# ECEN689: Special Topics in High-Speed Links Circuits and Systems Spring 2010

## Lecture 31: CDR Architectures


Sam Palermo

Analog & Mixed-Signal Center

Texas A&M University

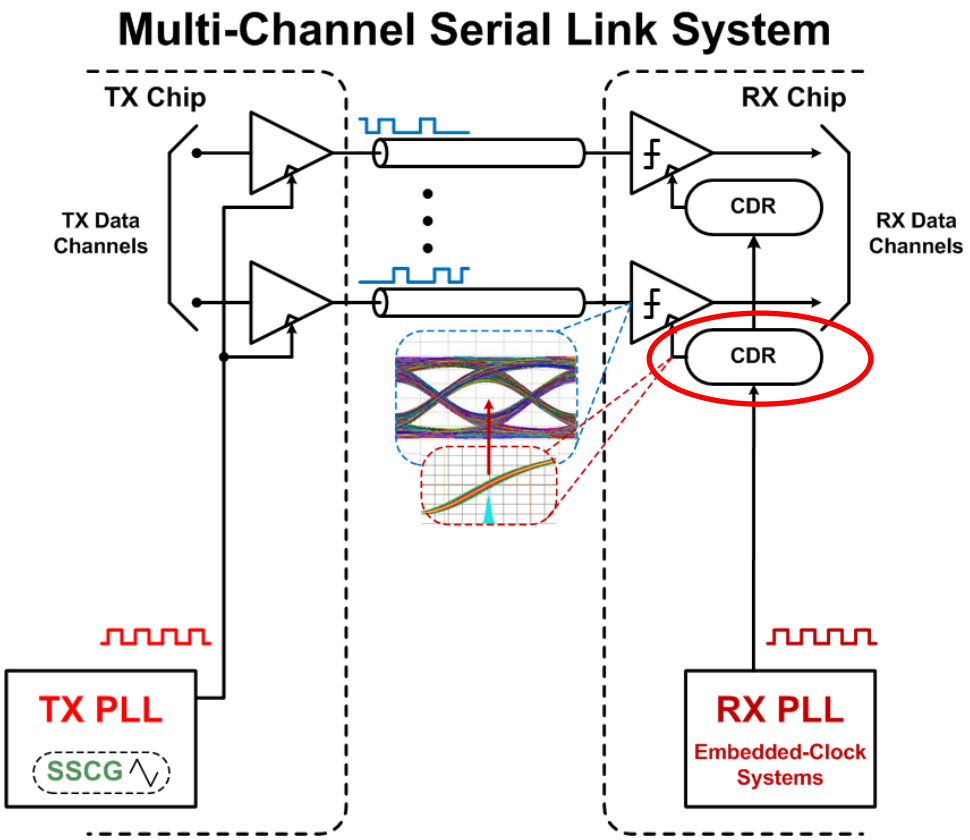
# Announcements

- Project Preliminary Report #2 due now
  - Feedback meetings on Friday 10:30-12
- Exam 2 is April 30
- Final Project Report Due May 4

# Agenda

- Analog & digital CDRs
- Analog dual-loop CDRs
- Digital dual-loop CDRs
- Phase Interpolators
- Delay-Locked Loops

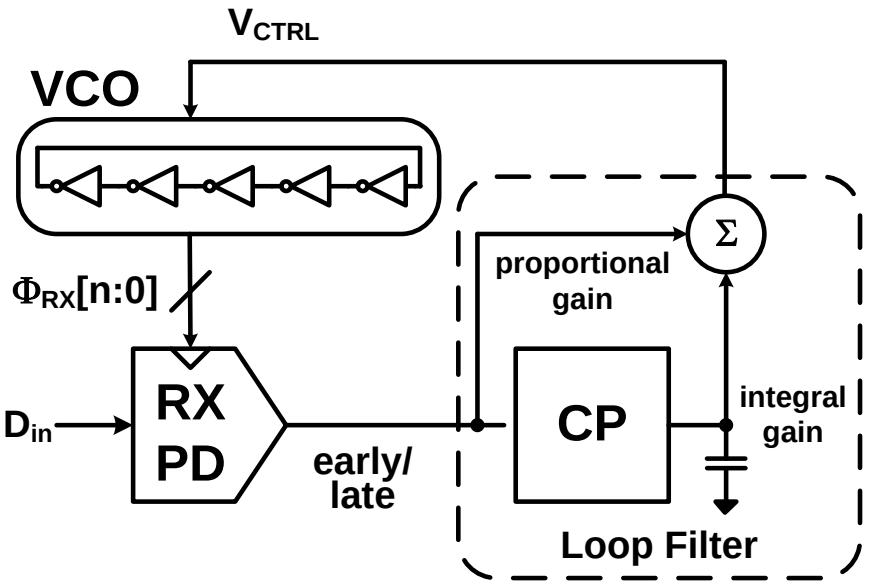
# Embedded Clock I/O Circuits



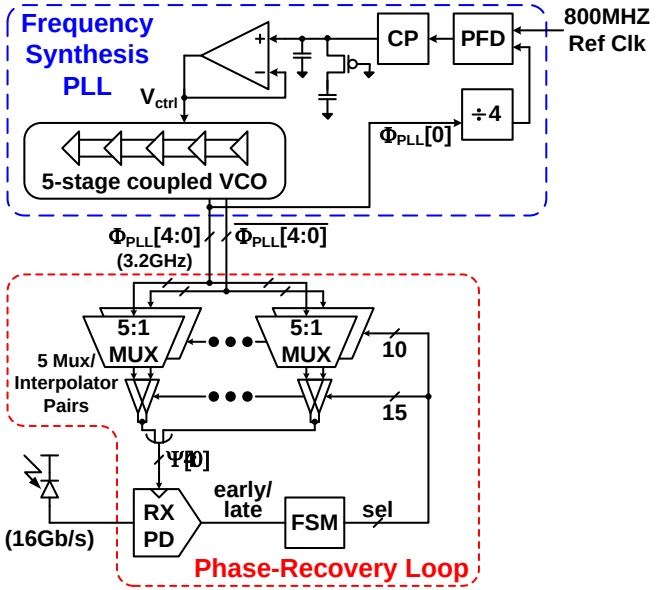

- TX PLL
- TX Clock Distribution
- CDR
  - Per-channel PLL-based
  - Dual-loop w/ Global PLL &
    - Local DLL/PI
    - Local Phase-Rotator PLLs
    - Global PLL requires RX clock distribution to individual channels

