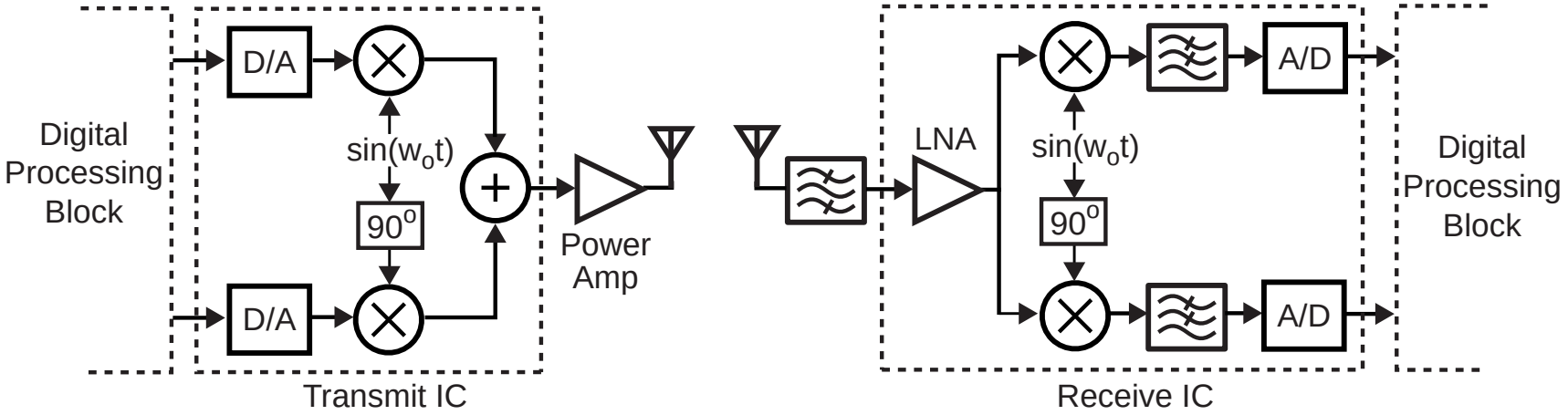
# *High Speed Communication Circuits and Systems*
# *Lecture 1*
# *Overview of Course*

**Michael Perrott**

**February 4, 2004**

# Wireless Systems

- **Direct conversion architecture**



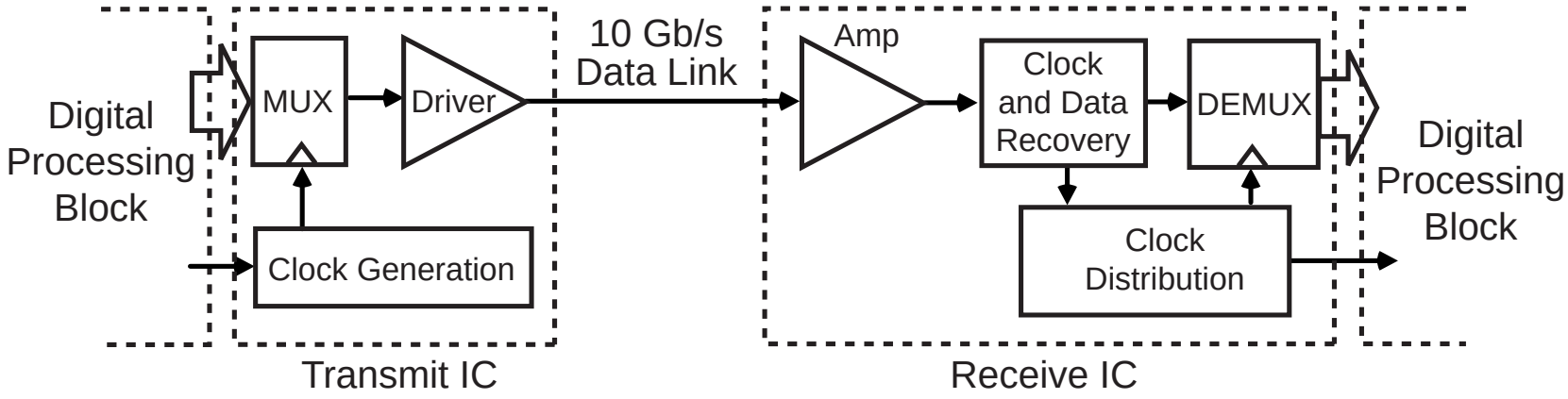

- **Transmitter issues**
  - **Meeting the spectral mask (LO phase noise & feedthrough, quadrature accuracy), D/A accuracy, power amp linearity**
- **Receiver Issues**
  - **Meeting SNR (Noise figure, blocking performance, channel selectivity, LO phase noise, A/D nonlinearity and noise), selectivity (filtering), and emission requirements**

# *Future Goals*

- **Low cost, low power, and small area solutions**
  - **New architectures and circuits!**
- **Increased spectral efficiency**
  - **Example: GSM cellphones (GMSK) to 8-PSK (Edge)**
    - Requires a linear power amplifier!
- **Increased data rates**
  - **Example: 802.11b (11 Mb/s) to 802.11a (> 50 Mb/s)**
    - GFSK modulation changes to OFDM modulation
- **Higher carrier frequencies**
  - **802.11b (2.5 GHz) to 802.11a (5 GHz) to ? (60 GHz)**
- **New modulation formats**
  - **GMSK, CDMA, OFDM, pulse position modulation**
- **New application areas**

# High Speed Data Links

- **A common architecture**




- **Transmitter Issues**
  - **Intersymbol interference (limited bandwidth of IC amplifiers, packaging), clock jitter, power, area**
- **Receiver Issue**
  - **Intersymbol interference (same as above), jitter from clock and data recovery, power, area**

# *Future Goals*

- **Low cost, low power, small area solutions**
  - **New architectures and circuits!**
- **Increased data rates**
  - **40 Gb/s for optical (moving to 120 Gb/s!)**
    - Electronics is a limitation (optical issues getting significant)
  - **> 5 Gb/s for backplane applications**
    - The channel (i.e., the PC board trace) is the limitation
- **High frequency compensation/equalization**
  - **Higher data rates, lower bit error rates (BER), improved robustness in the face of varying conditions**
  - **How do you do this at GHz speeds?**
- **Multi-level modulation**
  - **Better spectral efficiency (more bits in given bandwidth)**

# *This Class*

- **Circuit AND system focus**
  - **Knowing circuit design is not enough**
  - **Knowing system theory is not enough**
- **Circuit stuff**
  - **RF issues: transmission lines and impedance transformers**
  - **High speed design techniques**
  - **Basic building blocks: amplifiers, mixers, VCO's, digital components**
  - **Nonidealities: noise and nonlinearity**
- **System stuff**
  - **Macromodeling and simulation**
  - **Wireless and high speed data link principles**
  - **System level blocks: PLL's, CDR's, transceivers**

# *The Goal – Design at Circuit/System Level*

1. **Design architecture with analytical models**
   - May require new circuits – guess what they look like
2. **Verify architectural ideas by simulating with ideal macro-models of circuit blocks**
   - Guess macro-models for new circuits
3. **Add known non-idealities of circuit blocks (nonlinearity, noise, offsets, etc.)**
   - Go back to 1. if the architecture breaks!
4. **Design circuit blocks and get better macro-models**
   - Go back to 1. if you can't build the circuit!
   - Go back to 1. if the architecture breaks!
5. **Verify as much of system as possible with SPICE**
6. **Layout, extract, verify**
   - Do this soon for high speed systems - iteration likely!

# *Key System Level Simulation Needs*

- **You need a fast simulator**
  - **To design new things well, you must be able to iterate**
  - **The faster the simulation, the faster you can iterate**
- **You need to be able to add non-idealities in a controlled manner**
  - **Fundamental issues with architectures need to be separated from implementation issues**
    - An architecture that is fundamentally flawed should be quickly abandoned
- **You need flexibility**
  - **Capable of implementing circuit blocks such as filters, VCO's, etc.**
  - **Capable of implementing algorithms**
  - **Arbitrary level of detail**

# *A Custom C++ Simulator Will Be Used - CppSim*

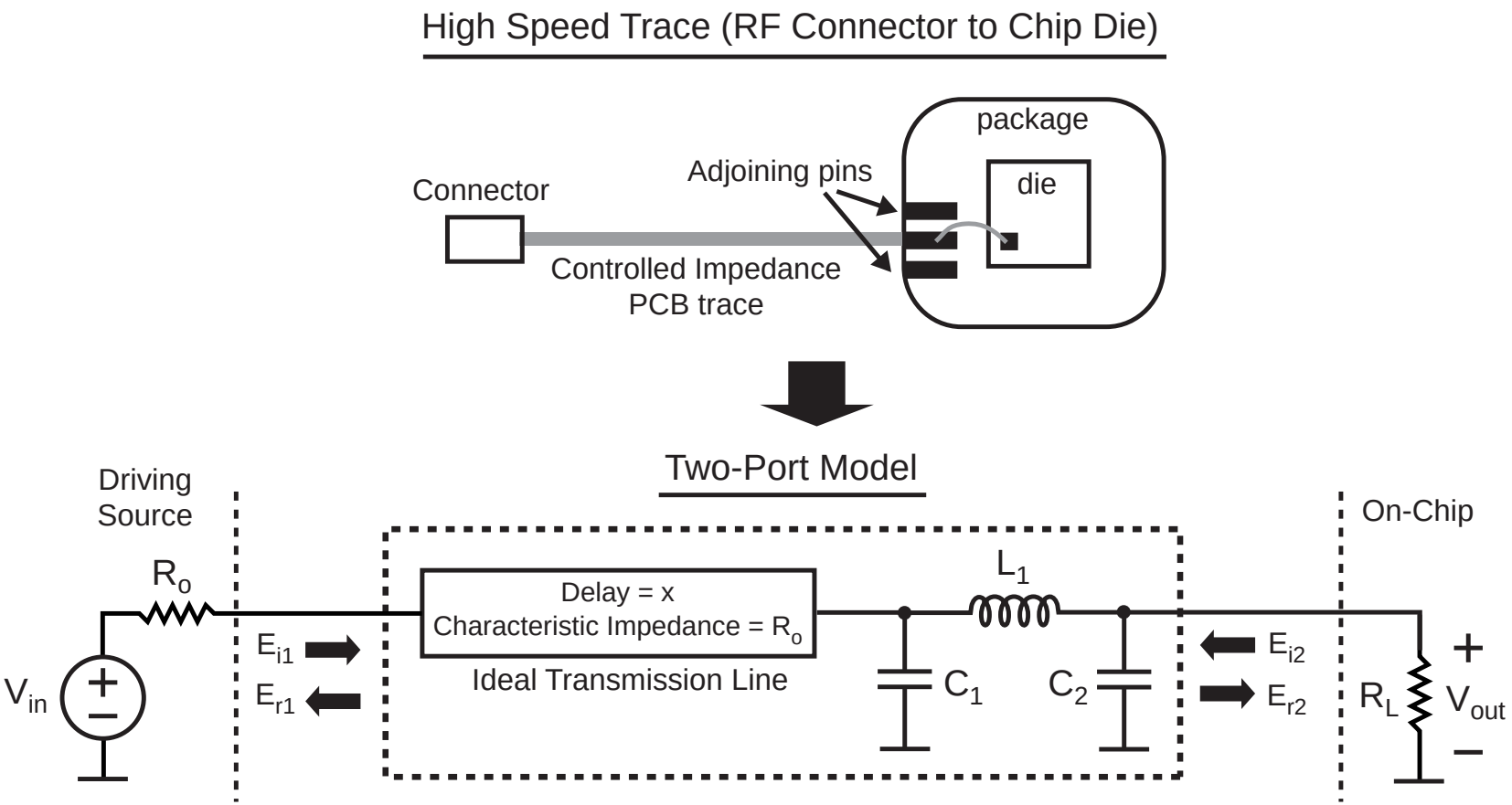
- **Blocks are implemented with C/C++ code**
  - **High computation speed**
  - **Complex block descriptions**
- **Users enter designs in graphical form using Sue2 schematic capture**
  - **System analysis and transistor level analysis in the same CAD framework**
- **Resulting signals are viewed in Matlab**
  - **Powerful post-processing and viewing capability**
- **Note: Hspice used for circuit level simulations**

