ECE 546
Lecture - 24
High-Speed Links

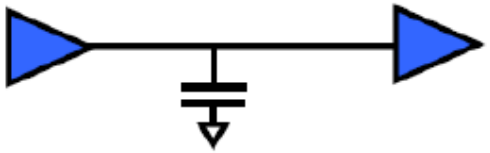
Spring 2026

Jose E. Schutt-Aine
Electrical & Computer Engineering
University of Illinois
jesa@illinois.edu

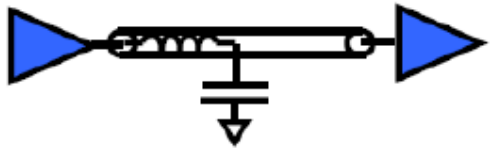
Inter-IC Communication Trends

Decade	Speeds	Transceiver Features
1980's	>10Mb/s	Inverter out, inverter in
1990's	>100Mb/s	Termination Source-synchronous clk.
2000's	>1 Gb/s	Pt-to-pt serial streams Pre-emphasis equalization
Future	>10 Gb/s	Adaptive Equalization, Advanced low power clk. Alternate channel materials

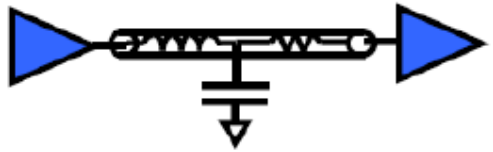
Lumped capacitance

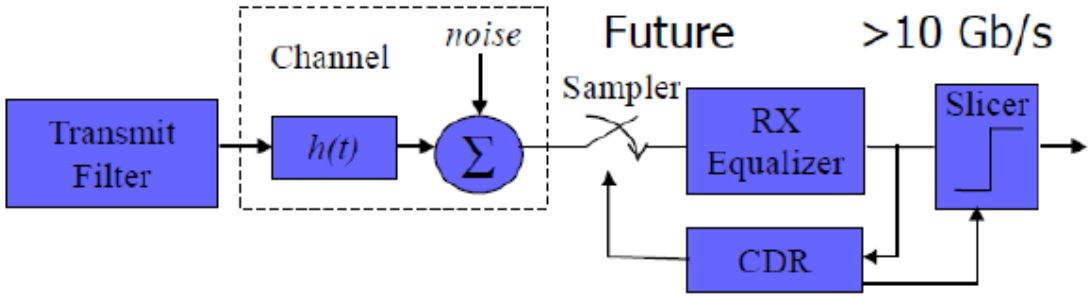


Transmission line



Lossy transmission line





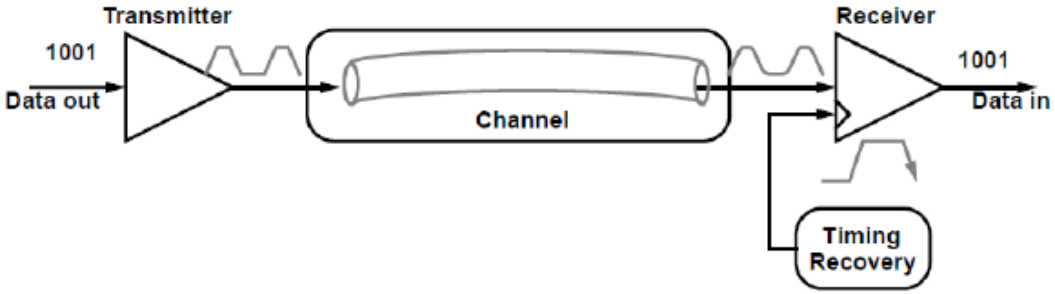
Slide Courtesy of Frank O'Mahony & Brian Casper, Intel

High-Speed Bus and Networks

- Memory Interfaces
  - DDR5 (6.4-8.8 Gbps/pin)
  - LPDDR5X (up to 8.5 Gbps)
  - GDDR6X (21-24 Gbps)
  - HBM3E (> 1 Tbps aggregate BW)
- Networking & Fabrics
  - Ethernet (up to 800 G (PAM4))
  - InfiniBand XDR (800 Gbps)
- Peripheral & Consumer Interfaces
  - USB4 v2 (80-120 Gbps)
  - HDMI 2.1 (48 Gbps)
  - Thunderbolt 4 (40 Gbps)
- Storage & Expansion
  - SATA III (6 Gbps)
  - UFS 4.0 (23.2 Gbps)
  - PCIe 4.0 (16 GT/s)
  - PCIe 5.0 (32 GT/s)
  - PCIe 6.0 (64 GT/s PAM4)

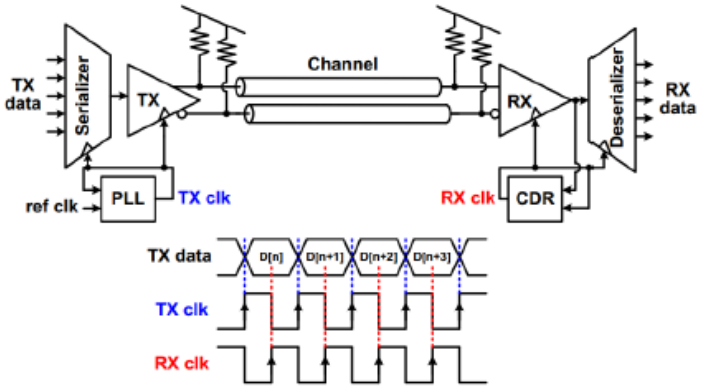
Basic Serial Link Architecture

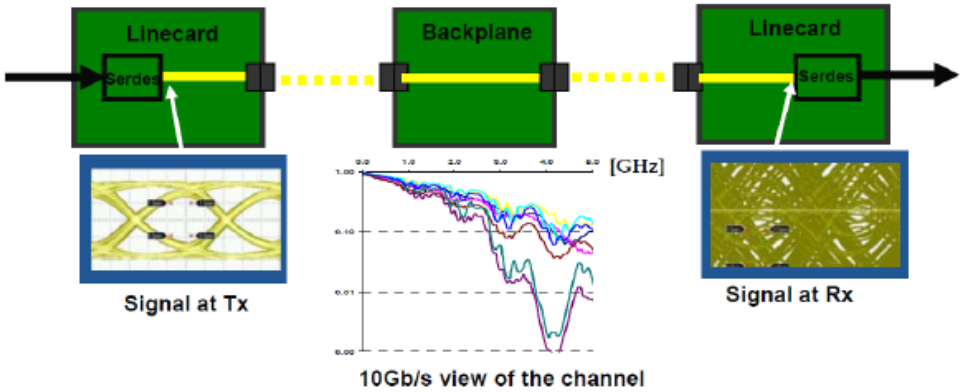
- **TX:** Generates train of analog pulses depending upon input digital data symbols and transmits them across the channel to the receiver.
- **Channel:** Electrical path between TX and RX blocks. Typically comprises of PCB traces, vias, connectors and other such I/O interface components.
- **RX:** Amplifies the recovered analog signal and samples it to output the corresponding digital bitstream.
- **Timing Recovery:** Circuit block responsible for deciding the correct sampling strobe point to sample and convert received data from analog to digital.



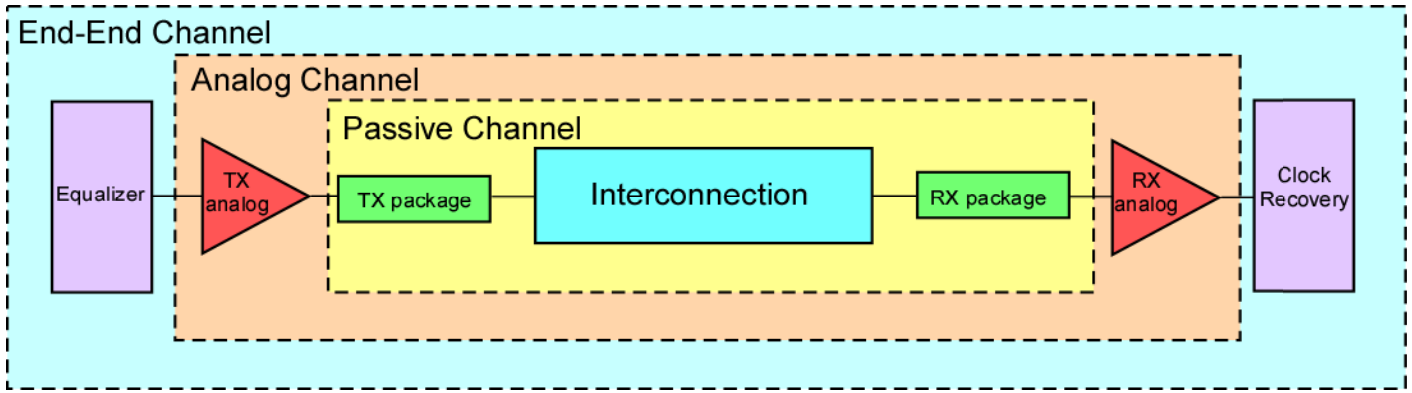
SerDes Serial Link Architecture

- **Serializer:** converts input parallel data-bits into serial for inter-IC transmission across lossy channel.
- **PLL:** Phase-locked loop used as a clock-synthesizer circuit that generates the high-speed master clock used for data transmission.
- **CDR:** Clock-Data Recovery circuit that performs the timing recovery function to recover TX clock to sample the received signal at the RX.
- **Deserializer:** converts recovered data-bits from serial to parallel form.





Serial Link

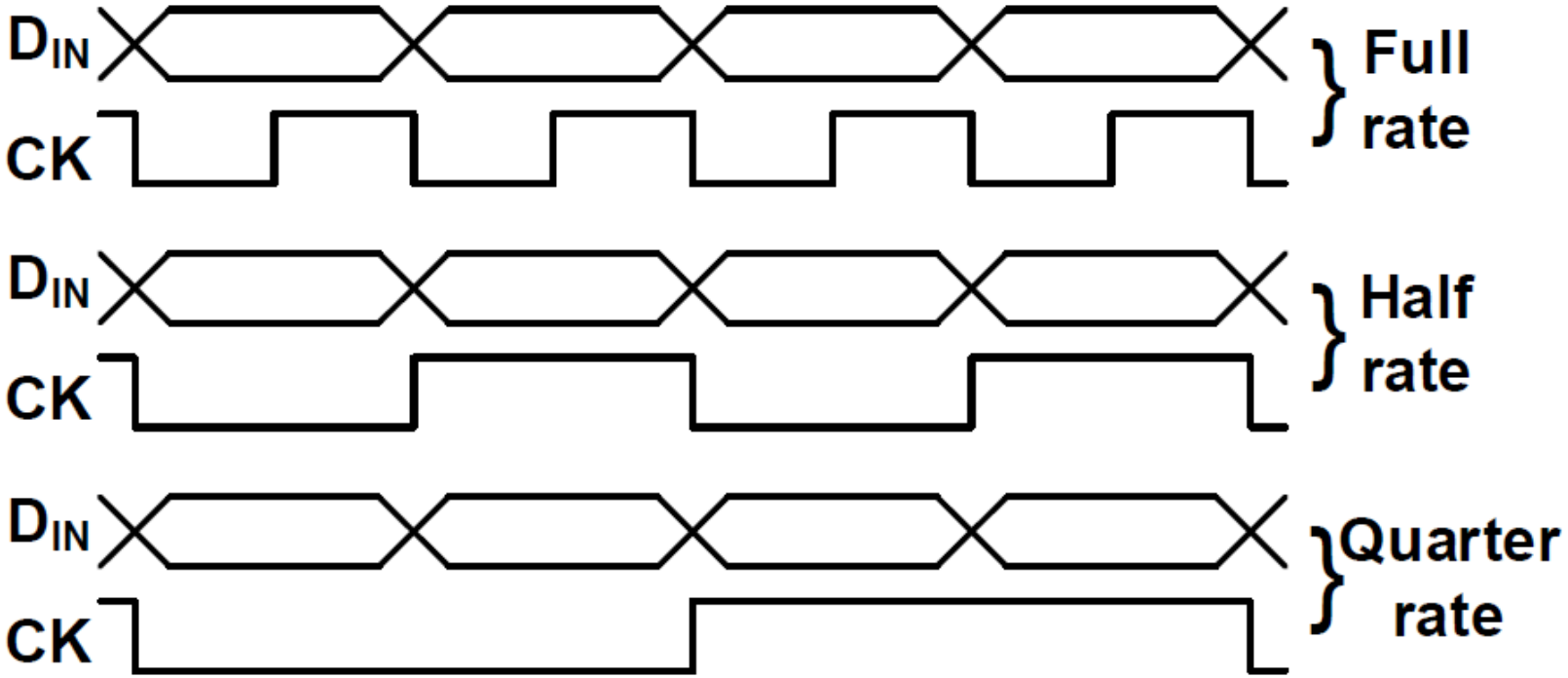


- Passive channel consists of linear elements (TL, package)

- Analog channel includes TX driver and RX termination network

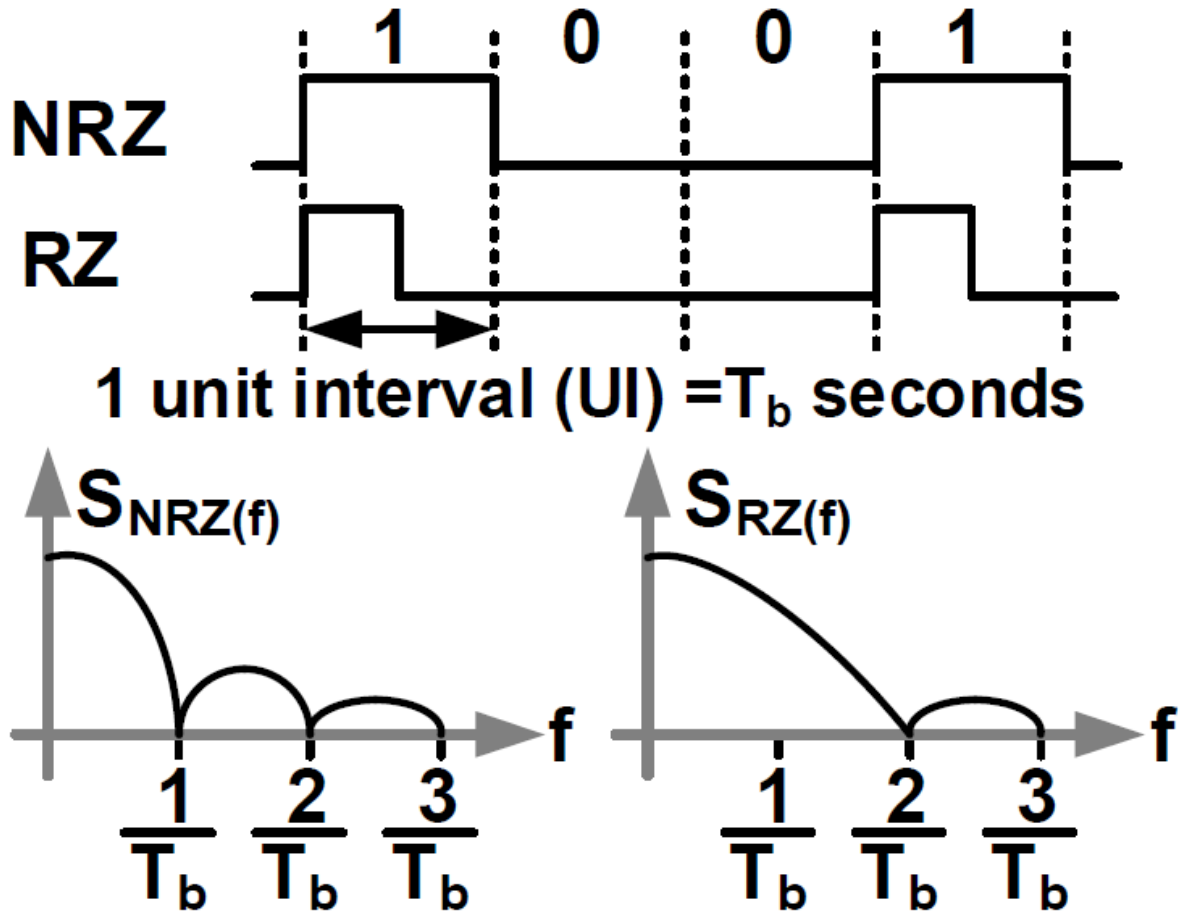
- End-to-end channel includes everything

Link Classification

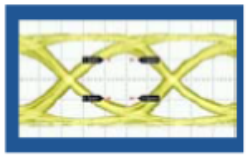


- Number of data bits per clock cycle
- Need multiple phases for half rate and quarter rate

