - Link optimization and monitor

Microcontroller and FirmWare

- SerDes are assembled in macros
  - 4 or 8 lane macros are available
- A common lane provide ancillary functionality
  - Central analog bias
  - First level of clock generation (internal reference clock)
  - Microcontroller and memory
- Size matters!
  - Using compact 32 bit microcontroller
    - ARM cortex M3, switching to RISC-V (with FPU)
  - 48kB code RAM + 32 kB data RAM
  - Silicon Area: 0.061 mm^2, power < 5 mW

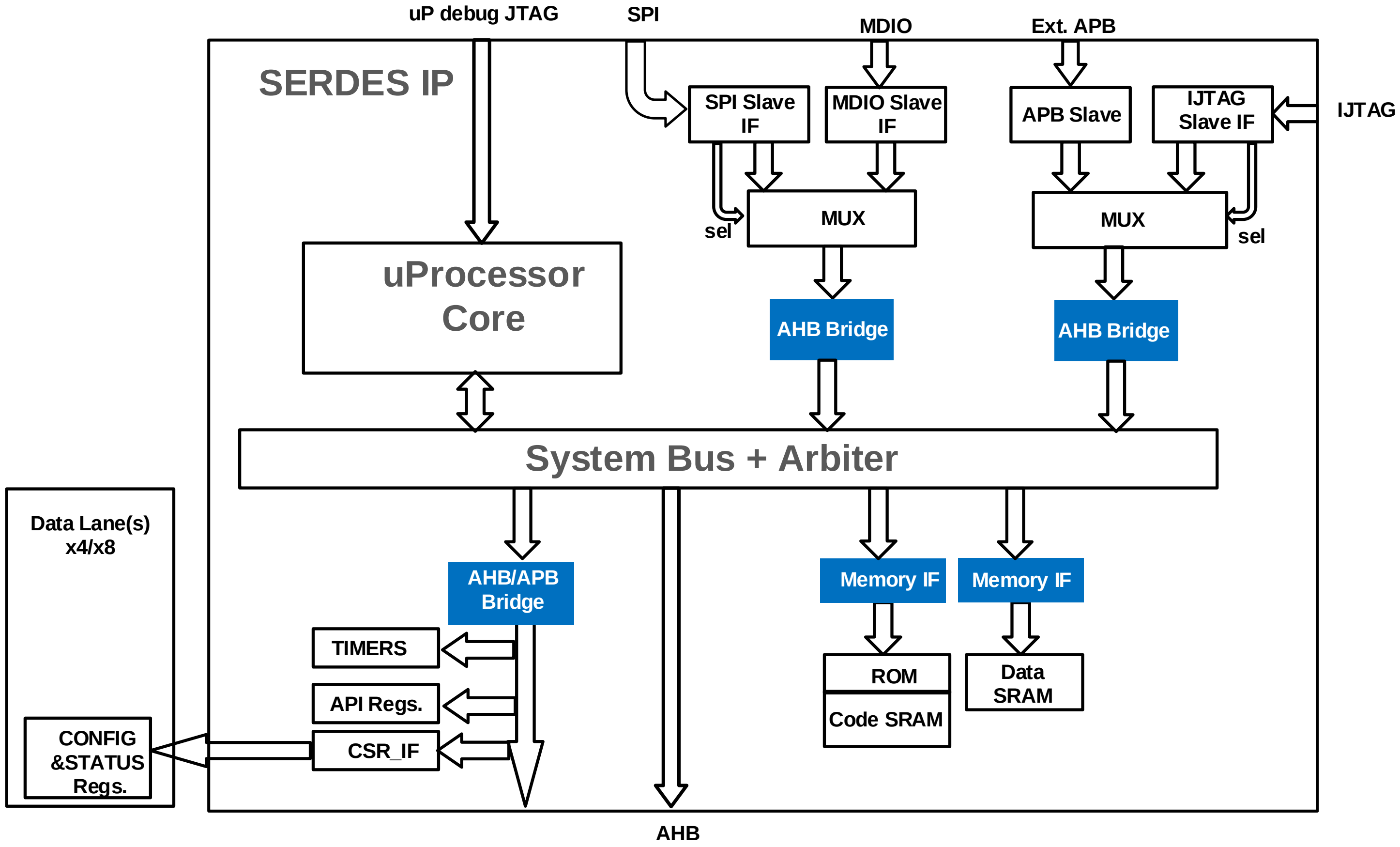




MicroController subsystem

- Communication with datalanes through register maps
  - A register map is a set of HW registers that is mapped to a memory space
  - But it is not a memory…
- Each data lanes contains several register maps
  - Depending on the HW block
  - RX, TX, CLK, …
- A set of interrupts is connected to specific HW components for low latency handling

SERDES MACRO TOP LEVEL BUS ACCESS VIEW



FirmWare Architecture

Paradigms

- Simplicity is a must
  - No RTOS
  - No external libraries…
    - sin(x), log(x),…? → LUT in ROM
- Yet, C++ is used
  - Not fully fledged in practice
  - Limited usage of inheritance, polymorphism, virtual methods
- …keeping efficiency
  - Each register is mapped into an object
  - Access to register fields is optimized