**CppSim and Sue2 are on Athena and freely downloadable at http://www-mtl.mit.edu/~perrott**

# *A Quick Preview of Homeworks and Projects*

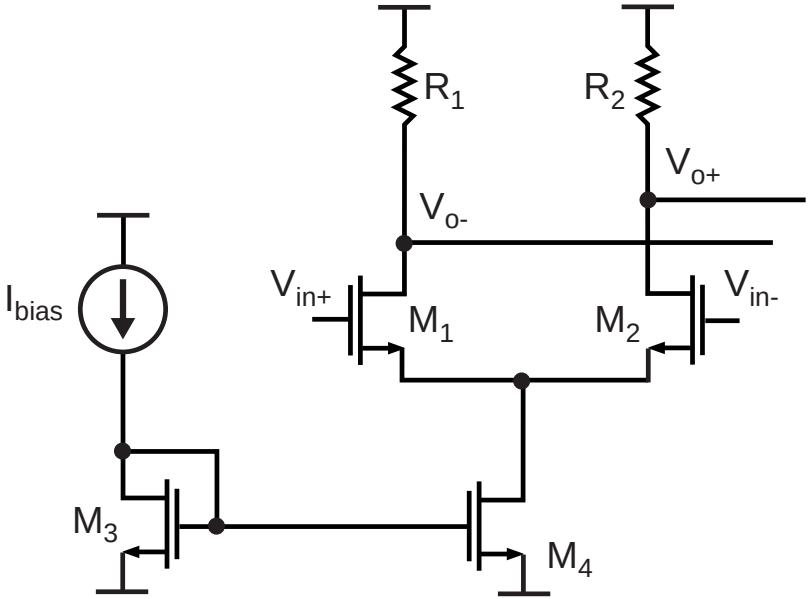
# HW1 – Transmission Lines and Transformers