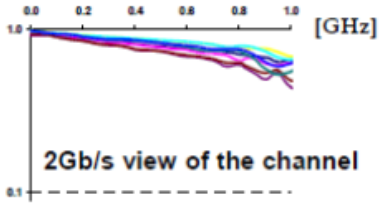
Signaling Protocol NRZ vs RZ

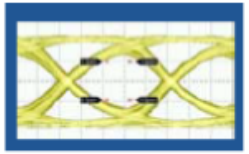


Channel

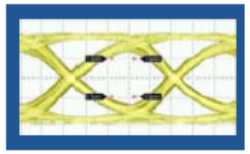


Signal at Tx

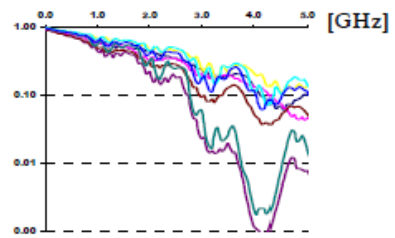


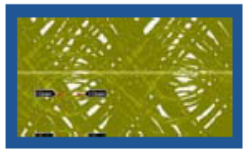


Signal at Rx

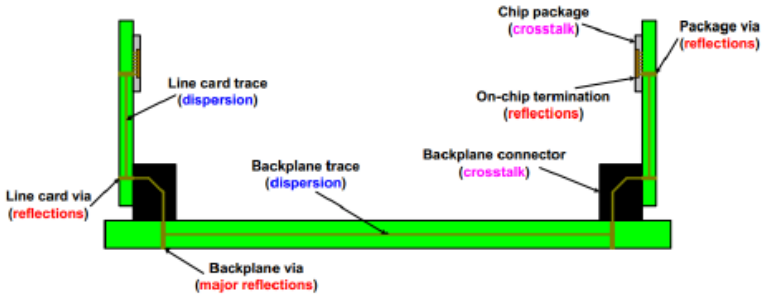


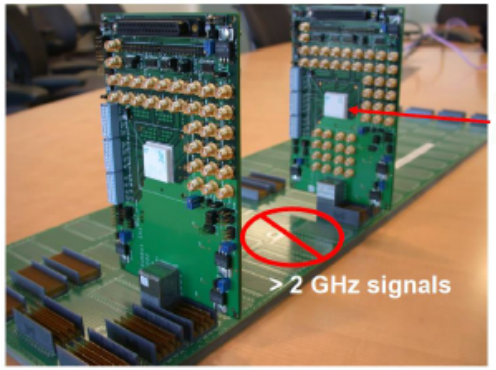
Signal at Tx

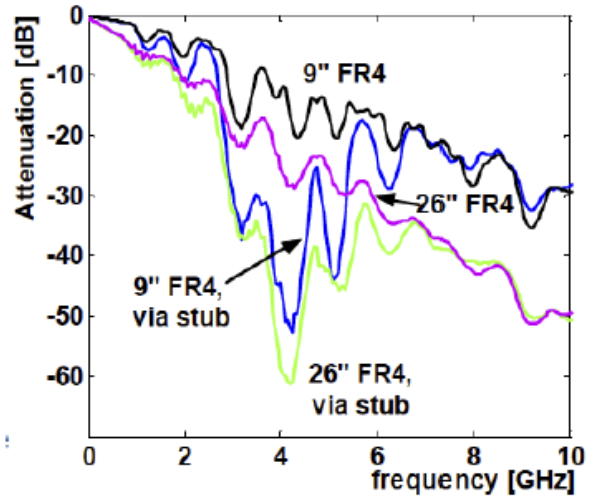




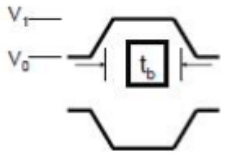
Signal at Rx





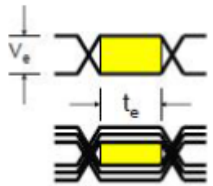


Eye-Diagrams



This is a "1"

This is a "0"

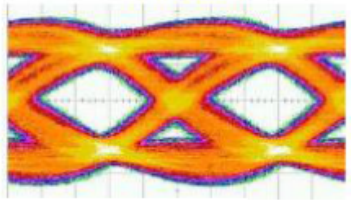


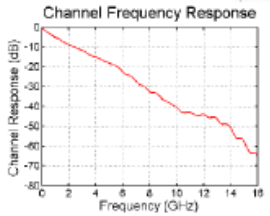
Eye Opening - space between 1 and 0

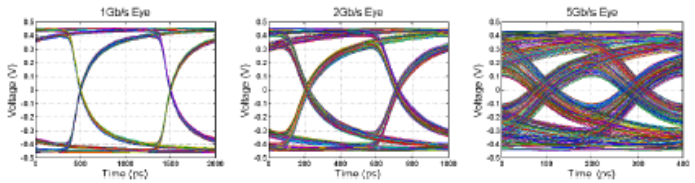
With voltage noise

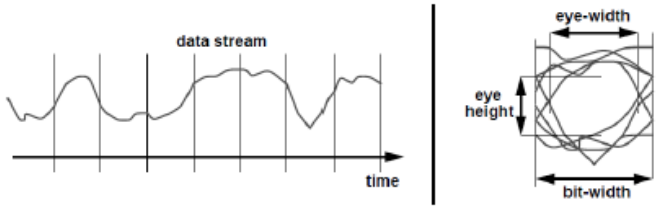
With timing noise

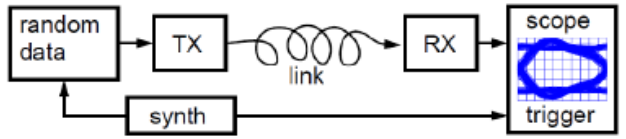
With Both!

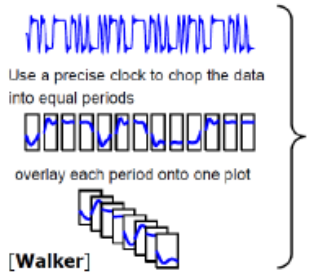


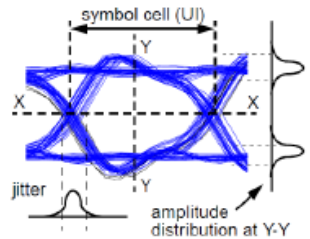


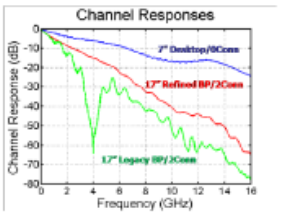


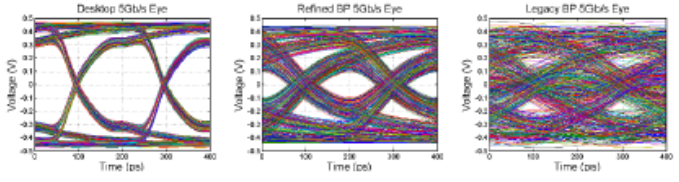












Timing Margin

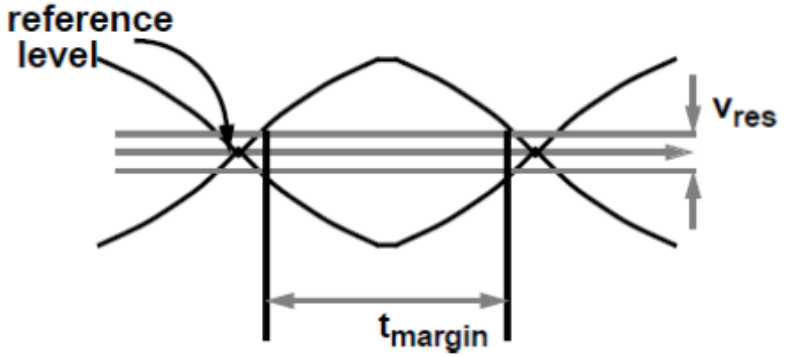
- $t_{margin} = t_{bit} - t_{os} - t_{jd} - t_{jc}$

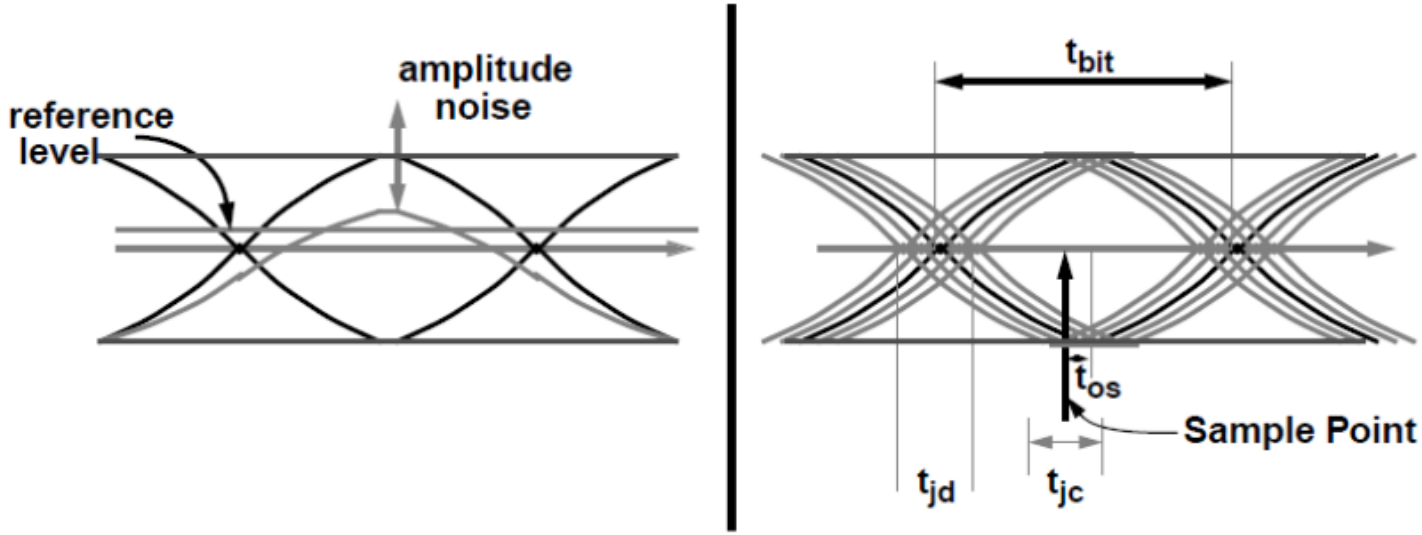
  Where,

  t_{bit} = bit-width of a symbol

  t_{os} = sampling error

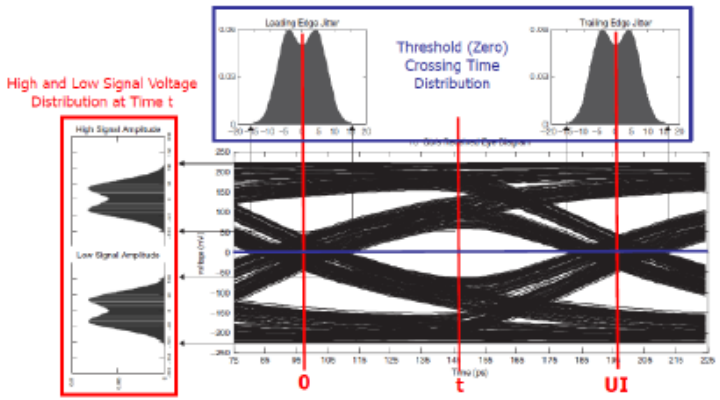
  t_{jd}/t_{jc} = data/clock jitter



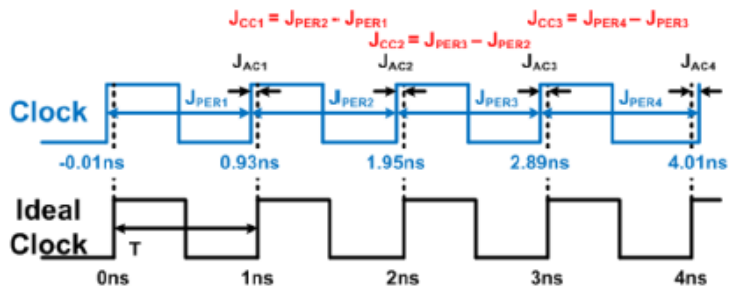


Timing Jitter

Jitter = time-domain variation in clock signal. Dominated by power-supply noise and substrate noise both of which don't scale with technology.

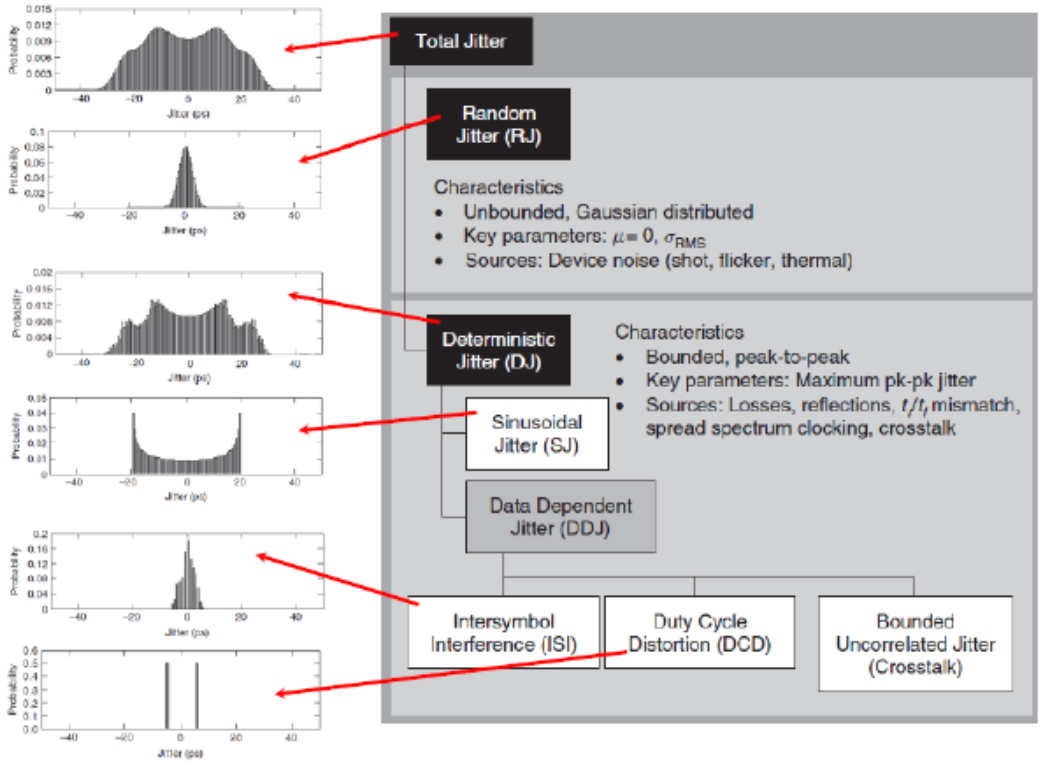


[Hall]



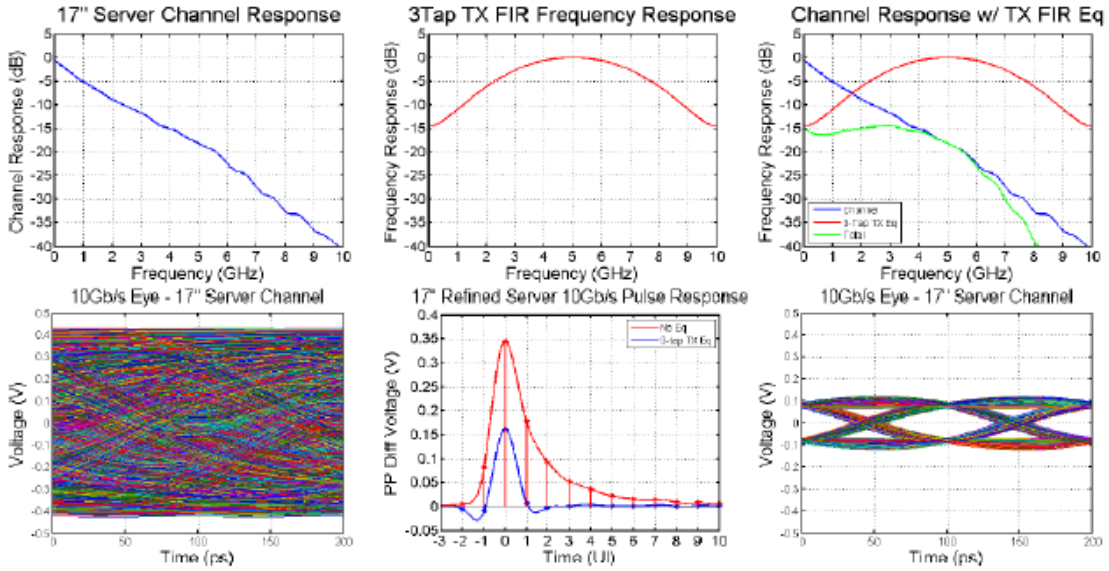
n	1	2	3	4	Mean	RMS	PP
J_{PER}	-0.06	0.02	-0.06	0.12	0.005	0.085	0.18
J_{CC}	0.08	-0.08	0.18	-	0.06	0.131	0.26
J_{AC}	-0.07	-0.05	-0.11	0.01	-0.055	0.05	0.12

