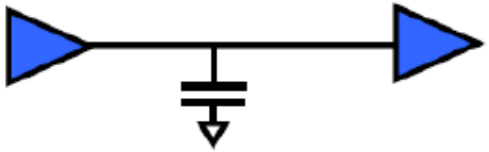
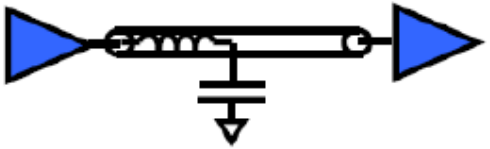
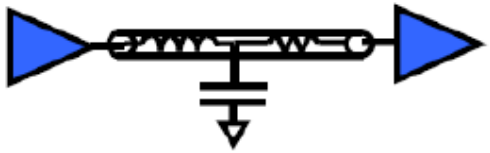
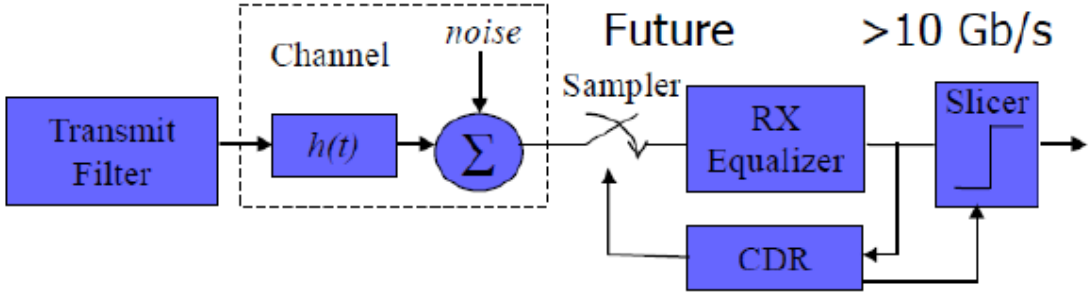
ECE 546
Lecture - 28
High-Speed Links

Spring 2024

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

Inter-IC Communication Trends

	Decade	Speeds	Transceiver Features
Lumped capacitance 	1980's	>10Mb/s	Inverter out, inverter in
Transmission line 	1990's	>100Mb/s	Termination Source-synchronous clk.
Lossy transmission line 	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

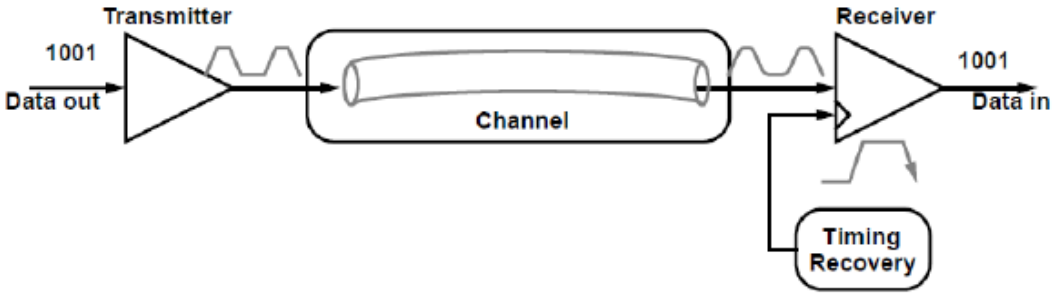
Slide Courtesy of Frank O'Mahony & Brian Casper, Intel

High-Speed Bus and Networks

- Memory Bus (Single-ended, Parallel)
  - DDR (4.266 Gbps)
  - LPDDR4 (4.266 Gbps)
  - GDDR (7 Gps)
  - XDR (differential, 4.8 Gbps)
  - Wide IO2, HBM
- Front Side Bus (Differential, Parallel)
  - QuickPath Interconnect (6.4 Gbps)
  - HyperTransport (6.4 Gbps)
- Computer IO (Differential, Parallel)
  - PCIe (8 Gbps)
  - InfiniBand (10 Gbps)
- Cable (Differential, Serial)
  - USB (4.266 Gbps)
  - HDMI (4.266 Gbps)
  - Firewire: Cat 5, Cat 5e, Cat 6
- Storage (Differential, Serial)
  - eMMC, UFS (6 Gbps)
  - SAS, STATA (6 Gbps)
  - FiberChannel (10 – 20 Gbps)
- Ethernet (Differential, Serial)
  - XAUI (10 Gbps)
  - XFI (10 Gbps)
  - CEI-6GLR
  - SONNET (10 Gbps)
  - 10GBase-x, 100GBase (25 Gbps)

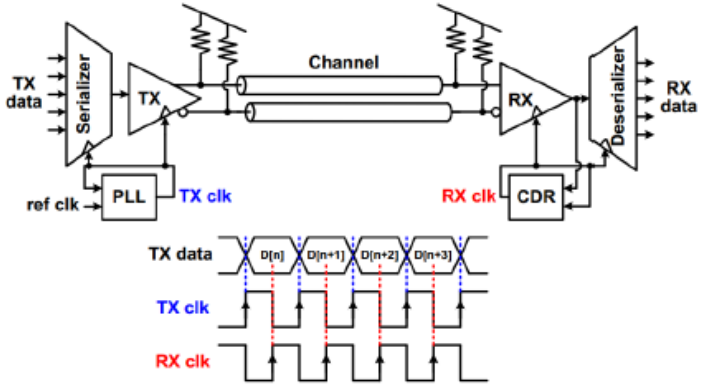
Basic Serial Link Architecture

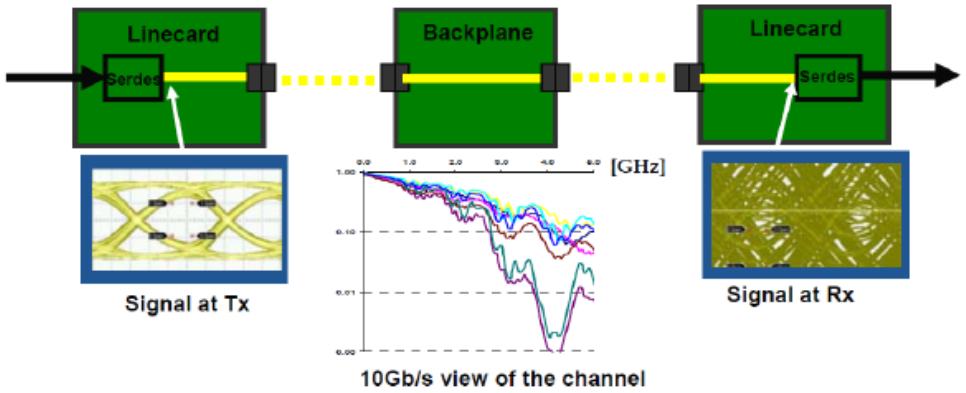
- **TX:** Generates train of analog pulses depending upon input digital data symbols and transmits them across the channel to the receiver.
- **Channel:** Electrical path between TX and RX blocks. Typically comprises of PCB traces, vias, connectors and other such I/O interface components.
- **RX:** Amplifies 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.



Basic 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.





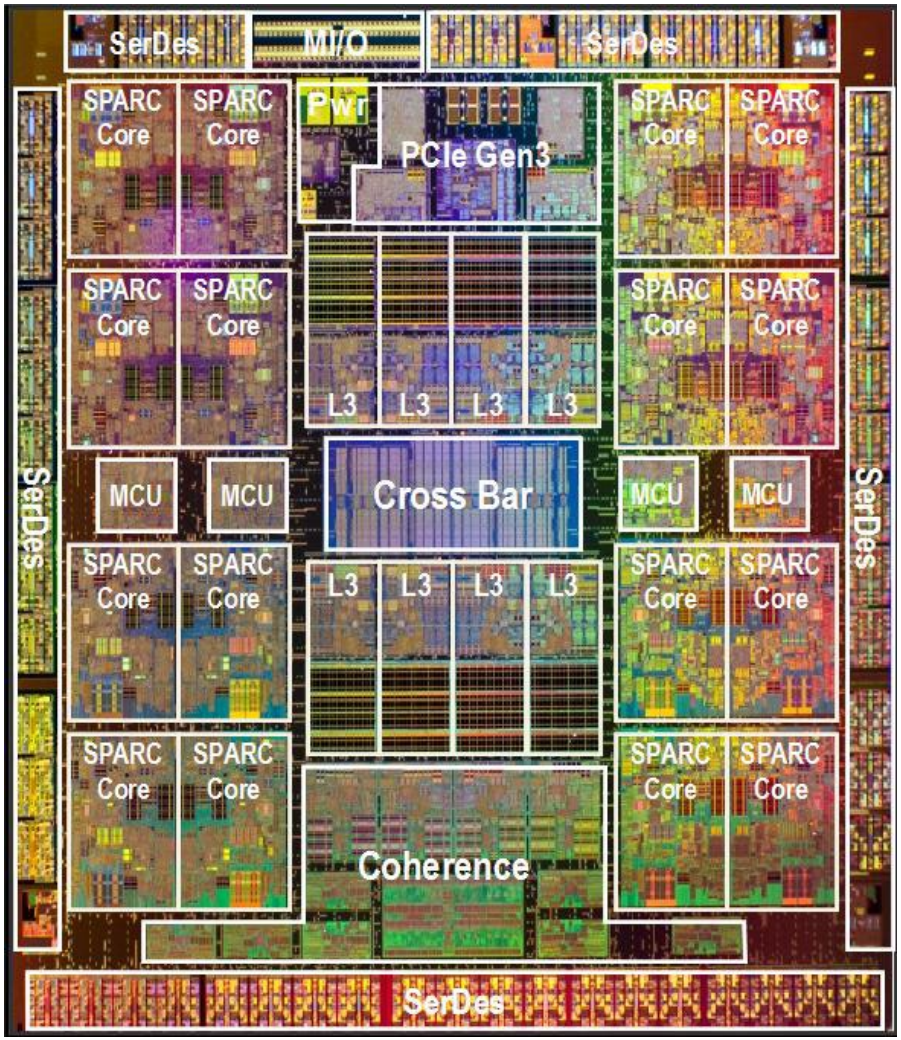
Why SERDES?

- 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.

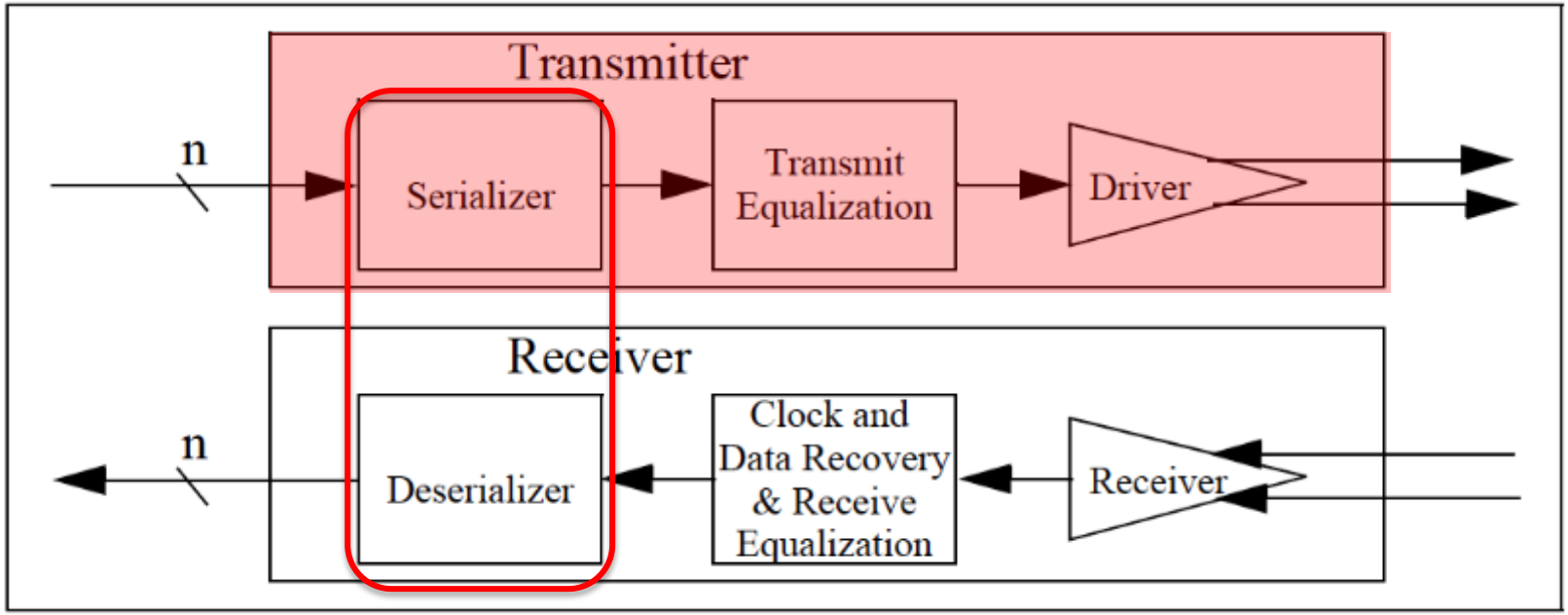
Serial Links in SoC: Oracle SPARC T5*



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

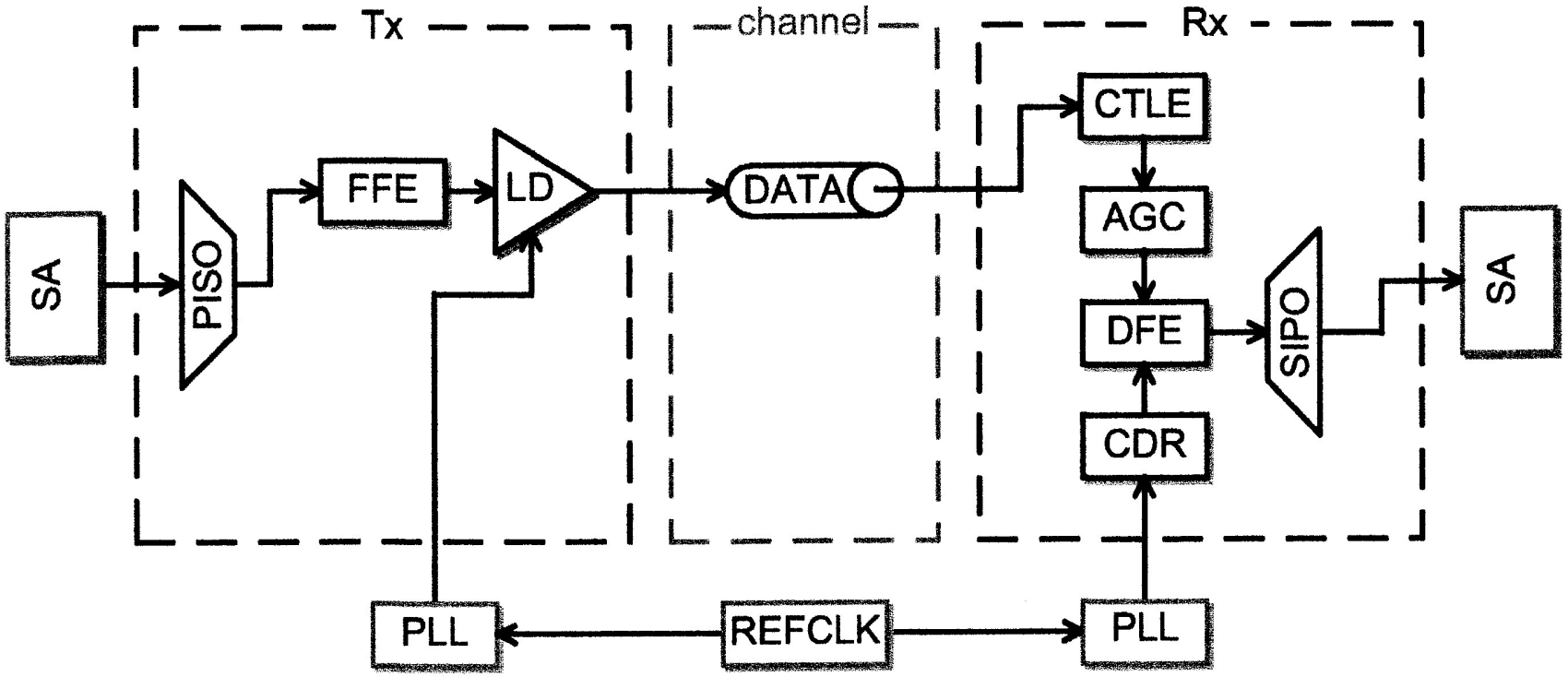
*** J. Hart et al., "A 3.6GHz 16-Core SPARC SoC Processor in 28nm", Proceedings of the 2013 IEEE International Solid-State Circuits Conference.**

Serializer/Deserializer Blocks



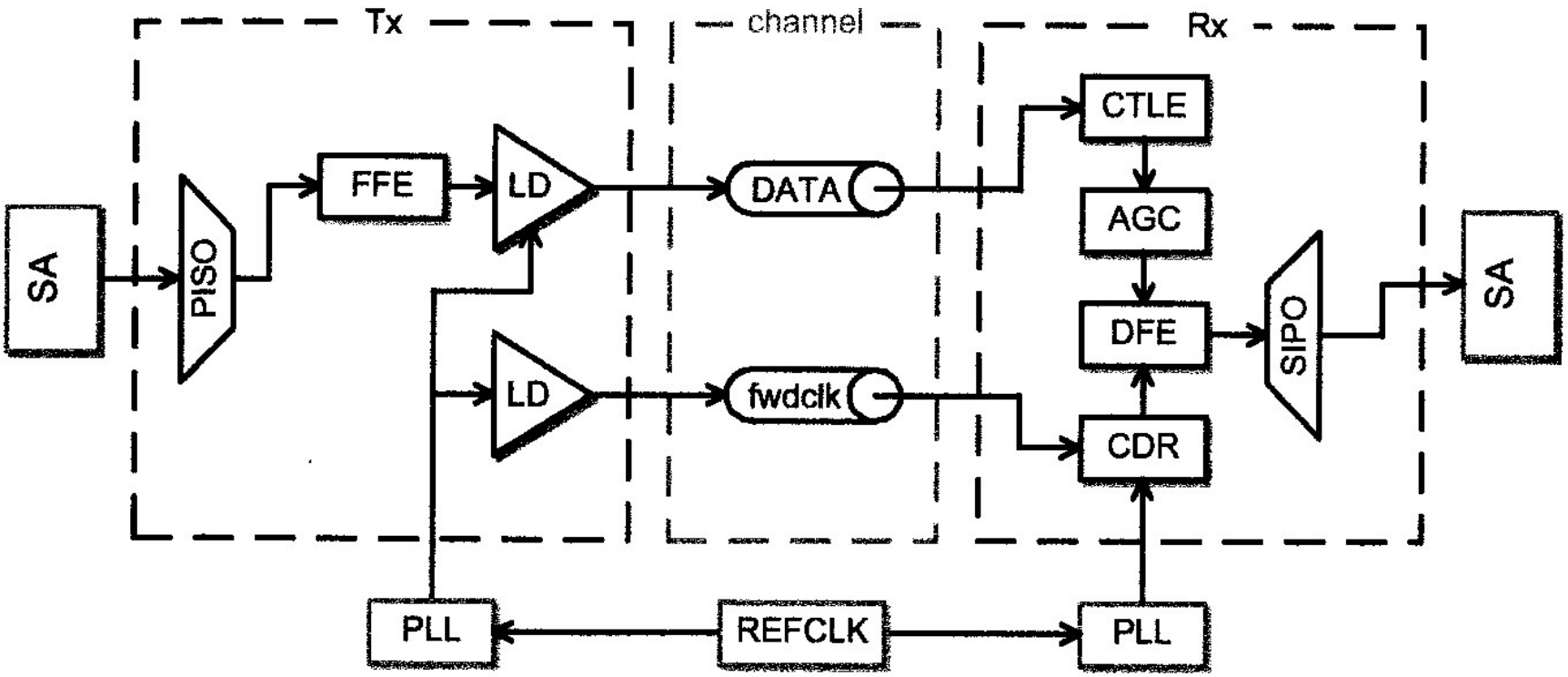
- Serializer:
- Deserializer

Embedded Clock Architecture



- Converts parallel data into serial data (Tx side)
- Applies equalization to the data stream
- Converts serial data into parallel data (Rx side)

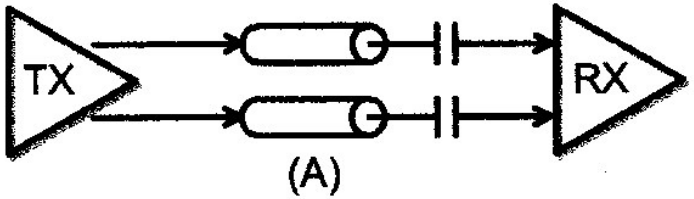
Forwarded Clock Architecture

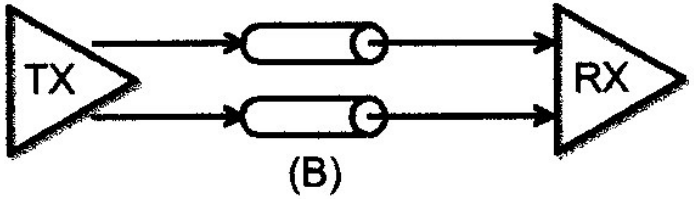


- Additional lane for delivering the clock
- Jitter introduced by the clock can be canceled at receiver
- Offers better jitter performance