- **High speed data link application:**

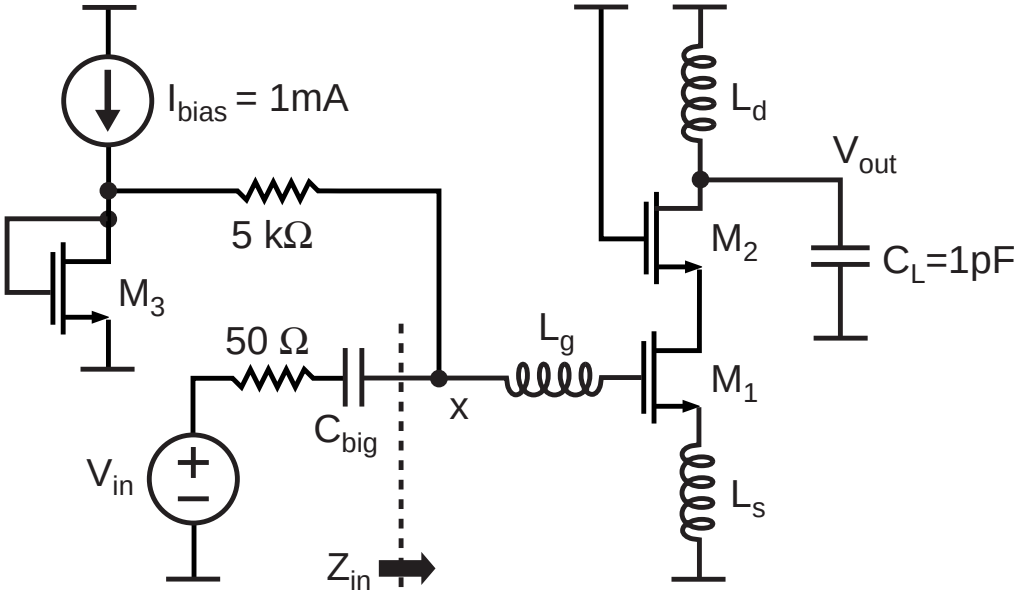


# HW2 – High Speed Amplifiers

- **Broadband**

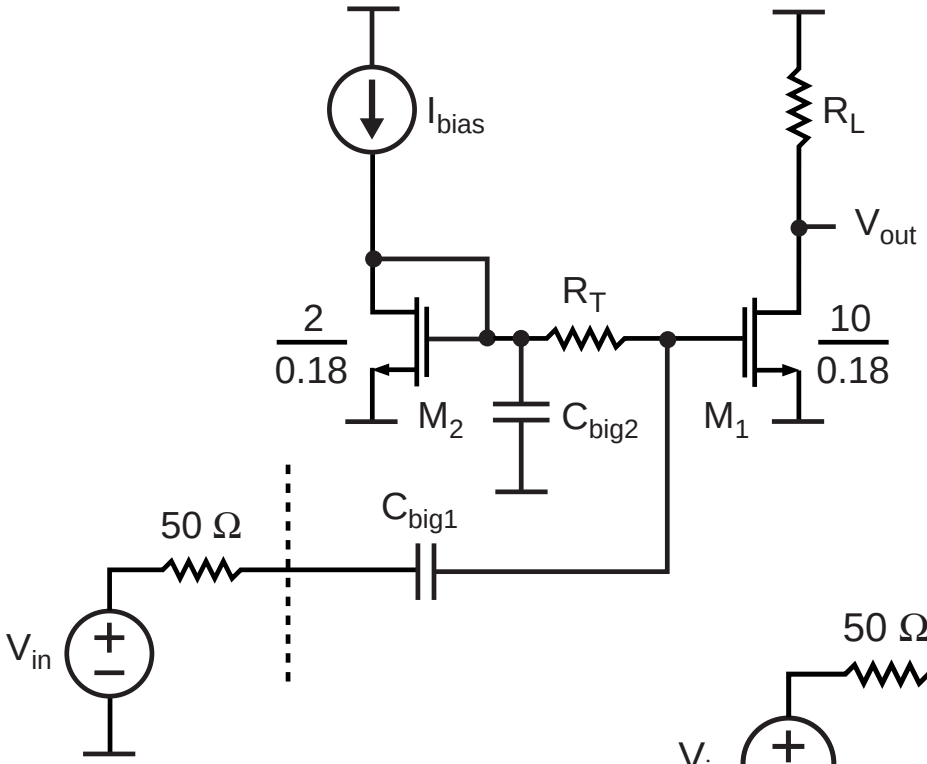



- **Narrowband**

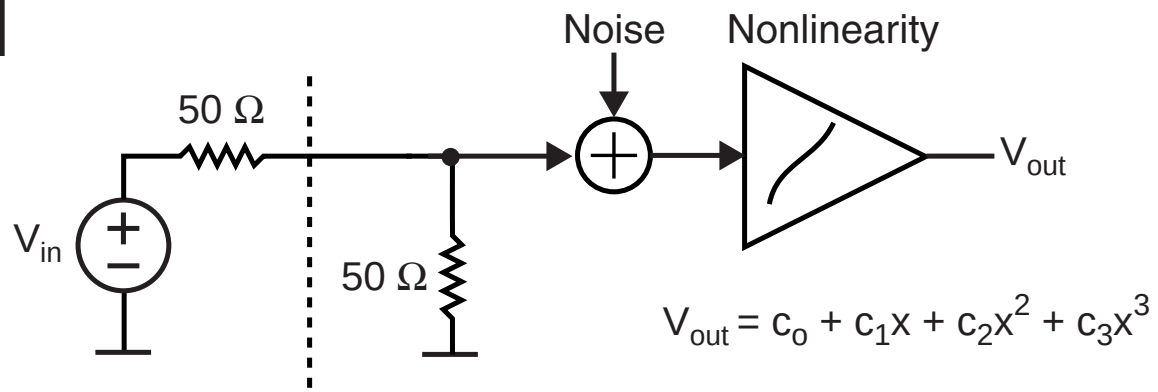


# HW3 – Amplifier Noise and Nonlinearity

- **Amplifier circuit**



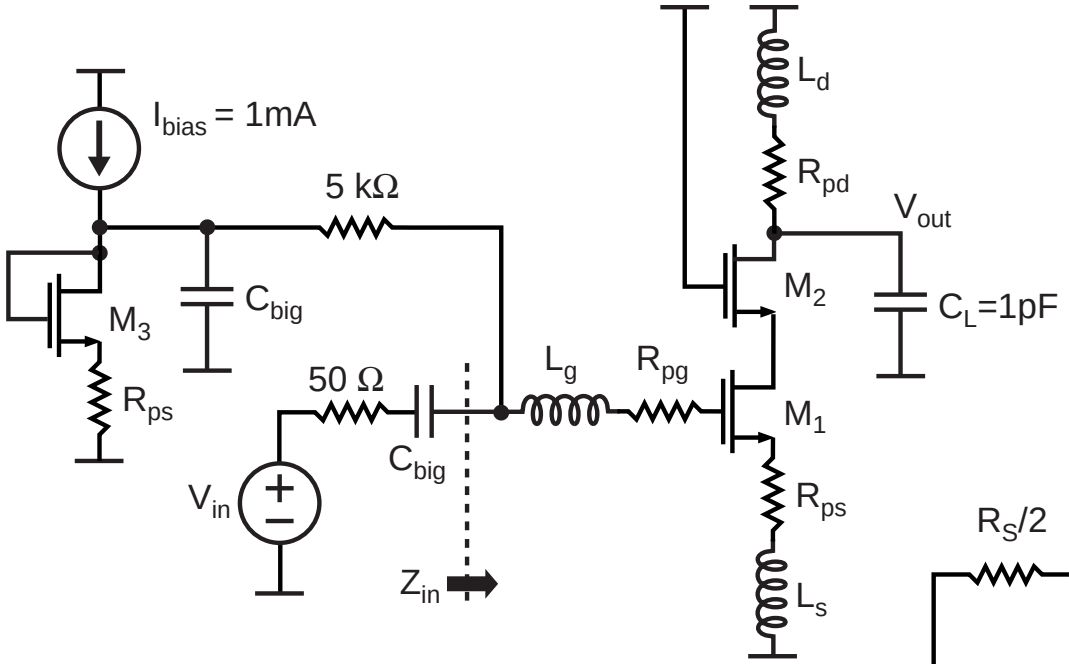

- **Model**



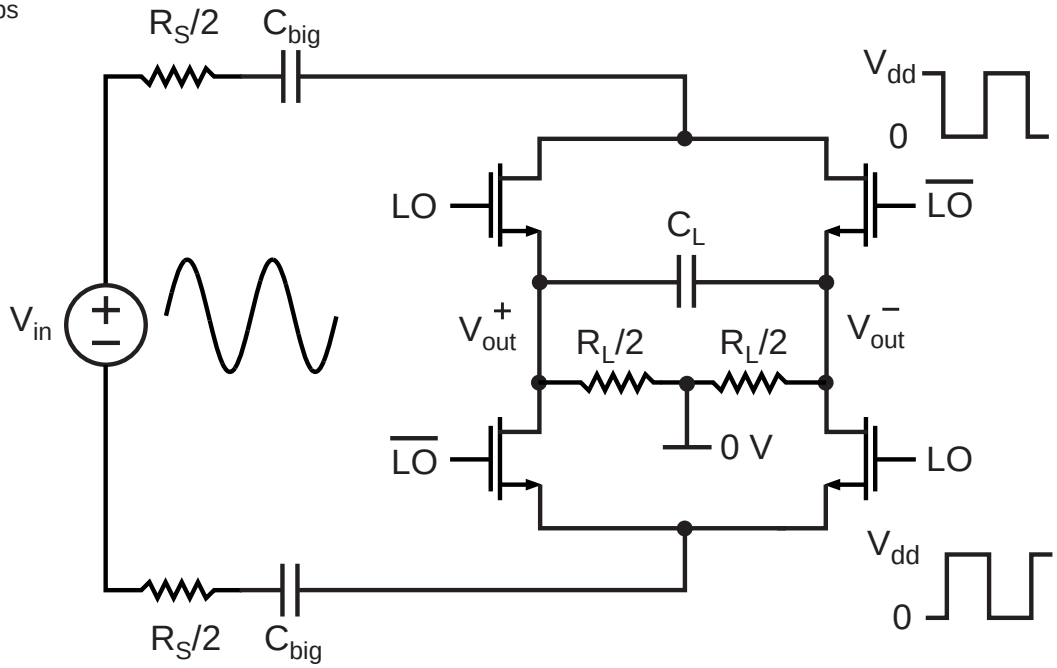

$$V_{out} = c_o + c_1 x + c_2 x^2 + c_3 x^3$$

# HW4 – Low Noise Amplifiers and Mixers

- **Narrowband LNA**

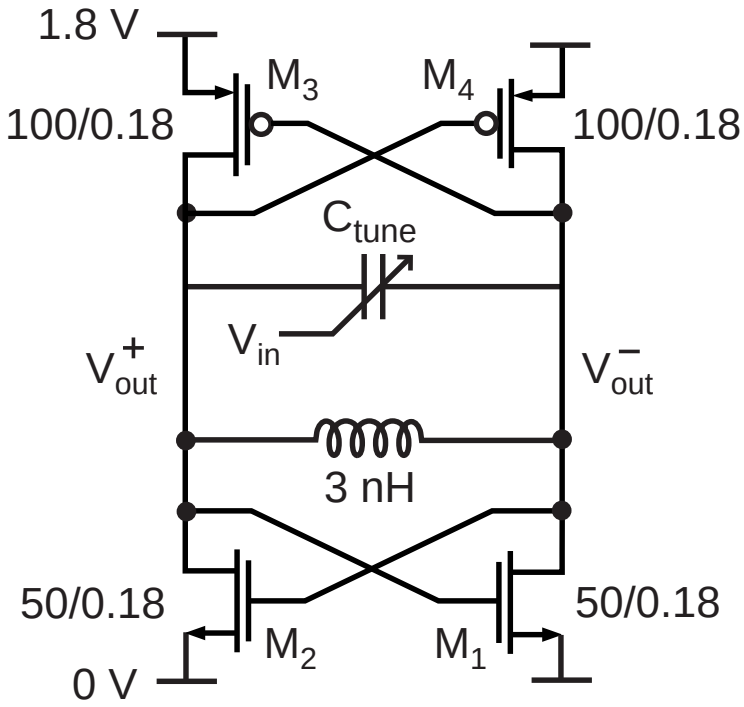



- **Passive Mixer**

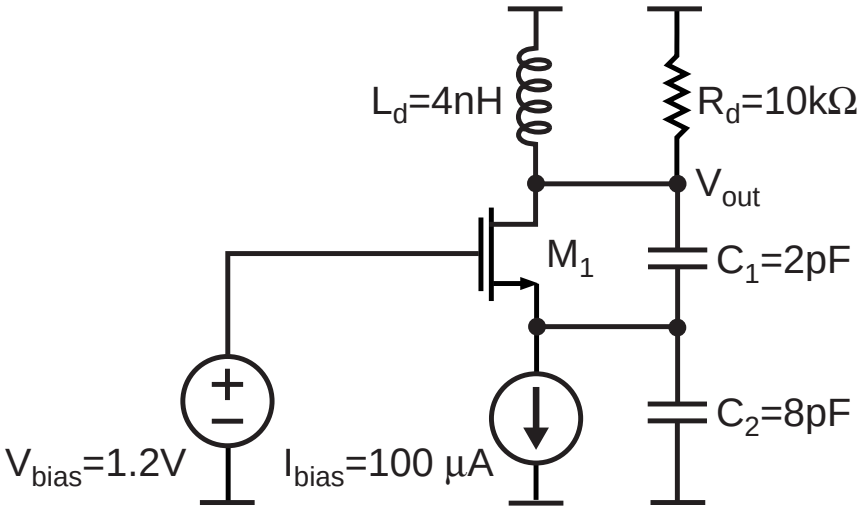


# HW5 – Voltage Controlled Oscillators

- **Differential CMOS**