# Embedded Clocking (CDR)

**PLL-based CDR**

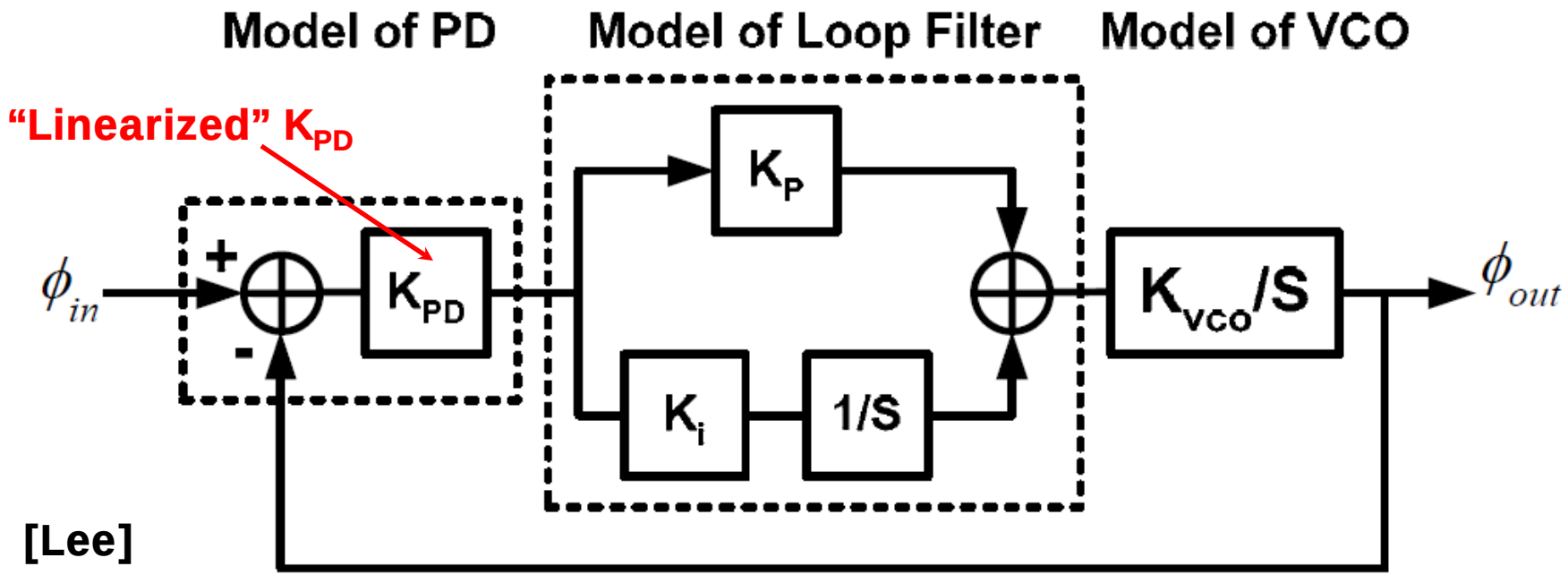



**Dual-Loop CDR**




- Clock frequency and optimum phase position are extracted from incoming data
- Phase detection continuously running
- Jitter tracking limited by CDR bandwidth
  - With technology scaling we can make CDRs with higher bandwidths and the jitter tracking advantages of source synchronous systems is diminished
- Possible CDR implementations
  - Stand-alone PLL
  - “Dual-loop” architecture with a PLL or DLL and phase interpolators (PI)
  - Phase-rotator PLL

# Analog PLL-based CDR

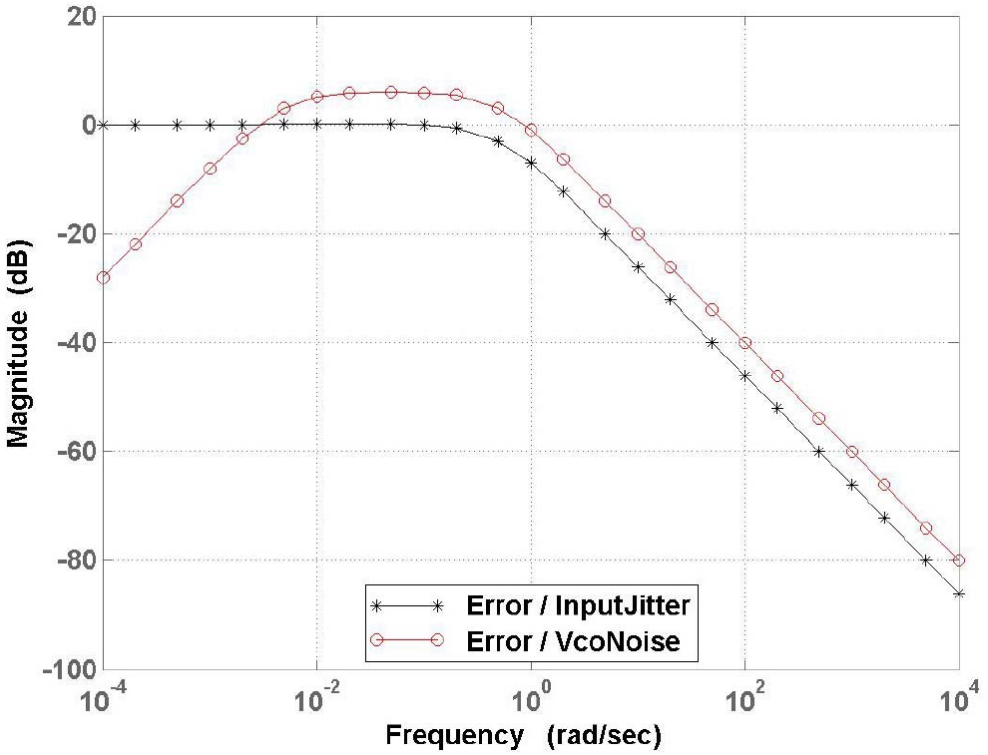



[Lee]

$$\frac{\phi_{out}}{\phi_{in}} = \frac{s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}{s^2 + s \cdot K_P \cdot K_{PD} \cdot K_{VCO} + K_i \cdot K_{PD} \cdot K_{VCO}}$$

$$K_P = I_C \cdot R \qquad K_i = \frac{I_C}{C} \qquad \omega_n = \sqrt{K_i \cdot K_{PD} \cdot K_{VCO}} \qquad \zeta = \frac{K_P}{K_i} \cdot \frac{\omega_n}{2}$$