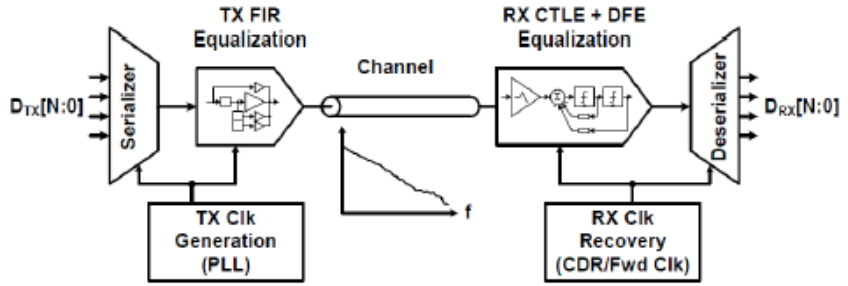
J_{PER} = time difference between measured period and ideal period
J_{CC} = time difference between two adjacent clock periods
J_{AC} = time difference between measured clock and ideal trigger clock



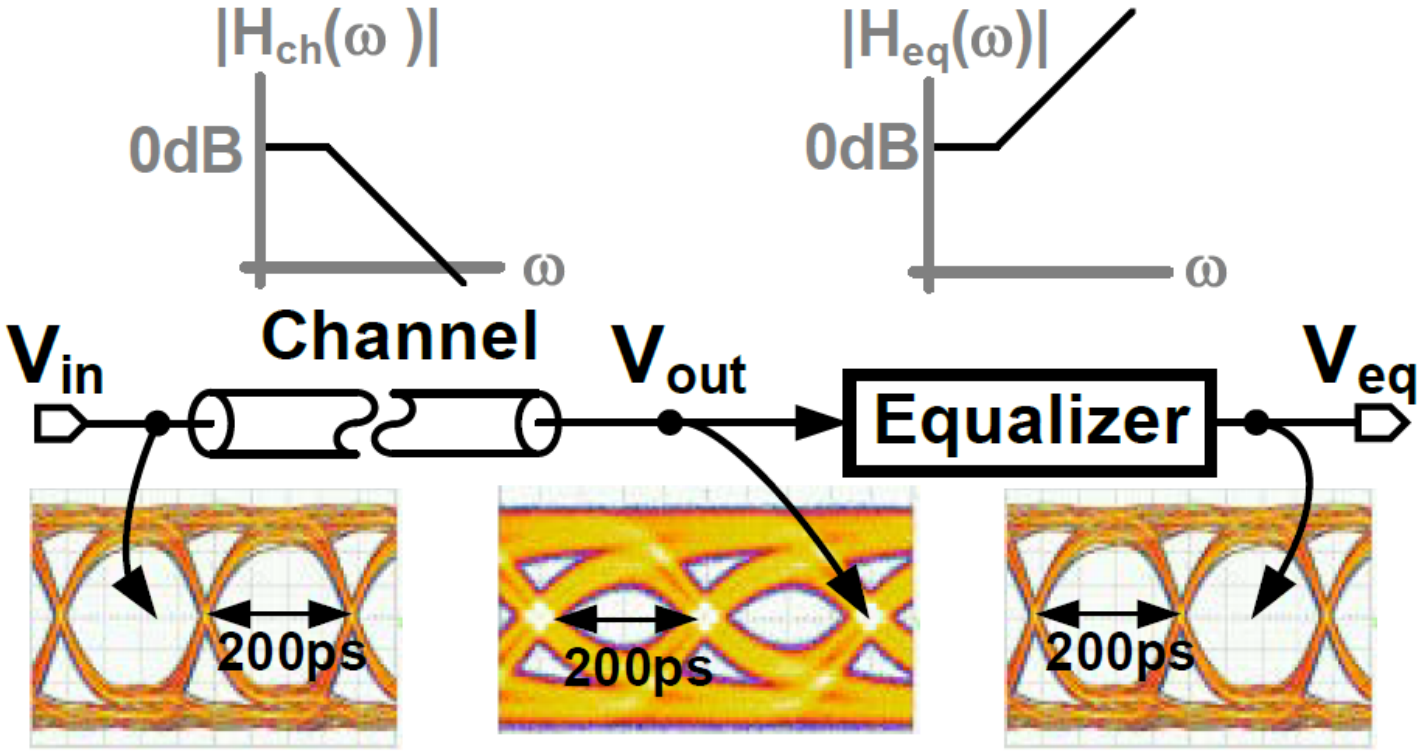
Equalization

- Goal: Counteract the channel degradation by trying to flatten the frequency response on both TX as well as RX fronts to remove time-domain ISI effects.





Equalization



FFE vs. DFE

FFE

- Can mitigate the pre-cursor channel response in low-BW channels.
- Can compensate ISI arising from transient TL loss over wide time-spans.

DFE

- Cannot equalize ISI arising from pre-cursor channel response.
- Can only compensate ISI from a fixed time-span.

FFE + DFE

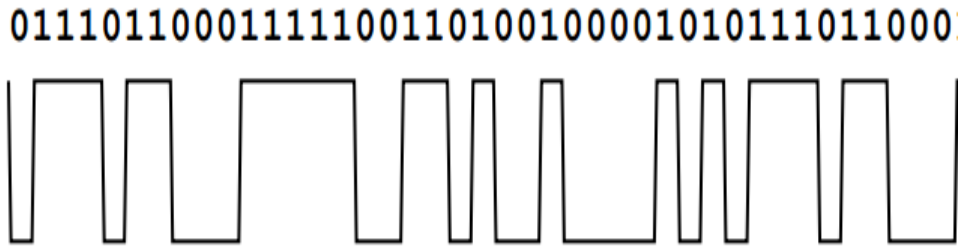
- Guarantees max. performance from the SerDes.
- Advantage:
  - DFE permits use of low-frequency de-emphasis at TX resulting in a larger received signal envelope, smaller signal/crosstalk ratio.
  - System capable of employing continuous adaptive equalization of its feedback taps to optimize performance.

CDR Circuit Overview

- Monitor data signal transitions and select optimal sampling phase for the data at midpoint between edges.
- Extracts clock information from incoming data stream and uses this regenerated clock to resample the data waveform and recover the data.
- Non-linear circuit and key block to limit jitter, noise within the SERDES circuit.

Basic Idea

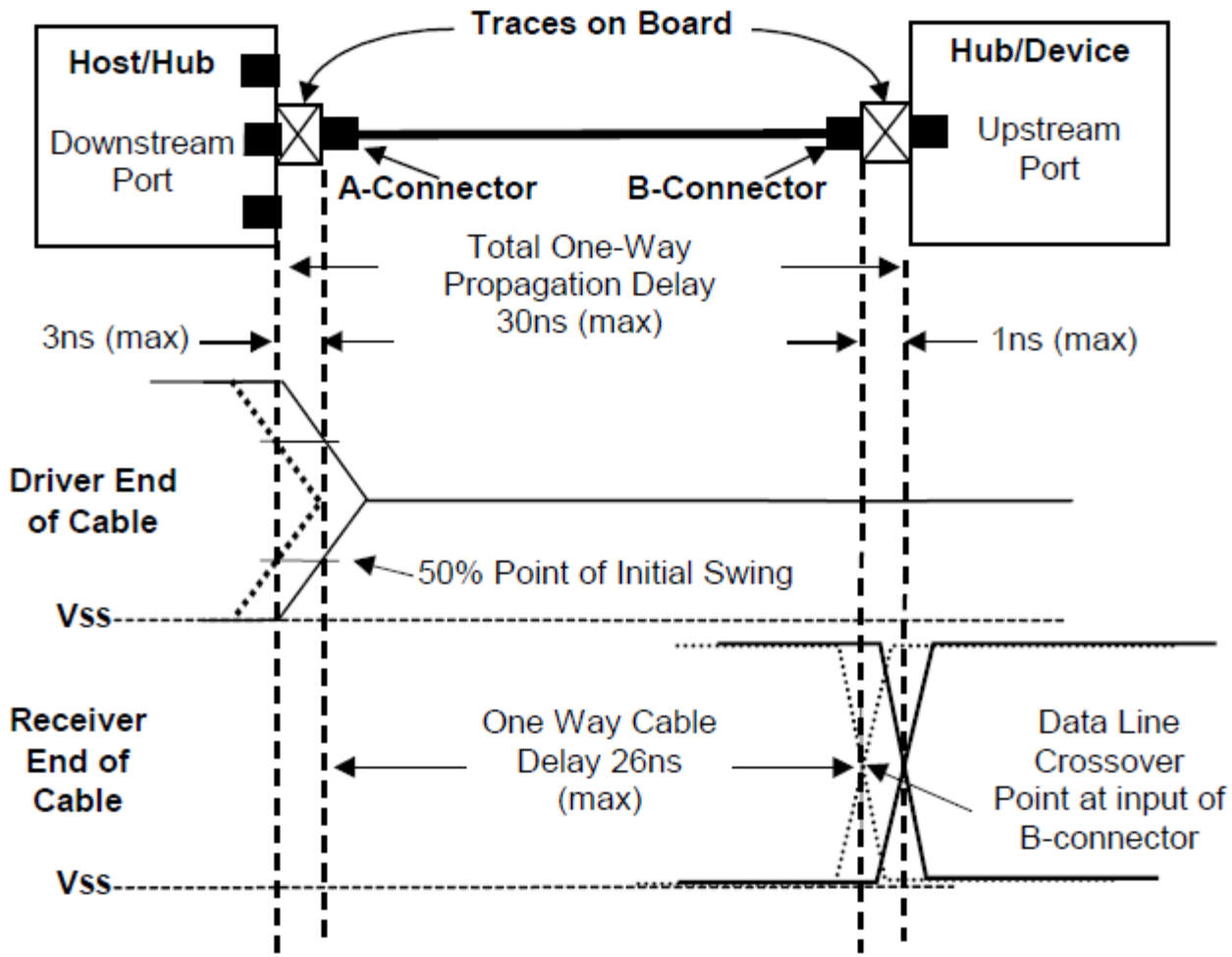
- Serial data transmission sends binary bits of information as a series of optical or electrical pulses



- The transmission channel (coax, radio, fiber) generally distorts the signal in various ways

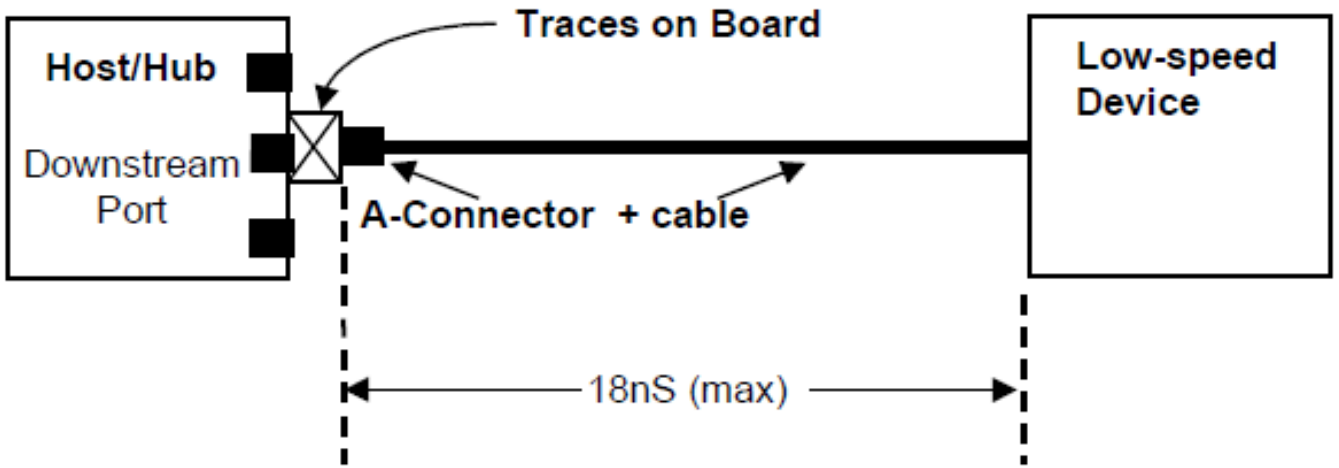
- From this signal we must recover both clock and data

Universal Serial Bus (USB)



Full-speed Cable Delay

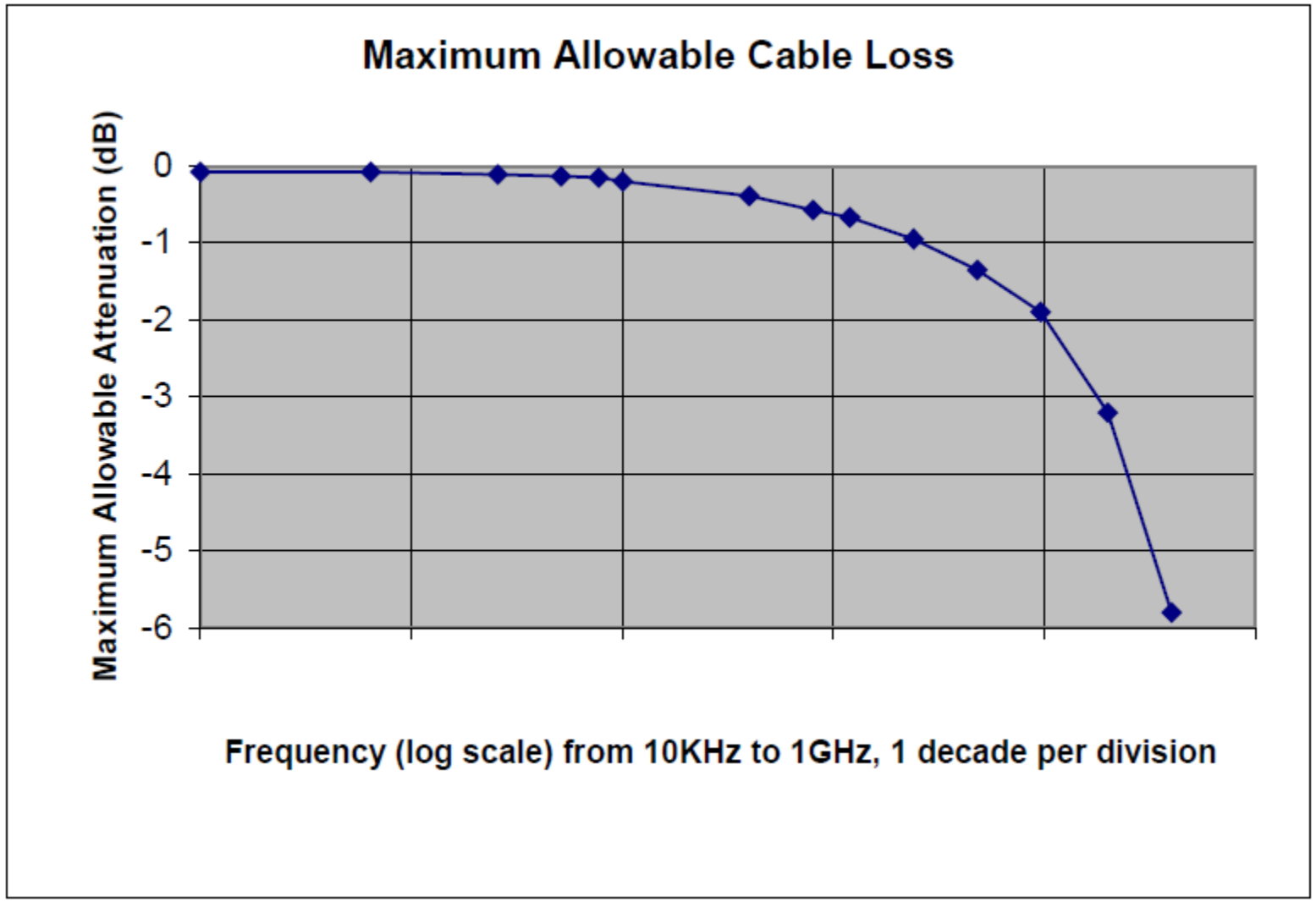
Universal Serial Bus (USB)