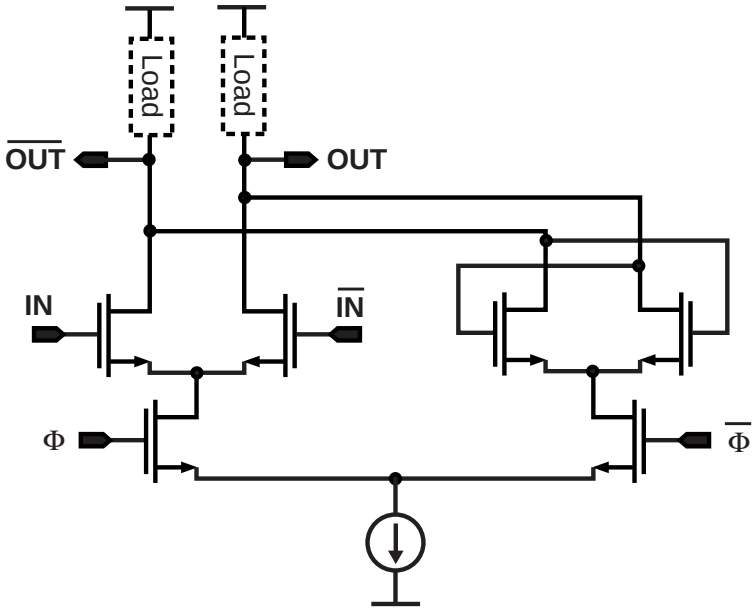



- **Colpitts**

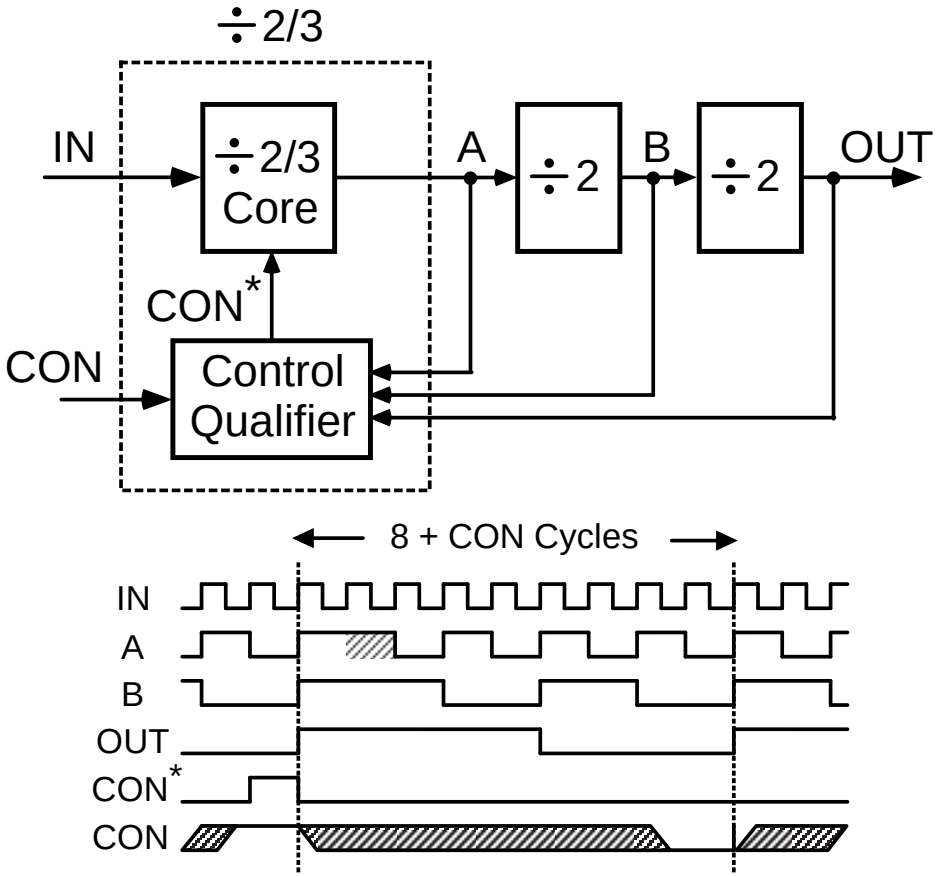


# Project 1 - High Speed Frequency Dividers

- **High speed latches/registers**

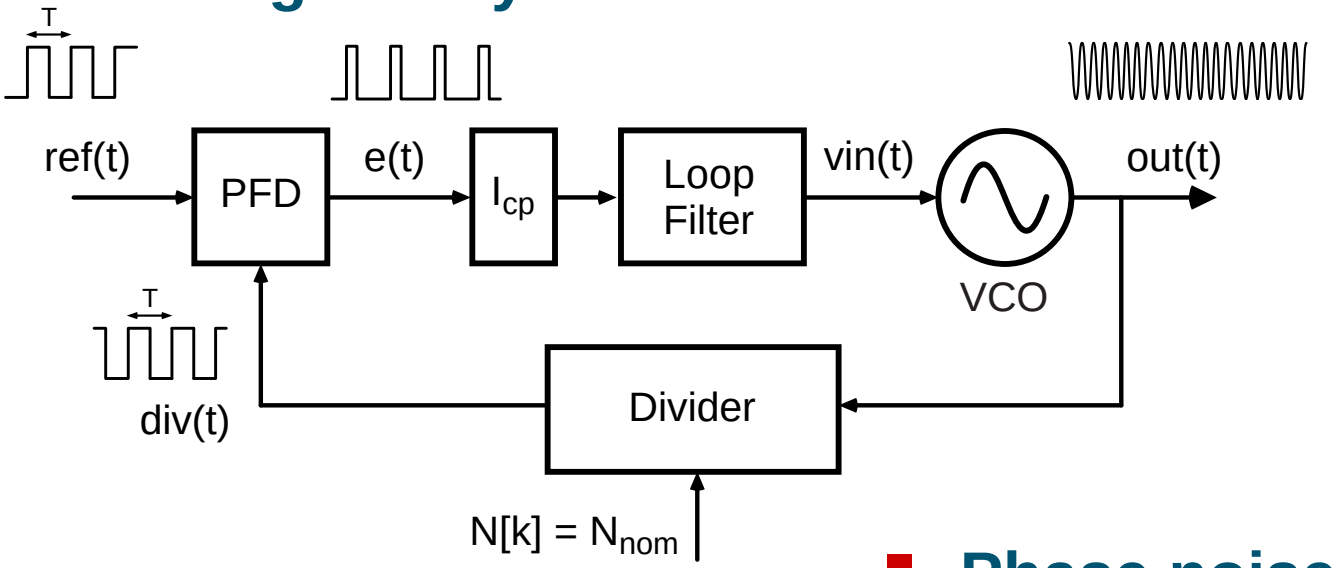



- **High speed dual-modulus divider**

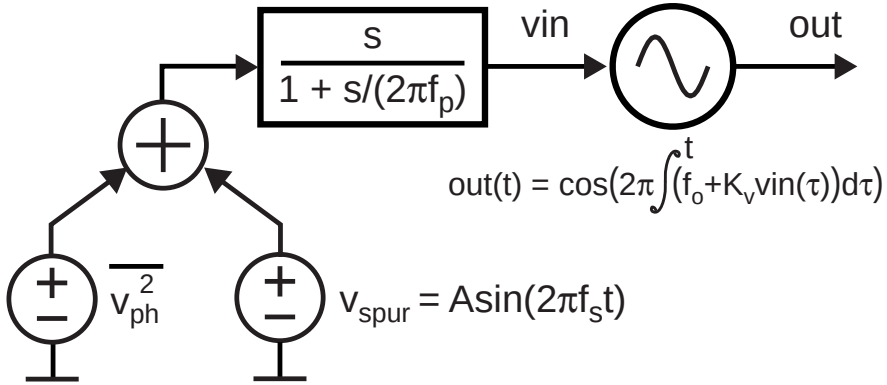


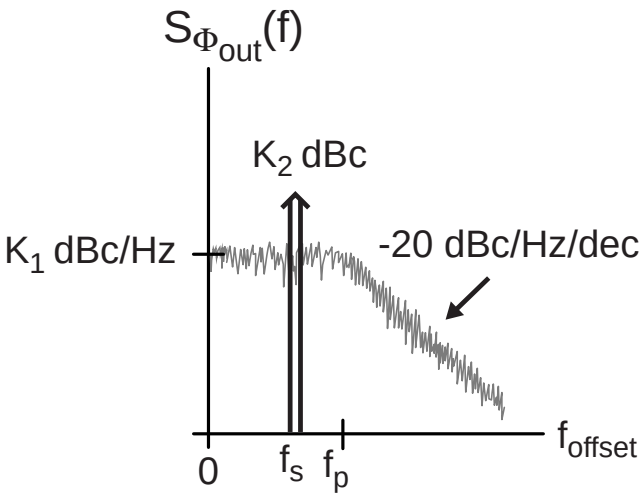
# HW6 – Phase Locked Loop Design

- **Integer-N synthesizer**

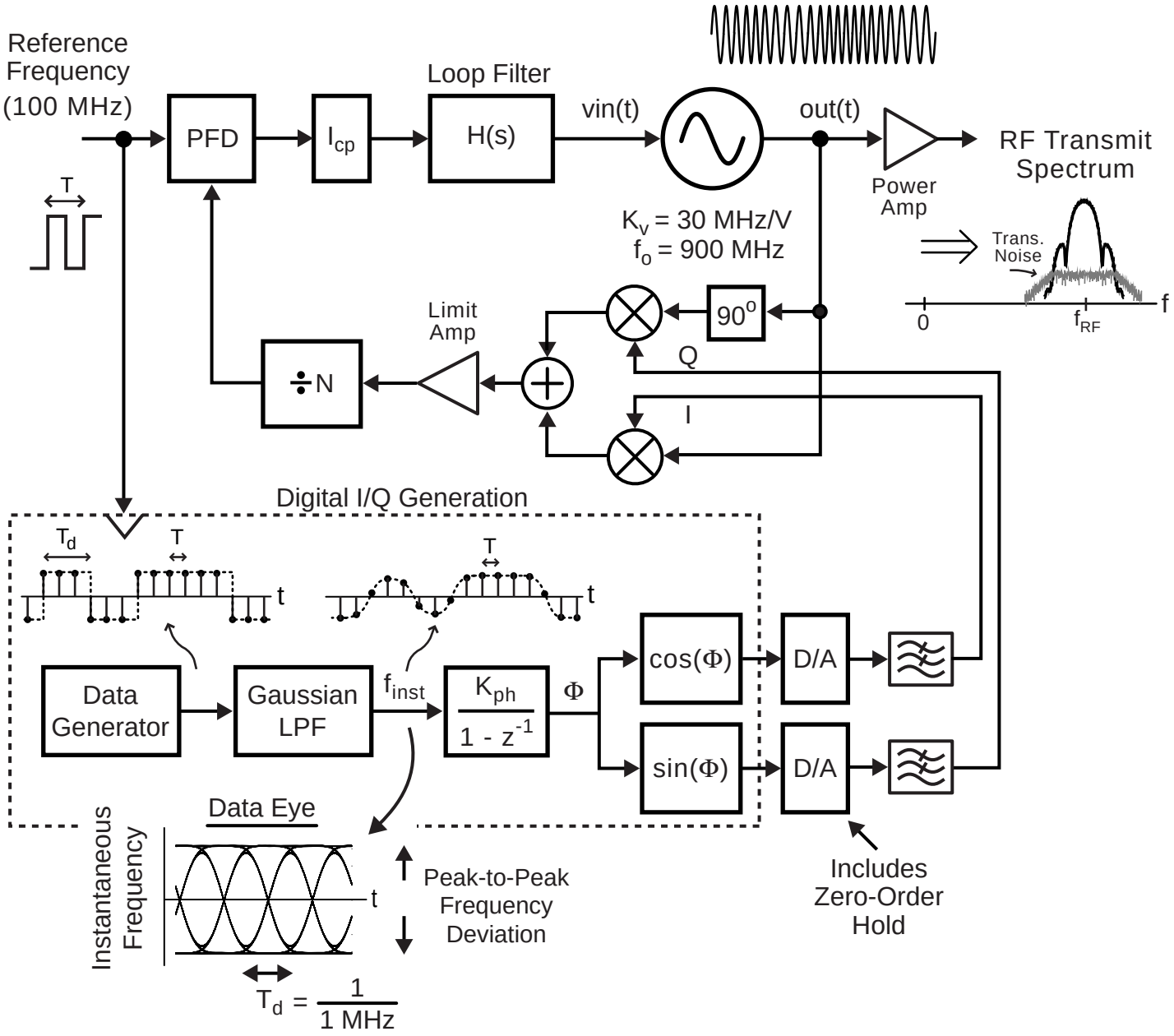



- **Phase noise simulation**

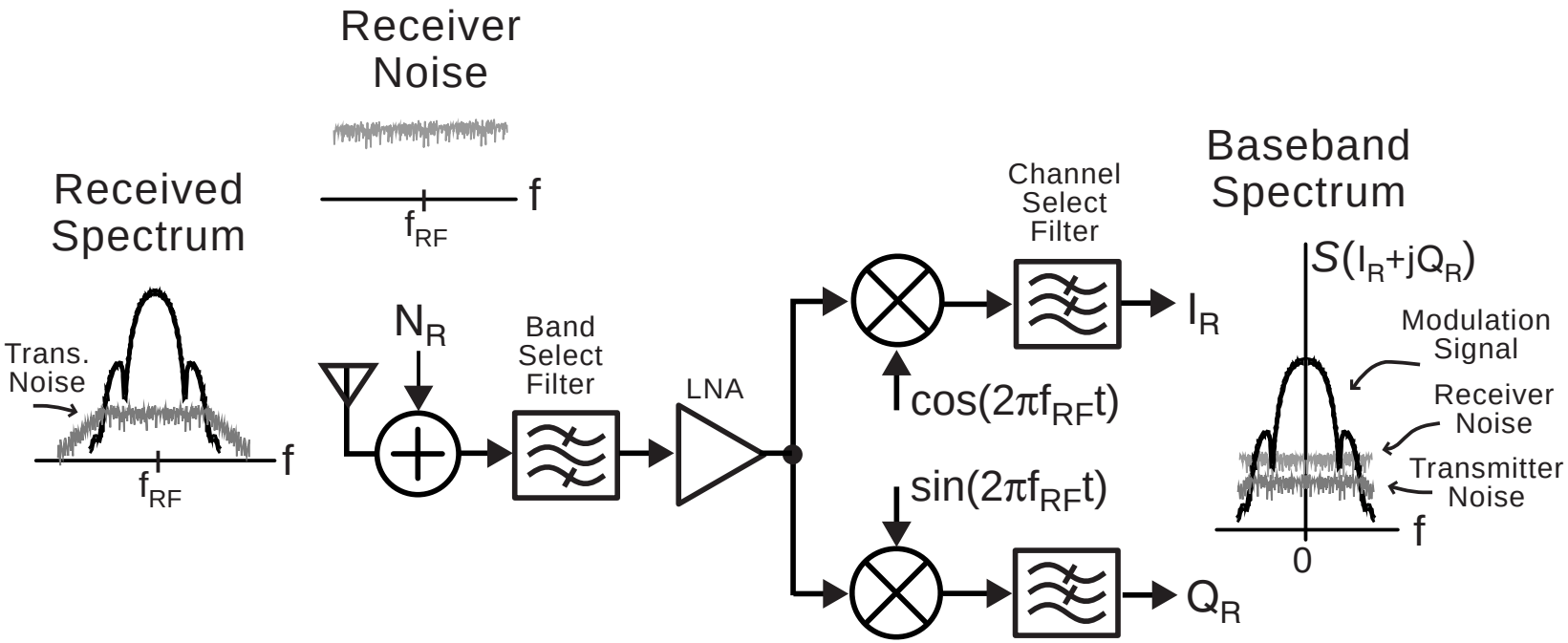


# Project 2 – GMSK Transmitter for Wireless Apps



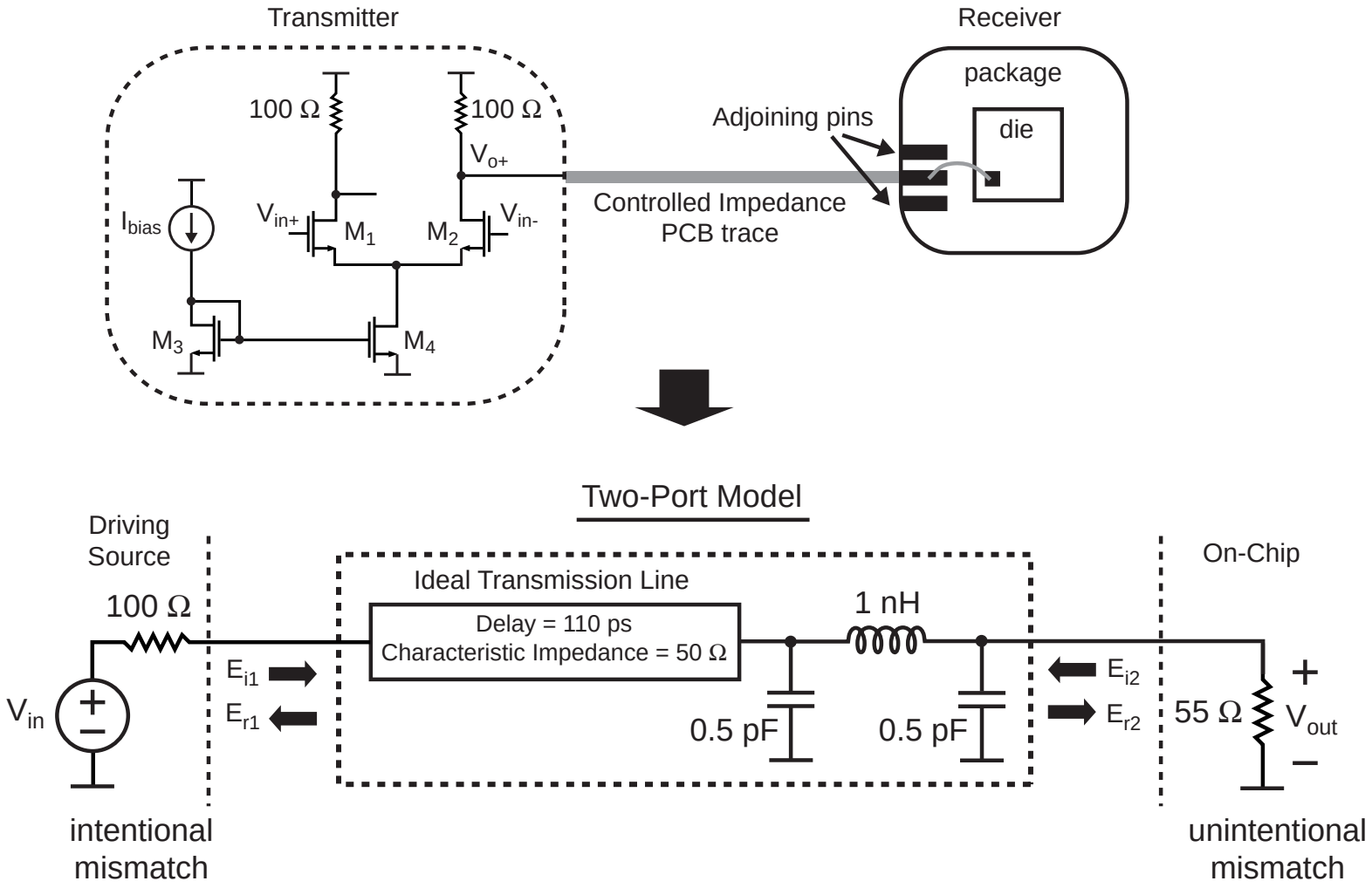
# Project 2 – Accompanying Receiver



# *Basics of Digital Communication*

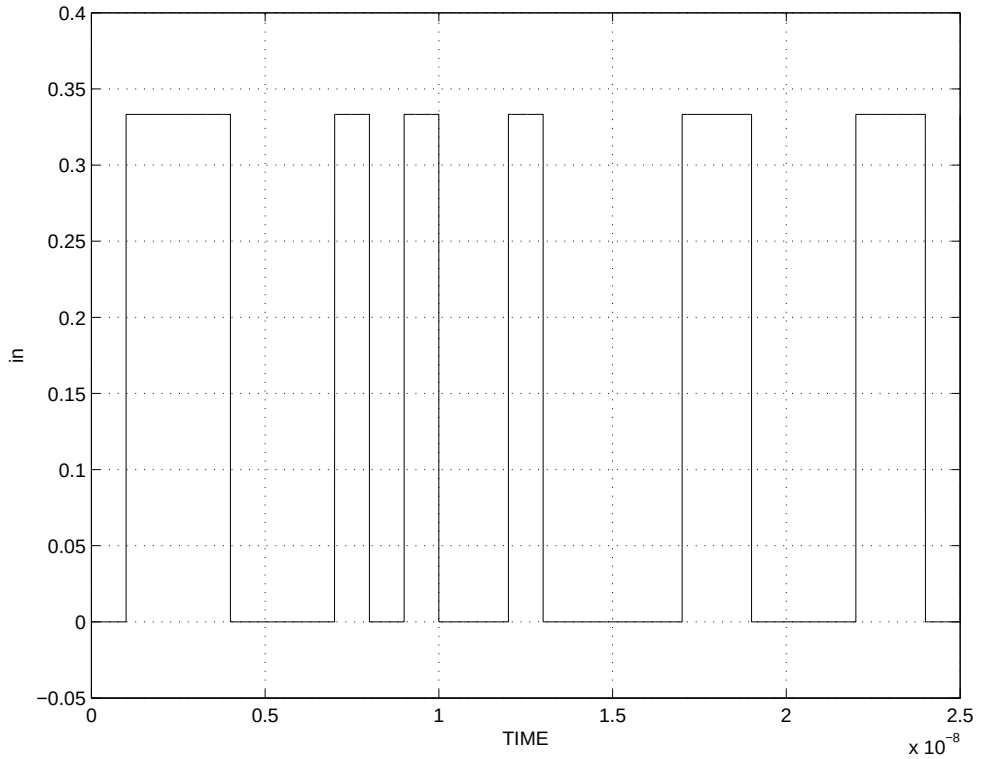