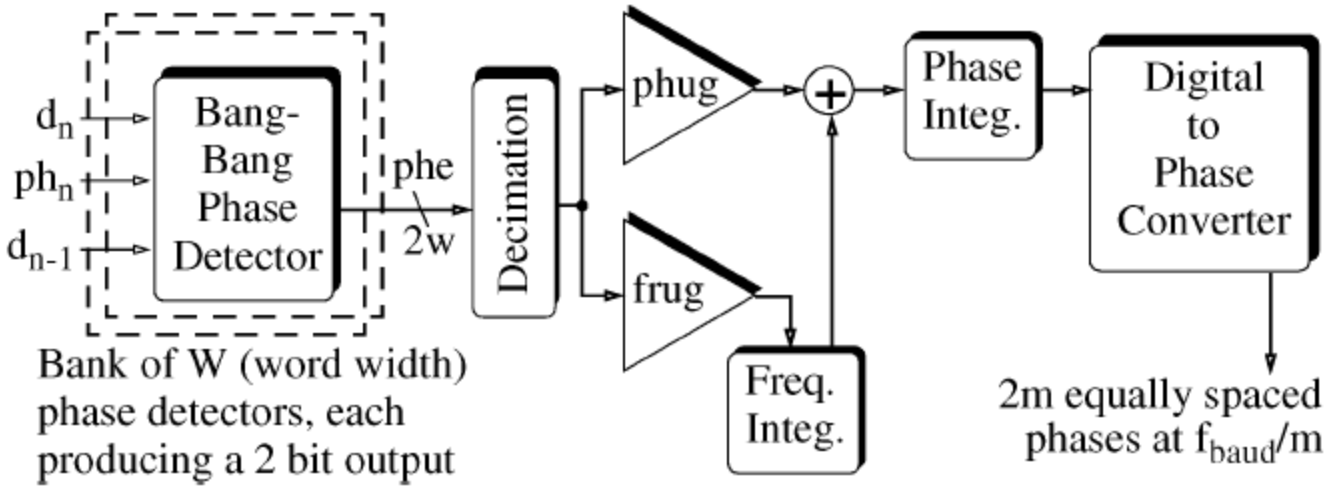
# Analog PLL-based CDR

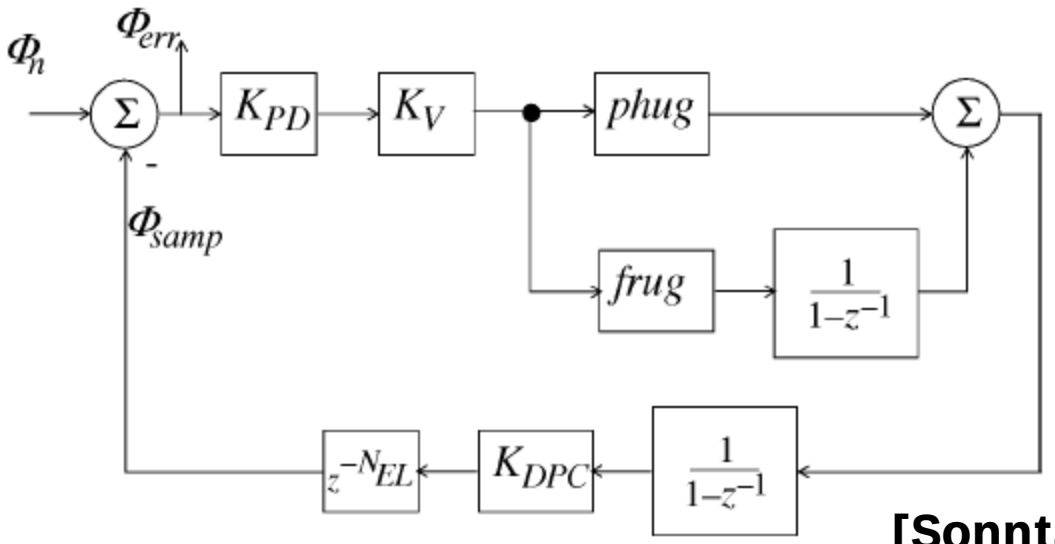



**[Lee]**

- CDR “bandwidth” will vary with input phase variation amplitude with a non-linear phase detector
- Final performance verification should be done with a time-domain non-linear model