One-way Propagation Delay
Low-speed Cable Delay

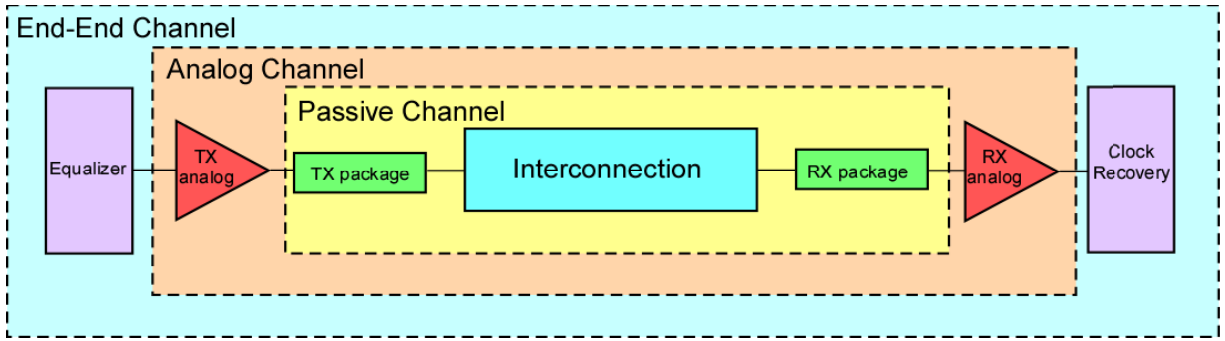
- The cable delay must be less than 5.2 ns per meter.
- The maximum delay allowed is 30 ns.
- Allocation for cable delay is 26 ns

Universal Serial Bus (USB)



Serial Channel Characterization

- Millions of bits of behavior are needed to adequately characterize serial links➔ long simulation times
- SERDES transmitters / receivers can be modeled as a combination of analog & algorithmic elements
- Serial channels can be characterized using S Parameter data and/or other passive interconnect models



High-Speed Serial Channels

High speed Serial channels are pushing the current limits of simulation. Models/Simulator need to handle current challenges

- Need to accurately handle very high data rates
- Simulate large number of bits to achieve low BER
- Non-linear blocks with time variant Systems
- TX/RX equalization and vendor specific device settings
- Coding schemes
- All types of jitter: (random, deterministic, etc.)
- Crosstalk, loss, dispersion, attenuation, etc…
- Clock Data Recovery circuits
- TX and RX may come from different vendors

Simulation Methods

Analysis Method	Advantages	Drawbacks
IBIS	Fast	Not accurate
Device Level	Accurate Nonlinear	Very slow IP liability
Fast convolution	Very fast Handles EQ Include bit patterns	Not Silicon Specific Assumes LTI
Statistical	Very Fast Handles EQ	Not silicon specific No bit patterns Assumes LTI
IBIS-AMI	Fast Handles Vendor EQ Includes Bit Patterns Not limited to LTI	Implementations vary

Simulation Environment for High-Speed Links

- Use Cadence Virtuoso for transistor level modeling and design of link blocks like TX driver, RX driver as well as the clocking circuits.
- Model the channel in Ansys HFSS and extract the S-parameter data to import it back into Virtuoso for full link simulation.
- Characterize link-performance by simulating the timing jitter as well as transient noise-profile within Virtuoso.
- Plot the eye-diagram at output of TX as well as output of channel to calculate the voltage margin and timing margin.
- In MATLAB, write a script to calculate the "worst-case" eye and estimate the BER performance of the link.
- Consider using CppSim simulator for system level modeling of the timing recovery circuit in order to accurately construct the transistor level model in Virtuoso.

Industry Standard: IBIS

- Provided as binary code
- Fast, efficient execution
- Protects vendor IP
- Extensible modeling capability
- Allows models to be developed in multiple languages
- Standardized execution interface
- Standardized control (.AMI) file

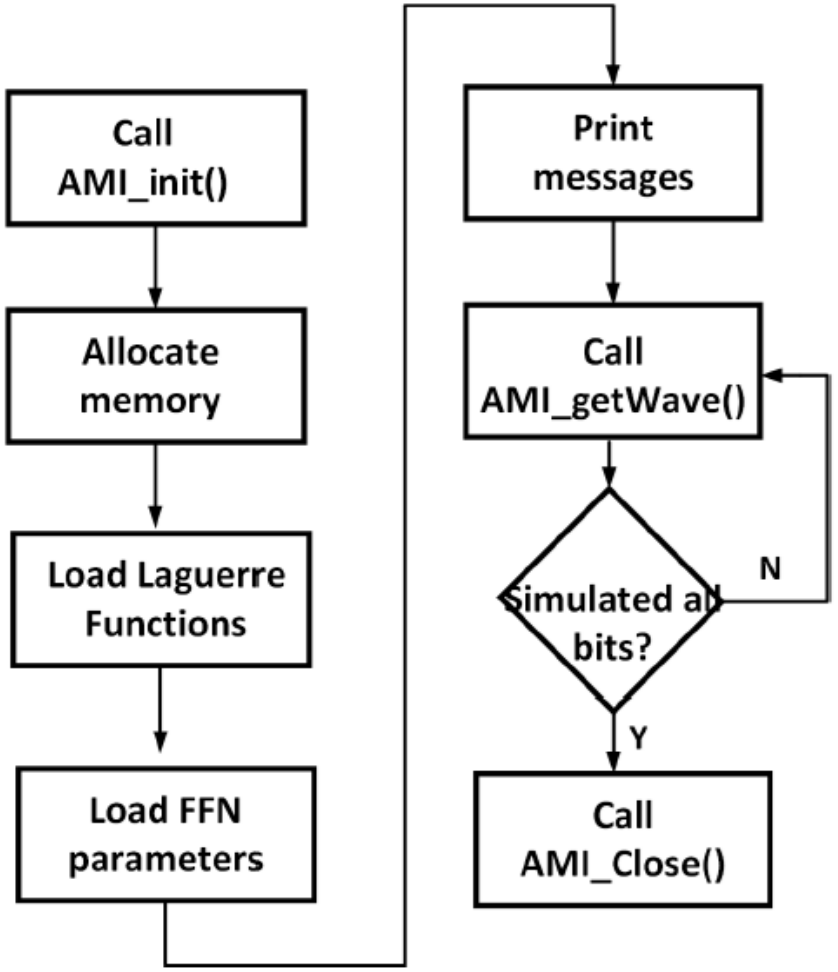
IBIS homepage: **http://www.eigroup.org/ibis/**

AMI

- AMI stands for Algorithmic Modeling Interface
  - faster signal processing algorithms
  - intellectual property protection
  - used in convolution transient engines
  - designed to be used with fixed time step data
  - introduced in IBIS 5.0 specs
  - in these specs the library is specified inside the IBIS wrapper

IBIS stands for “I/O Buffer Information Specification”; high-level buffer specification for circuit modeling
http://eda.org/pub/ibis/ver5.0/ver5_0.txt

AMI: Flow Diagram



AMI Challenges

- AMI models are compiled DLLs and text files
  - No graphical representation

- Package model standard not finalized
  - User needs to manually add IC/package parasitics to channel model

- Each IC vendor has different parameter set
  - No standards set
  - Each vendor must document their models

- No standard way to sweep parameters
  - Need to create multiple .AMI files
  - EDA tools need to parse arbitrary .AMI parameters

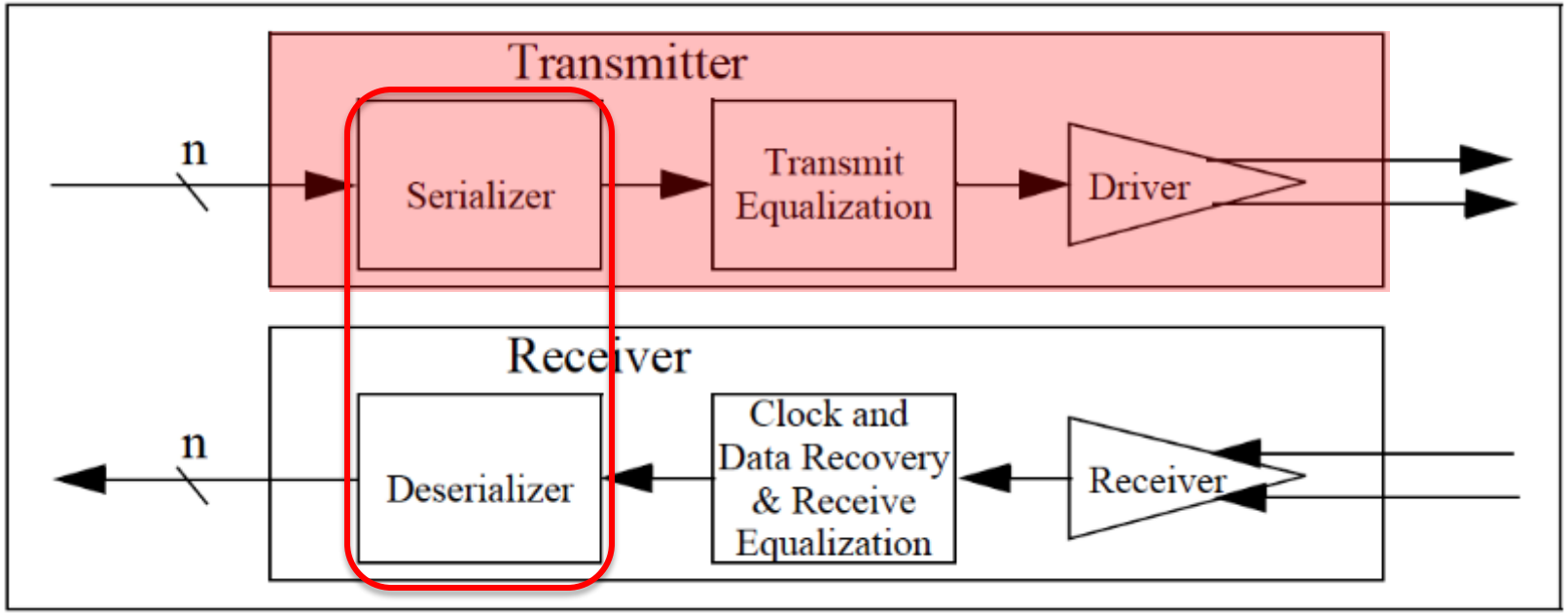
Why SERDES?

- Traditional parallel communication not suitable for inter-IC data transport in high-speed links.
  - High design overhead due to cross-talk, data-skew.
- Serial links are most cost-effective.
  - Parallel links = extra pins → Higher packaging costs.
  - Speed v/s cost tradeoff with serial links.
- Solution = SERDES!!!
  - Parallel communication still used in internal buses of ICs thus a need for SerDes.
  - Mitigate cost while maintaining high-speeds with a fast serial-parallel data conversion.

What is a SERDES?

- SERDES = SERializer – DESerializer
  - Used to transmit high speed IO-data over a serial link in I/O interfaces at speeds upwards of 2.5Gbps.
  - SerDes TX: transmit parallel data to receiver overhigh speed serial-link.
  - SerDes RX: receive data from serial-link and deliver parallel data to next-stage.
  - **Advantage**: Fast signaling, robust, high signal integrity.

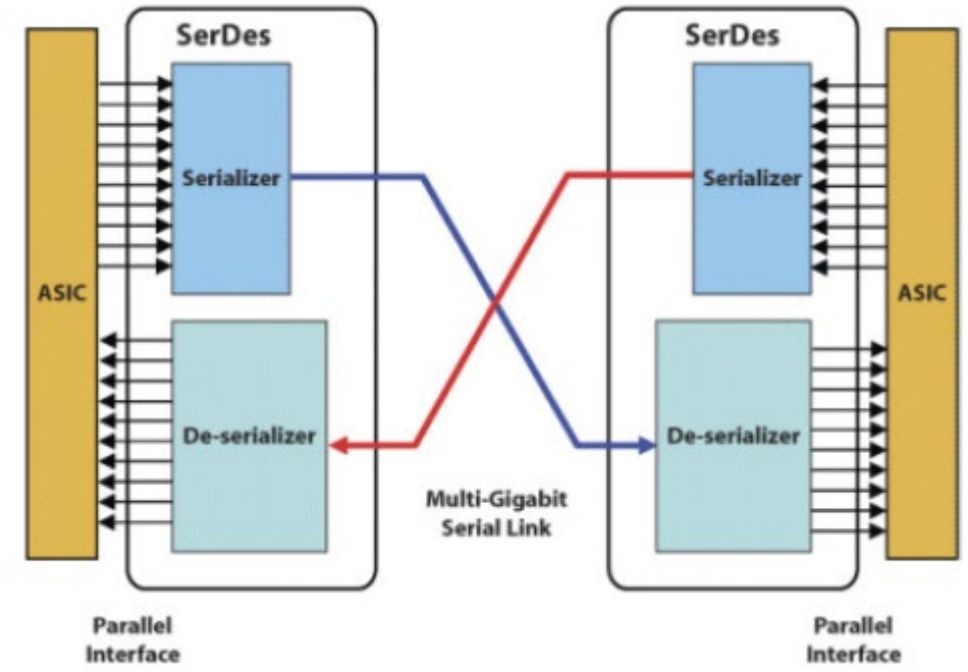
Serializer/Deserializer Blocks



- Serializer:
- Deserializer

SERDES

- Can be either a stand-alone or an IP core
- Converts parallel data into a serial data stream on TX side
- Converts the serial data back to parallel on RX side
- Timing skew solved by embedding clock signal into data stream
- PCI Express specifies a serial data rate of 2.5Gb/s



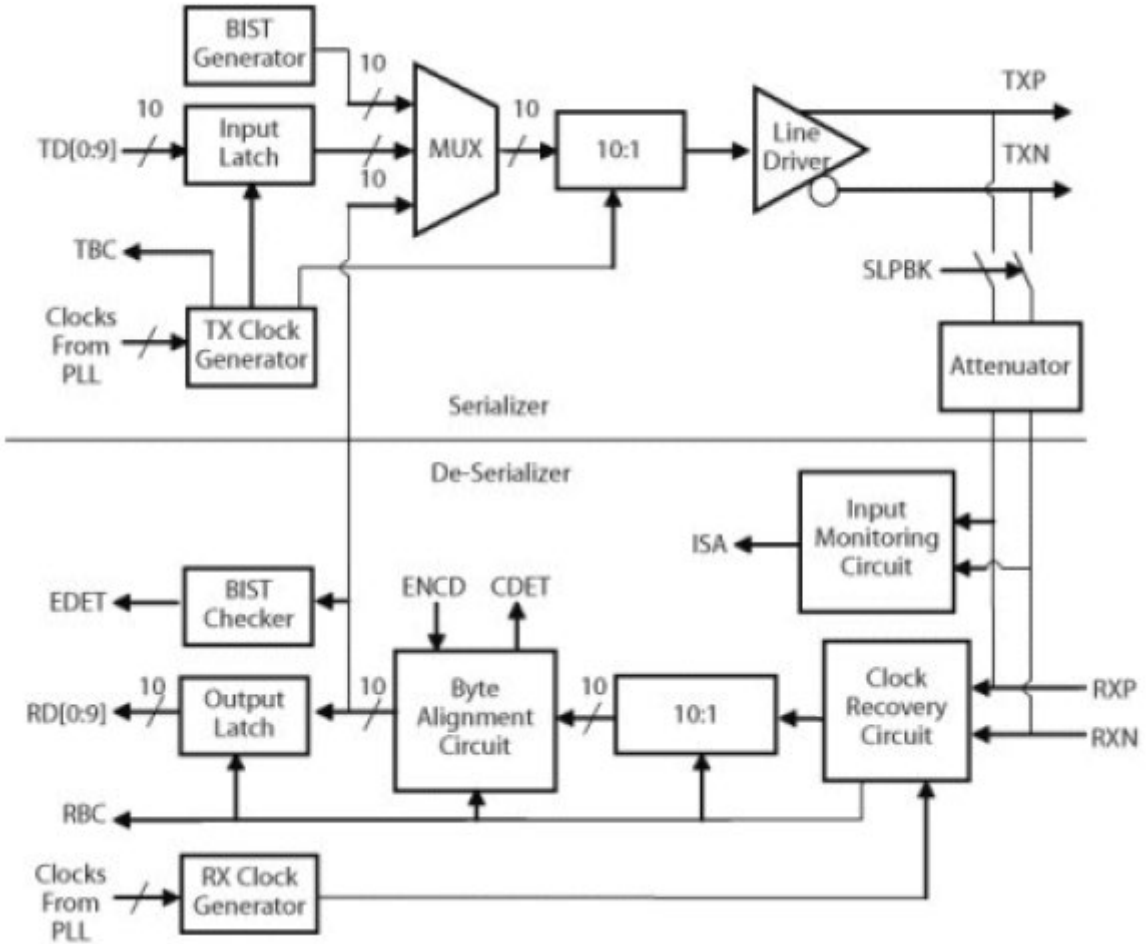
SERDES Characteristics

- Mixed-signal circuitry
- Must meet stringent electrical specifications
- Close to a year to complete the design.
- Another year + several design spins for mass-production
- More practical to license proven SerDes IP core
- Not all SerDes cores are created