# Example: A High Speed Backplane Data Link

- **Suppose we consider packaging issues at the receiver side (ignore transmitter packaging now for simplicity)**



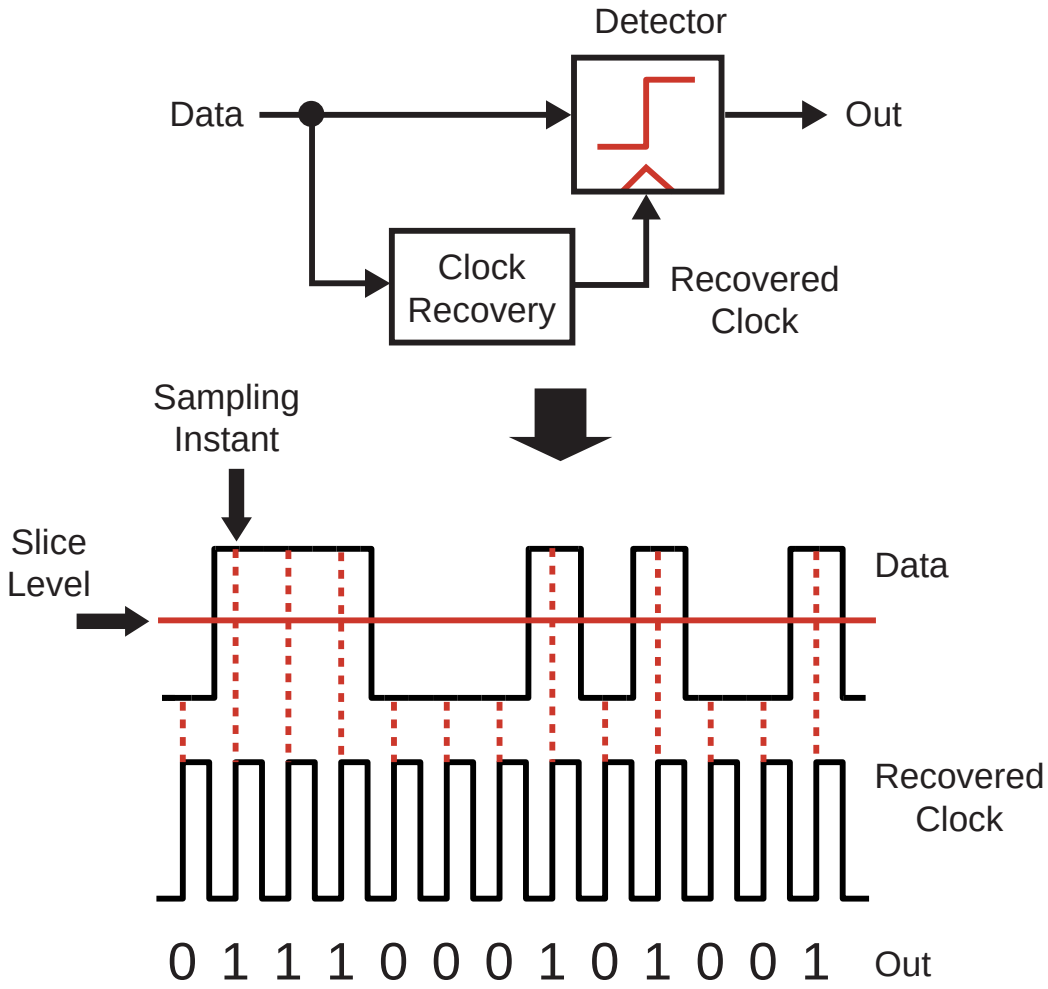
# *Modulation Format*

- **Binary, Non-Return to Zero (NRZ), Pulse Amplitude Modulation (PAM)**
  - **Send either a zero or one in a given time interval $T_d$**
  - **Time interval set by a low jitter clock**
  - **Ideal signal from transmitter:**



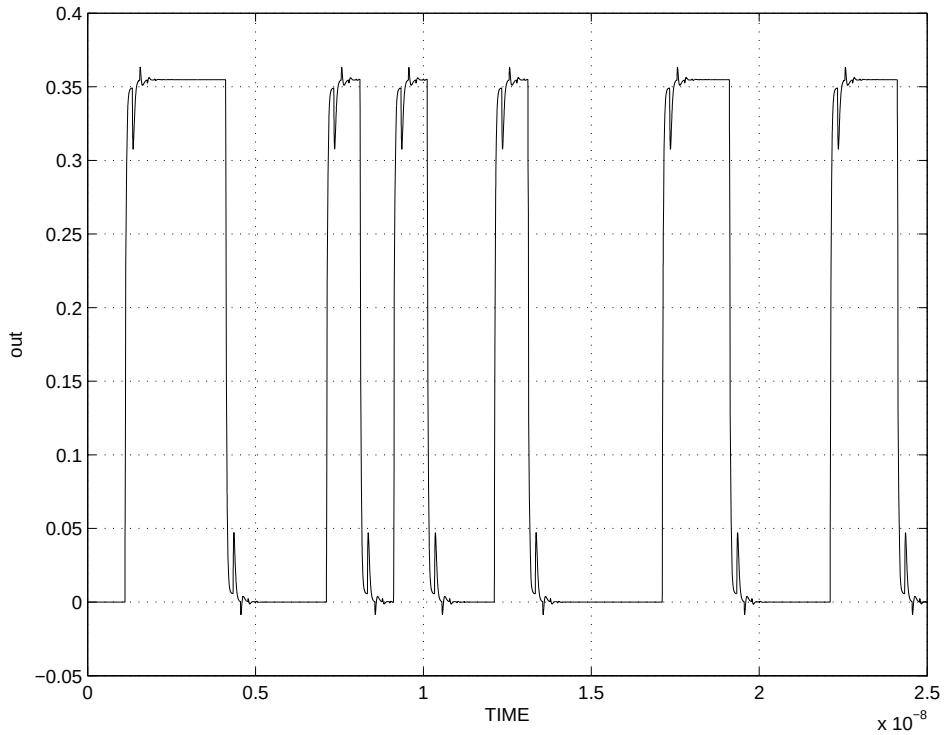
# Receiver Function

- Two operations
  - Recover clock and use it to sample data
  - Evaluate data to be 0 or 1 based on a slicer