DC and AC Coupling

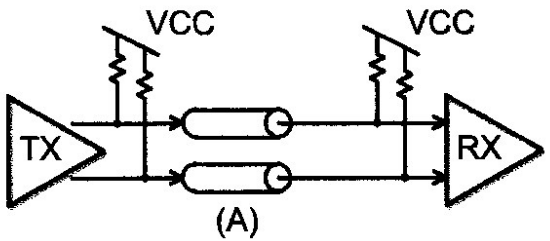
AC Coupled Link



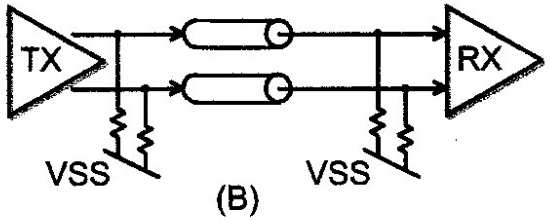


DC Coupled Link

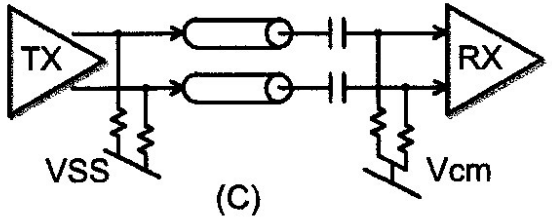
AC coupling has advantage of isolating common-mode voltage levels between RX and TX



Terminated to VCC

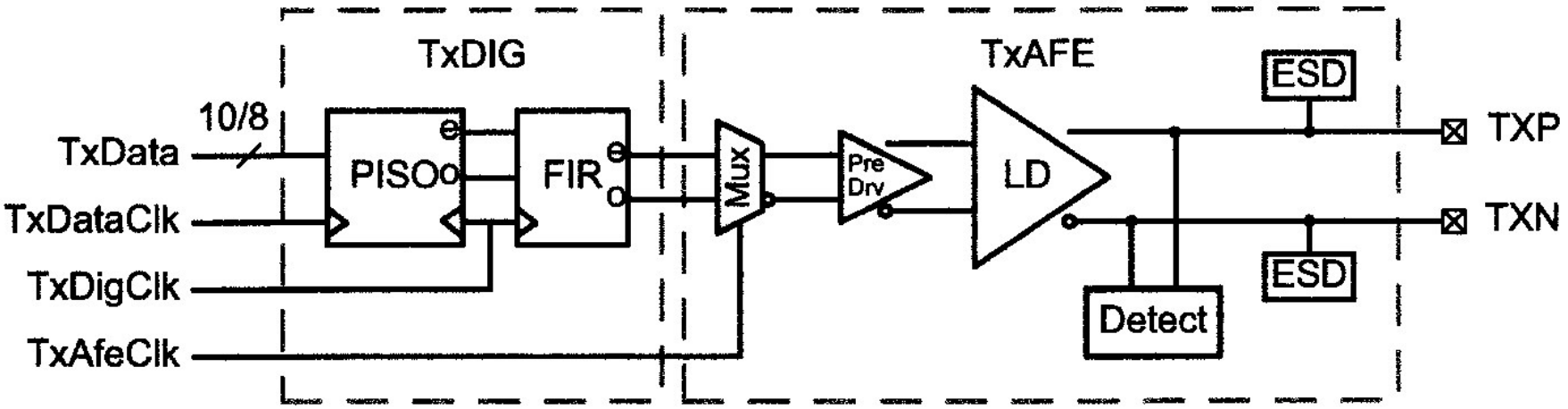


Terminated to VSS



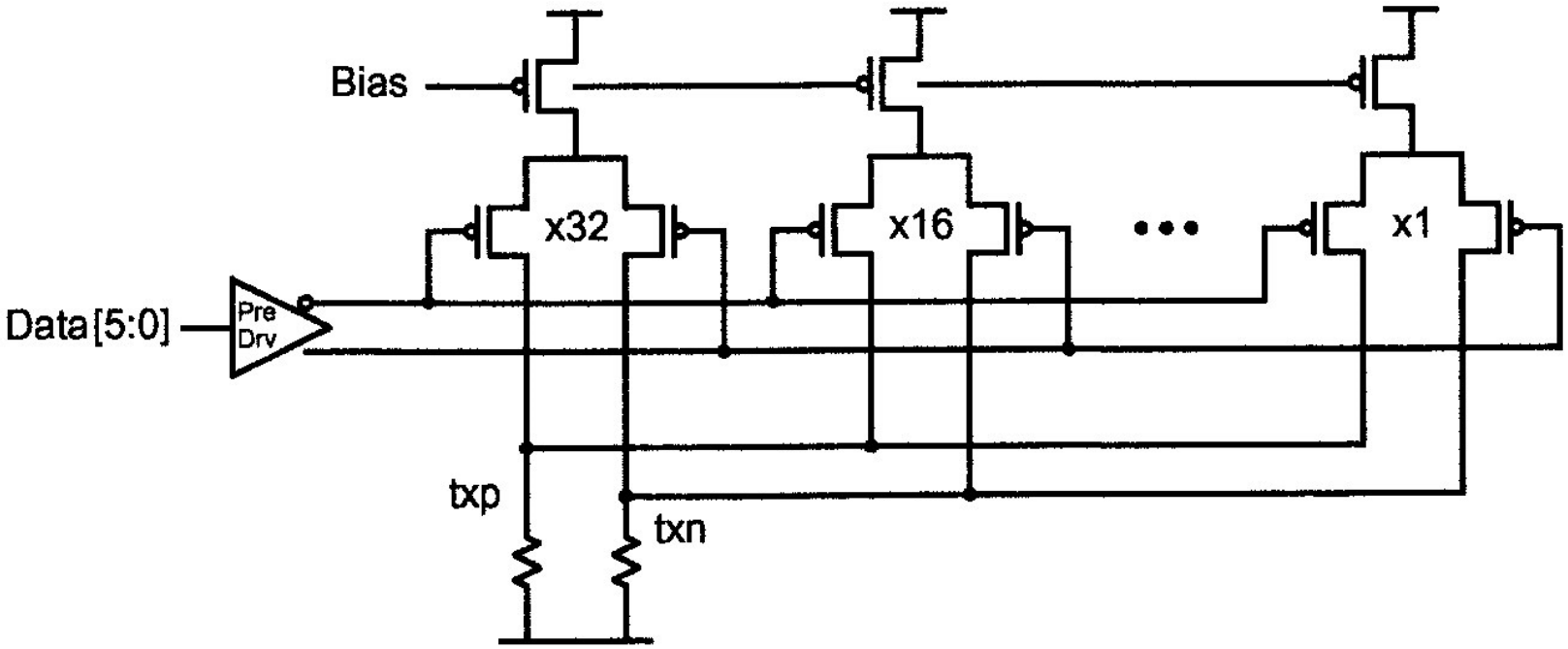
Terminated to Vcm

Transmitter



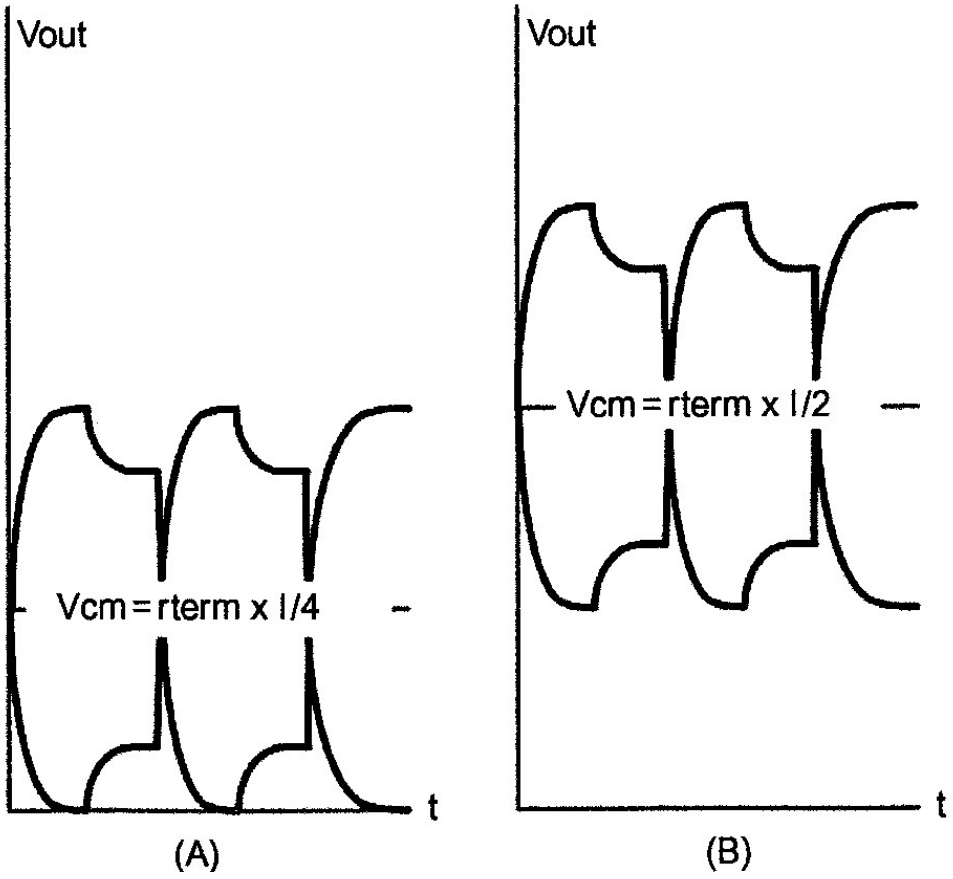
- Need large enough voltage swing
- Pre-driver is used to deliver large enough swing to Tx
- FFE can be realized anywhere along data path

Current-Mode Driver



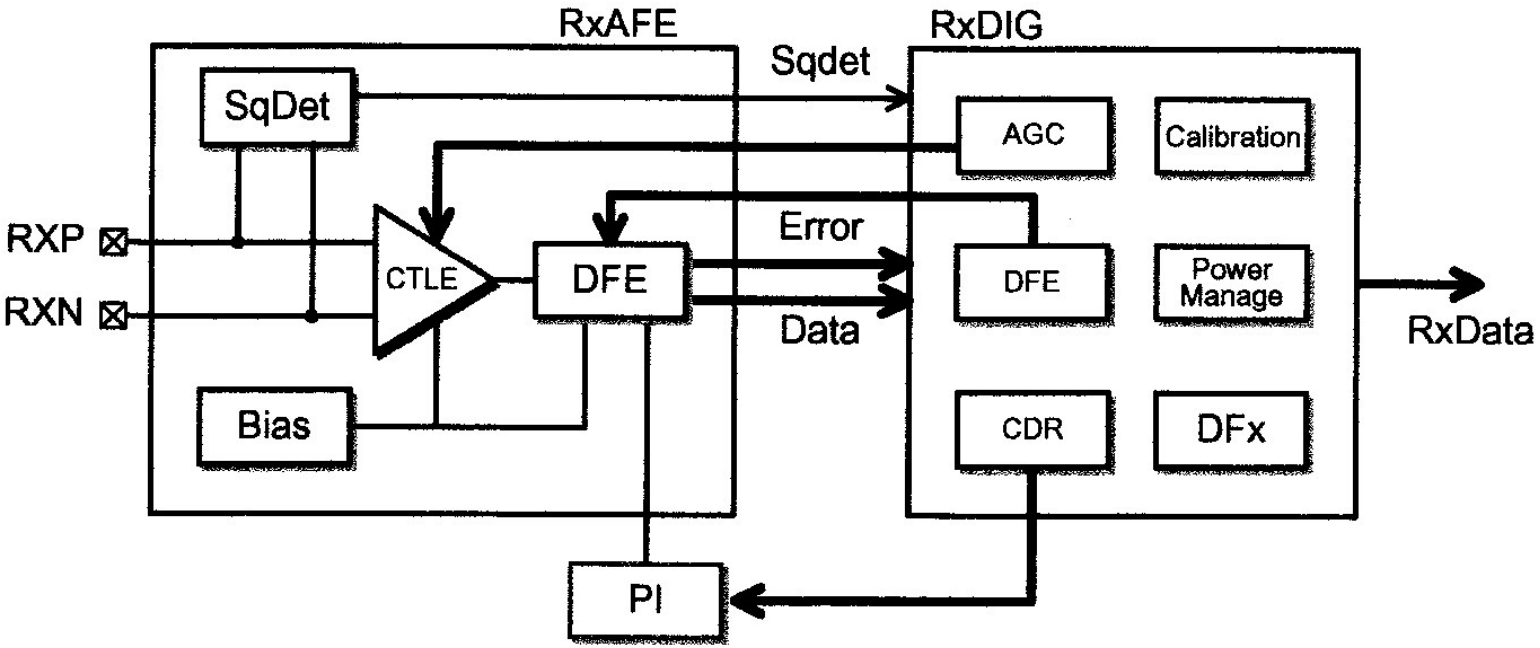
- Group of differential pairs
- Arranged in a binary weighted form
- Controlled by 6-bit equalized data

Current-Mode Driver



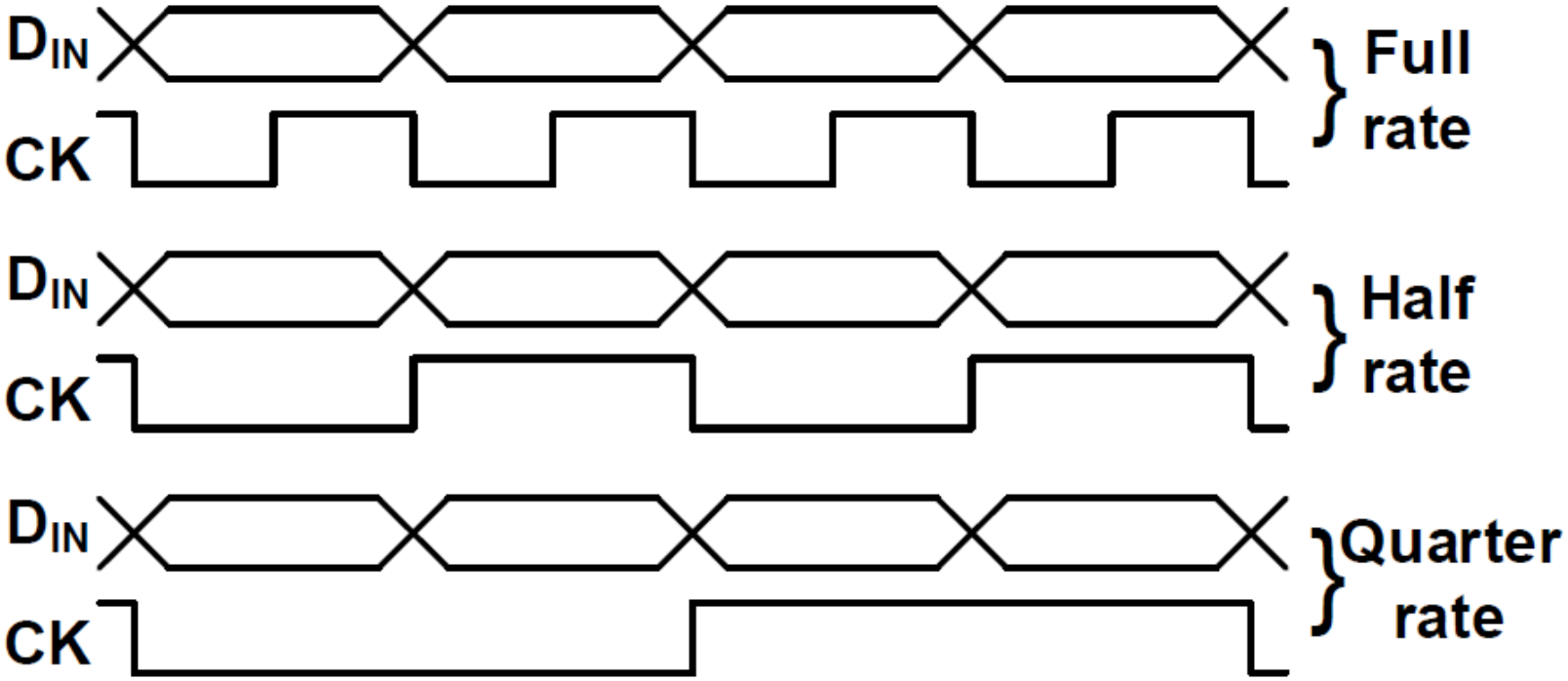
DC-Coupled Link **AC-Coupled Link**

Receiver



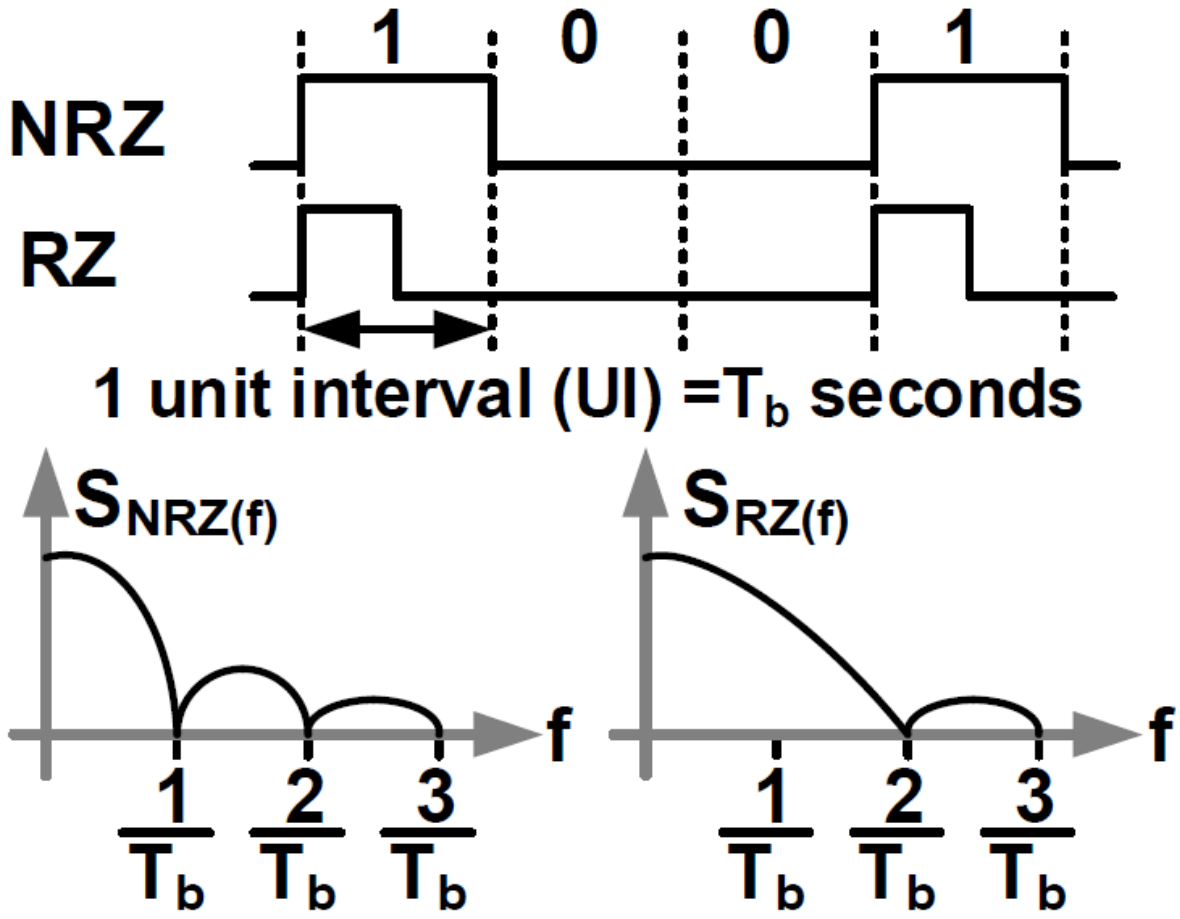
- Receives data
- Performs equalization
- Recovers data and clock

Link Classification



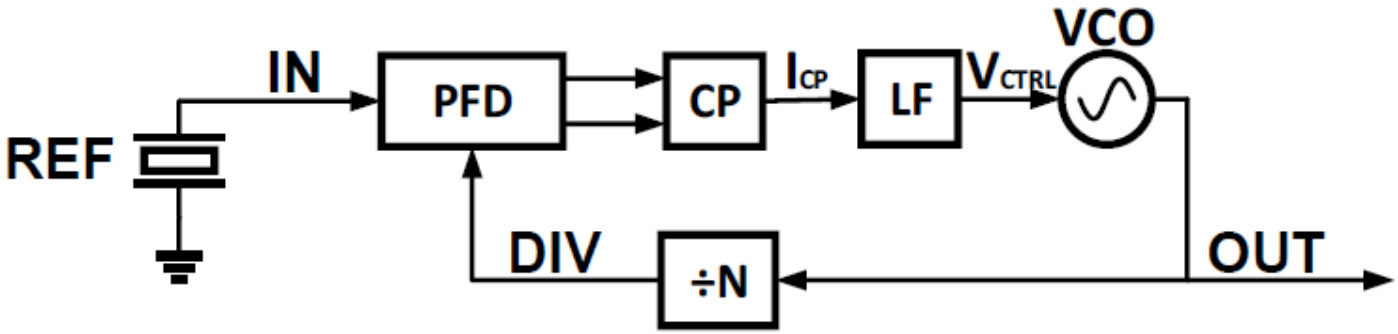
- Number of data bits per clock cycle
- Need multiple phases for half rate and quarter rate

Signaling Protocol NRZ vs RZ



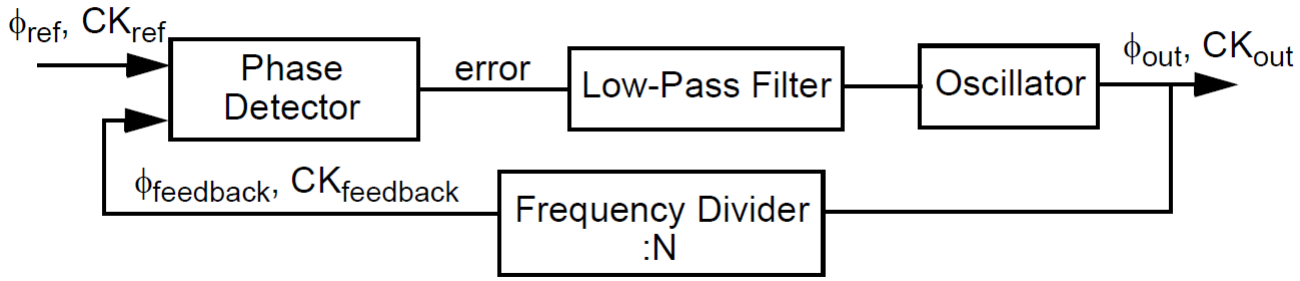
Clock Synthesizer

- PLL based clock-generating circuit needed to have a high-speed system master clock at the TX side.
- Reference clock generated by piezoelectric crystal but can only produce a stable, low-jitter clock in the MHz range.
- Basic Idea: take the reference signal and generate a scaled up clock at a higher frequency by eliminating static phase errors using a negative feedback control system in form of PLL.