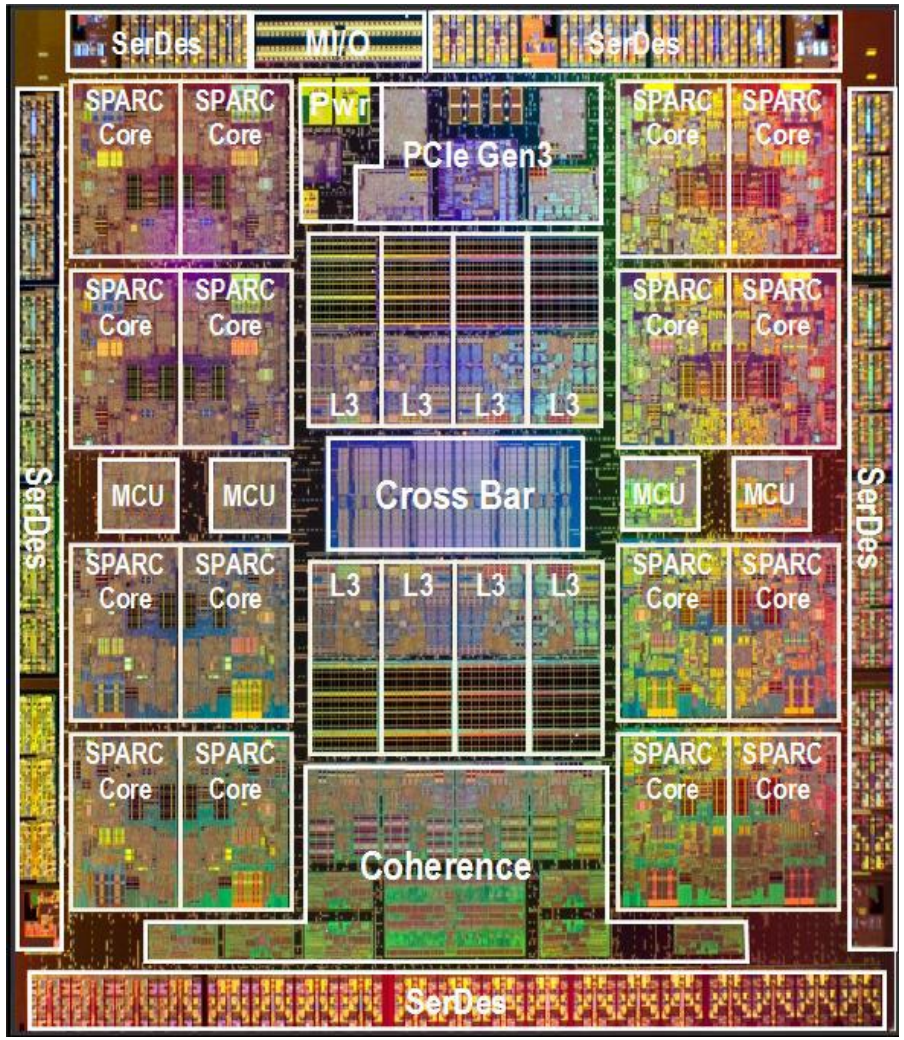
PCIExpress

- Parallel input data to Serializer is usually 10-bit wide
- 8b/10b encoding scheme to encode data.
- The 8b/10b encoder and decoder are pure digital circuitry
- Operate at one tenth of the serial data rate

PCIExpress SERDES



Serial Links in SoC: Oracle SPARC T5*



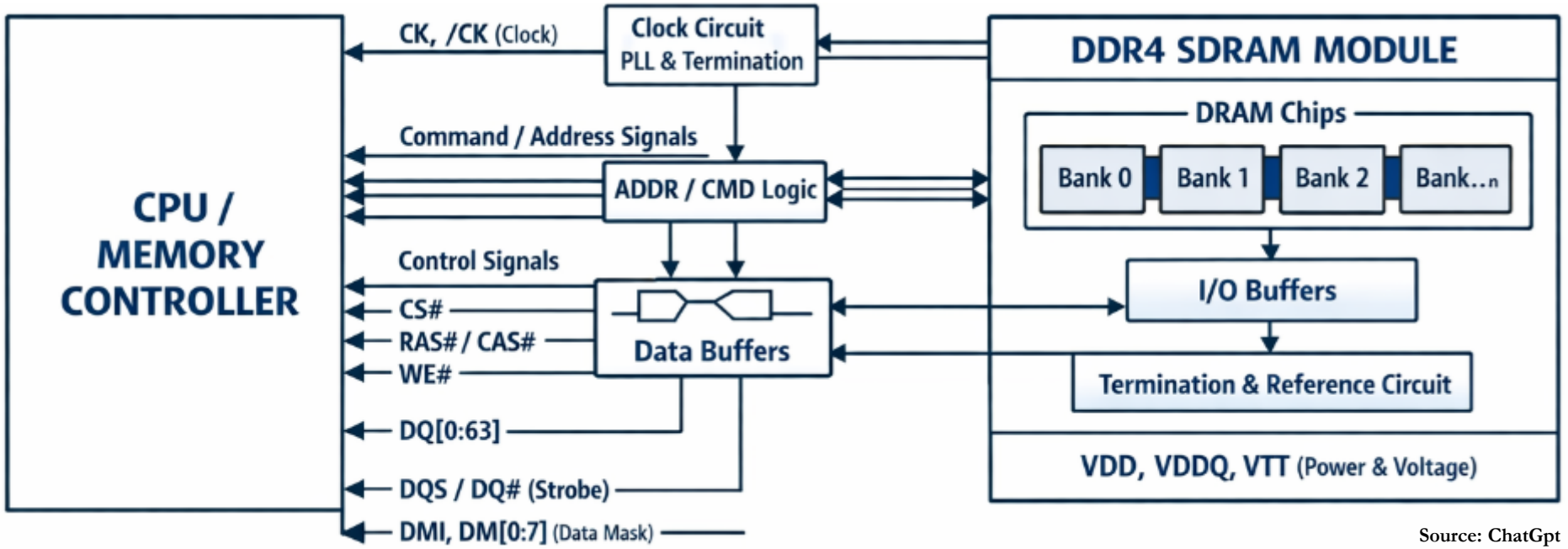
About 50 SerDes IPs
- **Single IP power & area**
- **Integrated with SoC**
- **Portable with SoC**

* J. Hart et al., "A 3.6GHz 16-Core SPARC SoC Processor in 28nm", Proceedings of the 2013 IEEE International Solid-State Circuits Conference.

Family of DDR

Generation	Max Data Rate	VDD	SI Limit
DDR3	1600 MT/s	1.5 V	Reflections
DDR4	3200 MT/s	1.2V	Loss + Xtalk
DDR5	6400 MT/s	1.1V	Loss + Jitter

DDR4 Memory Interface



Source: ChatGpt

DDR Architecture

Bank Groups

Bank groups increase concurrency but introduce command timing complexity, indirectly impacting SI through burst scheduling and SSN patterns.

Multi-Drop Channels

DDR4 DIMMs commonly use fly-by routing, creating controlled stubs.
Electrically:

- Stubs introduce impedance discontinuities
- Reflections must be absorbed via ODT

DDR5 Architecture

DDR5 introduces two independent 32-bit channels per DIMM, fundamentally altering channel EM behavior.

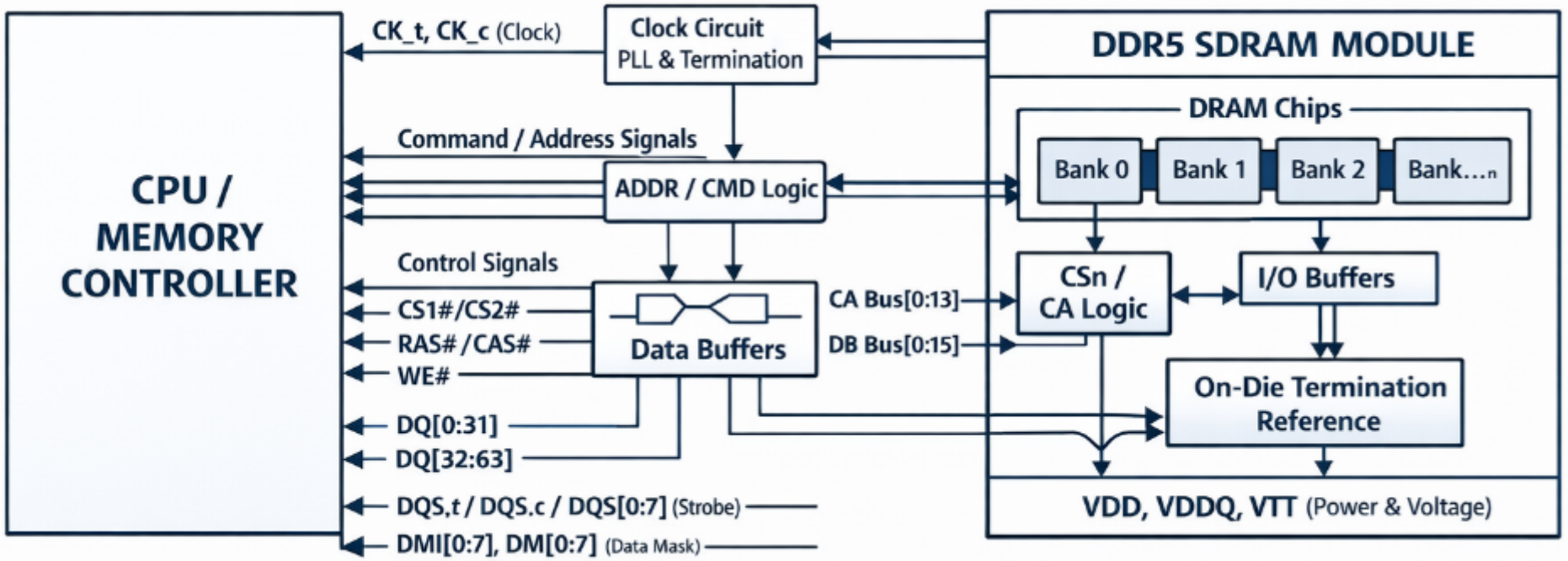
EM Advantages

- Shorter electrical channel per sub-channel
- Reduced loading
- Improved eye opening at high data rates

On-DIMM Voltage Regulation

- Local regulation reduces PDN inductance but introduces switching noise coupling challenges.

DDR5 Memory Interface



Source: ChatGpt

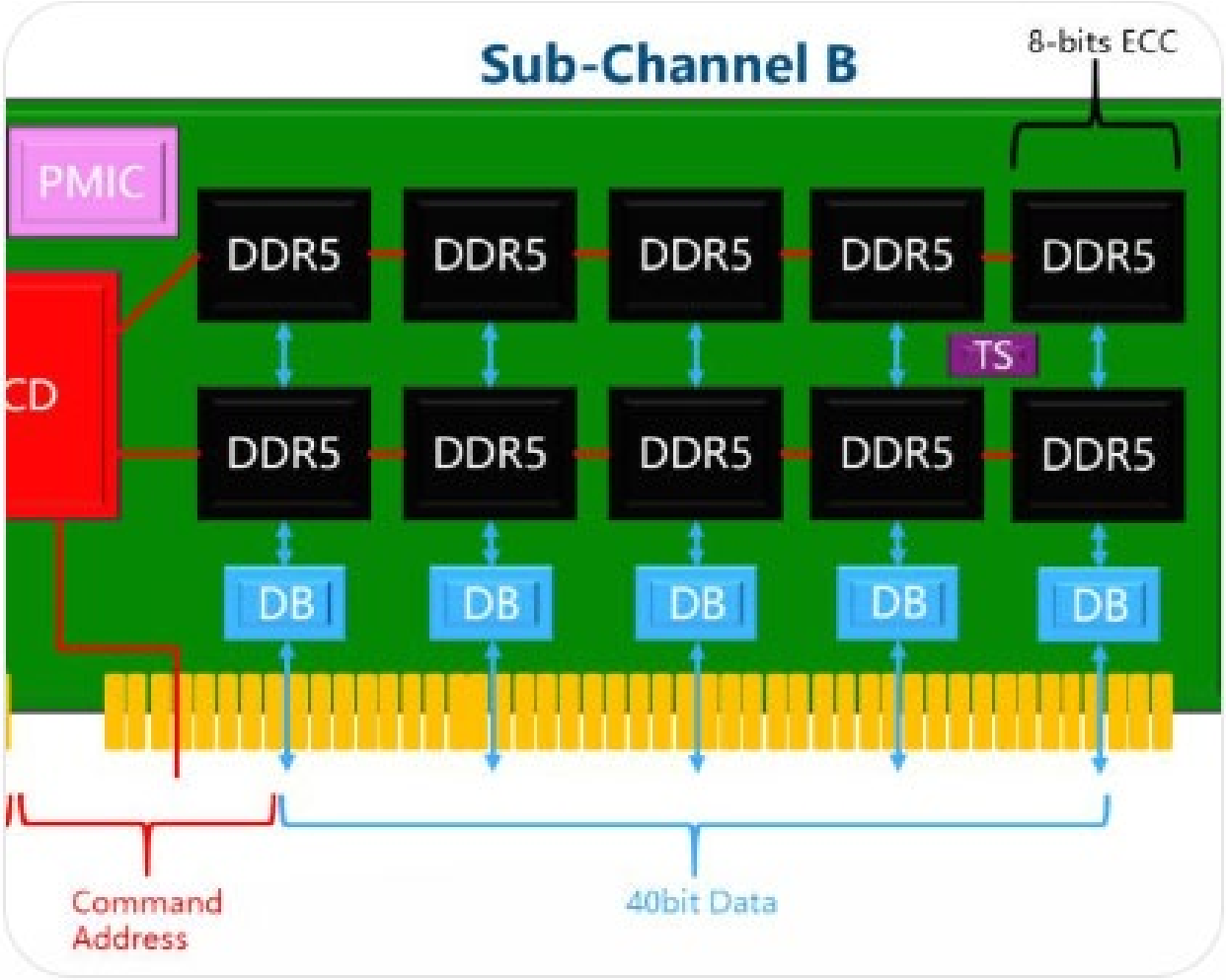
DDR5 Training

DDR5 needs training because the memory channel is so fast that fixed timing values are no longer reliable.

At multi-gigabit per pin data rates, small differences in trace length, package delay, voltage, temperature, noise, and DIMM variation can cause skew with small margin.

When system powers on, memory controller and DDR5 DRAM perform **training** to automatically calibrate the interface.