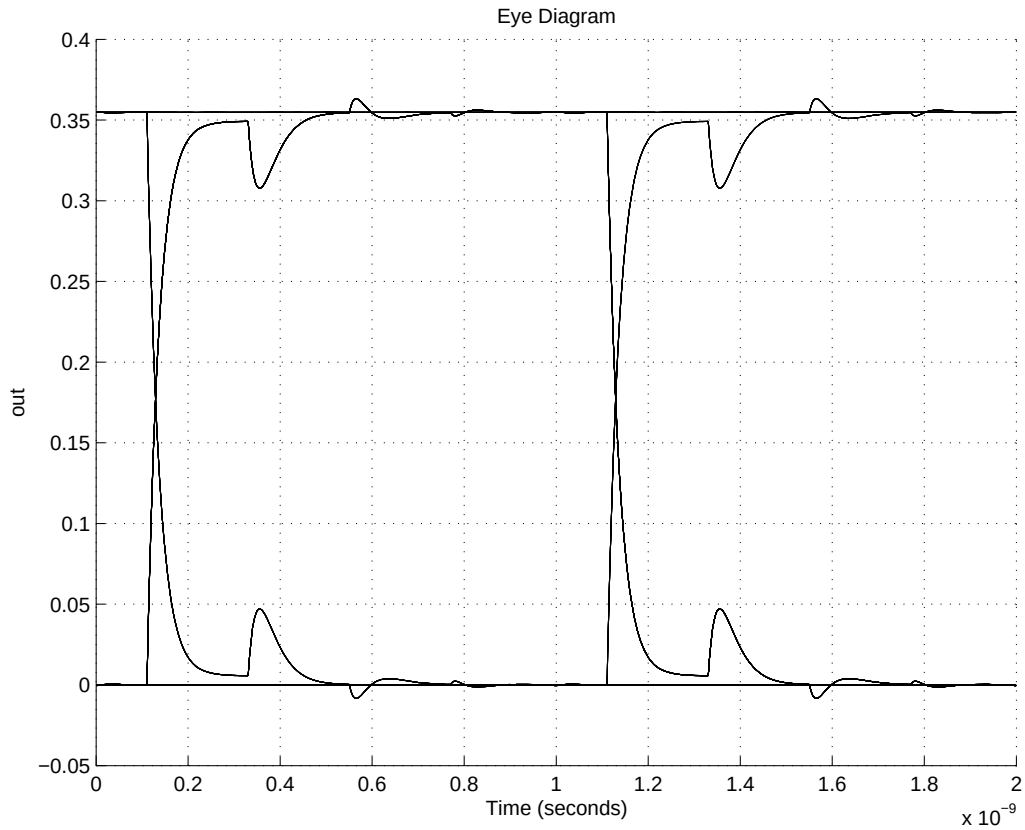
# *Issue: PC Board Trace is Not an Ideal Channel*

- **Chip capacitance and inductance limits bandwidth**
- **Transmission line effects cause reflections in the presence of impedance mismatch**
- **Example: transmit at 1 Gb/s across link in previous slide (assume bondwire inductance is zero)**
  - **Signal at receiver termination resistor**



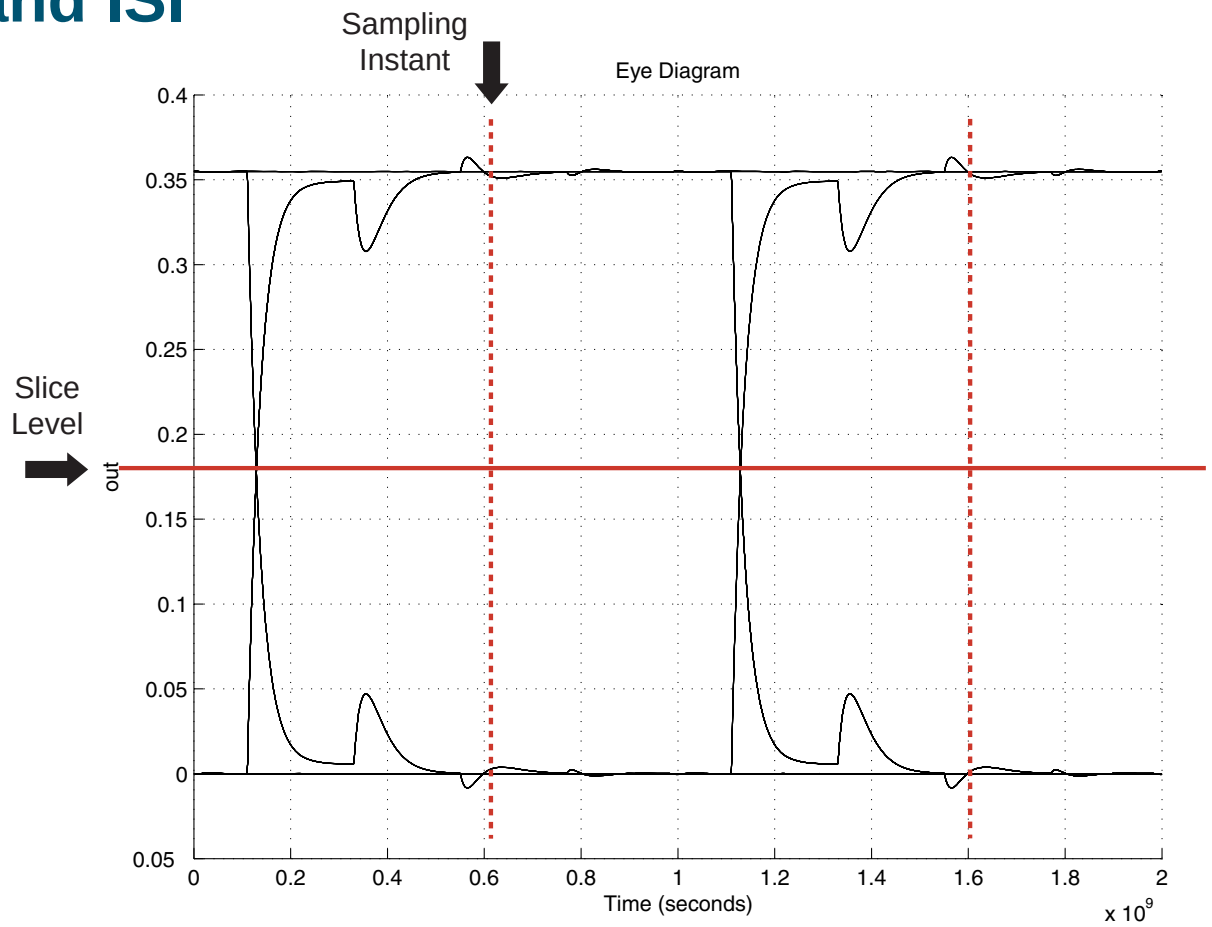
# *Eye Diagram for 1 Gb/s Data Rate*

- **Wrap signal back onto itself every $2*T_d$ seconds**
  - **Same as an oscilloscope would do**
- **Allows immediate assessment of the quality of the signal at the receiver (look at eye opening)**



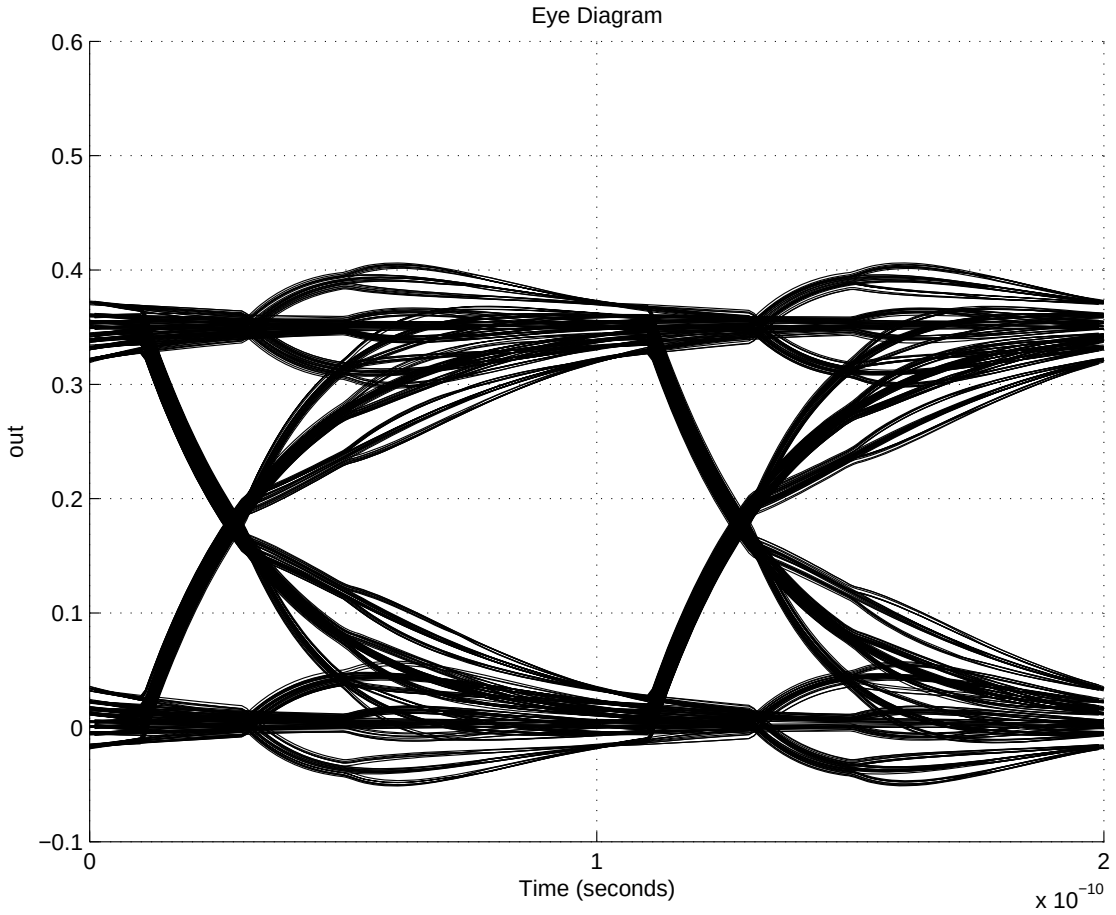
# Relationship of Eye to Sampling Time and Slice Level