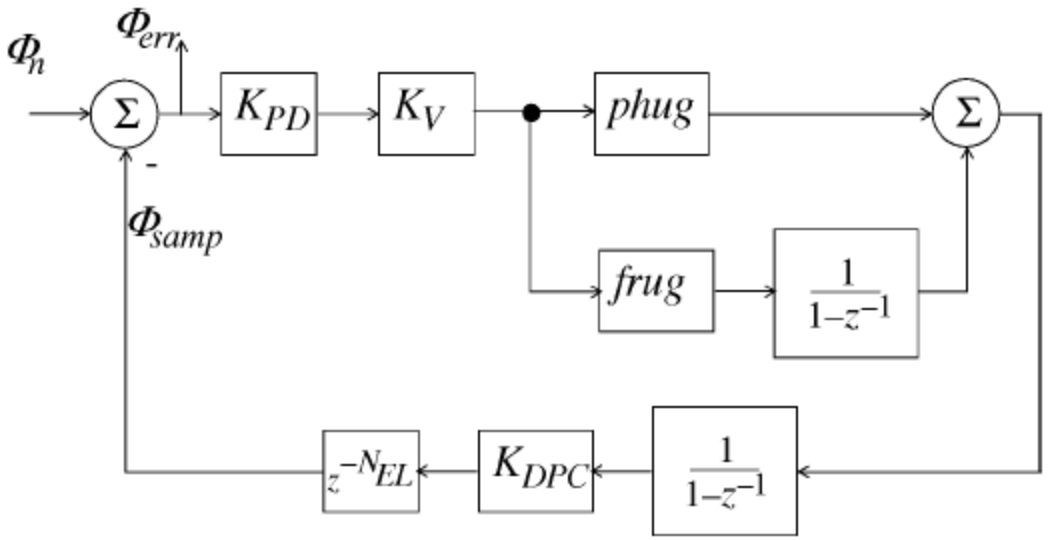
# Digital PLL-based CDR




**[Sonntag JSSC 2006]**

# Digital PLL-based CDR



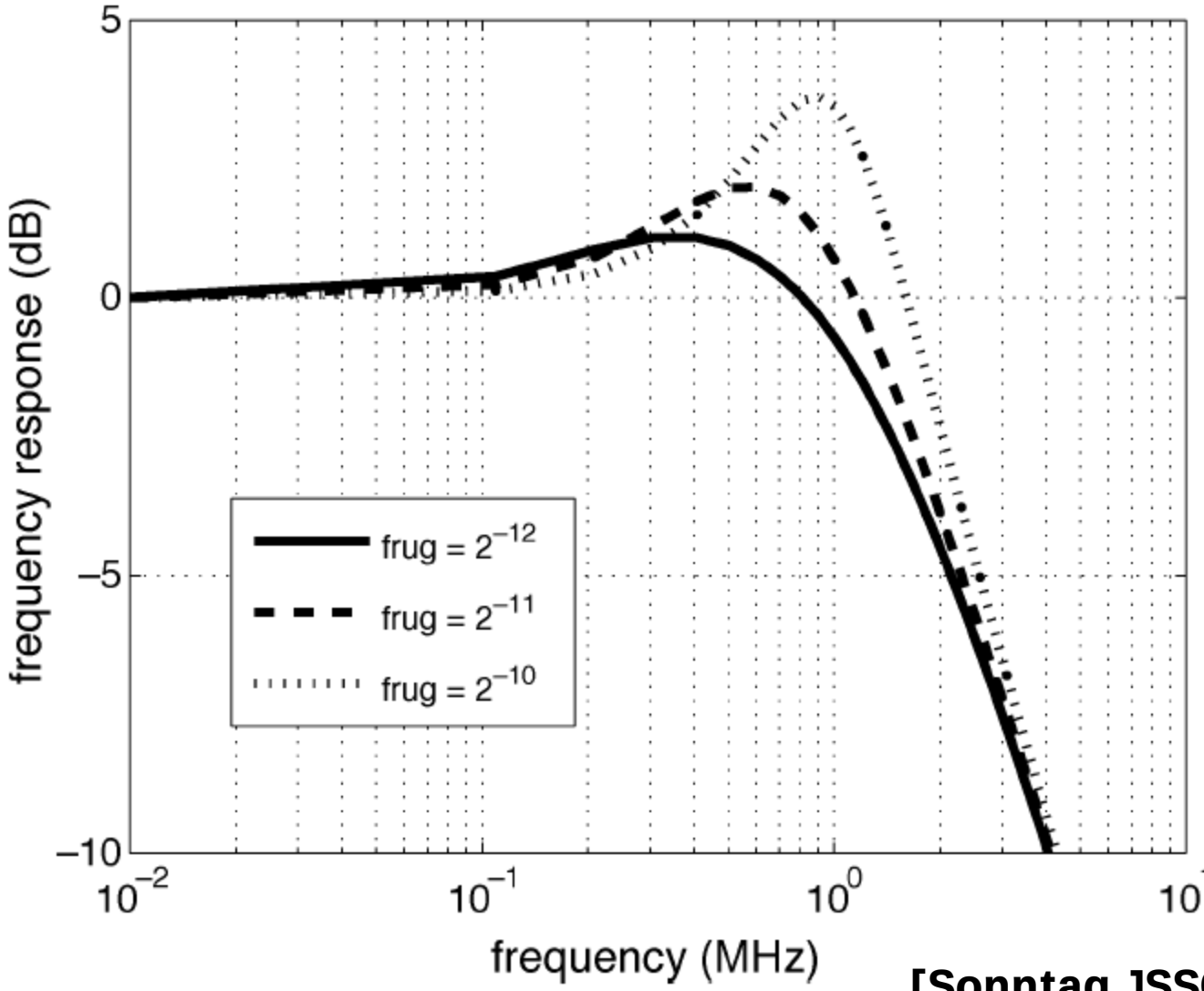

**Open-Loop Gain:**

$$L(z^{-1}) = \left(\frac{K_{\mathrm{PD}} K_{\mathrm{V}} K_{\mathrm{DPC}}}{1 - z^{-1}}\right)\left(phug + \frac{frug}{(1 - z^{-1})}\right) z^{-N_{\mathrm{EL}}}.$$

$$\Phi_{\mathrm{samp}}/\Phi_{\mathrm{in}} = \left(L(e^{-j\omega})\right) / \left(1 + L(e^{-j\omega})\right)$$