Execution Model

- Collaborative multitasking
  - A Task Manager executes processes with round-robin algorithm
    - Each process explicitly handles its state
    - No preemption is possible (except for interrupts)
  - Upon invocation, a process
    - executes
    - interacts with hardware
    - (possibly) changes state
    - returns control
- Asynchronous scheme
- Simple scheme
  - Task are statically created

Hardware support

- A number of hardware **monitors** are available
  - Data are too fast to be visible at the firmware level
  - Statistics are computed by hardware blocks
- For example:
  - SNR computation
  - BER computation
  - Histogram computation
  - Threshold monitors
- The **integration** of all these monitors (more generally **sensors**, including process spread and temperature) is used to implement the algorithms that control and optimize the serdes operation

Firmware Tasks

- Configure and Calibrate the system at startup
- Manage Datalanes
  - Configure SerDes for operation at different datarates/modulations
  - Calibrate analog circuits at startup
  - Configure and control hardware loops
  - Bring the link up
  - Optimize link performance
  - After link is up
    - Implement background optimization loops
    - Monitor system operation and performance
    - Manage destructive events

Firmware Tasks

- Manage low level link protocols
  - Assisted by digital blocks that process protocol frames
- Autonegotiation
  - Initial phase (fixed speed) where two serdes negotiate the final signaling speed
- Link Training
  - Follows autonegotiation
  - Allows controlling the TXFIR status
  - Complex optimization algorithms are implemented during this phase
    - Optimization of analog front end
    - Optimization of DSP, sampling time, …

Firmware Tasks

Why is it so challenging?

- **Interface and control** very high performance analog and digital sub-systems
- Implement complex **calibration algorithms** to cope with parameter dispersion
- **Optimize** the link under uncertainty and time constraints
- Guarantee **consistent and repeatable** performance
- Operate with **third party** link partners
- Exploit the intrinsic **performance** of the AFE+DSP
- Very limited resources, **efficiency** is a must
- Participate with **architecture** optimization and evolution on next nodes
  - Innovation, algorithms, strategy, …

External Interface, or more Software

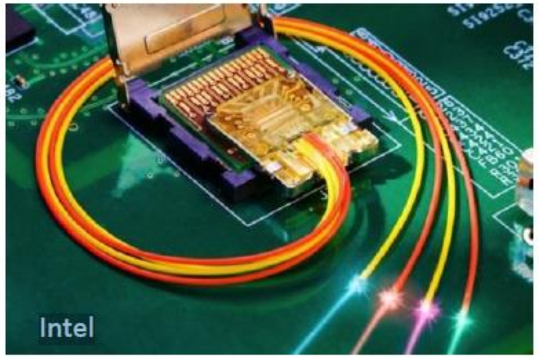
- A set of services is provided to upper levels/integration as an API
- This is a paradigm shift WRT previous analog serdes
  - Direct access to registers
- API and host side drivers become a part of the system
- The API becomes also the interface with the development environment and customer GUI
- Python is used as command line interface to the serdes
  - A higher abstraction level is provided encapsulating the HW and the FW API
  - Debugging and characterization is performed in Python
  - Integration with instrumentation, databases, libraries
- A GUI is built on top of Python to give access to complex functionalities

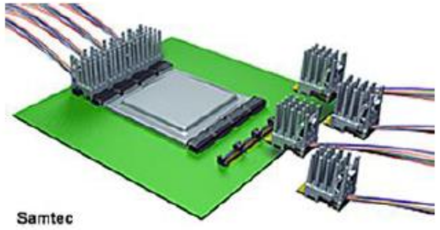
What is next

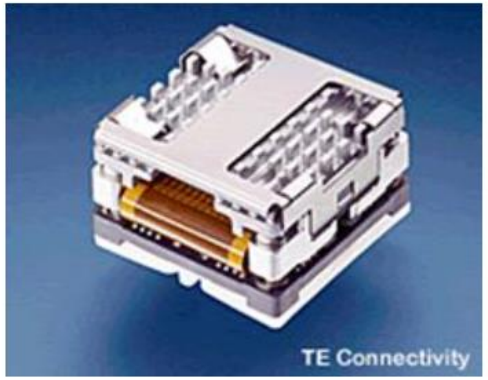
- 112G is coming
  - Increased baudrate cannot exploit faster technologies
    - 5nm is next, but…
  - CTLE stages become more problematic
    - Extension to new Nyquist
    - Parasitics and passives
  - The TX is also penalized for similar reasons
  - Channel attenuation becomes much worse
    - IL per unit length doubles
    - Roughness creates larger ILD
    - Packages much more complex
- Is on-board optics next to come?

Consortium for On-Board Optics (COBO)

- 112G electrical links create very challenging problems because of passives
  - Attenuation increases linearly with frequency
  - Problems with new materials
- Most likely optics will come closer to silicon
  - on-board optical modules
- On-board or embedded optics have been available for over a decade
- Growing use of optics
  - Datacenter networks
  - Consumer applications
  - Industrial, etc.







eSilicon®

Collaborate. Differentiate. Win.