- Horizontal portion of eye indicates sensitivity to timing jitter
- Vertical portion of eye indicates sensitivity to additional noise and ISI



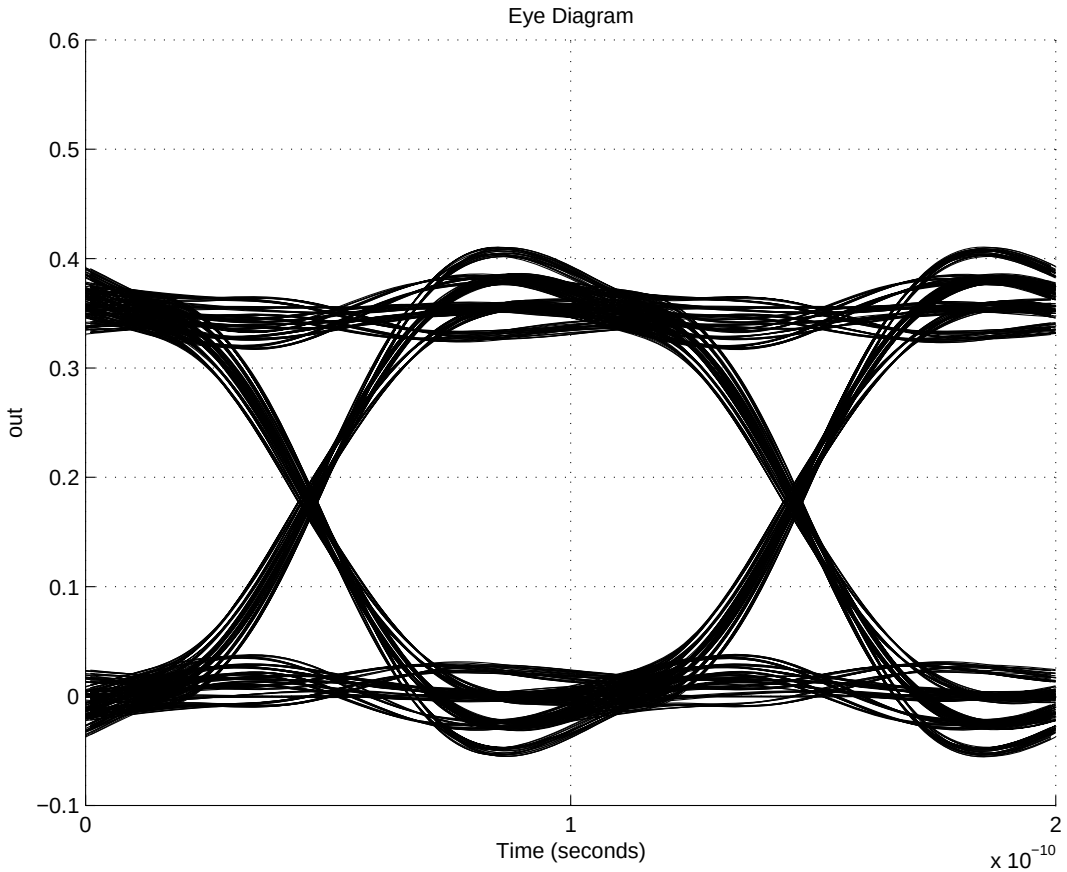
# What Happens if We Increase the Data Rate?

- Limited bandwidth and reflections cause intersymbol interference (ISI)
- Eye diagram at 10 Gb/s for same data link



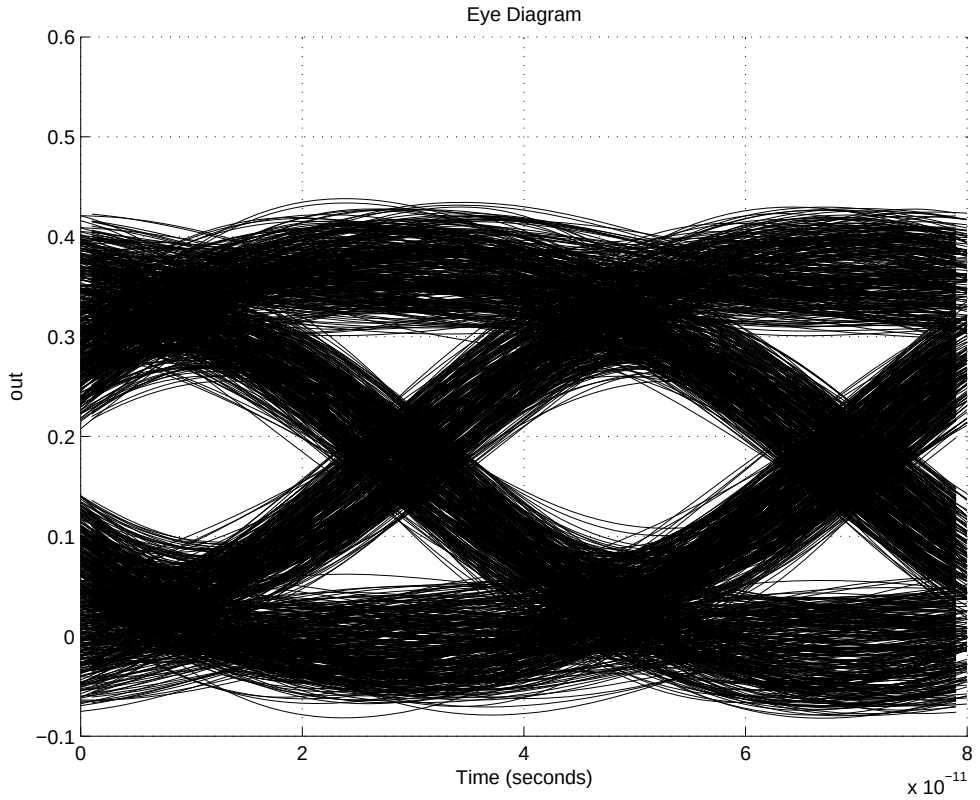
# What is the Impact of the Bondwire Inductance?

- Rule of thumb:  1 nH/mm for bondwire
  - Assume 1 nH
- Impact of inductance here increases bandwidth
  - less ISI occurs



# How High of a Data Rate Can The Channel Support?

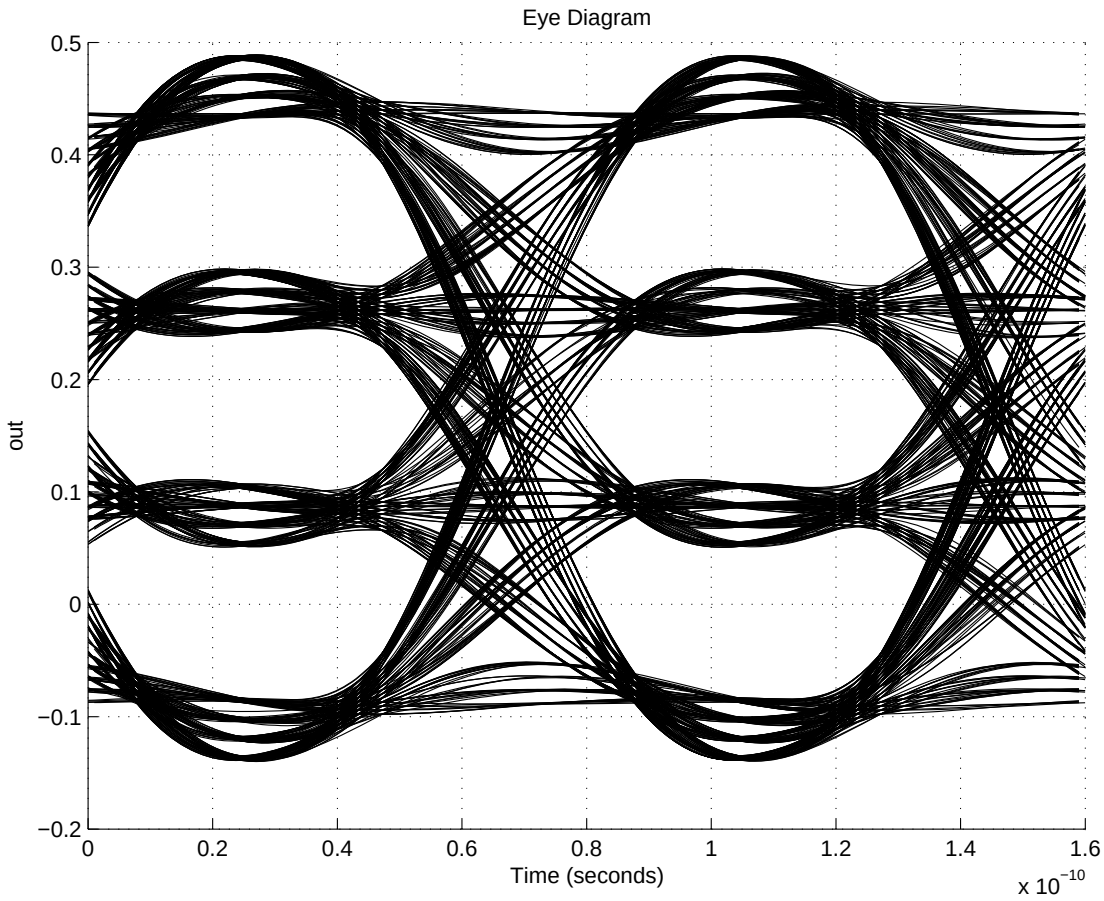
- **Raise it to 25 Gb/s**




- **However, we haven't considered other issues**
  - **PC board trace attenuates severely at high frequencies**
    - Bandwidth is < 5 GHz for 48 inch PC board trace (FR4)

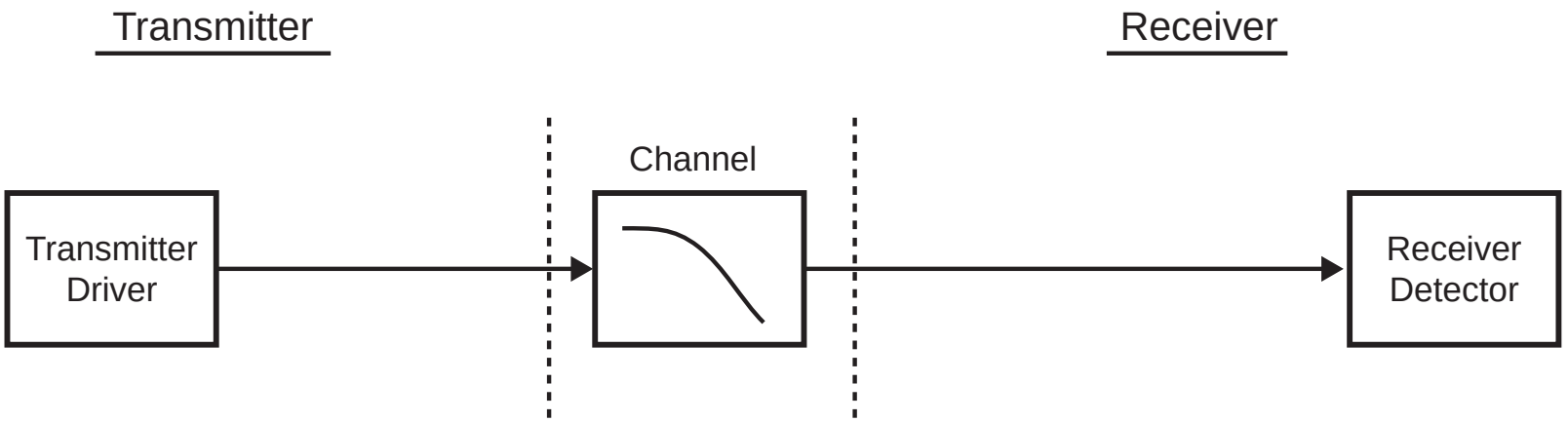
# *Multi-Level Signaling*

- **Increase spectral efficiency by sending more than one bit during a symbol interval**
  - **Example: 4-Level PAM at 12.5 Gb/s on same channel**
    - Effective data rate: 25 Gb/s