PLL Overview

Basic PLL Block Diagram:



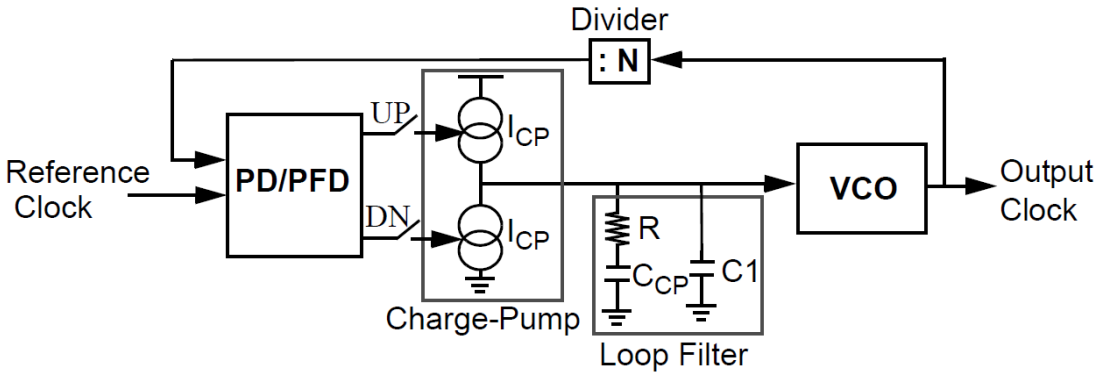
- Closed-loop feedback system that synchronizes the output CLK phase with that of the reference CLK.
- Tracks phase changes w/i the specified BW.
- Idea is that the PD (Phase Detector) will compare the reference CLK phase with that generated by the VCO.
  - Goal: Stabilize $\Delta\phi_e^{SS} \rightarrow 0$ such that VCO output CLK and reference CLK are locked at same frequency and phase.
  - Tracks low-frequencies but rejects high-frequencies.

Why need PLLs?

- Reduces jitter.
- Reduces clock-skew in high-speed digital ckts.
- Instrumental in frequency synthesizers.
- Essential building block of CDRs.

PLL Building Blocks

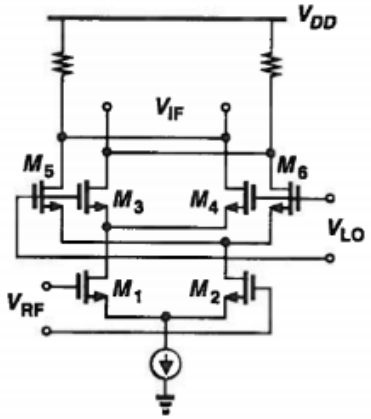
Basic PLL Components:



- PD/PFD ~ Phase/Phase+Frequency Detector
- CP ~ Charge pump circuit
- LF ~ Loop-Filter
- VCO ~ Voltage controlled oscillator
- Frequency Divider

PD/PFD Circuits

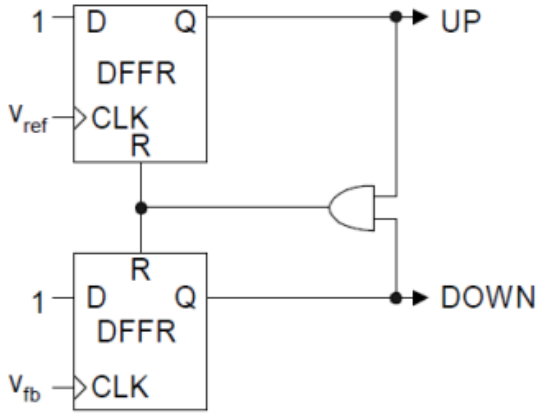
Common PD Implementations:



Gilbert-cell Mixer

XOR PD

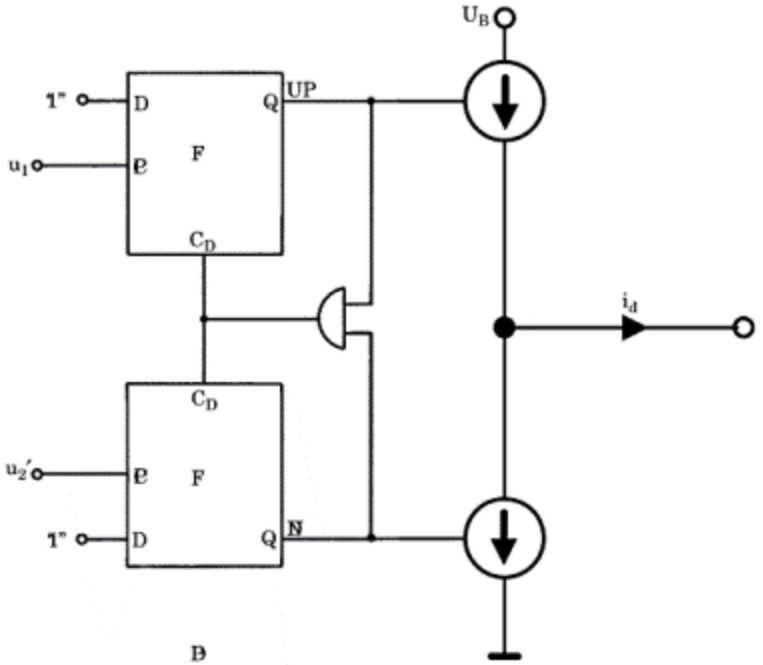
Common PFD Implementations:

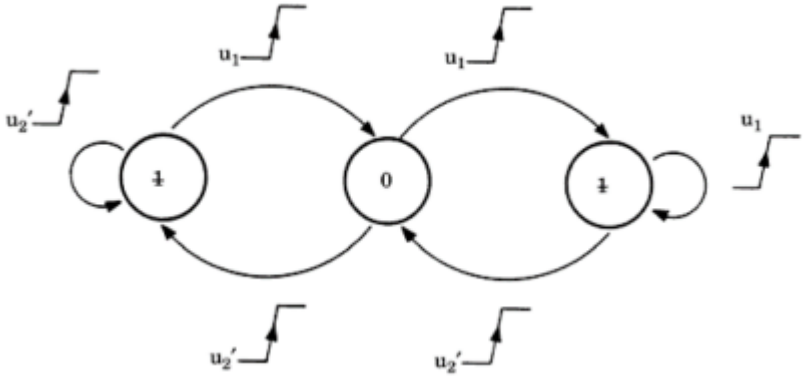


- PD/PFD are strictly digital circuits in high speed SerDes transceivers.
- Ideal PD is a "multiplier" in time-domain, ex: Mixer
- Analog PD → High Jitter, noise.
- XOR PD → sensitive to clock duty cycle
- PFD ~ best to lock phase and frequency!

PFD Theory

1. PFD is needed to adjust the control voltage for VCO according to the phase difference between the VCO output and reference frequency

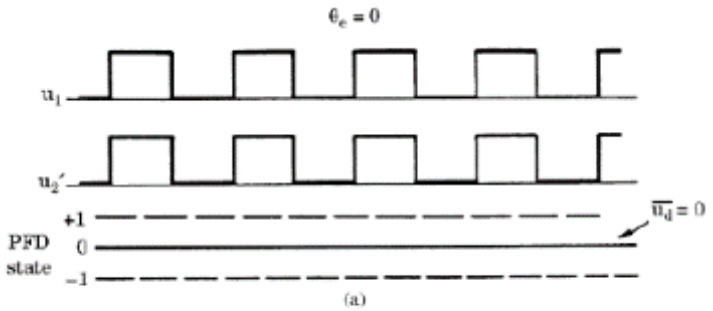




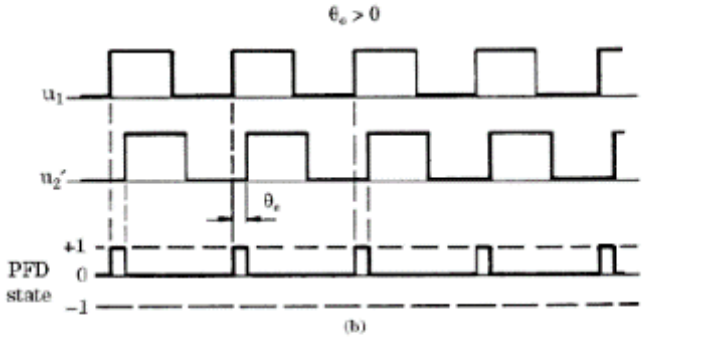
2. PFD can be seen as a state machine with three states. It will change the control voltage of VCO according to its current state and phase/frequency difference will cause state transition.

PFD Analysis

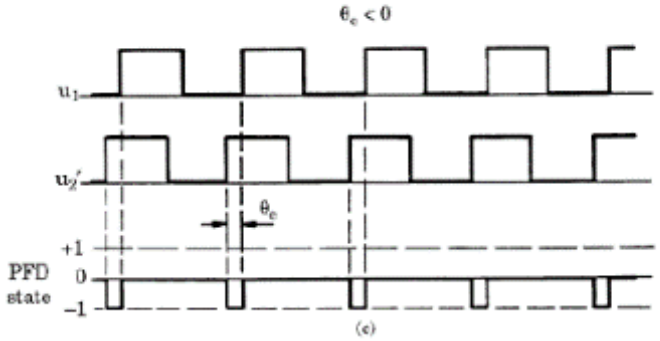
1. PFD is in state 0 with no phase difference.



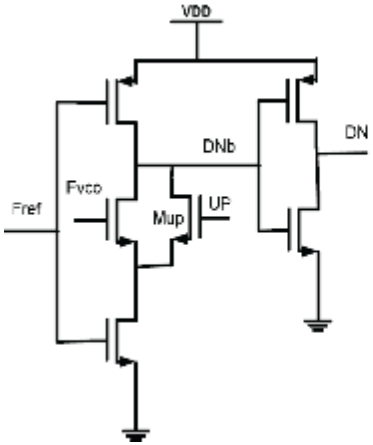
2. PFD is in state 1 with positive phase difference.



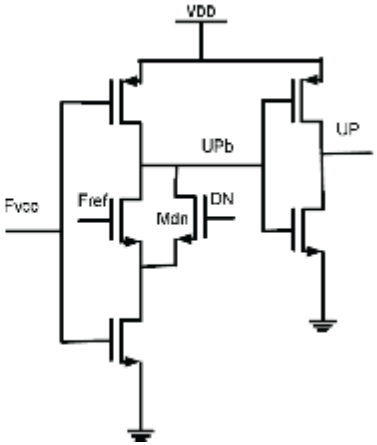
3. PFD is in state -1 with negative phase difference.



PFD Design Overview

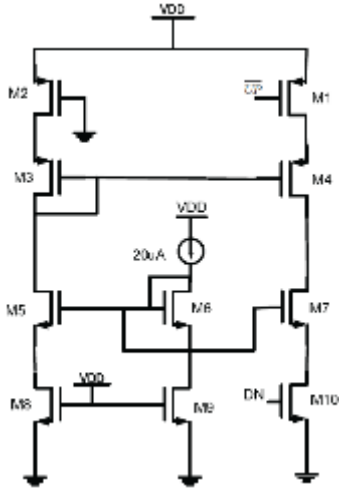


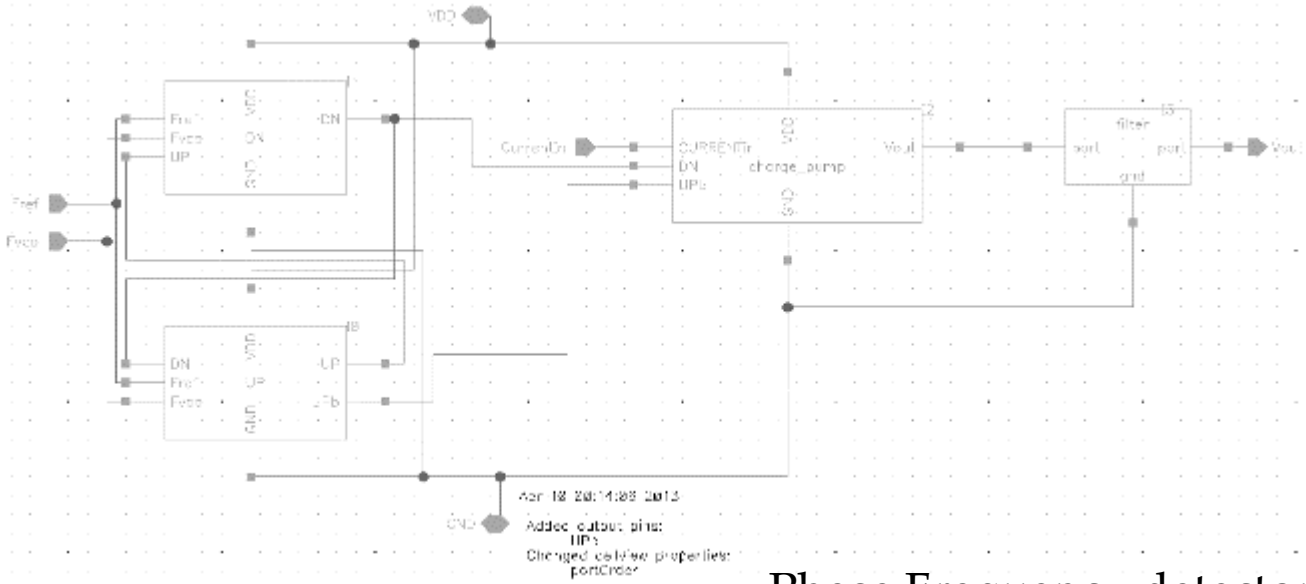
Down circuit



UP circuit

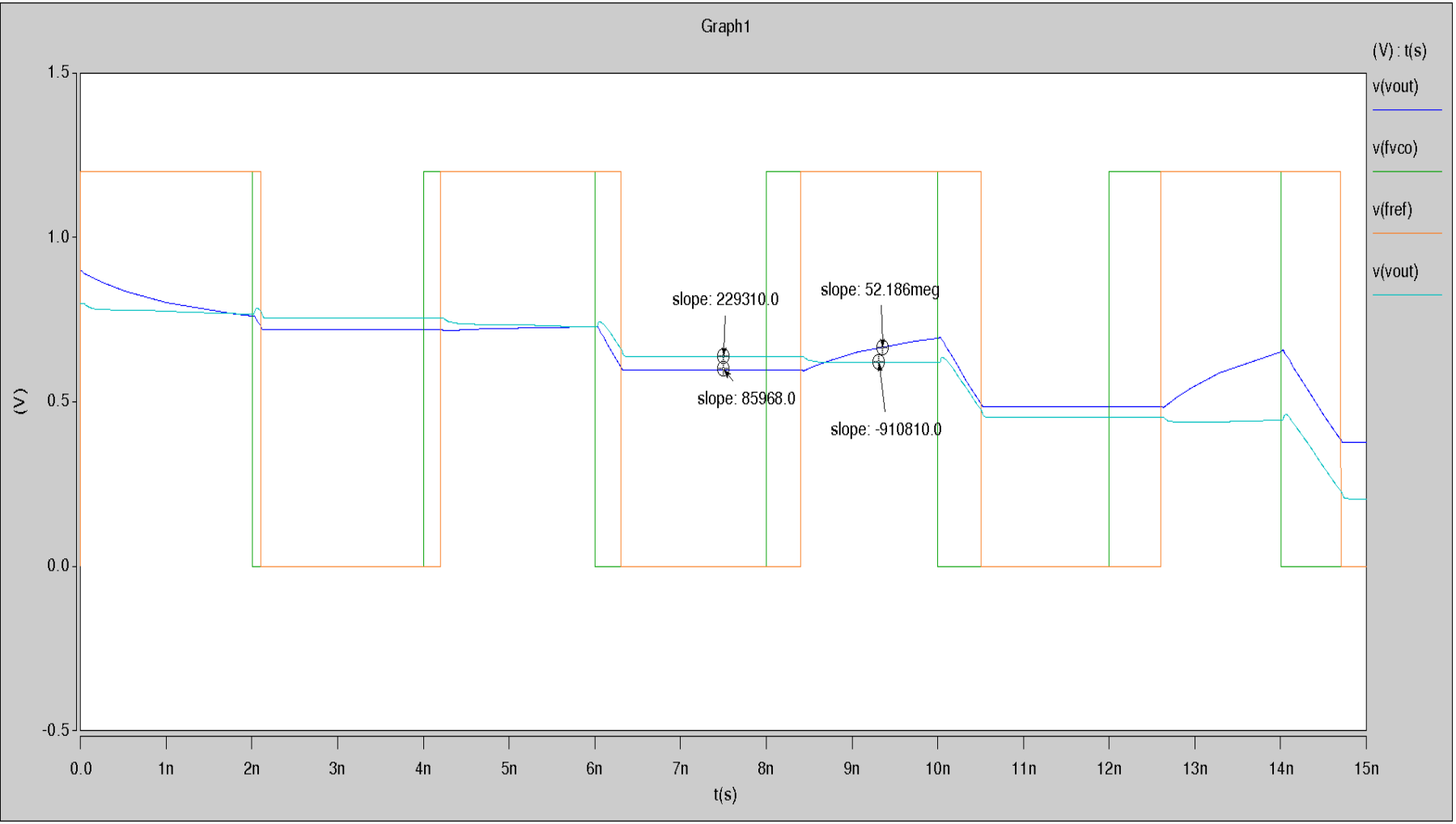
Charge pump





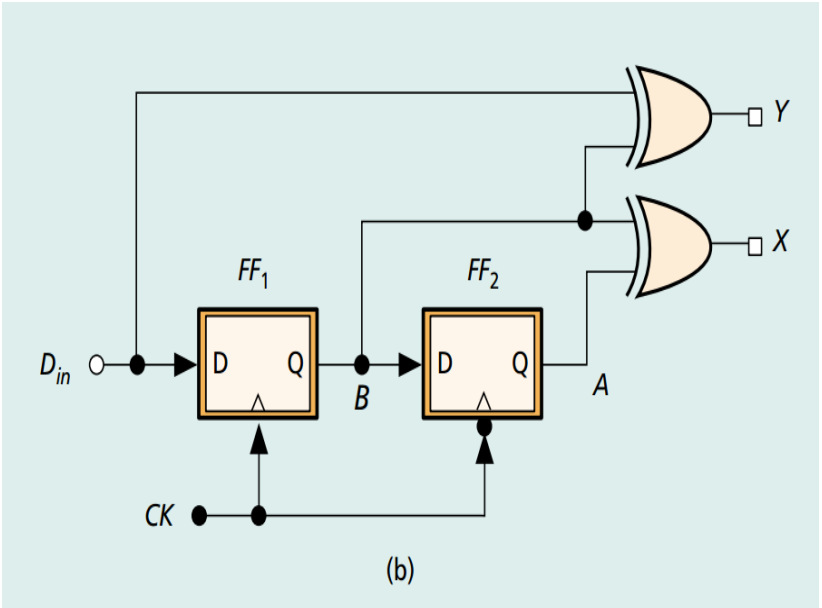
Phase Frequency detector

PFD Simulation



The Hogge Phase Detector

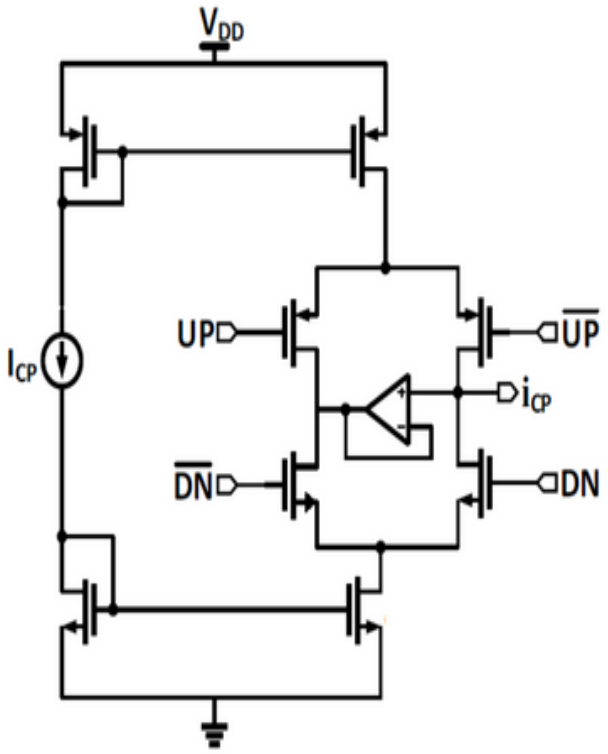
- Two Functions
  - Transition detection
  - Phase Detection



The Charge Pump

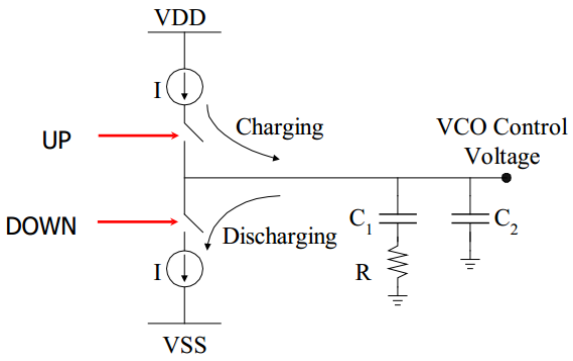
- Combination of current source and sink
- Converts PD output