DDR5 Layout



Source: ChatGpt

Crosstalk Mitigation Techniques

Objective

- Alleviate crosstalk noise in parallel buses
- Requires knowledge of coupling parameters

Approach

- Use Encoding

Multiconductor Theory

- Line bundle can be described by matrices per unit length
- $$\mathbf{Z}=\mathbf{R}+\mathrm{j}\omega\mathbf{L},\ \mathbf{Y}=\mathbf{G}+\mathrm{j}\omega\mathbf{C}$$
- Telegrapher's equations in frequency domain reveal coupling

$$\frac{d^2\mathbf{V}}{dz^2}=(\mathbf{ZY})\mathbf{V} \qquad \frac{d^2\mathbf{I}}{dz^2}=(\mathbf{YZ})\mathbf{I}$$

- Goal: introduce modal variables, diagonalizing the line equations
- Issue: For lines with discontinuities, Z and Y change over length

- Only interested in voltages/currents at ends of the channel
- Start by describing the channel by its ABCD-parameters (one choice):

$$\begin{bmatrix} v_S \\ i_S \end{bmatrix}=\begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} v_R \\ i_R \end{bmatrix}$$

Eigenvalues and Eigenvectors

$$ELCE^{-1} = \Lambda_m^2$$

gives

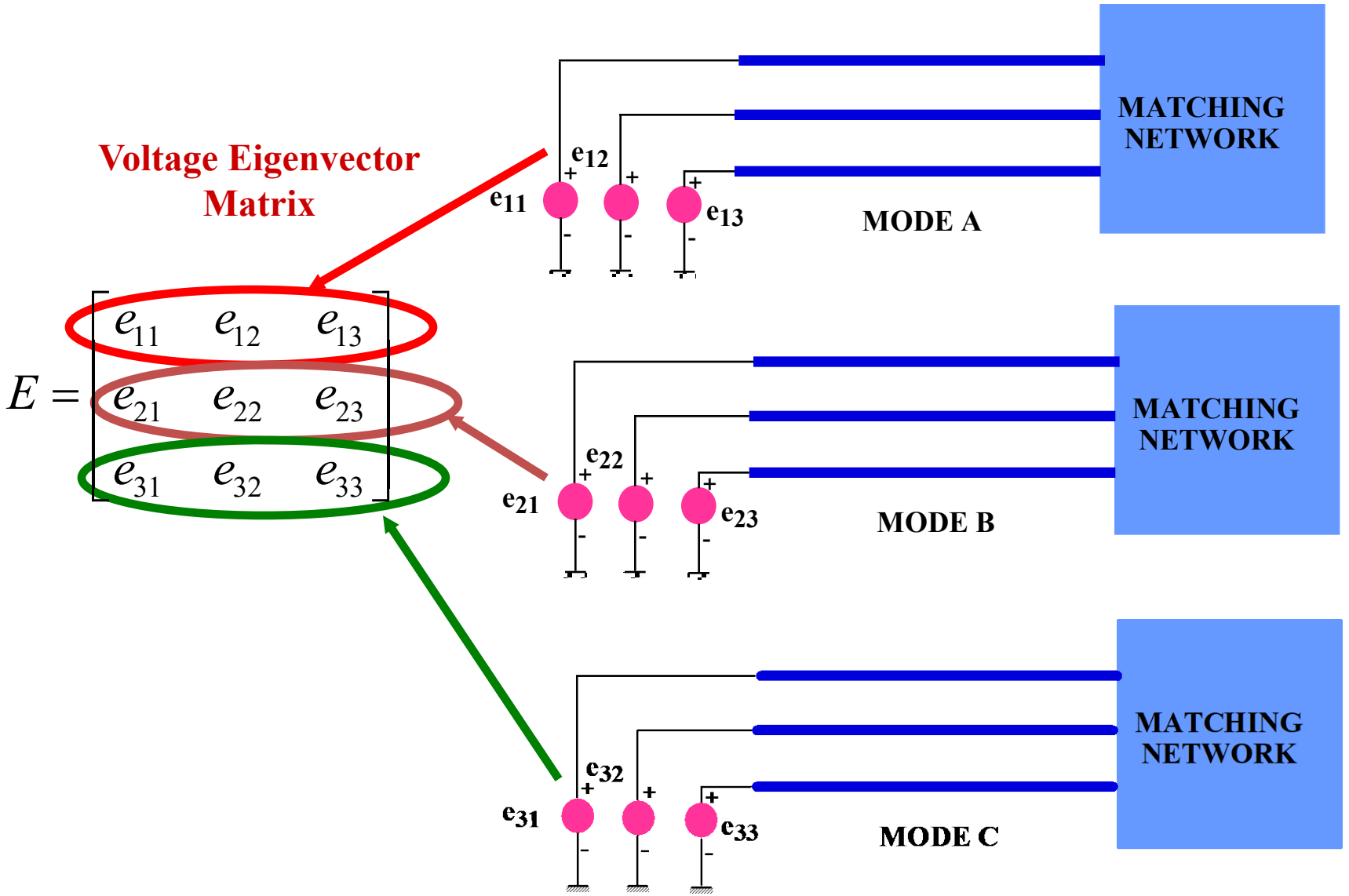
$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \qquad \Lambda_m = \begin{bmatrix} \frac{1}{v_{m1}} & 0 & 0 \\ 0 & \frac{1}{v_{m2}} & 0 \\ 0 & 0 & \frac{1}{v_{m3}} \end{bmatrix}$$

$$HCLH^{-1} = \Lambda_m^2$$

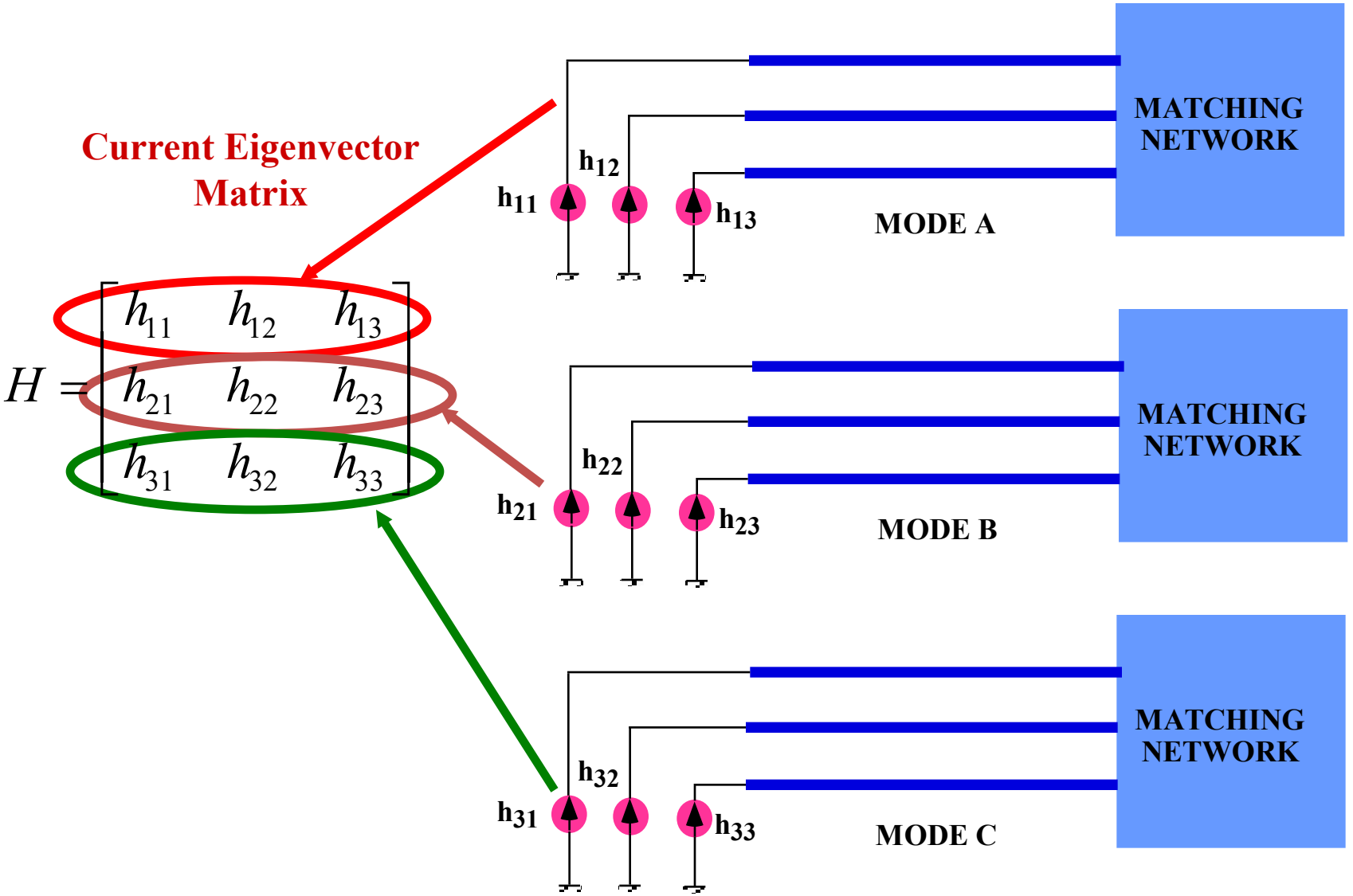
gives

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \qquad \Lambda_m = \begin{bmatrix} \frac{1}{v_{m1}} & 0 & 0 \\ 0 & \frac{1}{v_{m2}} & 0 \\ 0 & 0 & \frac{1}{v_{m3}} \end{bmatrix}$$

Modal Voltage Excitation

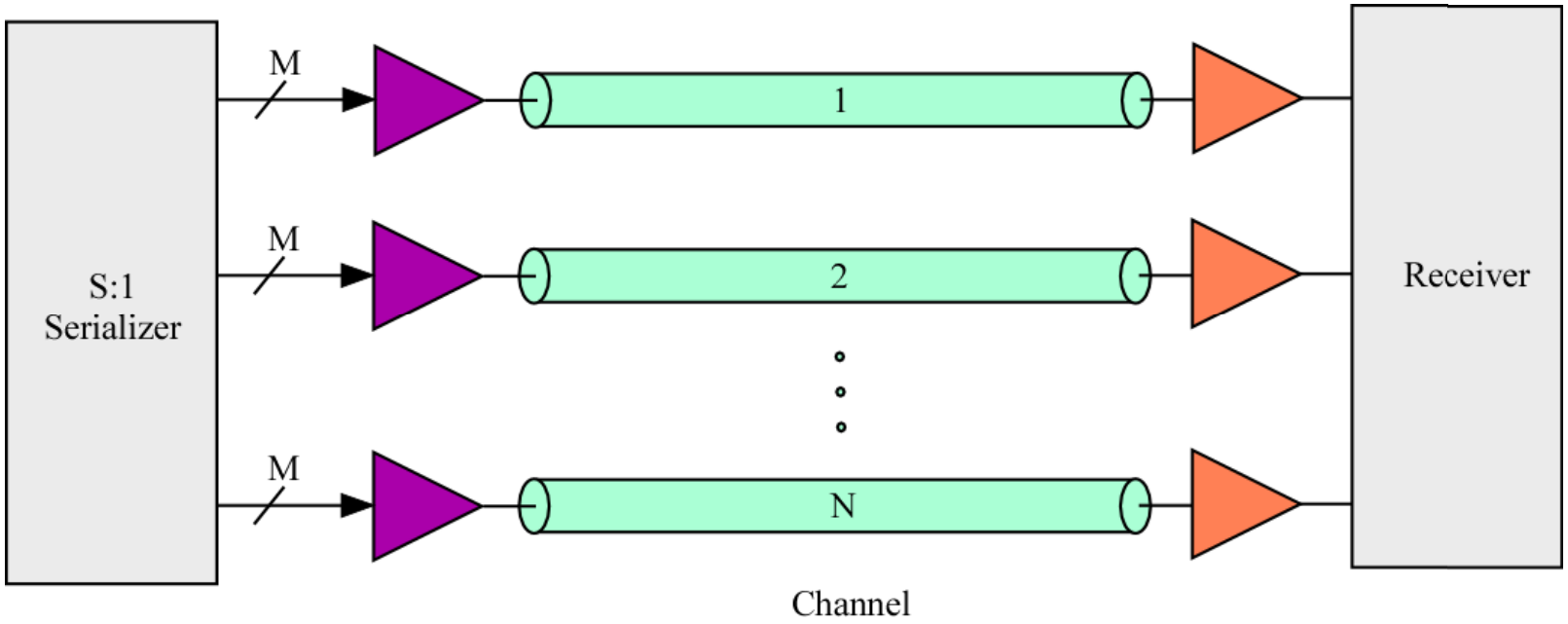


Modal Current Excitation



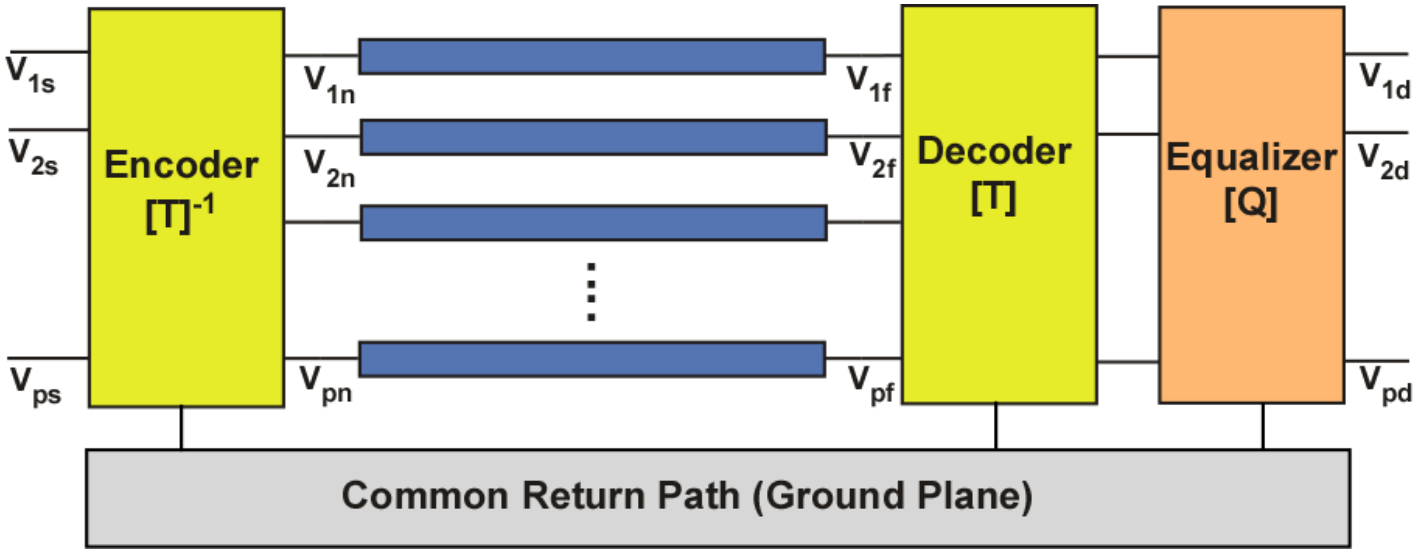
Crosstalk – Uniform Channel

Channel consists of uniform transmission lines



Crosstalk can be described by multi-conductor TL theory

Crosstalk Mitigation in Parallel Buses



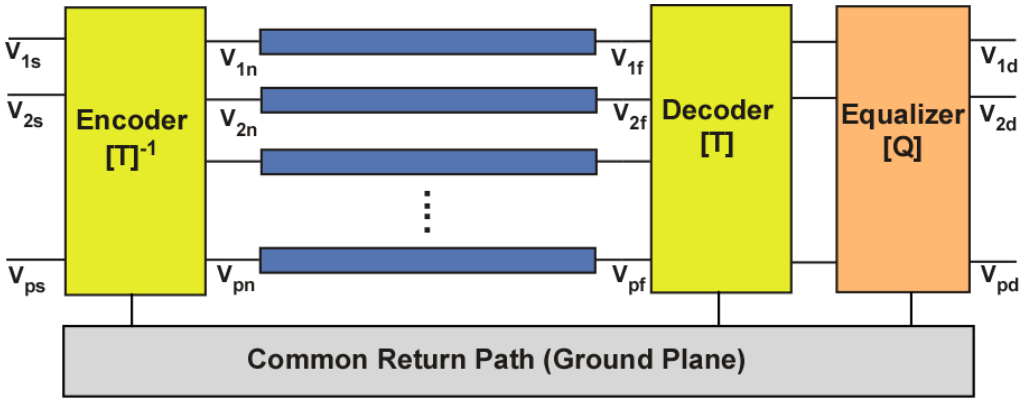
$$\mathbf{V_{sn}} = \begin{bmatrix} V_{1s} \\ V_{2s} \\ . \\ V_{ps} \end{bmatrix} \qquad \mathbf{V_{ln}} = \begin{bmatrix} V_{1n} \\ V_{2n} \\ . \\ V_{pn} \end{bmatrix} \qquad \mathbf{V_{lf}} = \begin{bmatrix} V_{1f} \\ V_{2f} \\ . \\ V_{pf} \end{bmatrix} \qquad \mathbf{V_{df}} = \begin{bmatrix} V_{1d} \\ V_{2d} \\ . \\ V_{pd} \end{bmatrix}$$