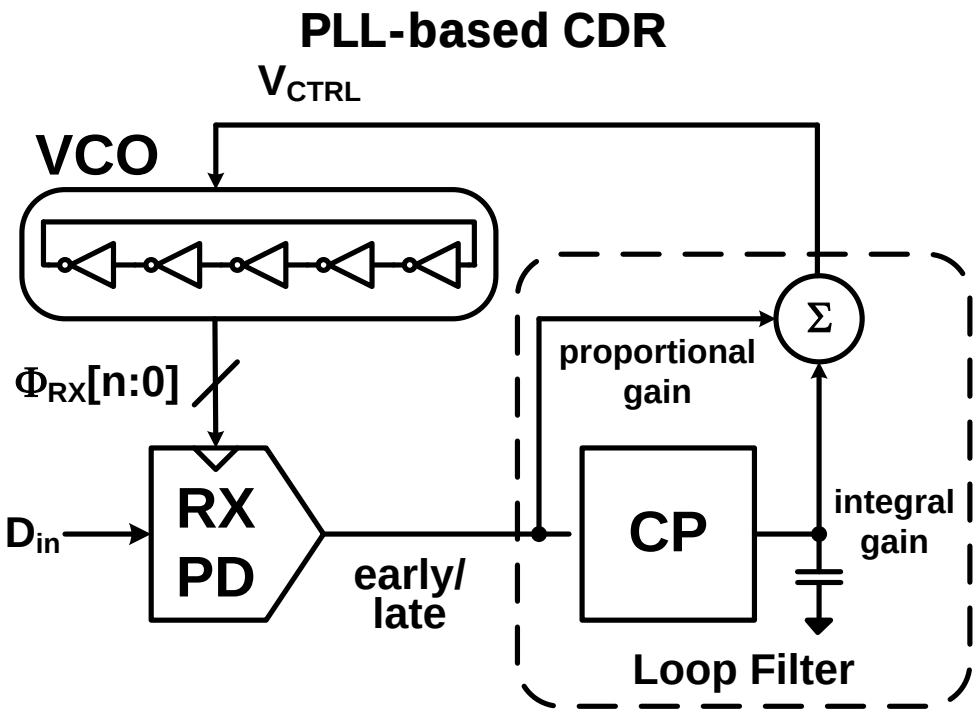
**[Sonntag JSSC 2006]**

# Digital PLL-based CDR



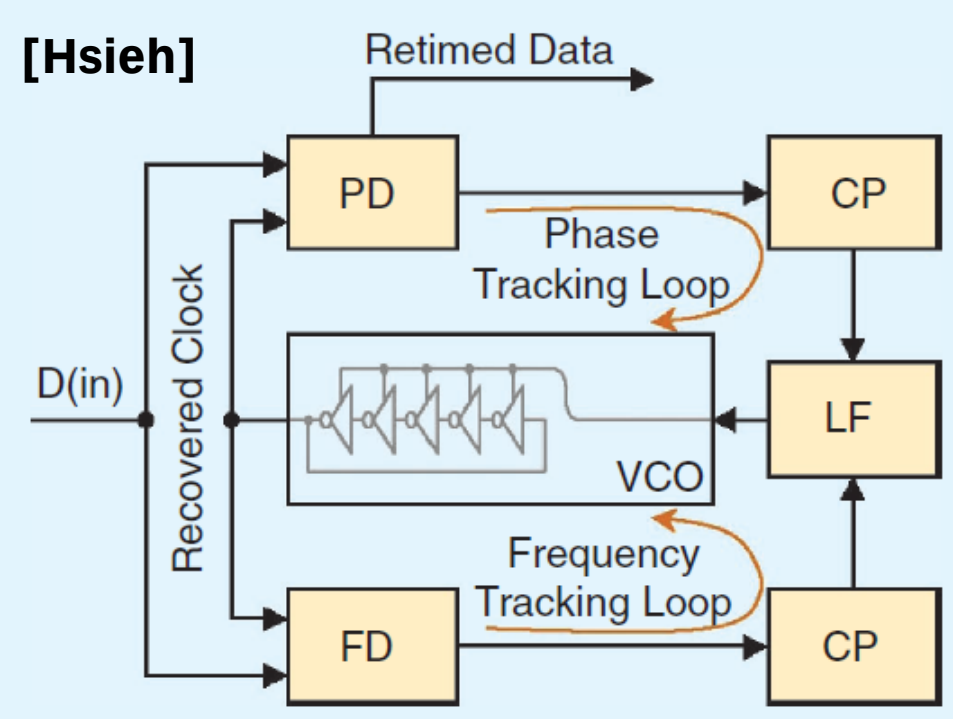

**[Sonntag JSSC 2006]**

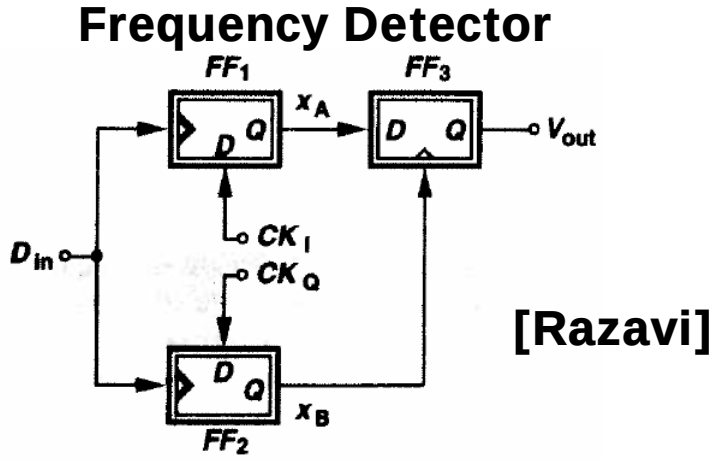
# Single-Loop CDR Issues

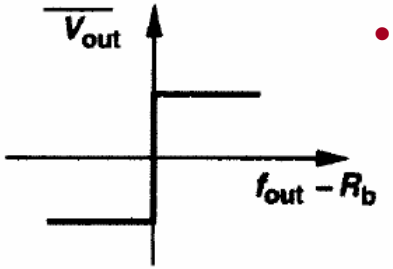



- Phase detectors have limited frequency acquisition range
  - Results in long lock times or not locking at all
  - Can potentially lock to harmonics of correct clock frequency
- VCO frequency range varies with voltage and temperature