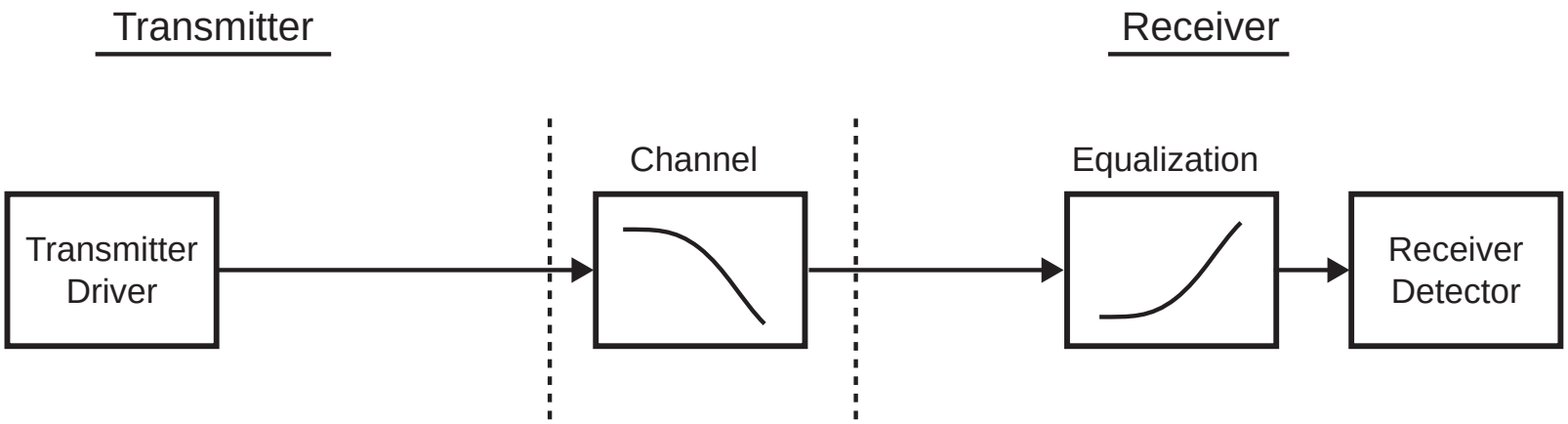
# How Else Can We Reduce ISI?

- **Consider a system level view of the link**
  - **Channel can be viewed as having an equivalent frequency response**
    - Assumes linearity and time-invariance (accurate for most transmission line systems)



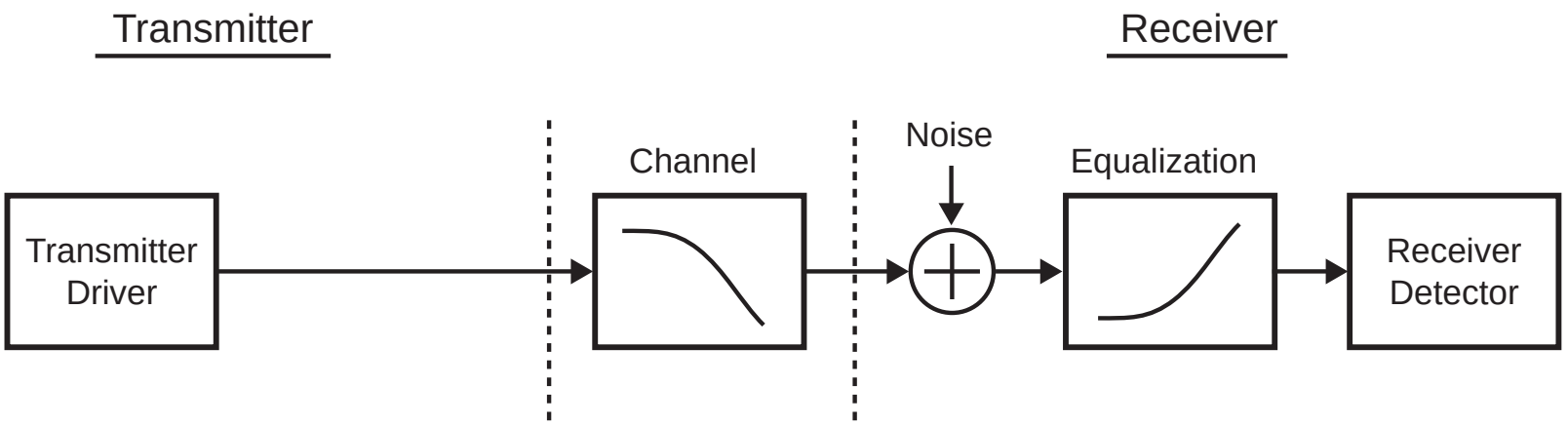
# Equalization

- Undo channel frequency response with an inverse filter at the receiver
  - Removes ISI!
  - Can make it adaptive to learn channel



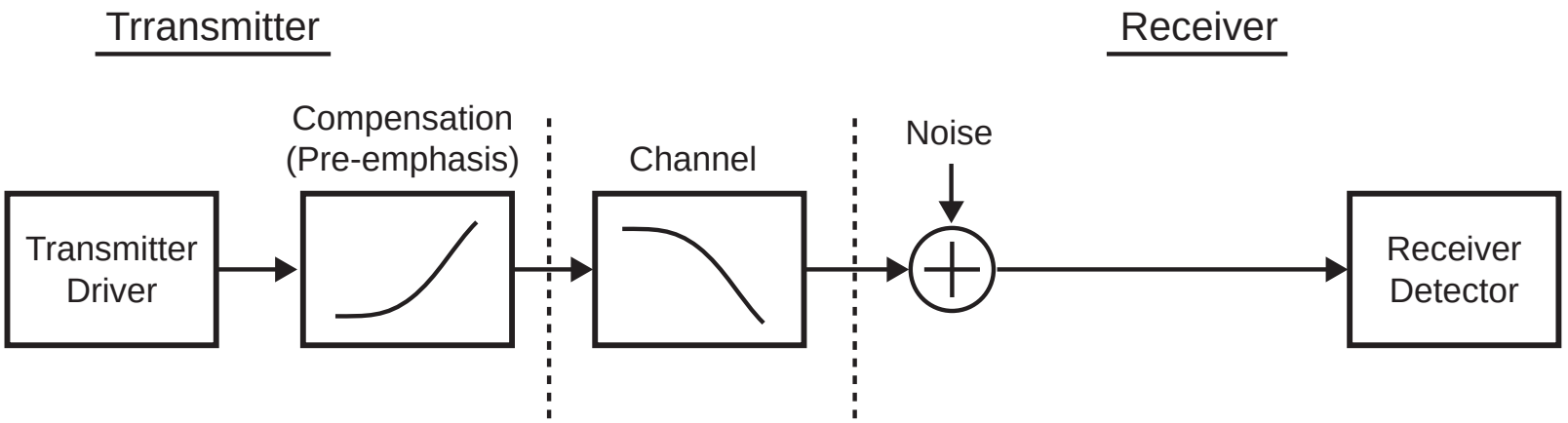
# The Catch

- **Equalization enhances noise**
  - **Overall SNR may be reduced**
- **Optimal approach is to make ISI and noise degradation about equal**



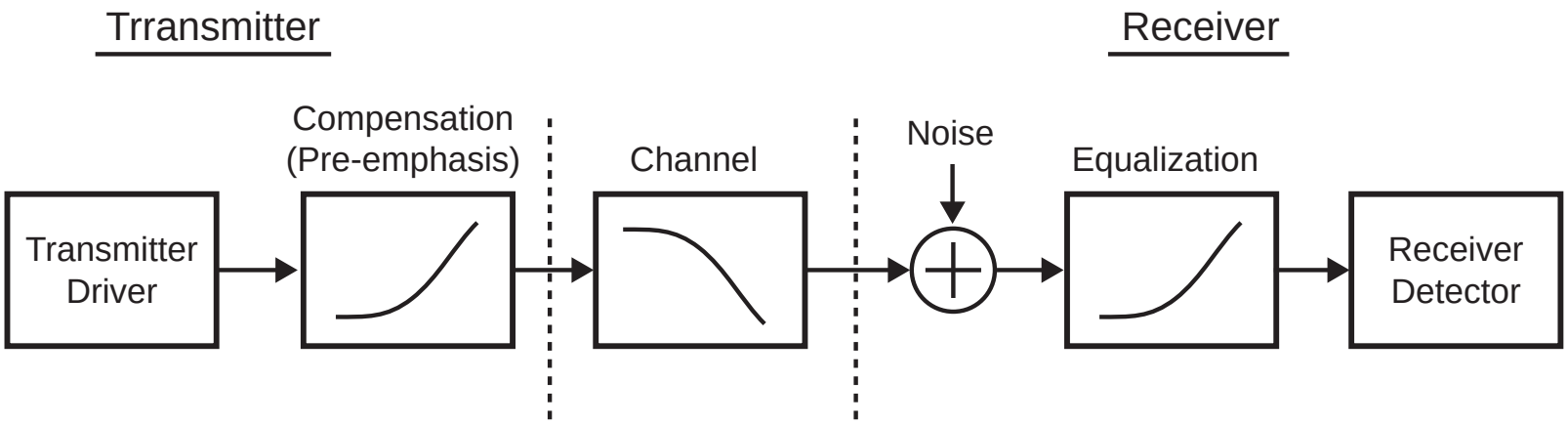
# *Alternative – Pre-emphasize at Transmitter*

- **Put inverse filter at transmitter instead of receiver**
  - **No enhancement of noise, but …**
  - **Need feedback from receiver to learn channel**
  - **Requires higher dynamic range/power from transmitter**



# Best Overall Performance

- **Combine compensation and equalization**
  - **Starting to see this for high speed links**



# What are the Issues with Wireless Systems?

- Noise
  - Need to extract the radio signal with sufficient SNR
- Selectivity (filtering, processing gain)
  - Need to remove interferers (which are often much larger!)
- Nonlinearity
  - Degrades transmit spectral mask
  - Degrades selectivity for receiver
- Multi-path (channel response)
  - Degrades signal – nulls rather than ISI usually the issue
  - Can actually be used to advantage!

**We will look at BOTH broadband data links and wireless systems in this class**