UP	DN	Io
0	0	0
1	0	Icp
0	1	-Icp

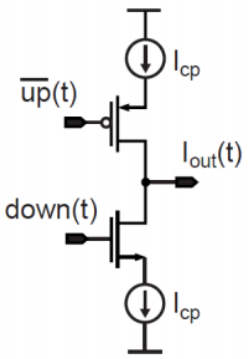


Charge-Pump Circuit

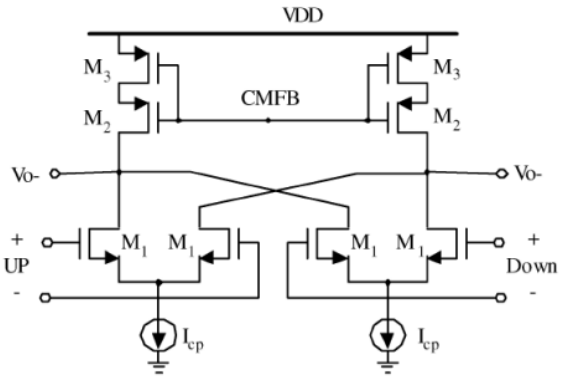
Common CP Implementations:



Single-Ended



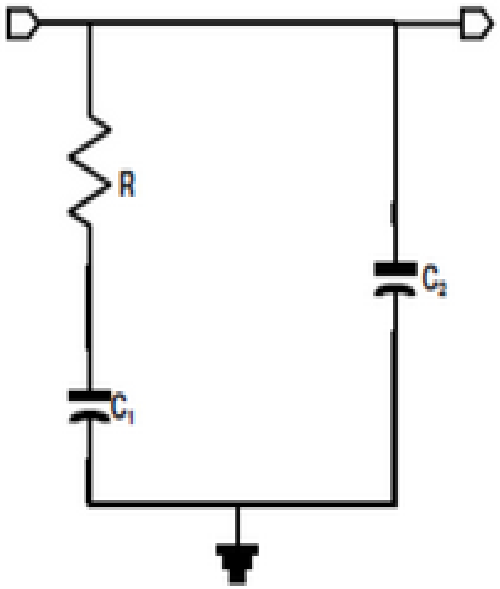
Fully Differential



- Used in conjunction with PFD over PD+LF combo. b/c:
  - Higher capture/lock acquisition range of PLL
  - $\Delta\phi_e^{SS} = 0$ provide no device mismatch exists.
  - Provide infinite gain for a static phase-error

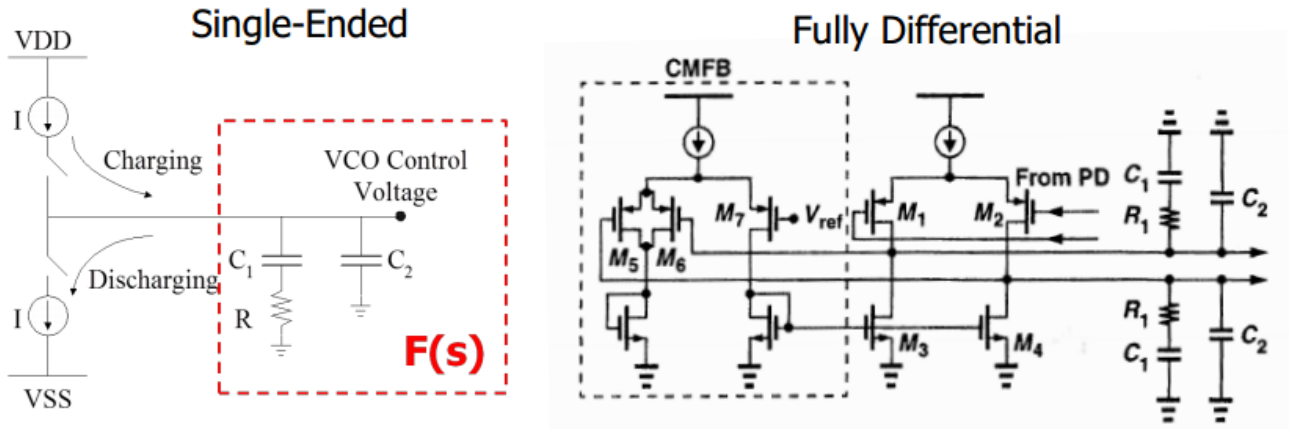
The Loop Filter

- Low-pass for rejection of high frequency noise
- Forms the control voltage of the VCO



Loop-Filter

Common LF Implementations:

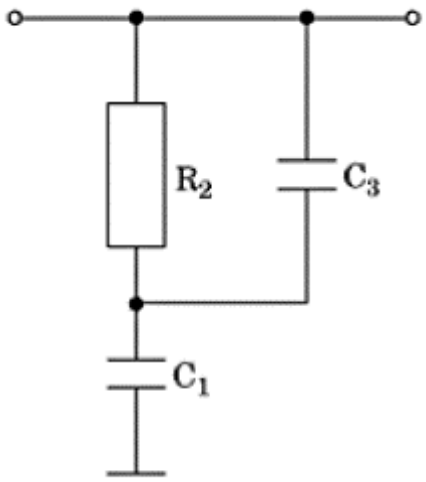


- Extracts average of PD error signals generate VCO control voltage.
- Integrates low-frequency phase-errors on C1 to set avg. freq.
- R adds thermal noise, C_1 determines loop BW, C_2 smoothens control voltage ripple.

Loop-Filter Design

1. Needed to filter out high frequency noise generated by PFD

2. Due to the superior performance of PFD, only a passive second order RC low pass filter is needed.



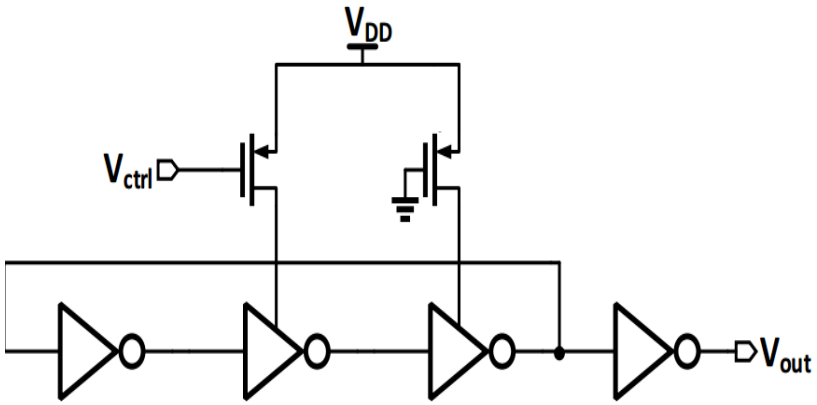
Low pass filter for current input

Where $R_2 = 70.18K\Omega, C_1 = 72.56\text{f}F, C_3 = 18.136\text{f}F$;

Assuming $K_0 = 4.5\,GHz/V$; $K_P = 3.183\mu A/rad$, $N = 8, \omega_T = 25\text{MHz}$

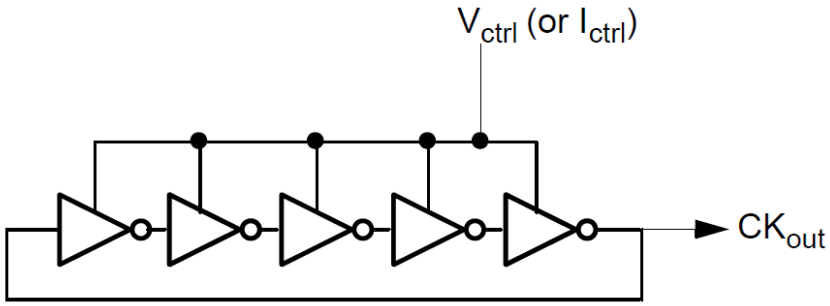
Voltage Controlled Oscillator

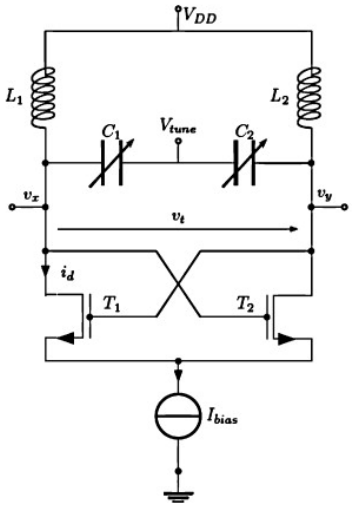
- Generates an output with oscillation frequency proportional to the control voltage
- Helps the CDR accumulate phase and achieve lock



VCO

Common VCO Implementation:

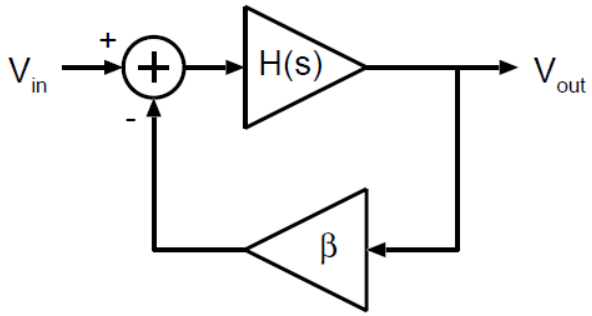




LC-Tank Oscillator

- Extracts average of PD error signals generate VCO control voltage.
- PLL acts like a High-pass filter with respect to VCO jitter.
- VCO always has one pole!

Oscillators Overview



- Closed-Loop Transfer function:

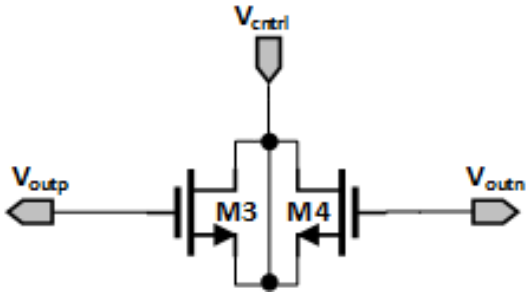
$$-\frac{V_{out}}{V_{in}}(s) = \frac{H(s)}{1+\beta H(s)}, where\ s = j\omega$$

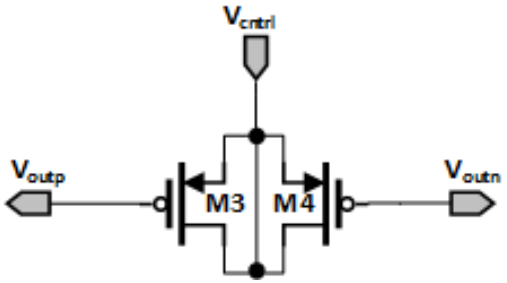
- Barkhausen's criteria for oscillation:
  - $|\beta H(j\omega_0)| = 1$
  - $\arg(\beta H(j\omega_0)) = -180^0.$
- ω_0 = oscillation-frequency.

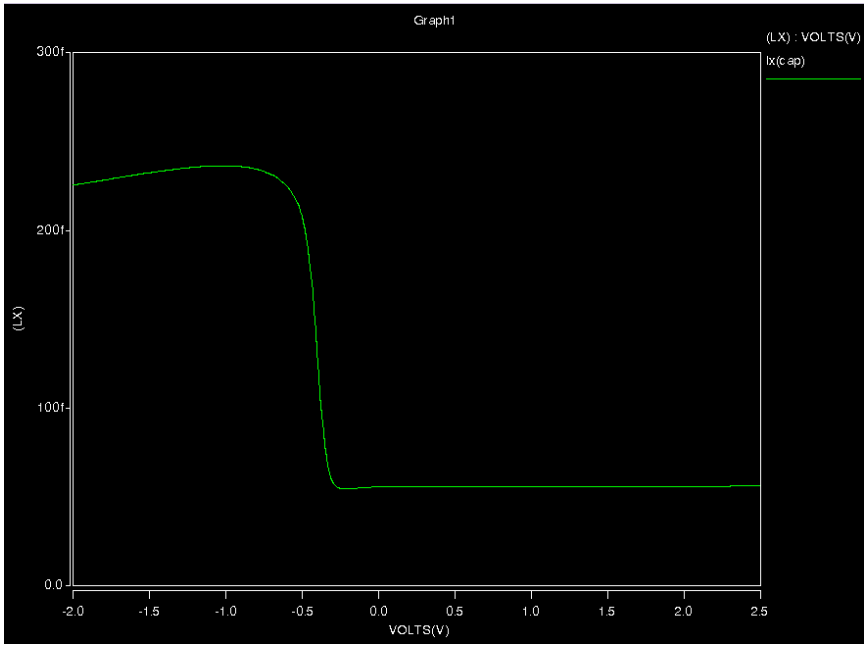
Ring v/s Tank Architecture

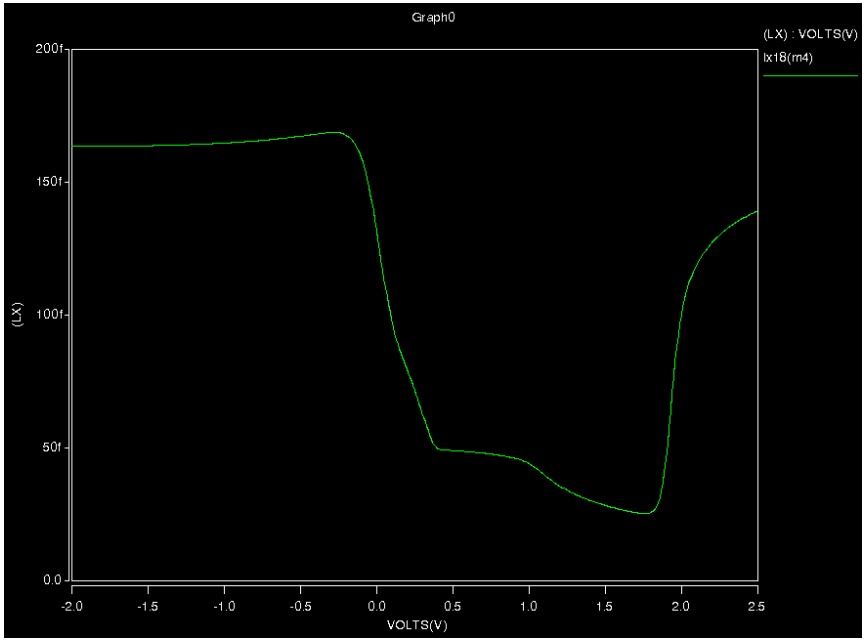
Ring Structure	LC-Tank Structure
1. Low-power, highly integrated.	1. High-power, not integrable.
2. Occupies smaller die-area.	2. Occupies large die-area.
3. Poor-performance at high-frequency due to large phase-noise + jitter.	3. Great phase-noise and jitter performance at high frequency.
4. Can only accept digital signals.	4. Can accept analog and digital signals.