Crosstalk Mitigation in Parallel Buses

$$\mathbf{V}_{mn} = \mathbf{E}\mathbf{V}_{ln}$$

where $\mathbf{V}_{ln}$ is the *line* voltage vector and $\mathbf{V}_{mn}$ is the *modal* voltage vector at the near end. $\mathbf{E}$ is the voltage eigenvector matrix associated with the multi-conductor system. In general, $\mathbf{E}$ will be complex and a function of frequency. The modal voltage vector at the far end, $\mathbf{V}_{mf}$ will be given by:

$$\mathbf{V}_{mf} = \mathbf{X}_m \mathbf{V}_{mn}$$



Crosstalk Mitigation in Parallel Buses

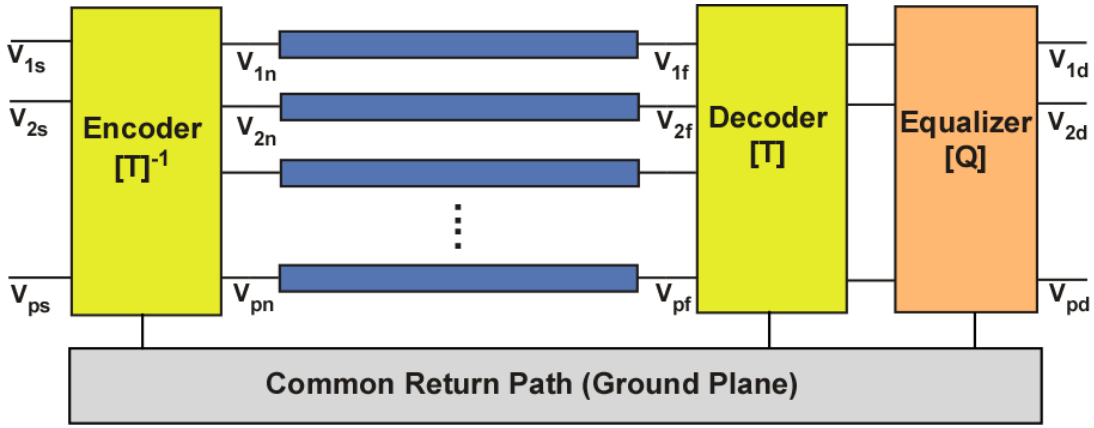
$\mathbf{X_m}$ is the complex propagation matrix function given by

$$\mathbf{X_m} = \begin{bmatrix} e^{-\alpha_1 l - j\beta_1 l} & & & \\ & e^{-\alpha_2 l - j\beta_2 l} & & \\ & & . & \\ & & & e^{-\alpha_p l - j\beta_p l} \end{bmatrix}$$

in which $\alpha_i + j\beta_i$ is the complex propagation constant, associated with the ith mode and l is the length of the lines. In terms of near-end signals this reads

Crosstalk Mitigation in Parallel Buses

$$\mathbf{V_{mf} = X_m E V_{ln}}$$



The far-end line voltage vector, $\mathbf{V_{lf}}$ can be recovered using:

$$\mathbf{V_{lf} = E^{-1} V_{mf} = E^{-1} X_m E V_{ln}}$$

Crosstalk Mitigation in Parallel Buses

Now, assume that the information signals are encoded with the encoder **T** such that the signals are mapped to the orthogonal modes, as follows:

$$\mathbf{V_{ln}} = \mathbf{T^{-1}V_{sn}}$$

At the far end the decoded voltage vector would be given by:

$$\mathbf{V_{df}} = \mathbf{QTV_{lf}}$$

where **Q** is an equalization matrix representing any equalizer box that might be implemented at the output of the channel, we get

Crosstalk Mitigation in Parallel Buses

$$\mathbf{V_{df}} = \mathbf{QTE^{-1}X_mET^{-1}V_{sn}}$$

If we choose **T=E** we obtain

$$\mathbf{V_{df}} = \mathbf{QX_mV_{sn}}$$

$$\begin{bmatrix} V_{1d} \\ V_{2d} \\ . \\ V_{pd} \end{bmatrix} = \mathbf{Q} \begin{bmatrix} e^{-\alpha_1 l - j\beta_1 l} & & & \\ & e^{-\alpha_2 l - j\beta_2 l} & & \\ & & . & \\ & & & e^{-\alpha_p l - j\beta_p l} \end{bmatrix} \begin{bmatrix} V_{1s} \\ V_{2s} \\ . \\ V_{ps} \end{bmatrix}$$

Crosstalk Mitigation in Parallel Buses

If in addition, we implement an equalizer with property

$$\mathbf{Q} = \begin{bmatrix} e^{+\alpha_1 l} & & & \\ & e^{+\alpha_2 l} & & \\ & & . & \\ & & & e^{+\alpha_p l} \end{bmatrix}$$

this gives

$$\begin{bmatrix} V_{1d} \\ V_{2d} \\ . \\ V_{pd} \end{bmatrix} = \begin{bmatrix} e^{-j\beta_1 l} & & & \\ & e^{-j\beta_2 l} & & \\ & & . & \\ & & & e^{-j\beta_p l} \end{bmatrix} \begin{bmatrix} V_{1s} \\ V_{2s} \\ . \\ V_{ps} \end{bmatrix} = \begin{bmatrix} e^{-j\frac{\omega l}{v_{m1}}} & & & \\ & e^{-j\frac{\omega l}{v_{m2}}} & & \\ & & . & \\ & & & e^{-j\frac{\omega l}{v_{mp}}} \end{bmatrix} \begin{bmatrix} V_{1s} \\ V_{2s} \\ . \\ V_{ps} \end{bmatrix}$$

Crosstalk Mitigation in Parallel Buses

$$\begin{bmatrix} V_{1d} \\ V_{2d} \\ . \\ V_{pd} \end{bmatrix} = \begin{bmatrix} e^{-j\beta_1 l} & & & \\ & e^{-j\beta_2 l} & & \\ & & . & \\ & & & e^{-j\beta_p l} \end{bmatrix} \begin{bmatrix} V_{1s} \\ V_{2s} \\ . \\ V_{ps} \end{bmatrix} = \begin{bmatrix} e^{-j\frac{\omega l}{v_{m1}}} & & & \\ & e^{-j\frac{\omega l}{v_{m2}}} & & \\ & & . & \\ & & & e^{-j\frac{\omega l}{v_{mp}}} \end{bmatrix} \begin{bmatrix} V_{1s} \\ V_{2s} \\ . \\ V_{ps} \end{bmatrix}$$

in which we used the relation $\beta_i = \omega/v_{mi}$. This shows that if the proper encoder, decoder and equalizer can be implemented, ***all signals can be perfectly reconstructed, with no crosstalk, no attenuation and no dispersion.***

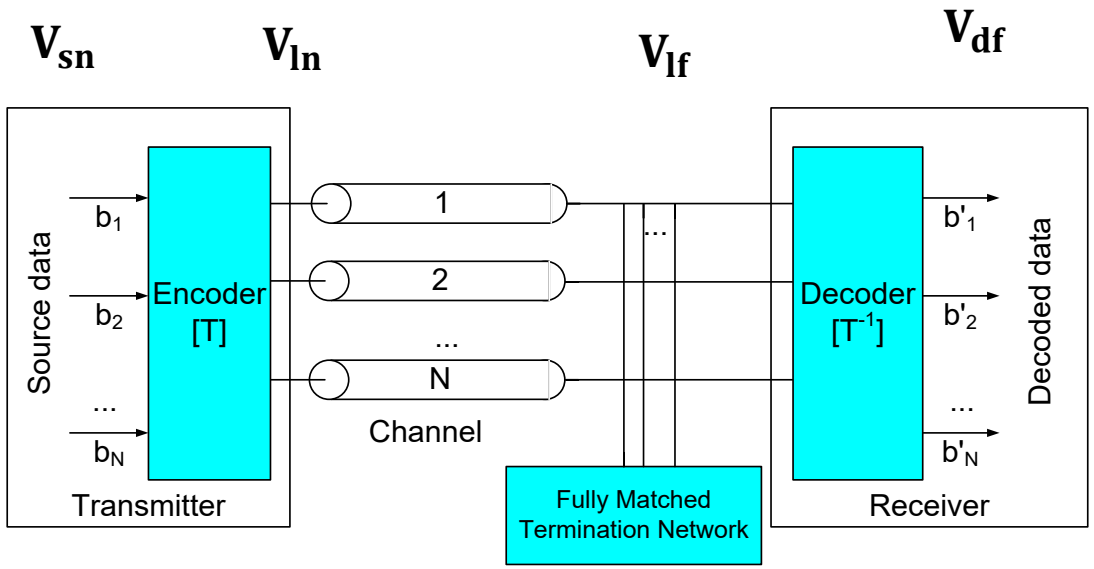
In the special case where the lines are lossless, $\alpha_i = 0$, Q= I (the identity matrix) and no equalization is needed. Also E is real and does not depend on frequency.

Modal Signaling System – Ideal Lines

$$\mathbf{V_{mn} = EV_{ln}}$$

$$\mathbf{V_{mf} = X_m V_{mn}}$$

$$\mathbf{V_{mf} = X_m E V_{ln}}$$



$$\mathbf{V_{lf} = E^{-1} V_{mf} = E^{-1} X_m E V_{ln}}$$

E: Eigenvector matrix

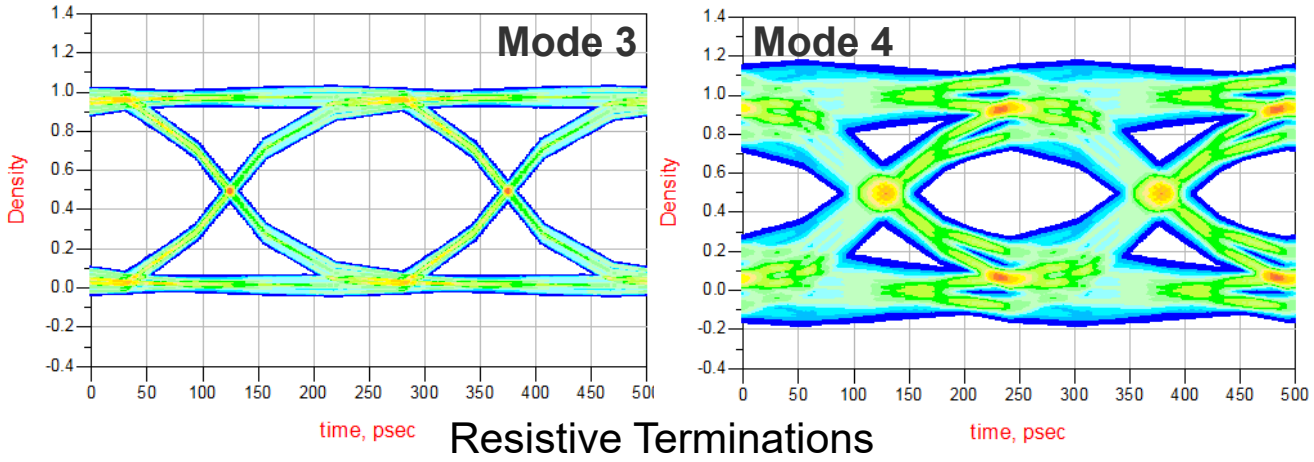
$$\mathbf{V_{df} = T E^{-1} X_m E T^{-1} V_{sn}}$$

X_m: Propagation matrix (diagonal)

If we choose $T=E^{-1}$ all signals are perfectly reconstructed

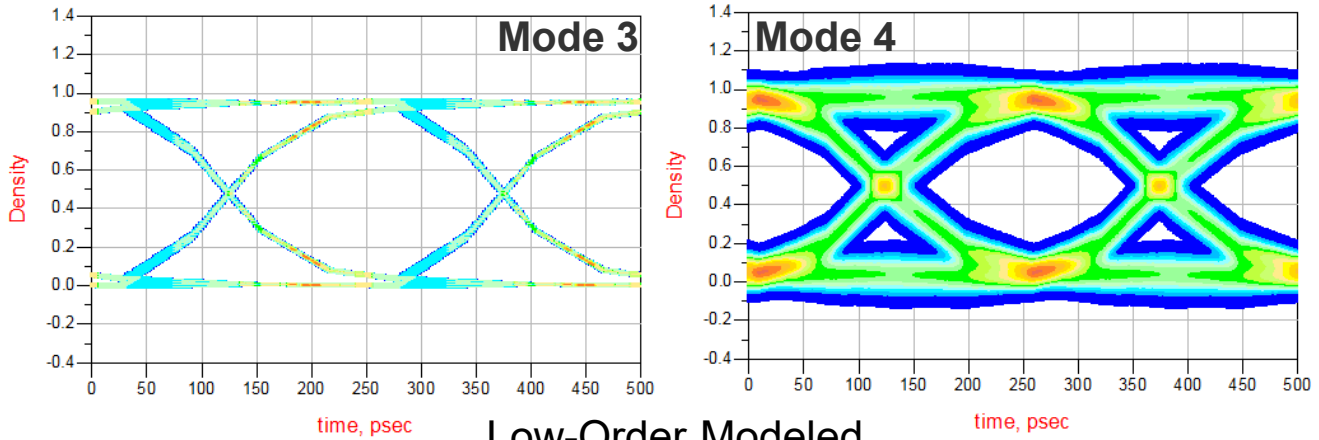
Performance Comparison of the Termination Networks

- Statistical eye diagrams of 4 Gb/s NRZ, t_r=67ps, all modes switching
  - Only 2 out of 4 channels shown
- *Note: channel for which uncoded eye was closed*



Resistive Terminations

- Vertical eye opening increase of 39%
- Reduction in peak-to-peak jitter of 27%
- "Ground mode" #4 suffers from ISI of internal reflections

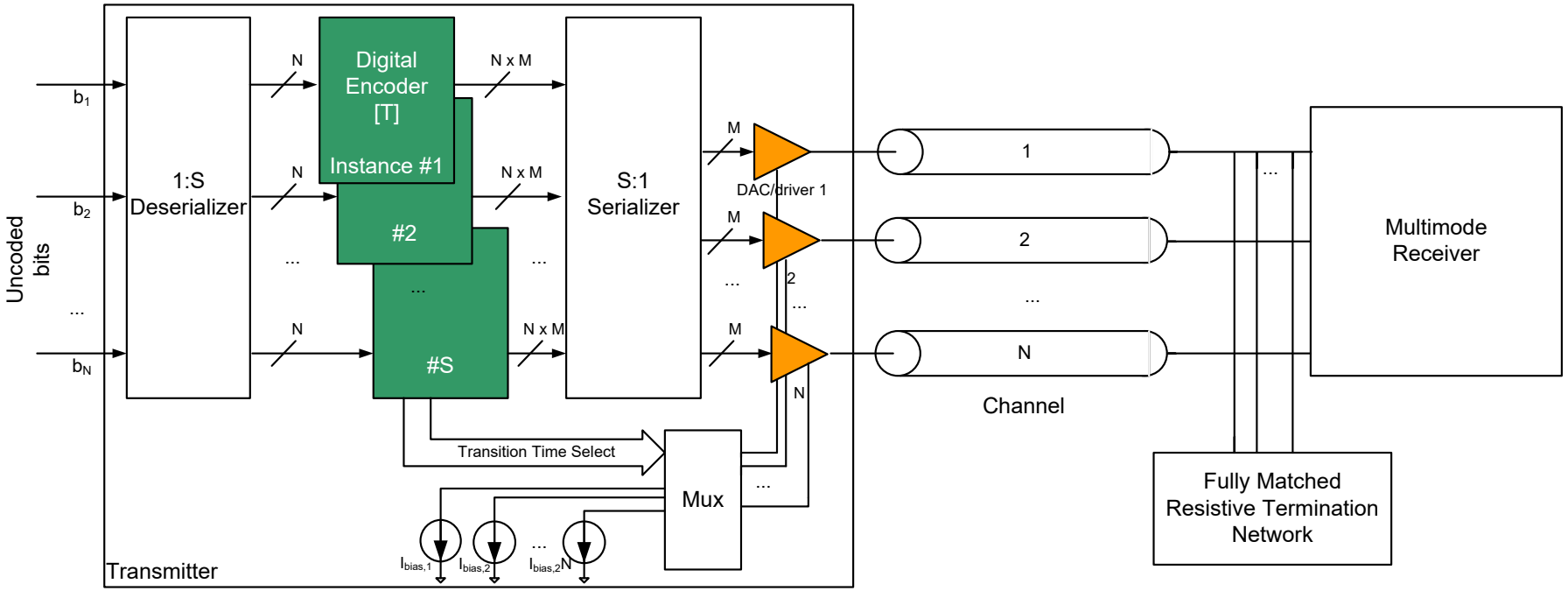


Low-Order Modeled

Milosevic, P., Schutt-Ainé, J.E., "Transceiver Design for High-Density Links With Discontinuities Using Modal Signaling", *IEEE Trans. Comp. Packaging. Manuf. Tech., vol. 3*, pp. 10-20, January 2013.