# Phase and Frequency Tracking Loops



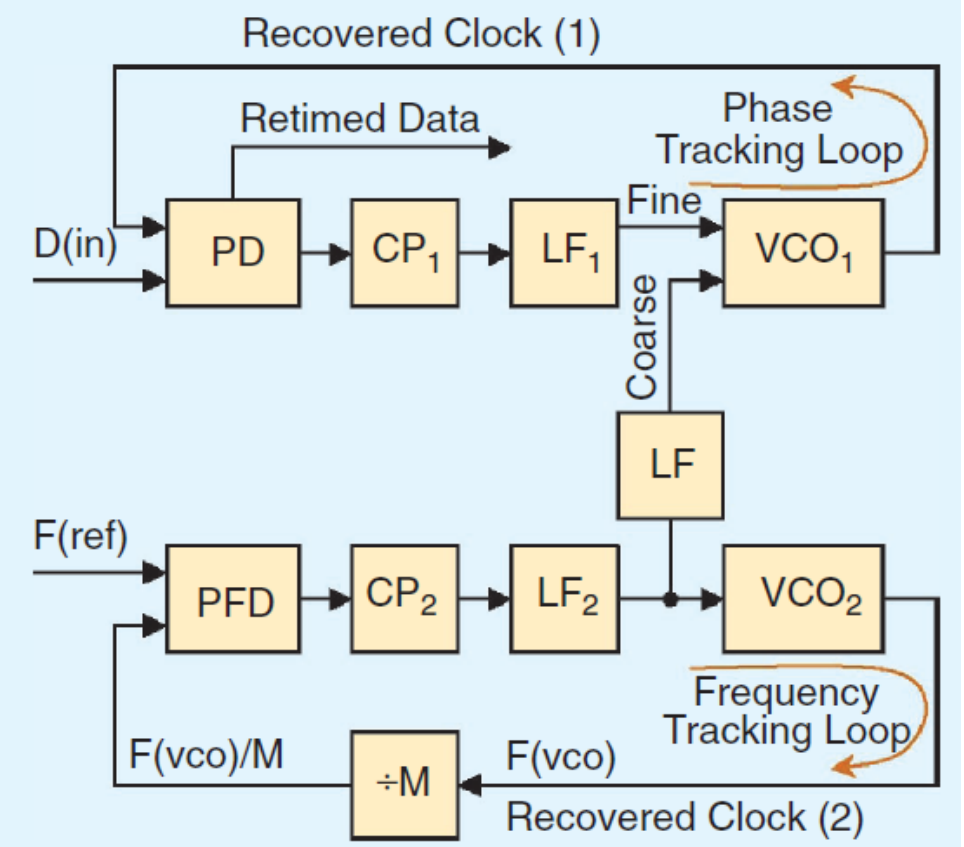




- Capture range ~<15% frequency offset

- Frequency tracking loop operates during startup or loss of phase lock
  - Ideally should be mostly off in normal operation
- Frequency loop bandwidth typically much smaller than phase loop bandwidth to prevent loop interaction

# Analog Dual-Loop CDR w/ Two VCOs

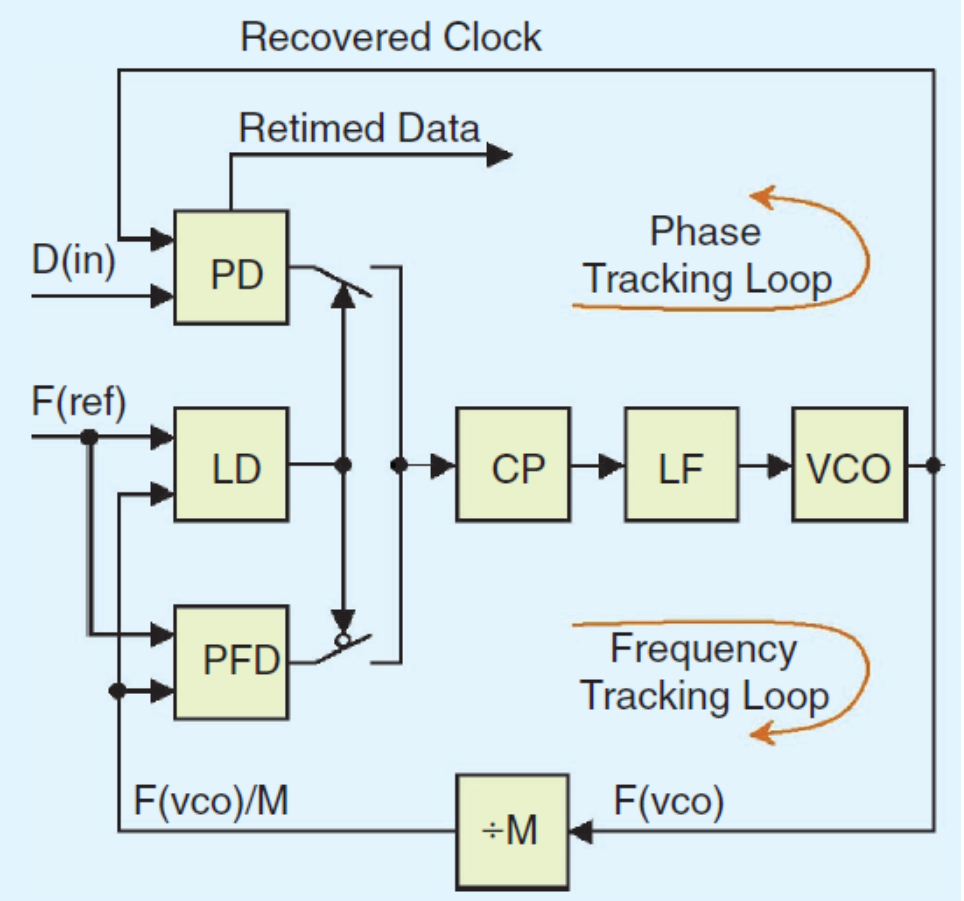
- Frequency synthesis loop with replica VCO provides a “coarse” control voltage to set phase tracking loop frequency
- Frequency loop can be a global PLL shared by multiple channels
- Issues
  - VCO matching
  - VCO pulling
  - Distributing voltage long distances




**[Hsieh]**

# Analog Dual-Loop CDR w/ One VCO

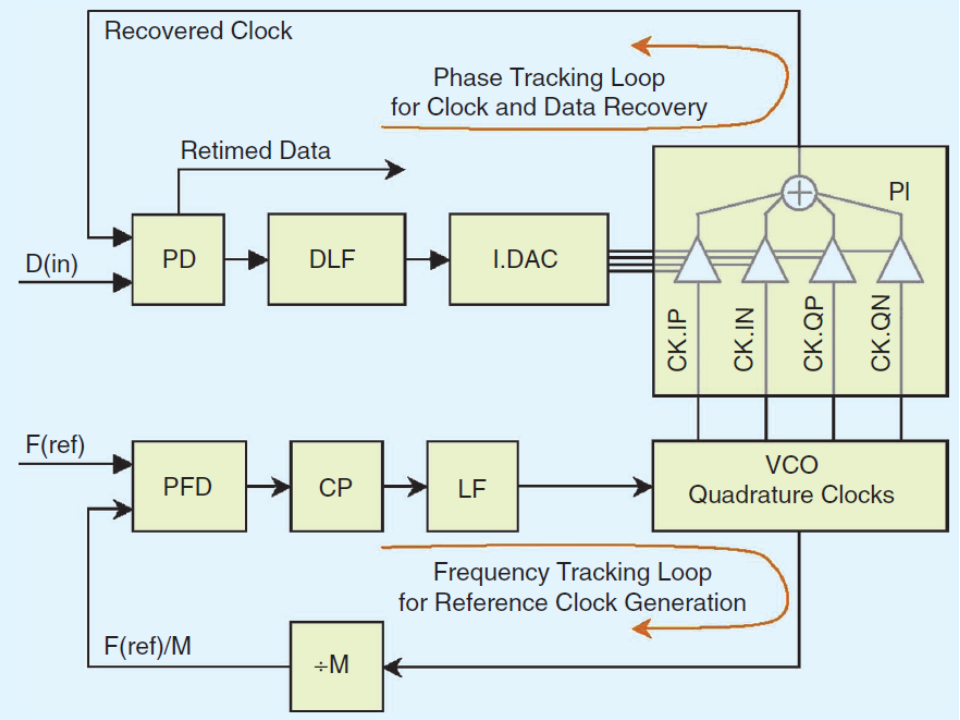
- Frequency loop operates during startup or loss of phase lock
  - Ideally should be mostly off in normal operation
- Input reference clock simplifies frequency loop design
- Care must be taken when switching between loops to avoid disturbing VCO control voltage and loose frequency lock