MOS Varactor

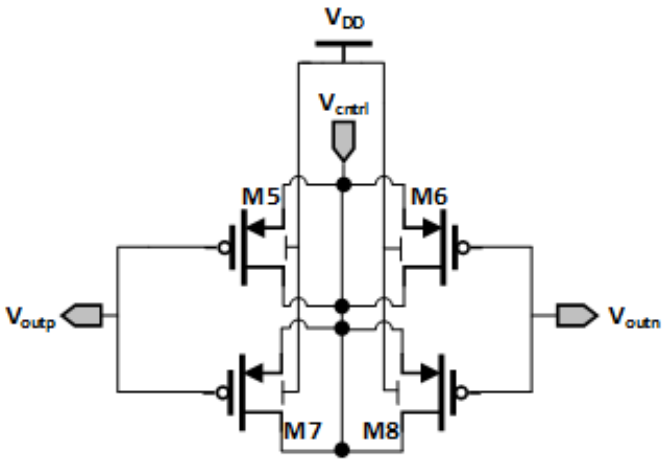


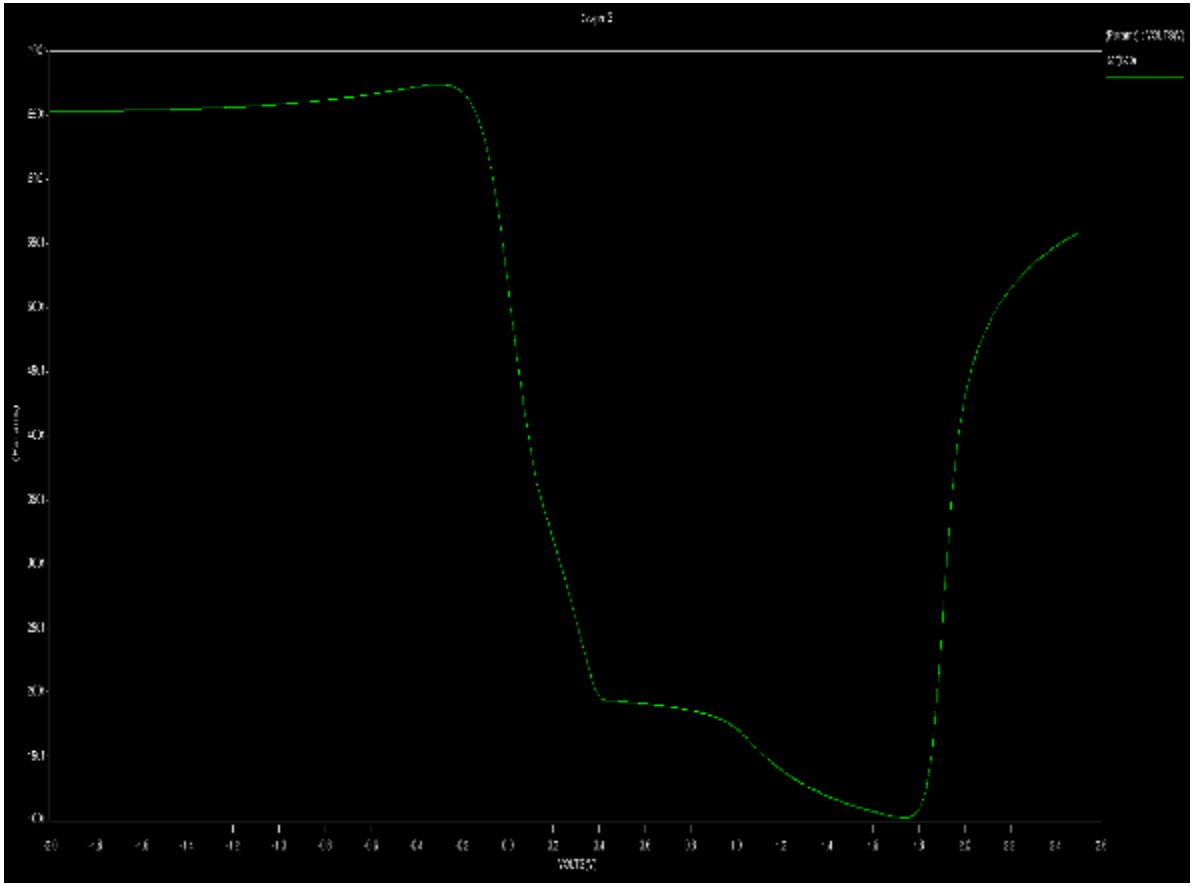




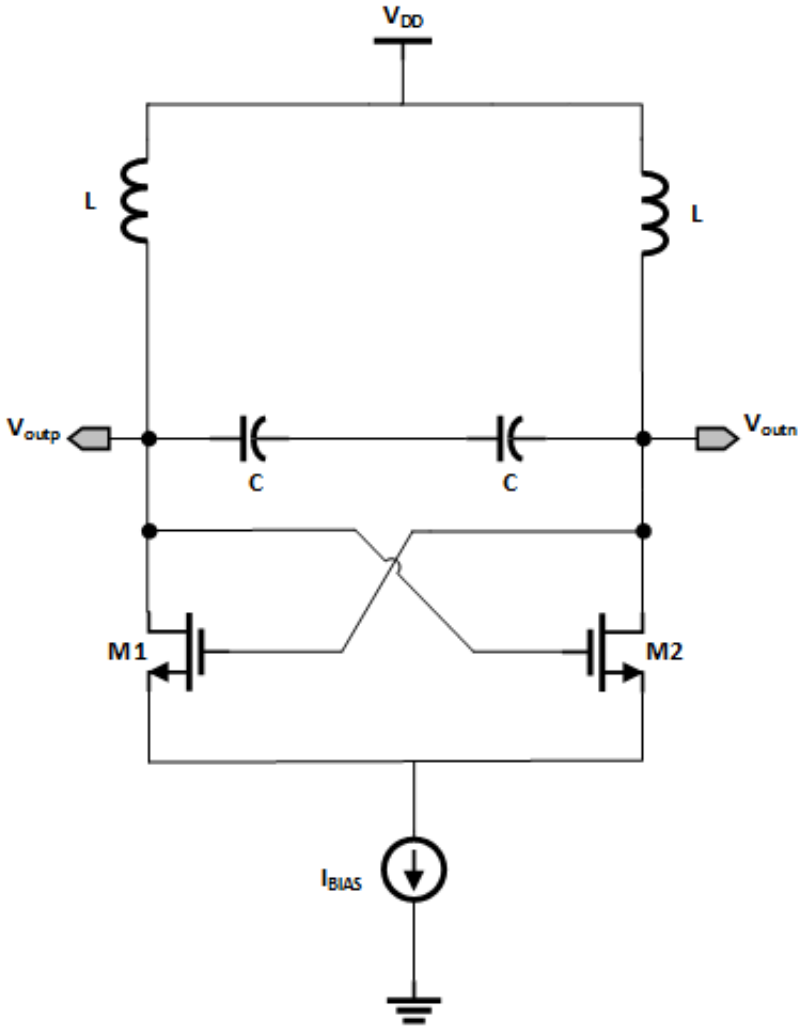


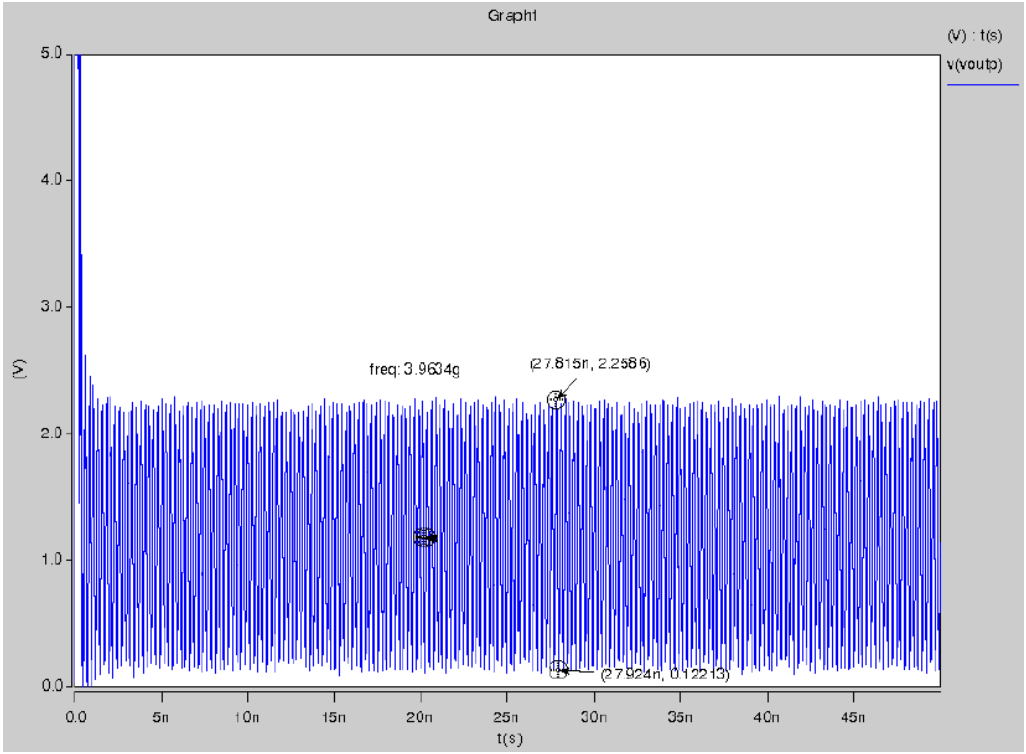
Cascode MOS Varactor



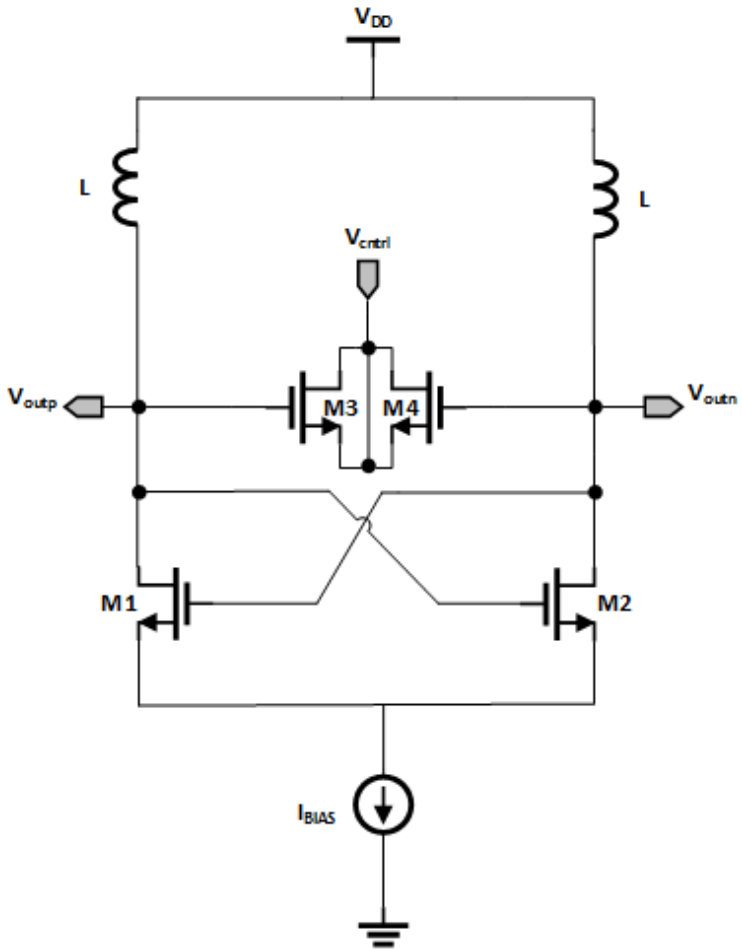


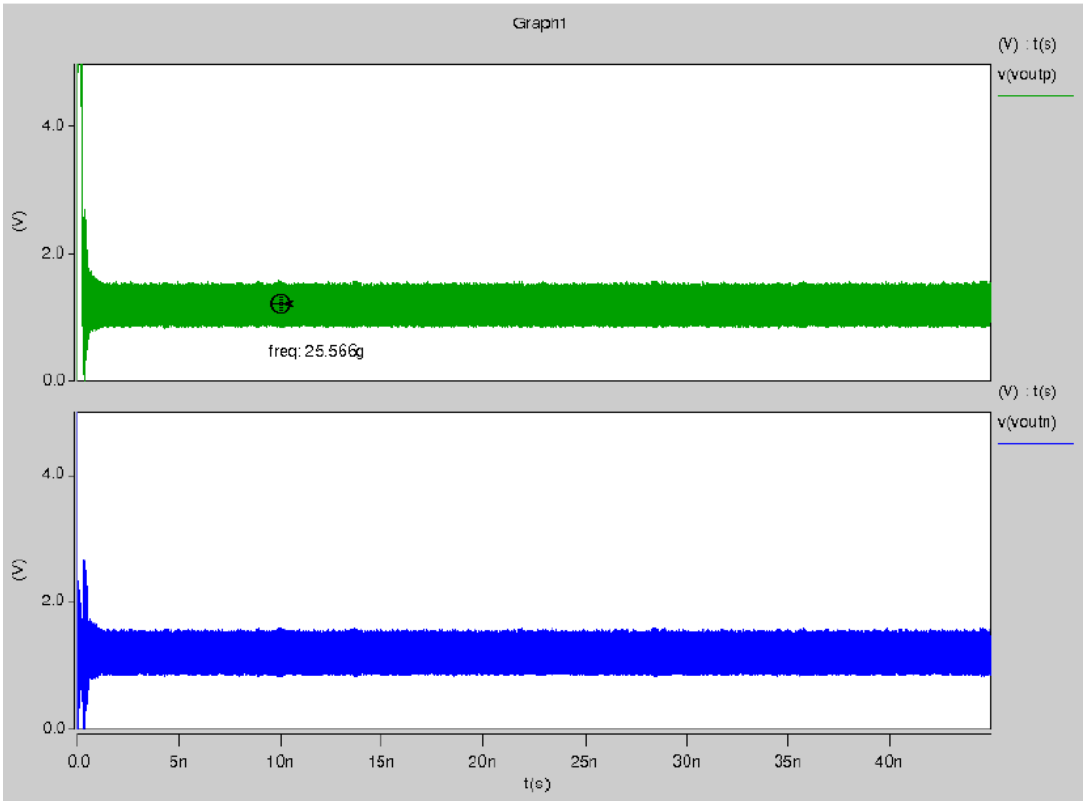
LC-Tank VCO Designs - I



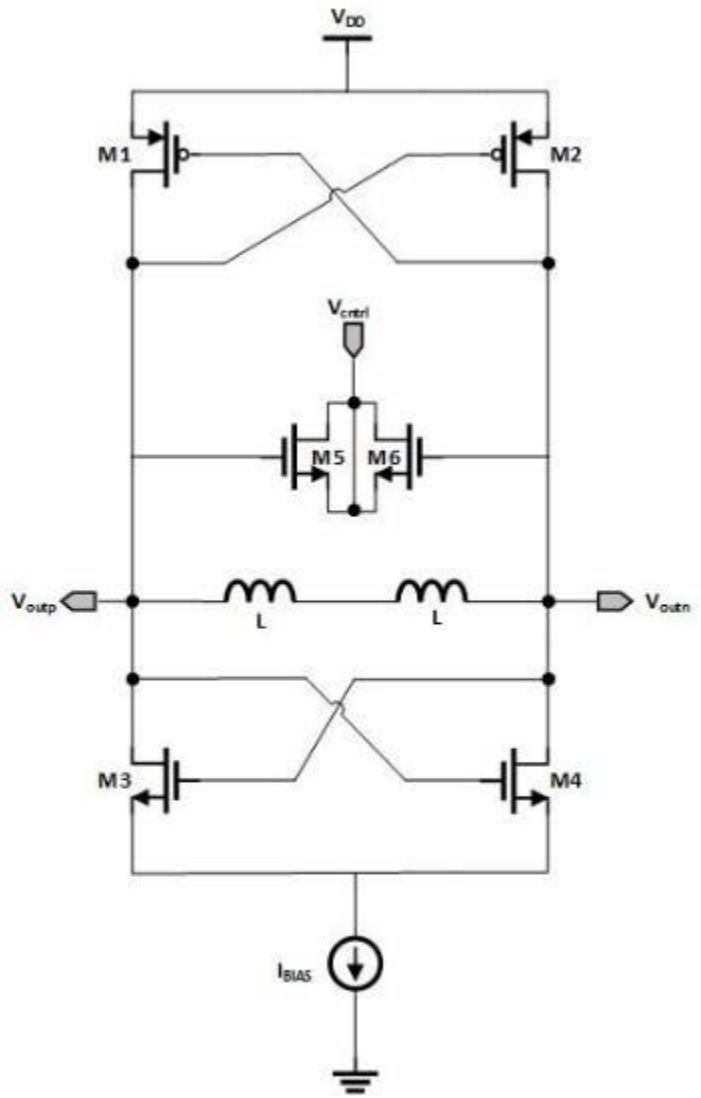


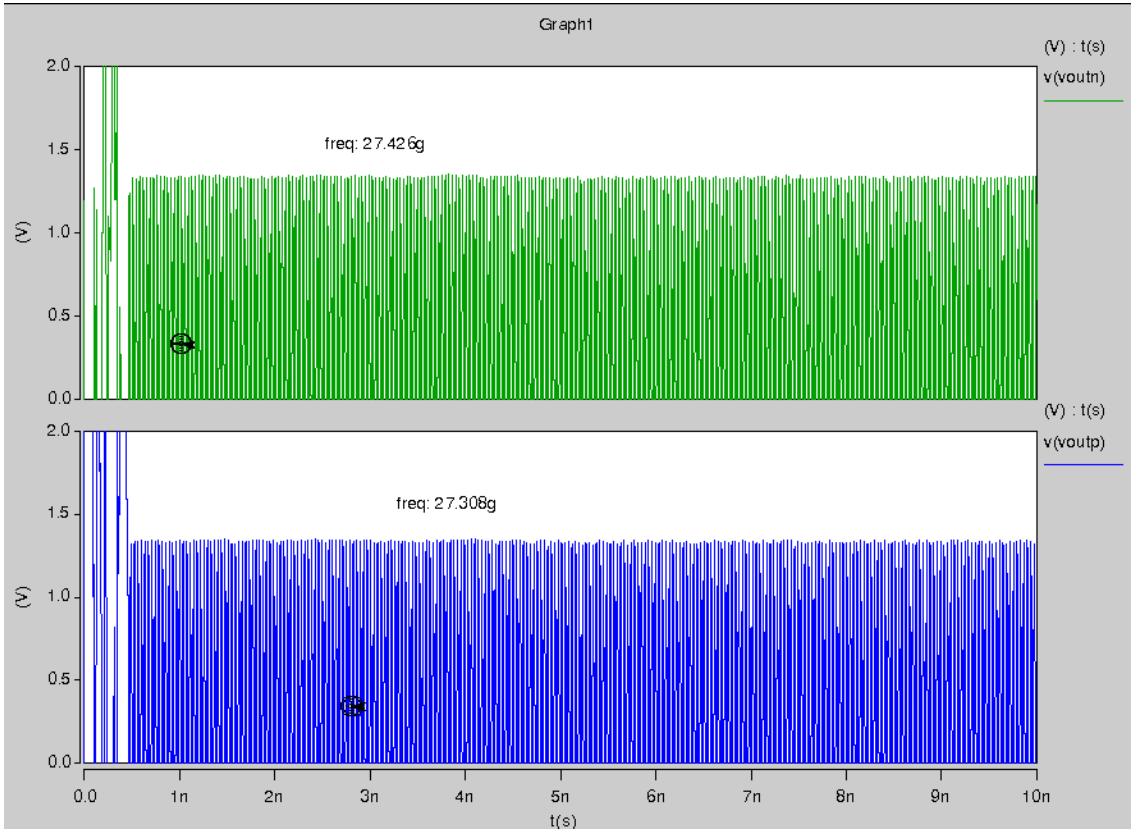
LC-Tank VCO Designs - II



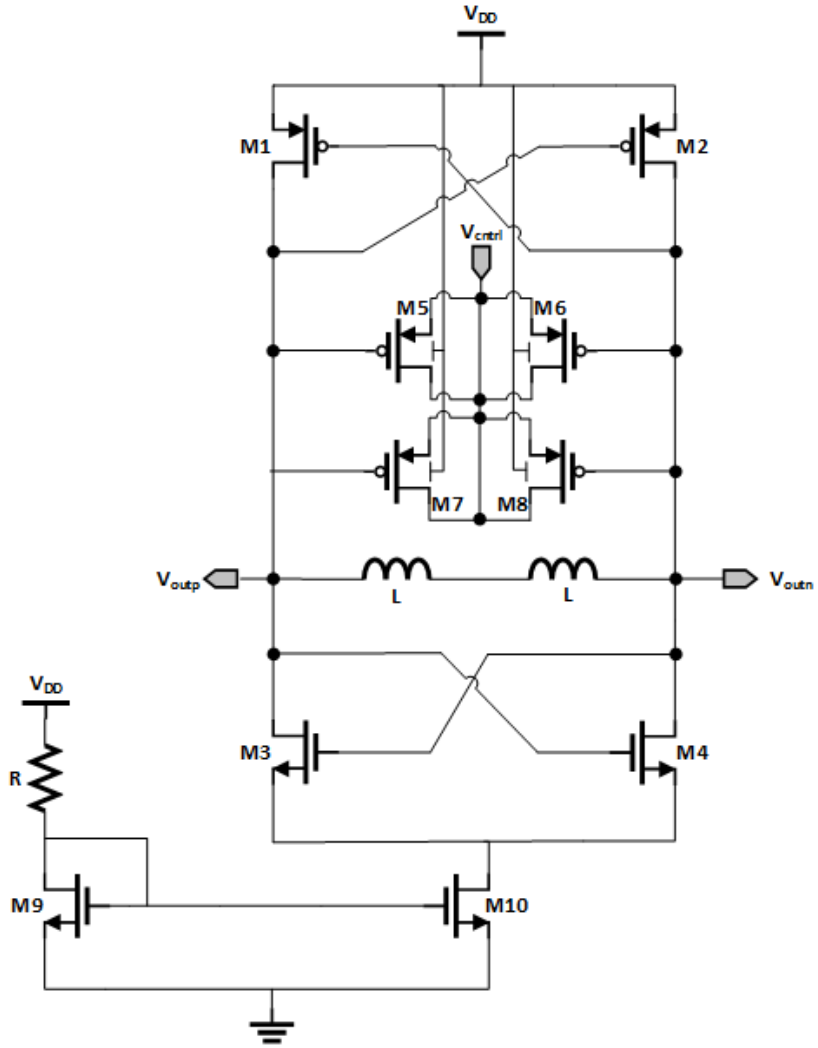


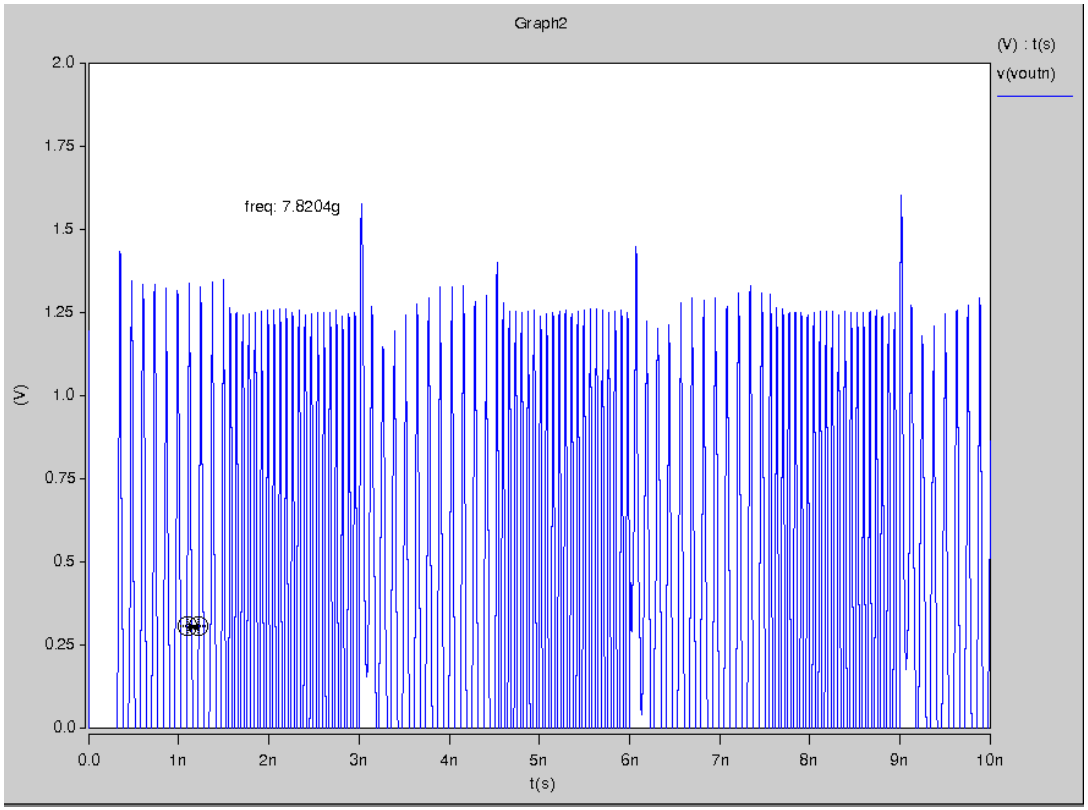
LC-Tank VCO Designs - III





LC-Tank VCO Designs - Final

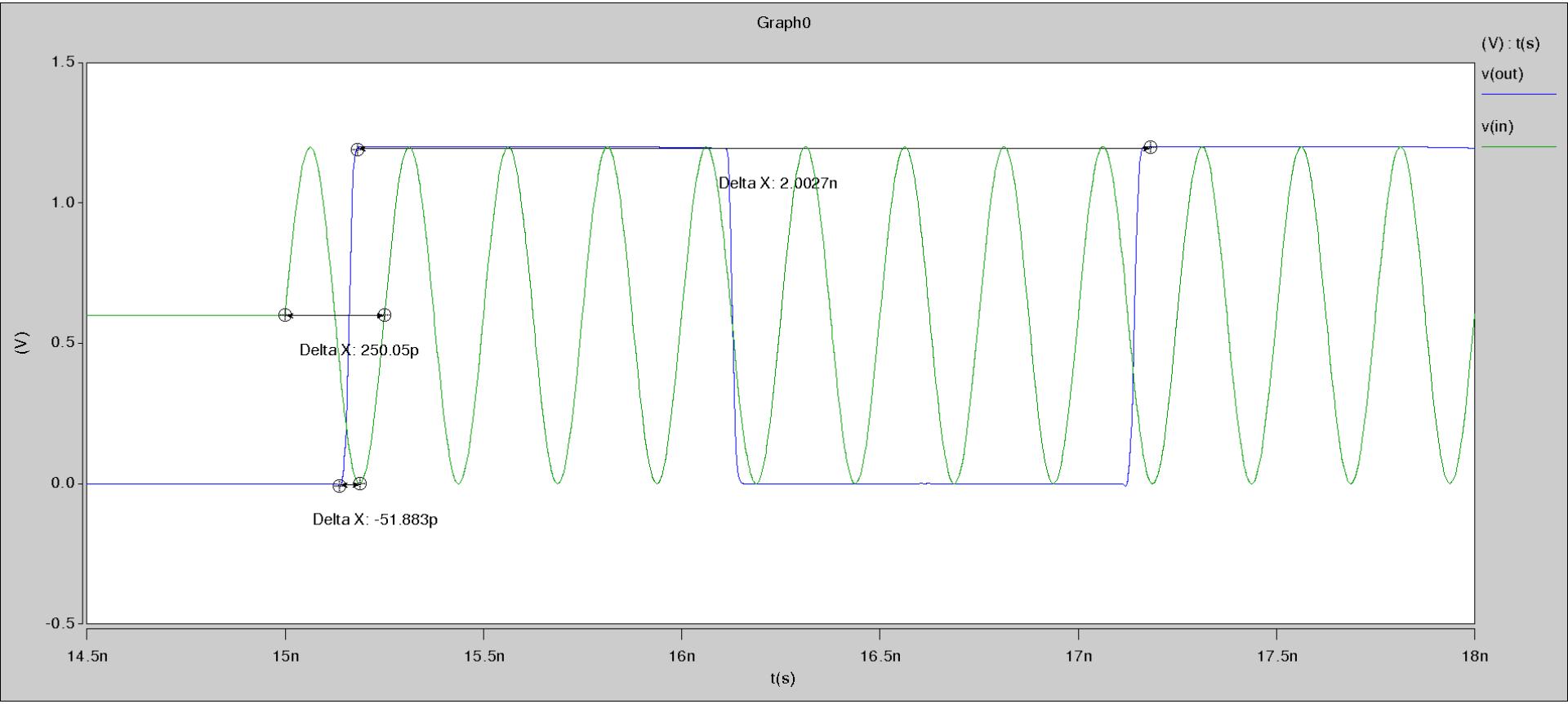




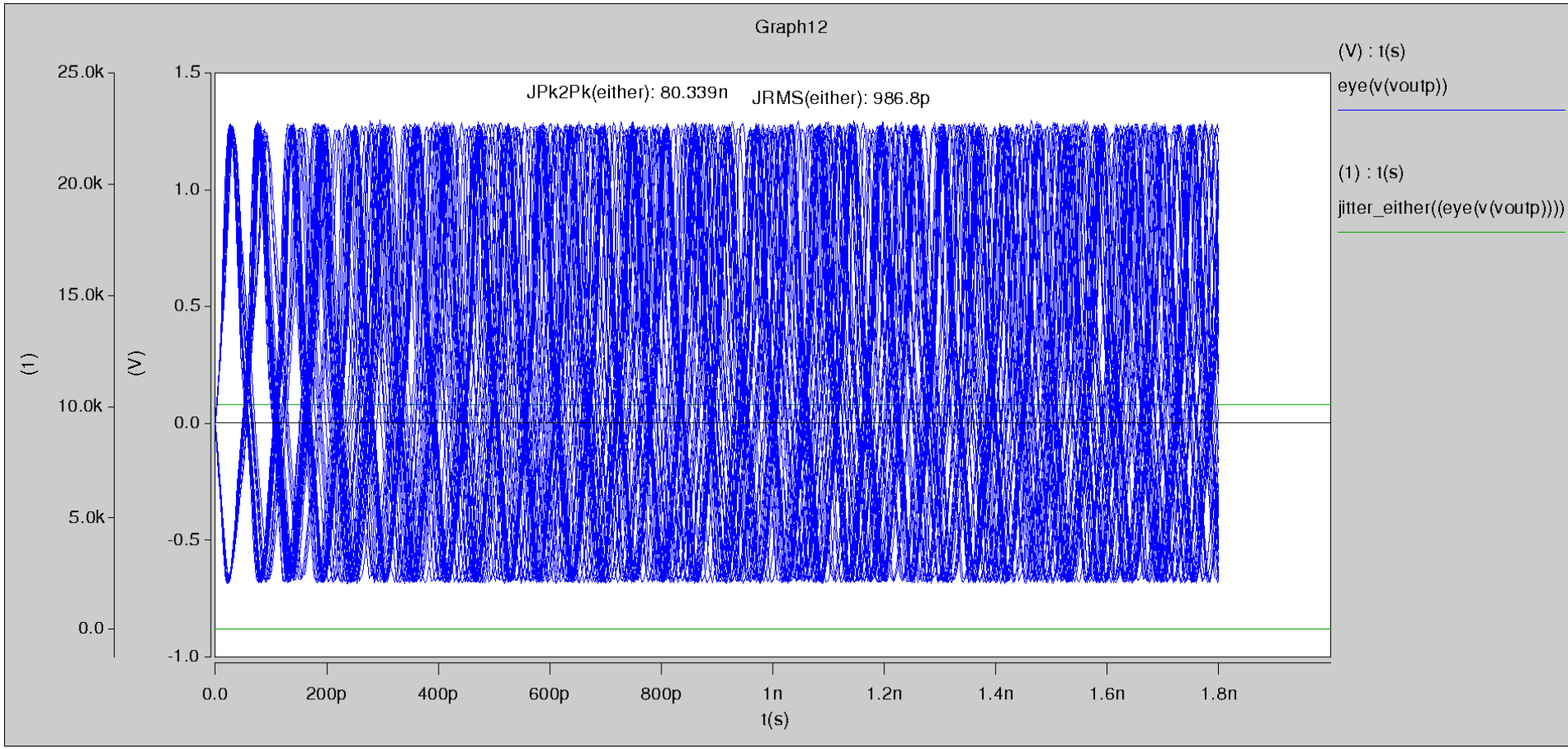
Final VCO Design Parameters

M1	L = 100n, W = 2u
M2	L = 100n, W = 2u
M3	L = 100n , W = 2u
M4	L = 100n, W = 2u
M5	L = 500n, W = 10u
M6	L = 500n, W = 10u
M7	L = 500n, W = 10u
M8	L = 500n, W = 10u
M9	L = 100n, W = 2u
M10	L = 50n, W = 2u
L	1.5nH, Q = 5
R	465 Ω

Fractional N-Divider Simulation



VCO Jitter Analysis



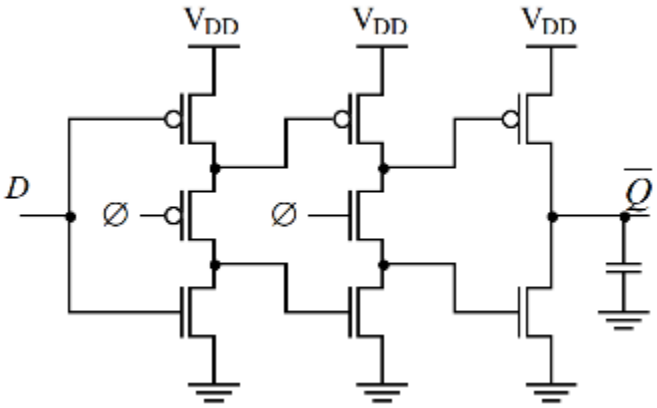