Physical Realization (1)
DSP-based Encoding

- DSP encoder directly calculates final transition values
- DAC/line drivers need to generate proper transition waveforms
- Most suited to Tx with DSP core (and SerDes) already in place

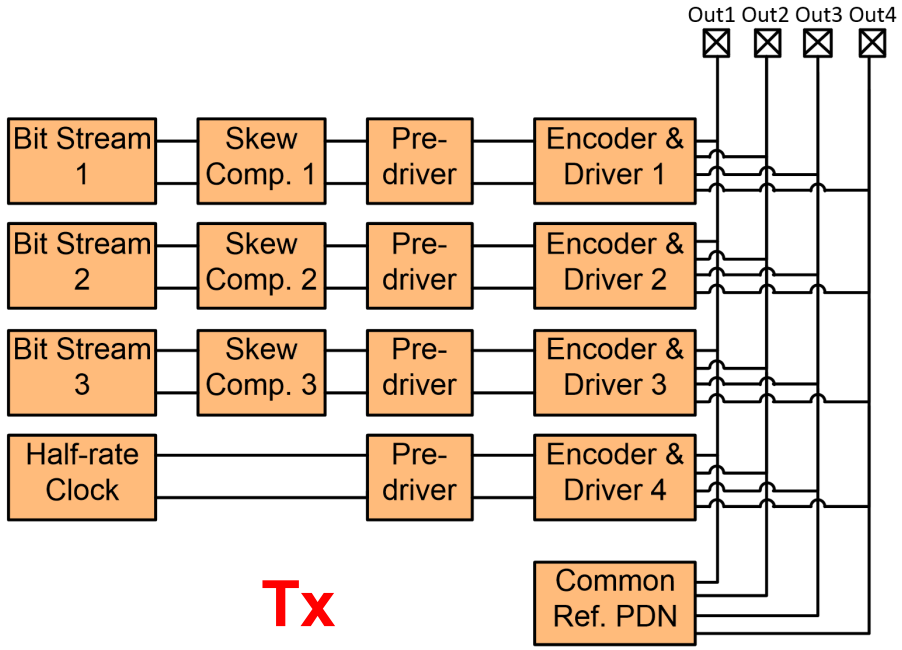


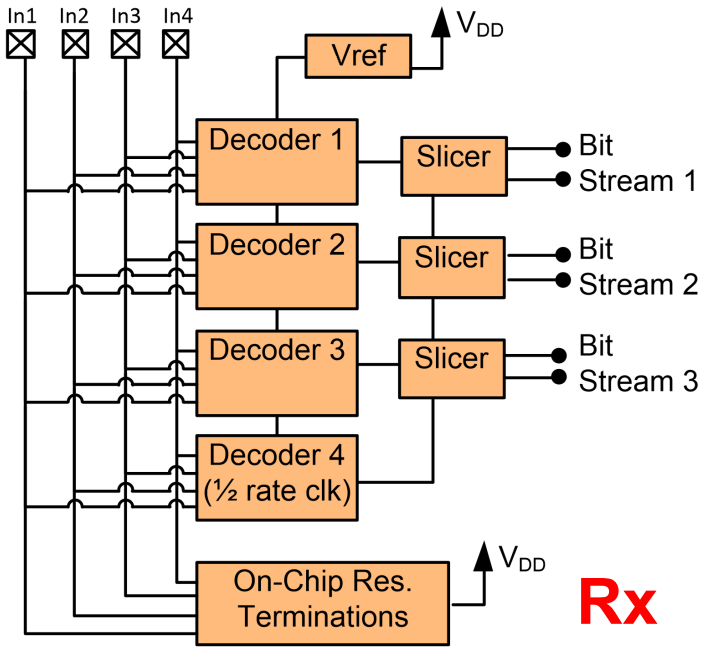
Milosevic, P., Schutt-Ainé, J.E., "Transceiver Design for High-Density Links With Discontinuities Using Modal Signaling", *IEEE Trans. Comp. Packaging. Manuf. Tech., vol. 3*, pp. 10-20, January 2013.

Physical Realization (2) Analog Frontend

- Channel: 4-line 4-inch pkg-PCB-pkg bus
- 3 bitstreams x 4 Gb/s = 12 Gb/s
- Forwarded clock uses ground mode
  - Half rate (2Gb/s) to alleviate limited bandwidth
  - This allows simple resistive terminations

P. Milošević and J. Schutt-Ainé, "Design of a 12Gb/s Transceiver for High-Density Links with Discontinuities using Modal Signaling" *Conf. on Electrical Performance of Electronic Packaging and Systems, 2011*

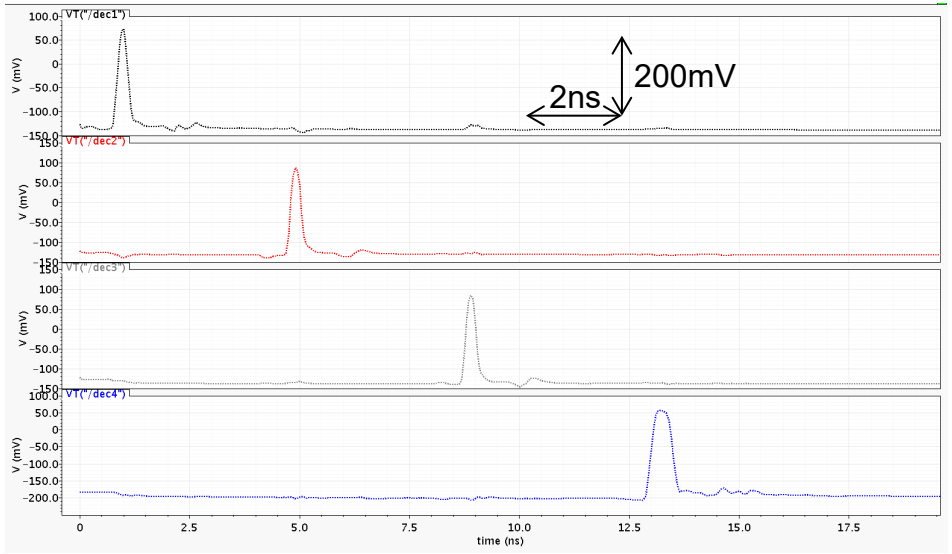




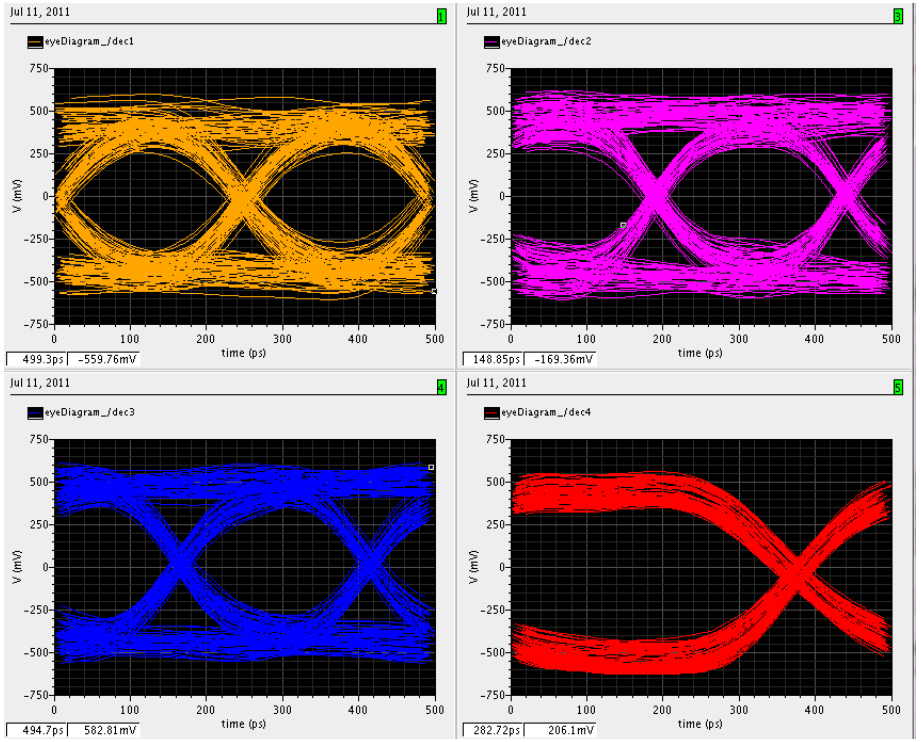
Milosevic, P., Schutt-Ainé, J.E., "Transceiver Design for High-Density Links With Discontinuities Using Modal Signaling", *IEEE Trans. Comp. Packaging. Manuf. Tech., vol. 3*, pp. 10-20, January 2013.

Modal Signaling – Circuit-level Results

- Process used: IBM 90 nm low-power digital RF, 1.2 V supply
- Encoder/Driver (w/o pre-drivers): 11.0 mW (0.92 mW/Gb/s), 6500μm^2
- Decoder overhead (w/o slicers): 14.5 mW (1.20 mW/Gb/s), 4300μm^2

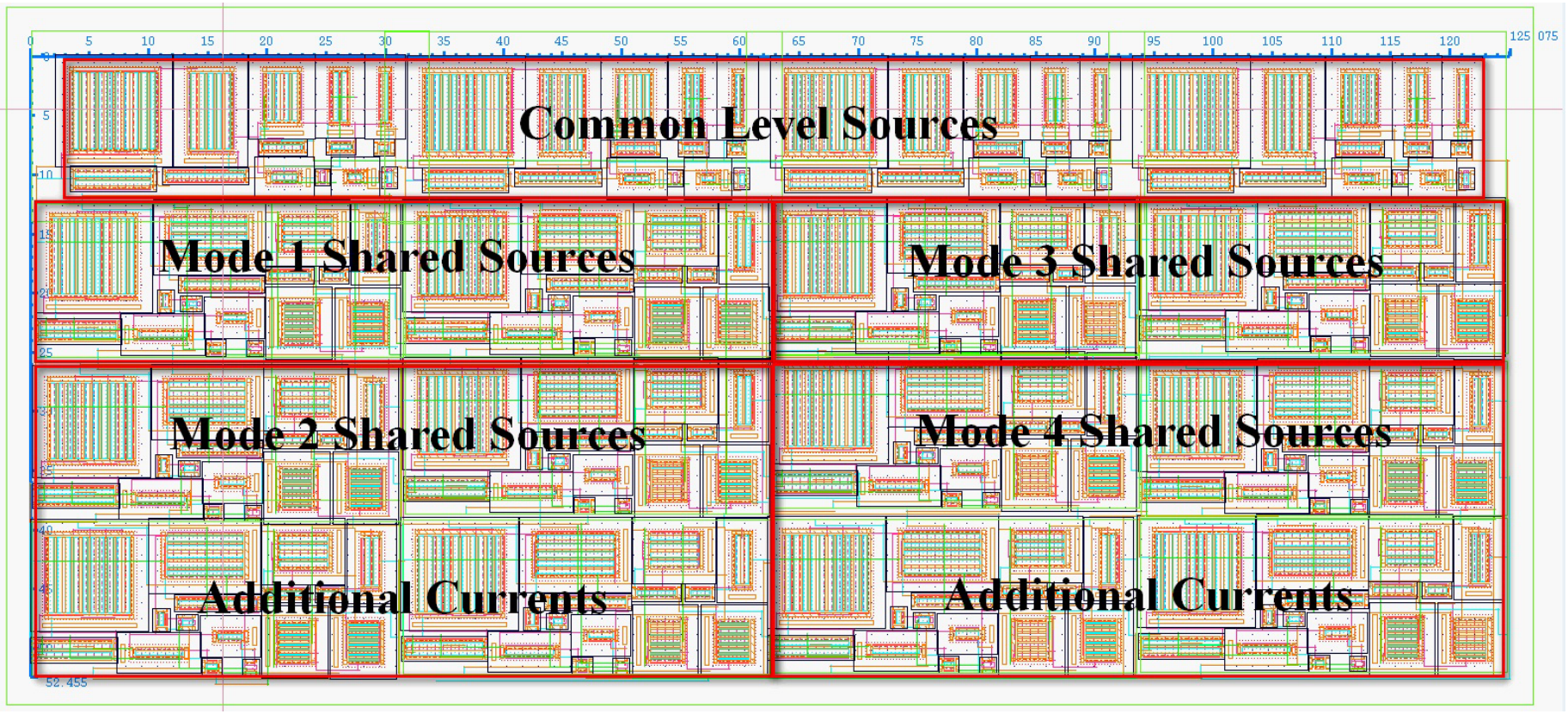


Unit pulse responses of signals over equivalent modal channels



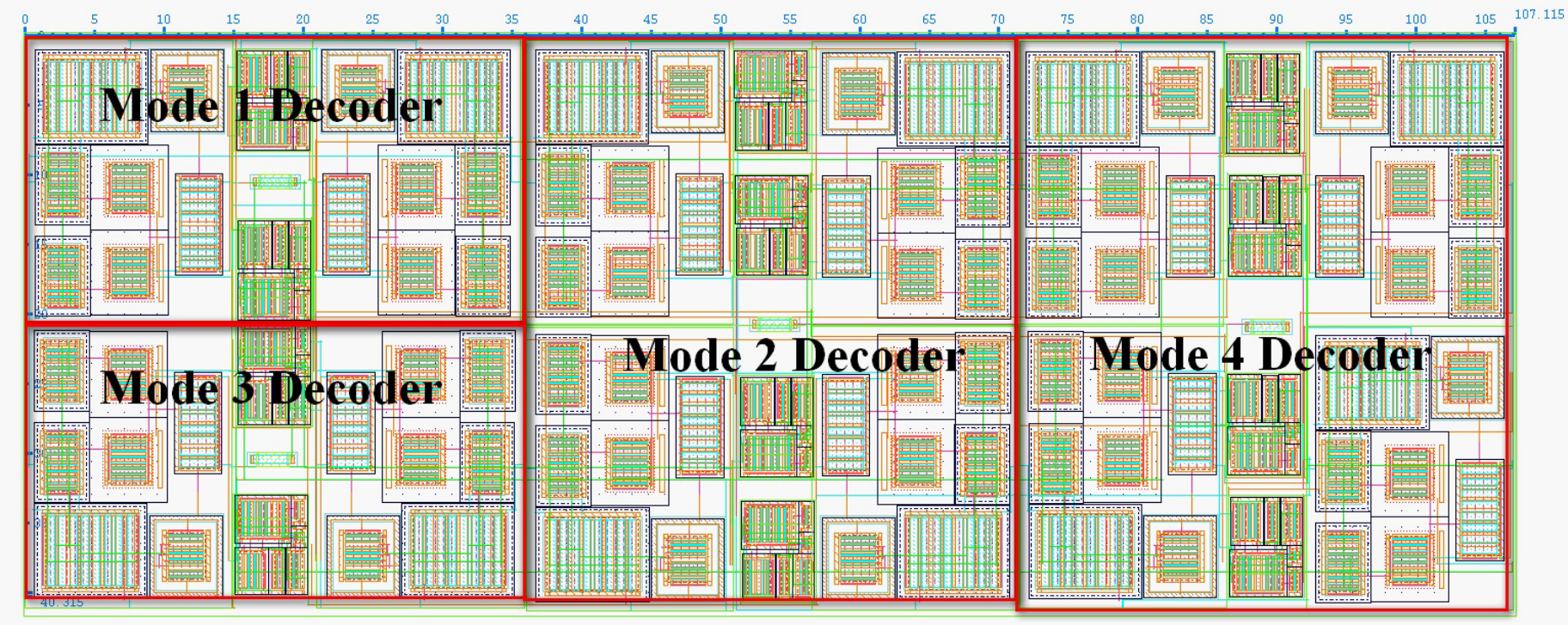
Normalized eye diagrams of decoded modal signals

Encoder Layout



Milosevic, P., Schutt-Ainé, J.E., "Transceiver Design for High-Density Links With Discontinuities Using Modal Signaling", *IEEE Trans. Comp. Packaging. Manuf. Tech., vol. 3*, pp. 10-20, January 2013.

Decoder Layout



Milosevic, P., Schutt-Ainé, J.E., "Transceiver Design for High-Density Links With Discontinuities Using Modal Signaling", *IEEE Trans. Comp. Packaging. Manuf. Tech., vol. 3*, pp. 10-20, January 2013.