**[Hsieh]**

# Phase Interpolator (PI) Based CDR

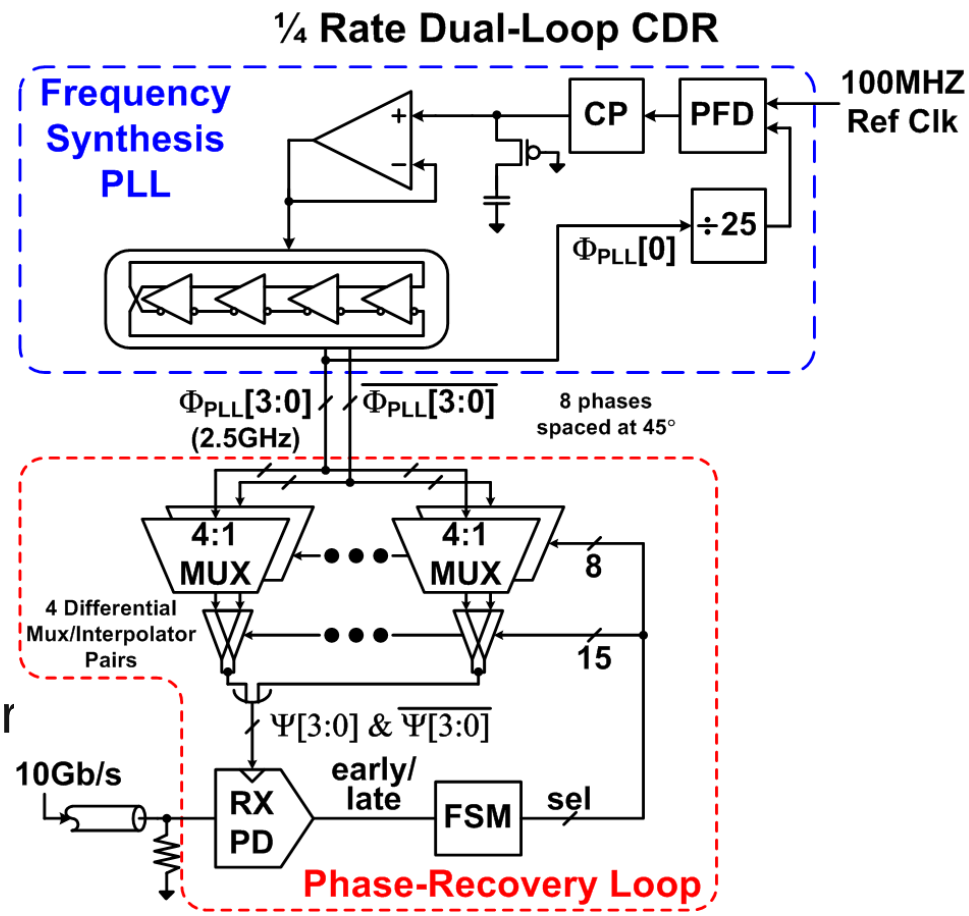
- Frequency synthesis loop produces multiple clock phases used by the phase interpolators
- Phase interpolator mixes between input phases to produce a fine sampling phase
  - Ex: Quadrature 90° PI inputs with 5 bit resolution provides sampling phases spaced by $90°/(2^5-1)=2.9°$
- Digital phase tracking loop offers advantages in robustness, area, and flexibility to easily reprogram loop parameters