Theoretical Design Overview

- $Q = Q_L = \frac{\omega_0 L}{R} = 5$
- $\omega_0 = \frac{1}{\sqrt{LC}}\sqrt{1 - \frac{R^2 C_{var}}{L}}$
- $Choose\ g_m \geq \frac{RC_{var}}{L}\ for\ each\ transistor.$
  - Recall, $g_m = \sqrt{\mu_n C_{ox}\left(\frac{W}{L}\right) I_{bias}}$
- $R_{PU} = -\frac{2}{g_{m1,2}}\ and\ R_{PD} = -\frac{2}{g_{m3,4}}$

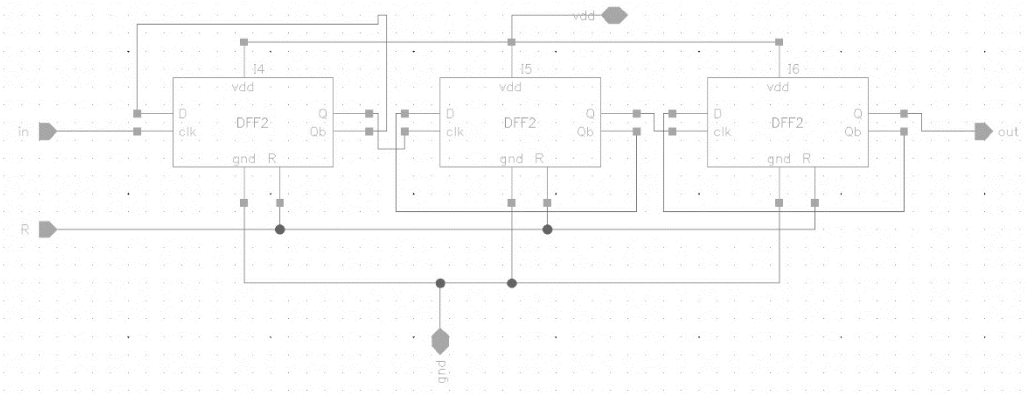
Fractional N-Divider Circuit

1. Needed to slow down the VCO's output so that PFD can compare it with reference frequency.

2. N D-FlipFlops cascaded together to achieve 2^N divider.

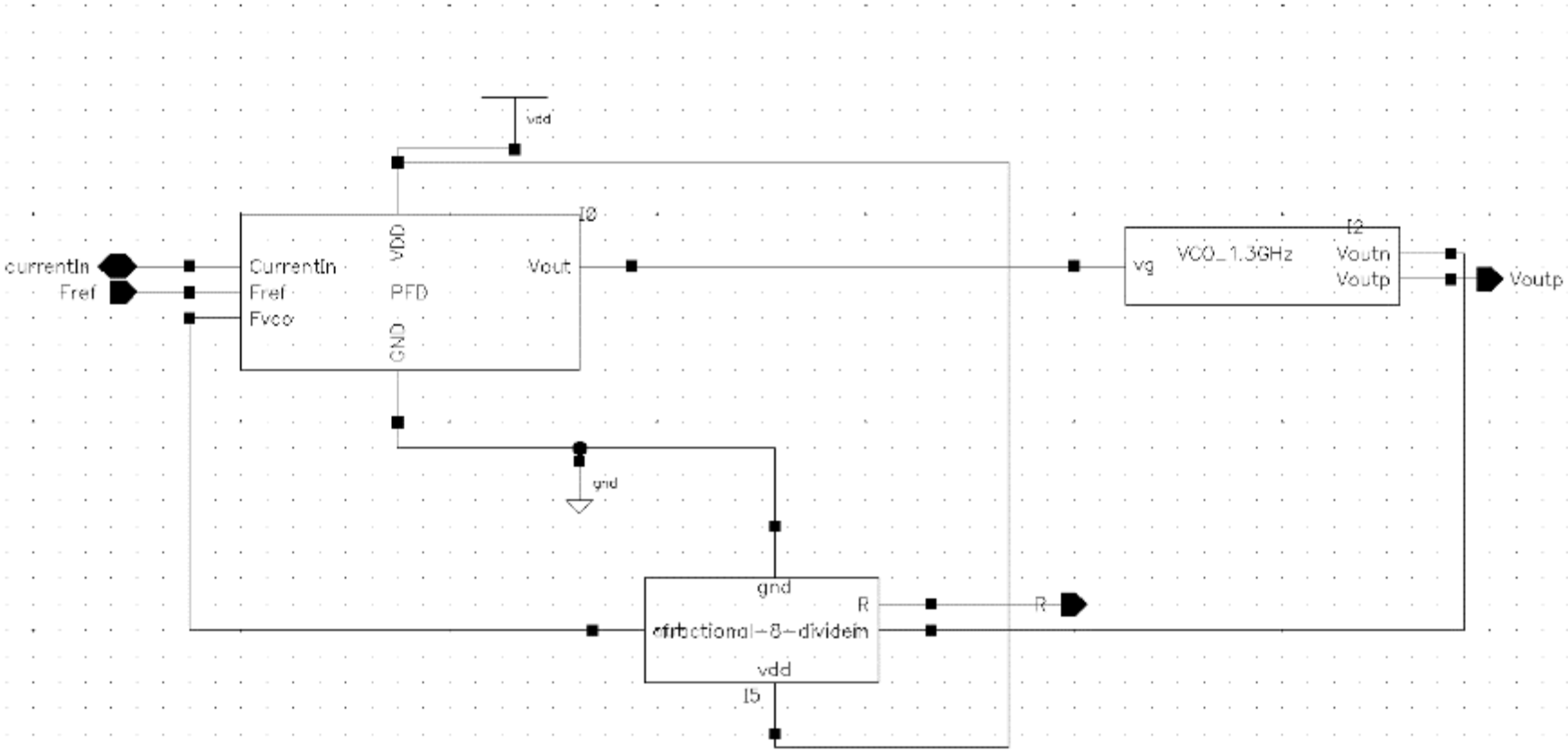


Positive edge-triggered DFF using split-output latches

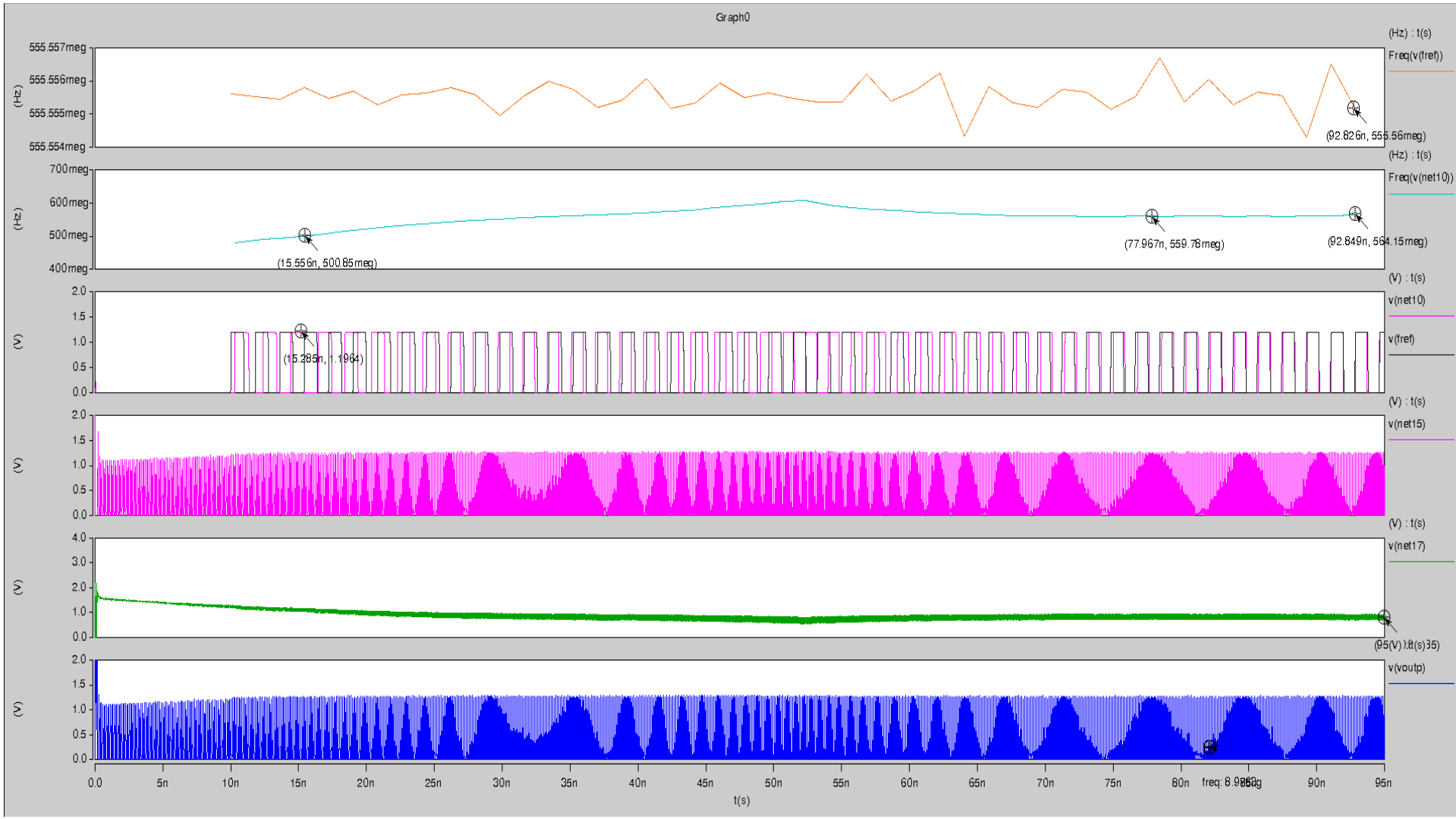


Fractional 8 Divider

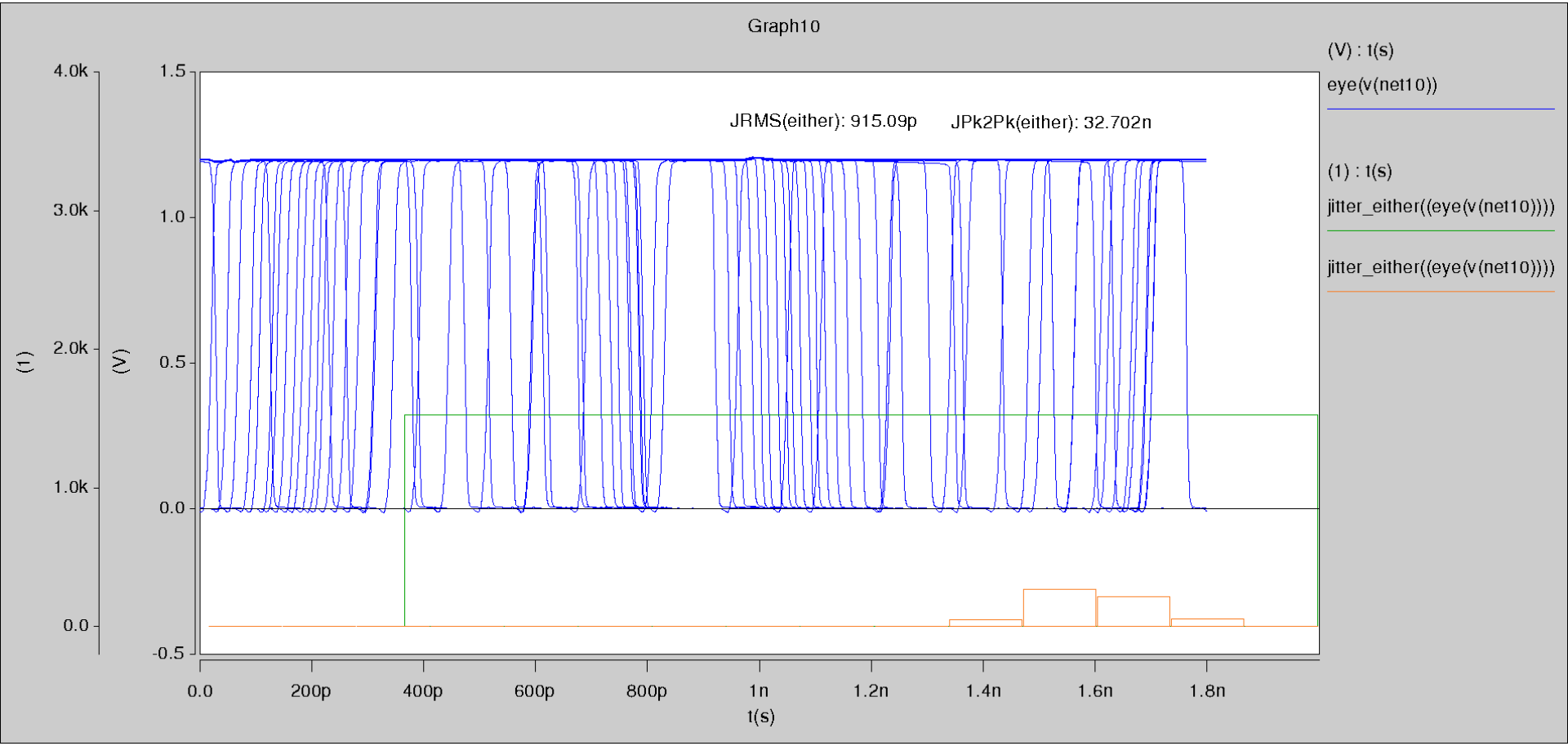
Complete PLL Circuit



Complete PLL Simulation

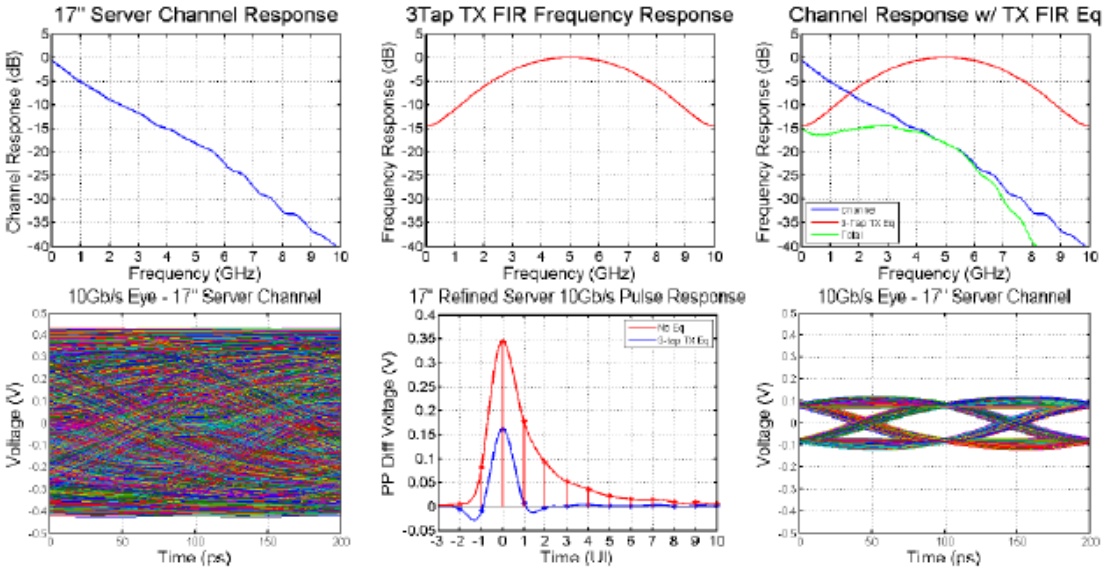


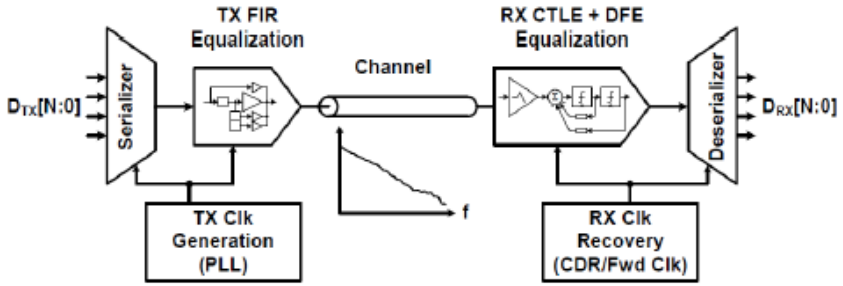
Complete PLL Jitter Analysis



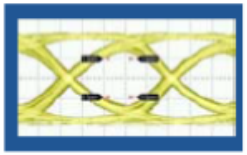
Equalization

- 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.

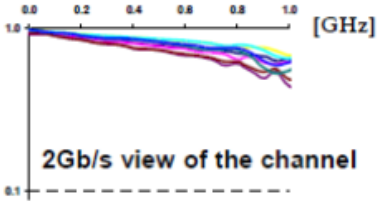


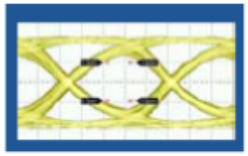


Channel

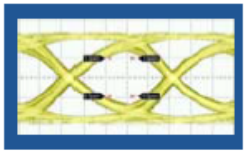


Signal at Tx

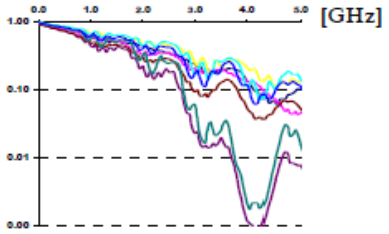


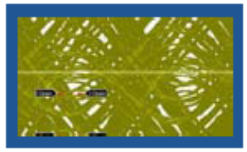


Signal at Rx

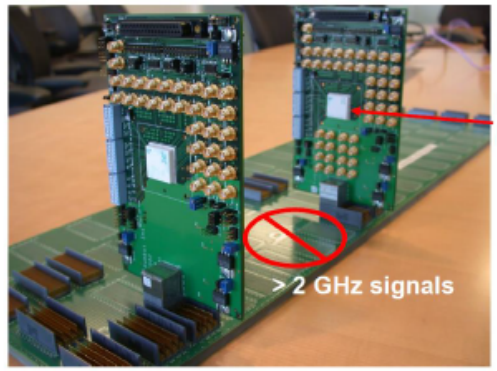


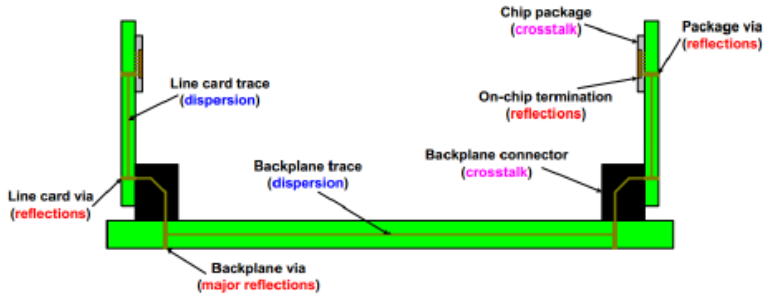
Signal at Tx

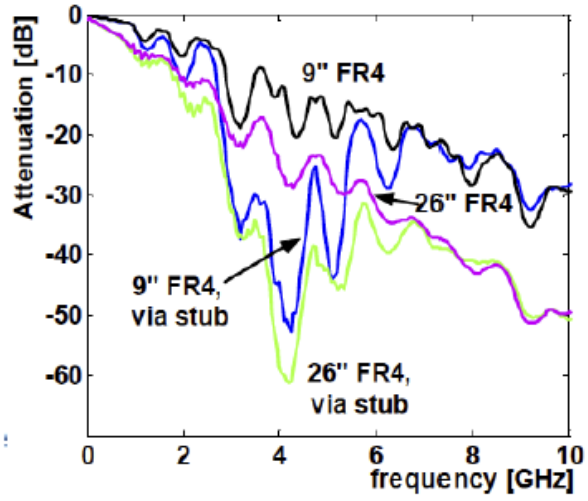




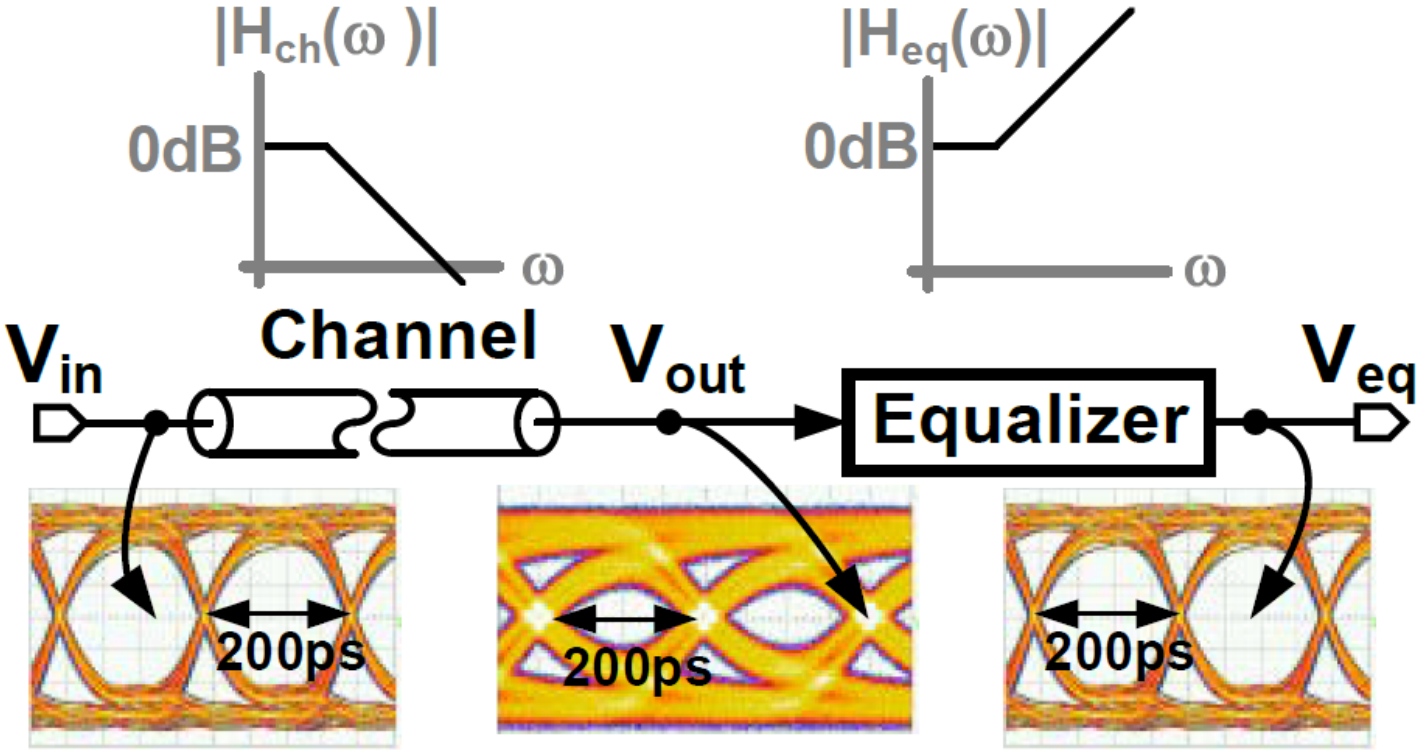
Signal at Rx







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.

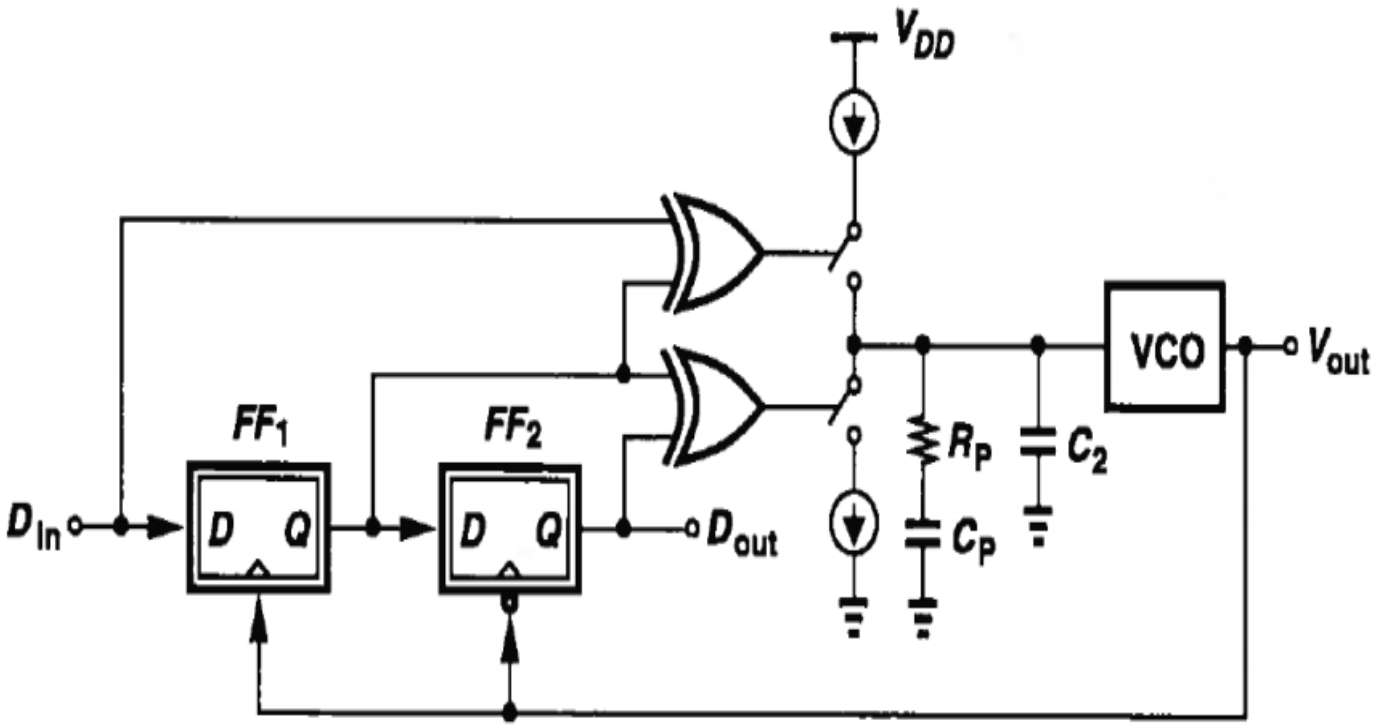
• DFE

- 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.

The CDR Circuit

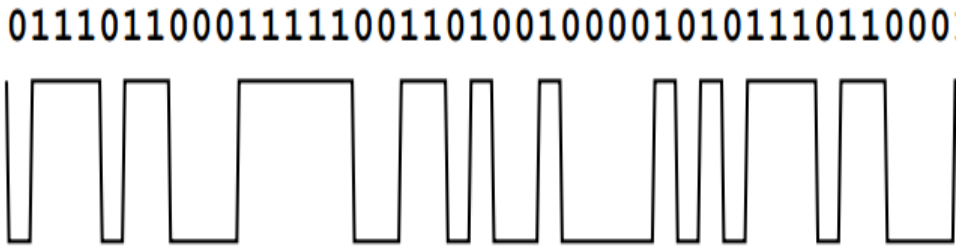


CDR Circuit Overview

- 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

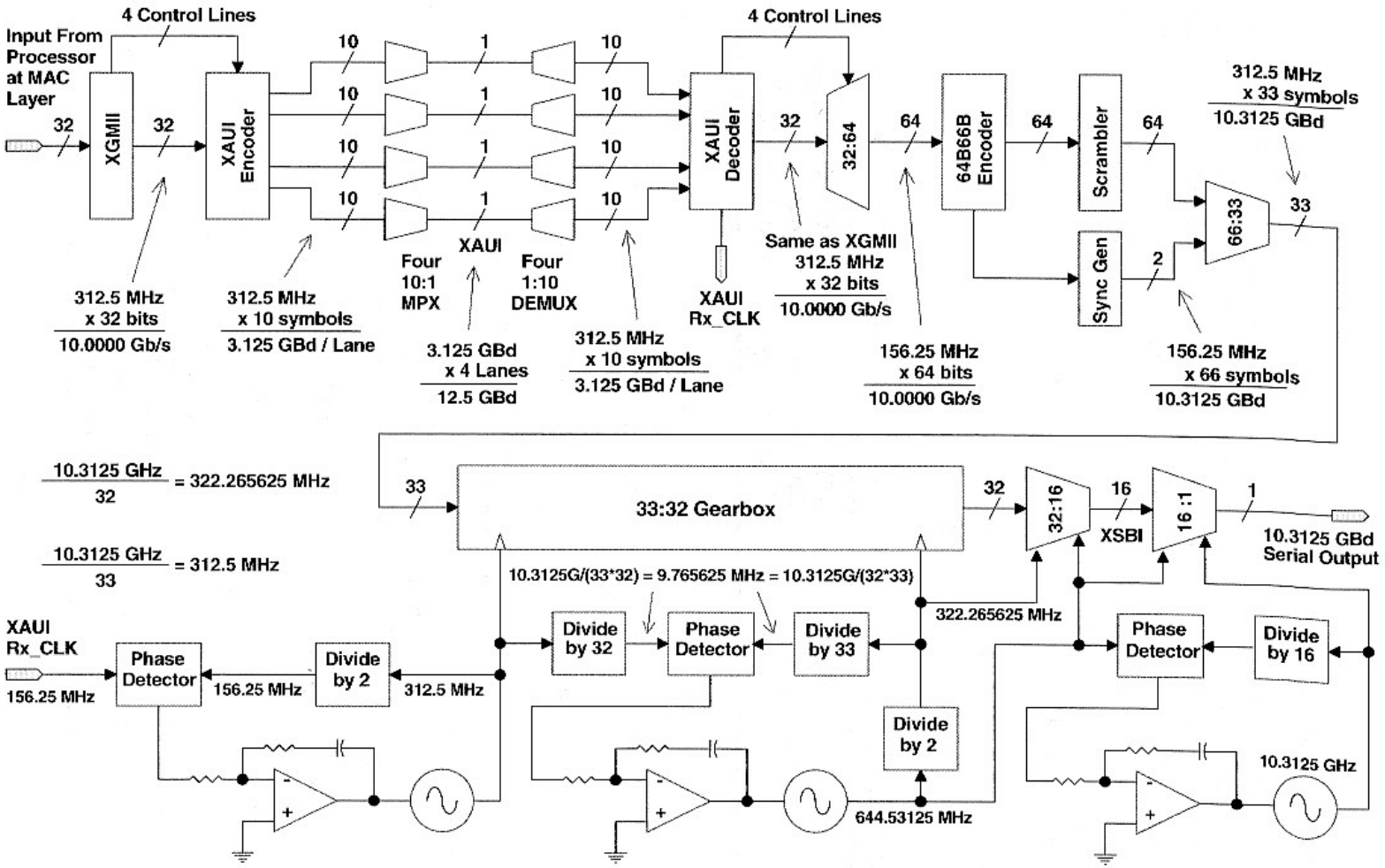
- Serial data transmission sends binary bits of information as a series of optical or electrical pulses



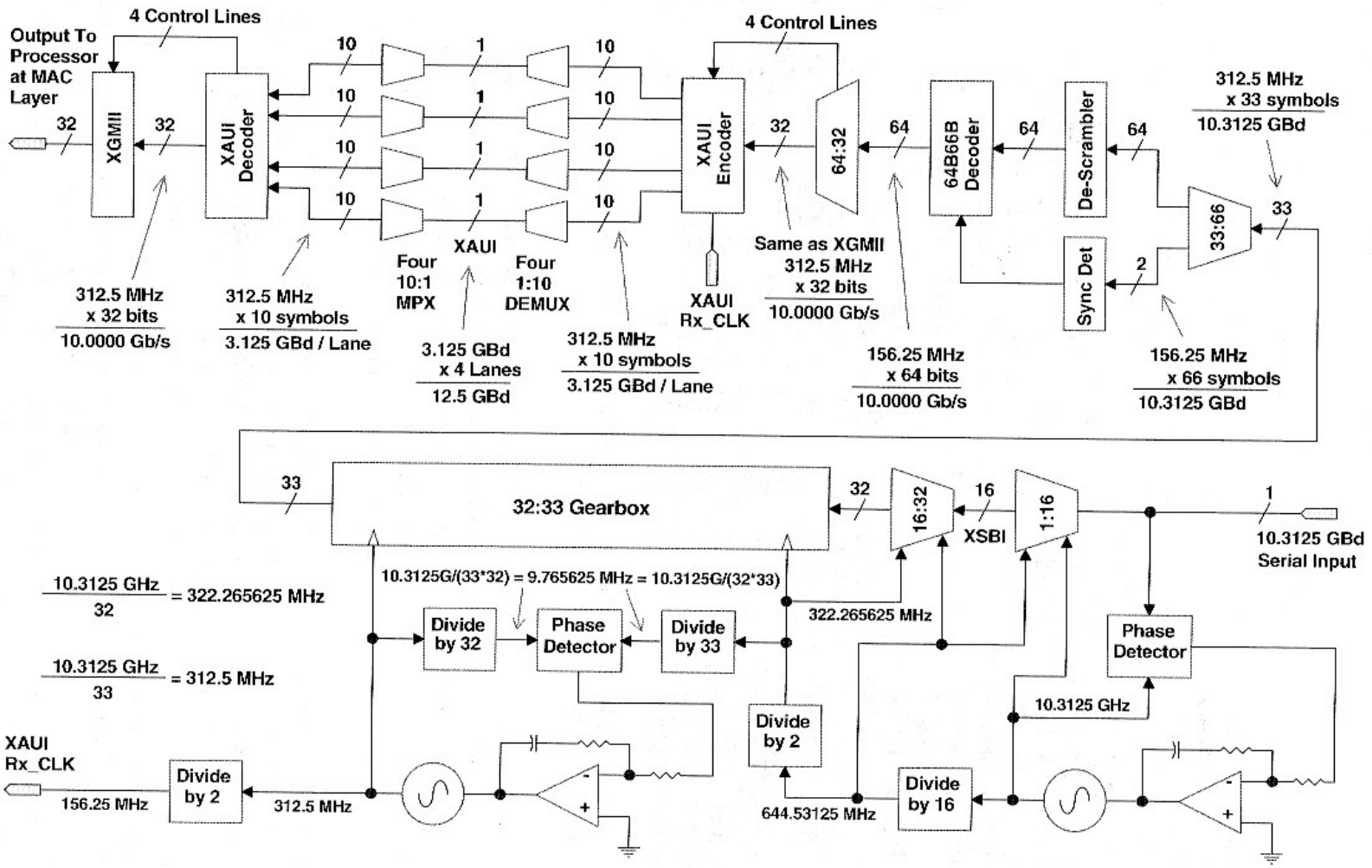
- The transmission channel (coax, radio, fiber) generally distorts the signal in various ways

- From this signal we must recover both clock and data

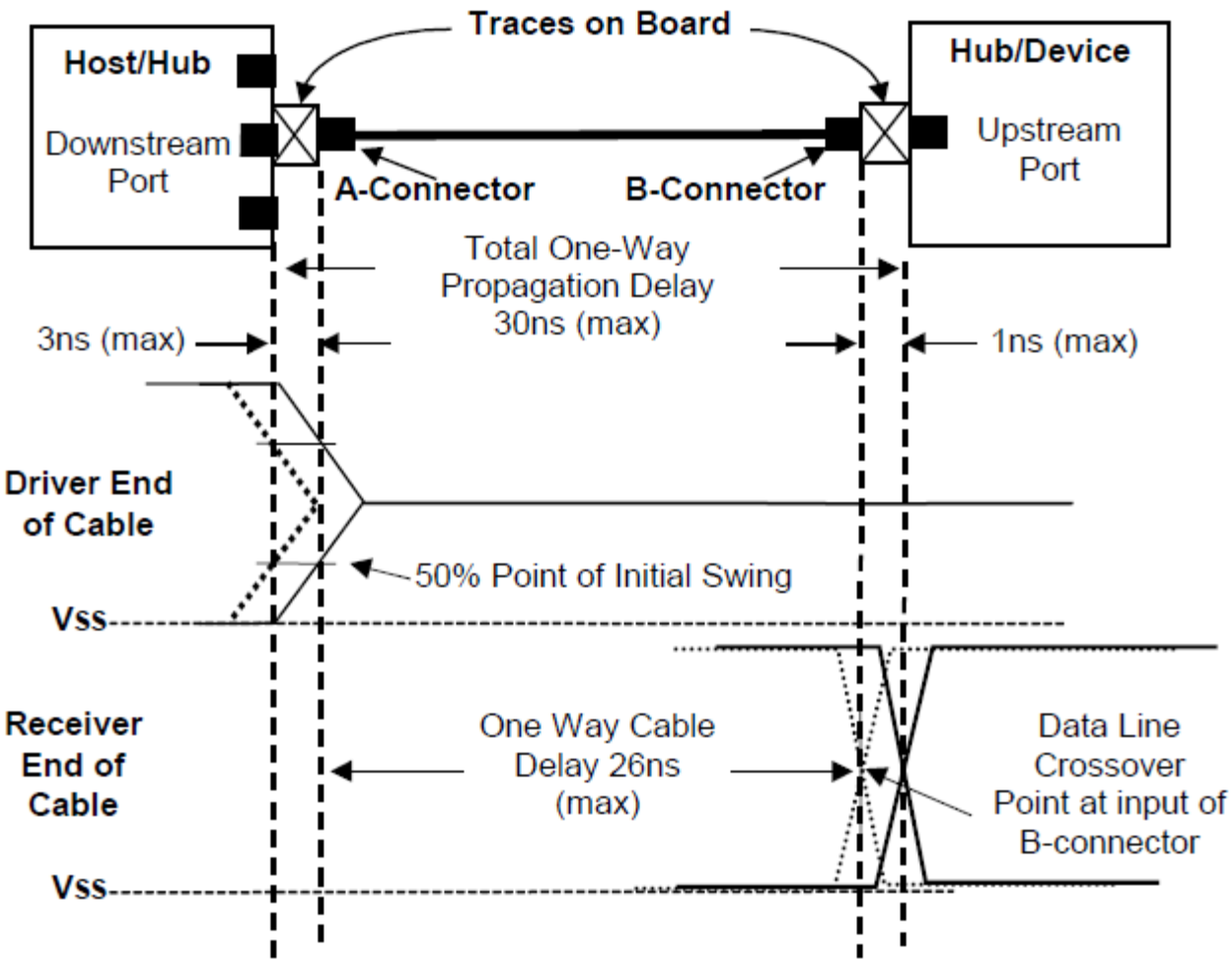
10 Gigabit Ethernet Serializer



10 Gigabit Ethernet Deserializer

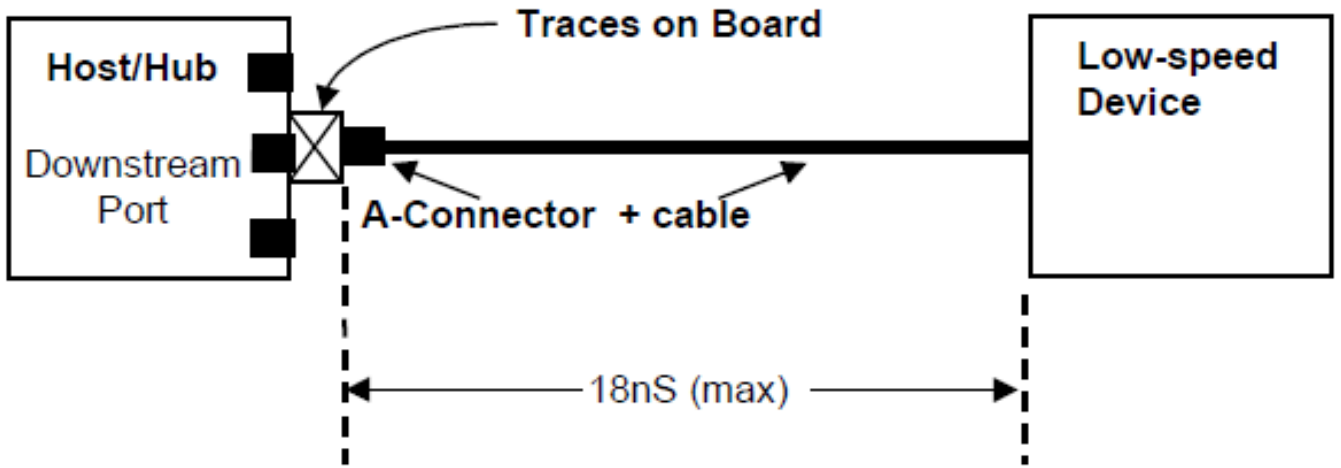


Universal Serial Bus (USB)



Full-speed Cable Delay

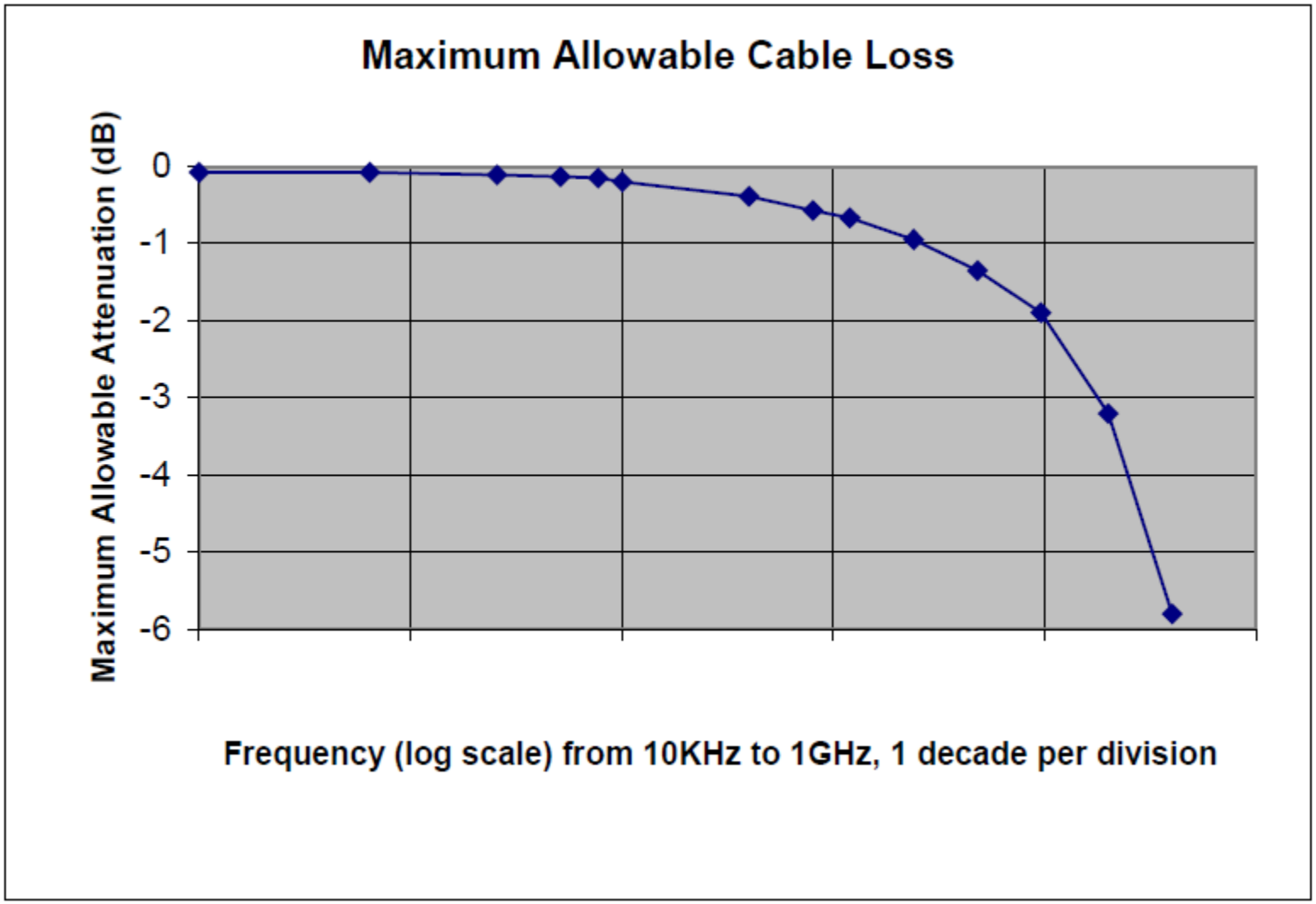
Universal Serial Bus (USB)



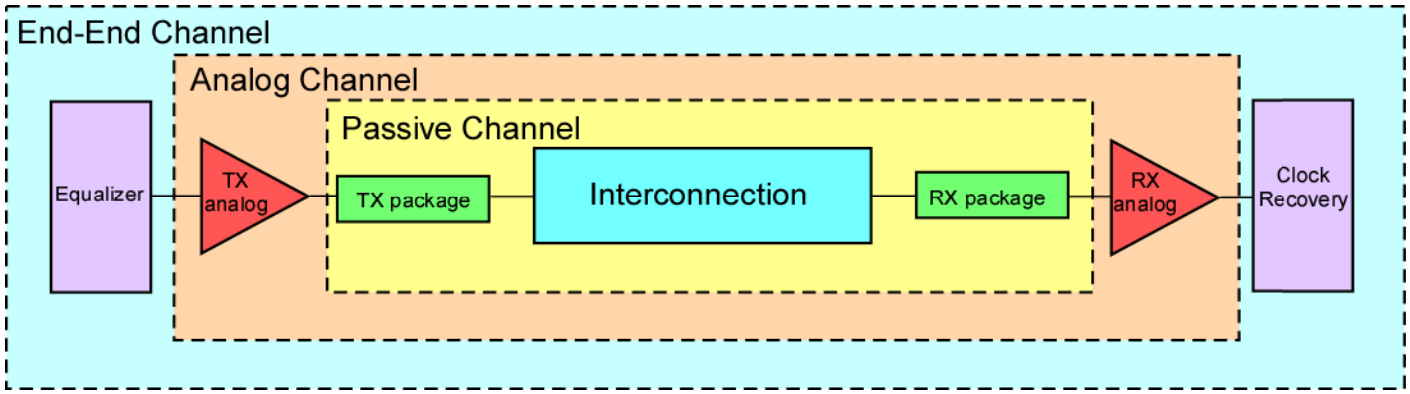
One-way Propagation Delay
Low-speed Cable Delay

- 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 Link



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

- Analog channel includes TX driver and RX termination network

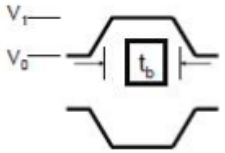
- End-to-end channel includes everything

High-Speed Serial Channels

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

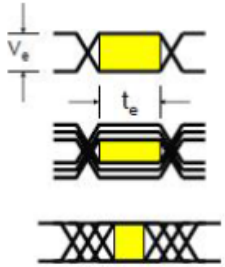
- Need to accurately handle very high data rates
- Simulate large number of bits to achieve low BER
- Non-linear blocks with time variant Systems
- TX/RX equalization and vendor specific device settings
- Coding schemes
- All types of jitter: (random, deterministic, etc.)
- Crosstalk, loss, dispersion,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

Eye-Diagrams



This is a "1"

This is a "0"

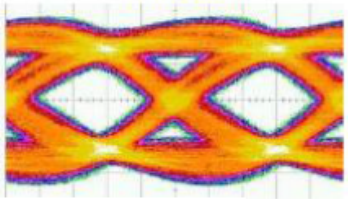


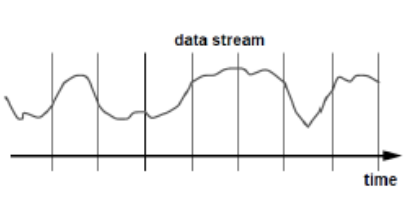
Eye Opening - space between 1 and 0

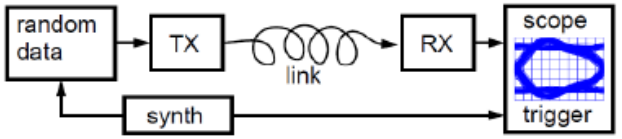
With Both!

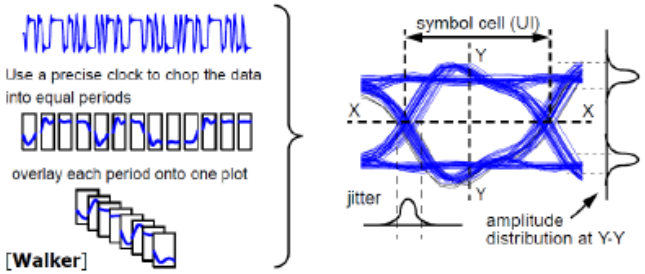
With voltage noise

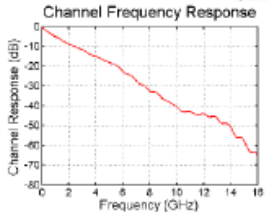
With timing noise

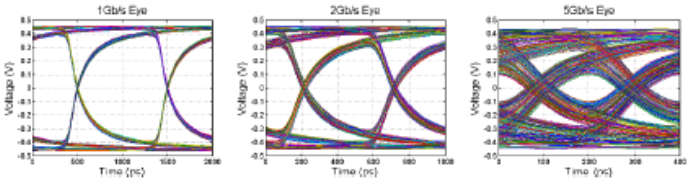


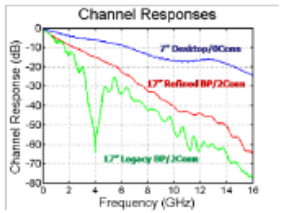


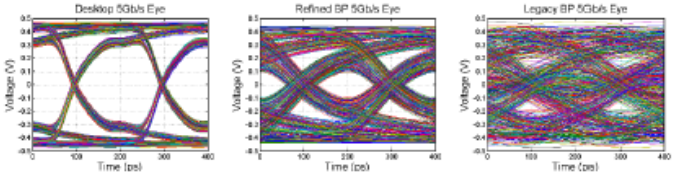












Timing Margin

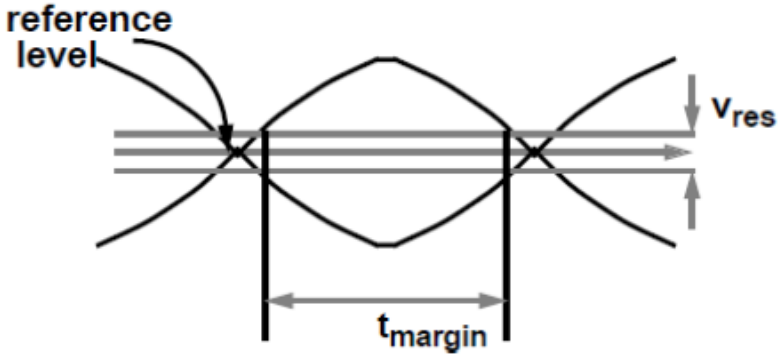
- $t_{margin} = t_{bit} - t_{os} - t_{jd} - t_{jc}$

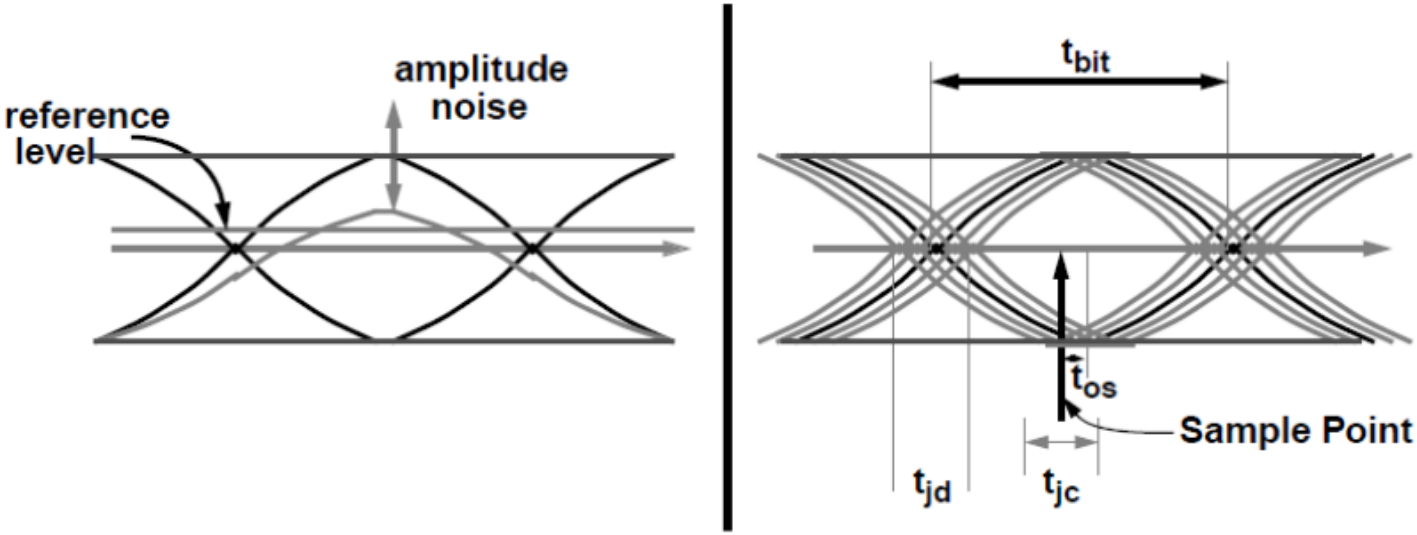
  Where,

  t_{bit} = bit-width of a symbol

  t_{os} = sampling error

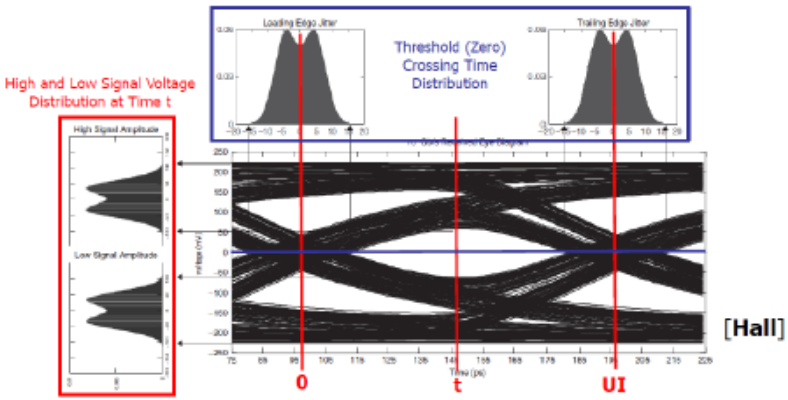
  t_{jd}/t_{jc} = data/clock jitter



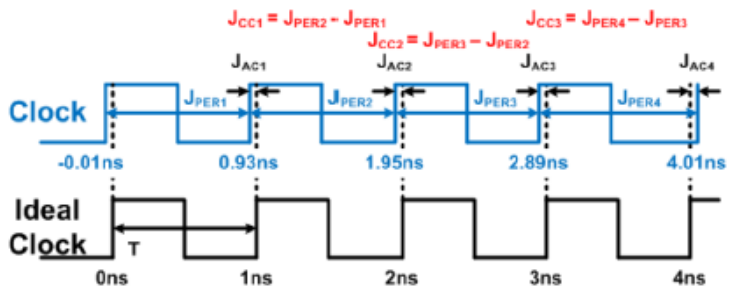


Timing Jitter

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

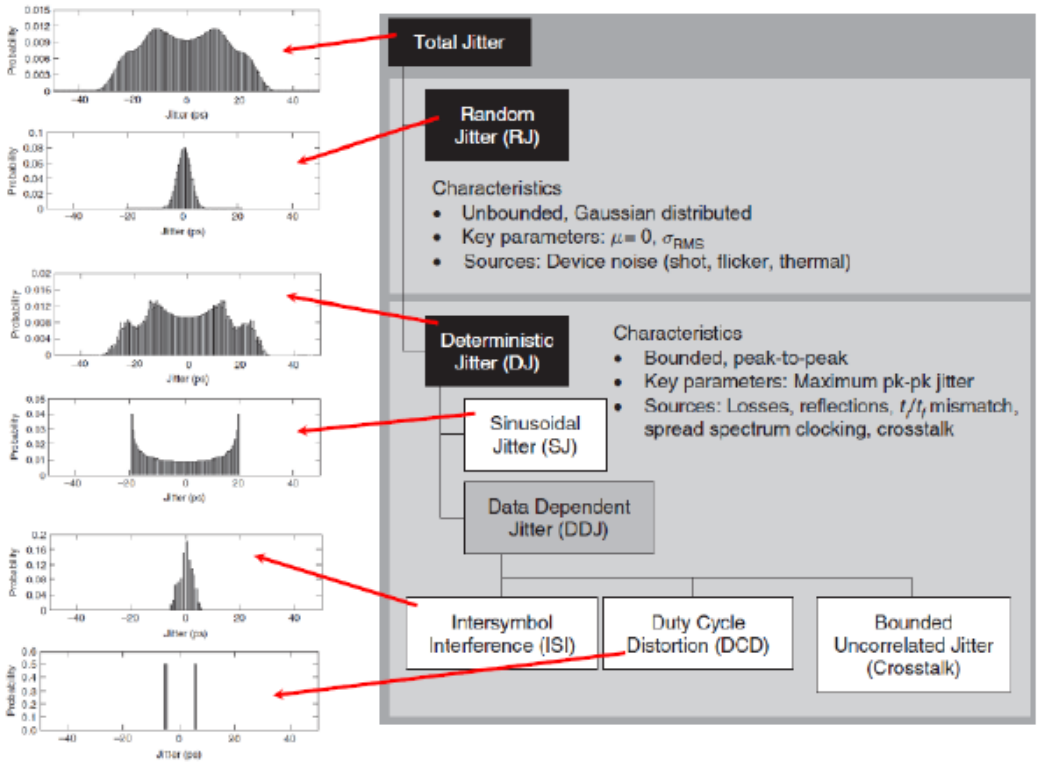


[Hall]



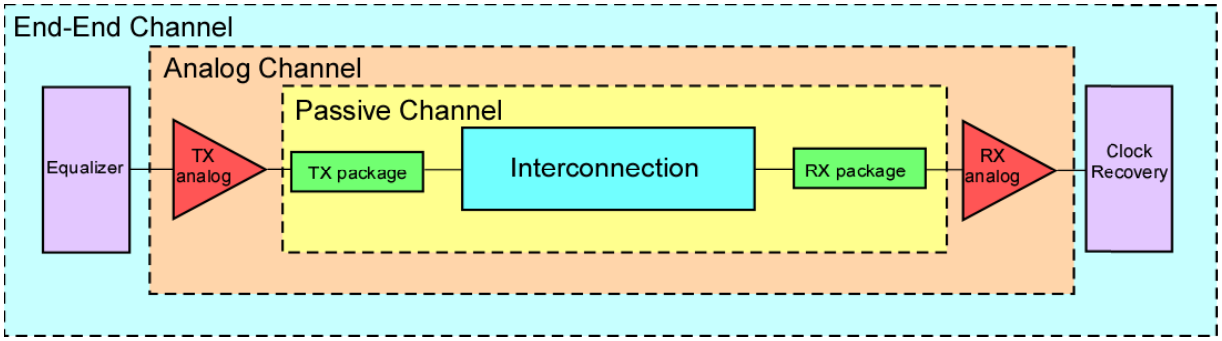
n	1	2	3	4	Mean	RMS	PP
J_{PER}	-0.06	0.02	-0.06	0.12	0.005	0.085	0.18
J_{CC}	0.08	-0.08	0.18	-	0.06	0.131	0.26
J_{AC}	-0.07	-0.05	-0.11	0.01	-0.055	0.05	0.12

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



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



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

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 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