**[Hsieh]**

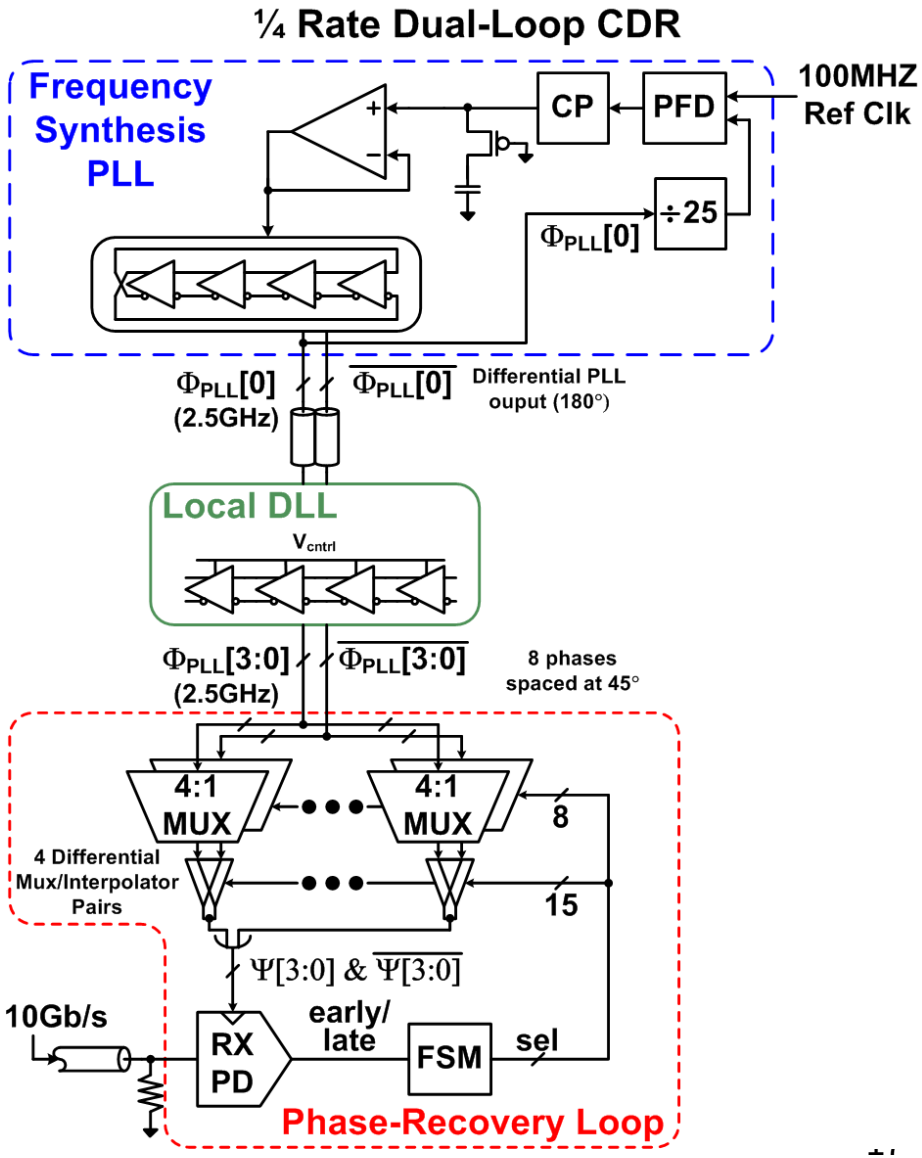
# Phase Interpolator (PI) Based CDR

- Frequency synthesis loop can be a global PLL
- Can be difficult to distribute multiple phases long distance
  - Need to preserve phase spacing
  - Clock distribution power increases with phase number
  - If CDR needs more than 4 phases consider local phase generation



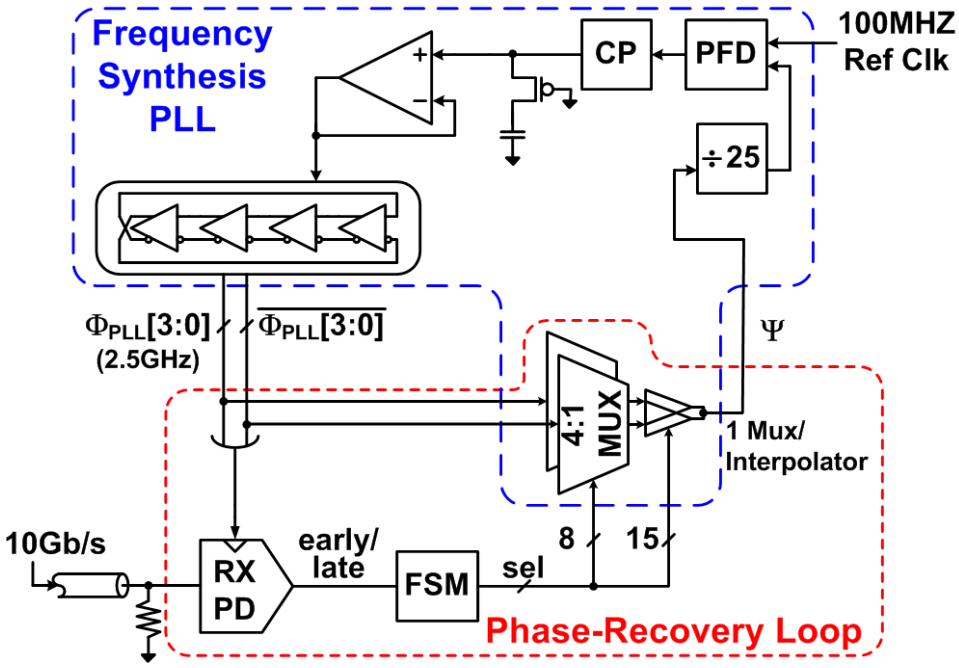
# DLL Local Phase Generation

- Only differential clock is distributed from global PLL
- Delay-Locked Loop (DLL) locally generates the multiple clock phases for the phase interpolators
  - DLL can be per-channel or shared by a small number (4)
- Same architecture can be used in a forwarded-clock system
  - Replace frequency synthesis PLL with forwarded-clock signals



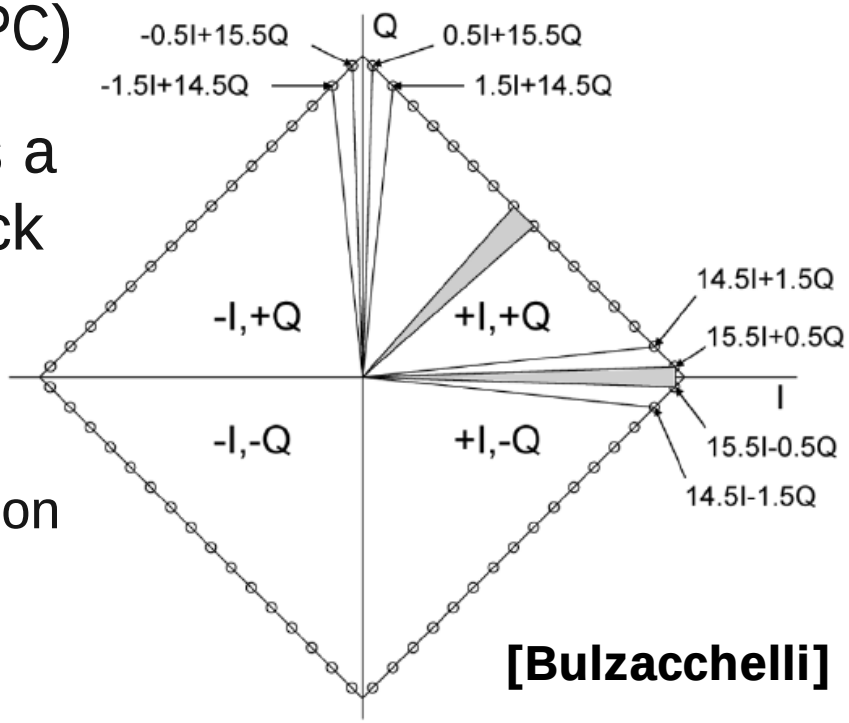
# Phase Rotator PLL

- Phase interpolators can be expensive in terms of power and area
- Phase rotator PLL places **one** interpolator in PLL feedback to adjust all VCO output phases simultaneously
- Now frequency synthesis and phase recovery loops are coupled
  - Need PLL bandwidth greater than phase loop
    - Useful in filtering VCO noise



# Phase Interpolators

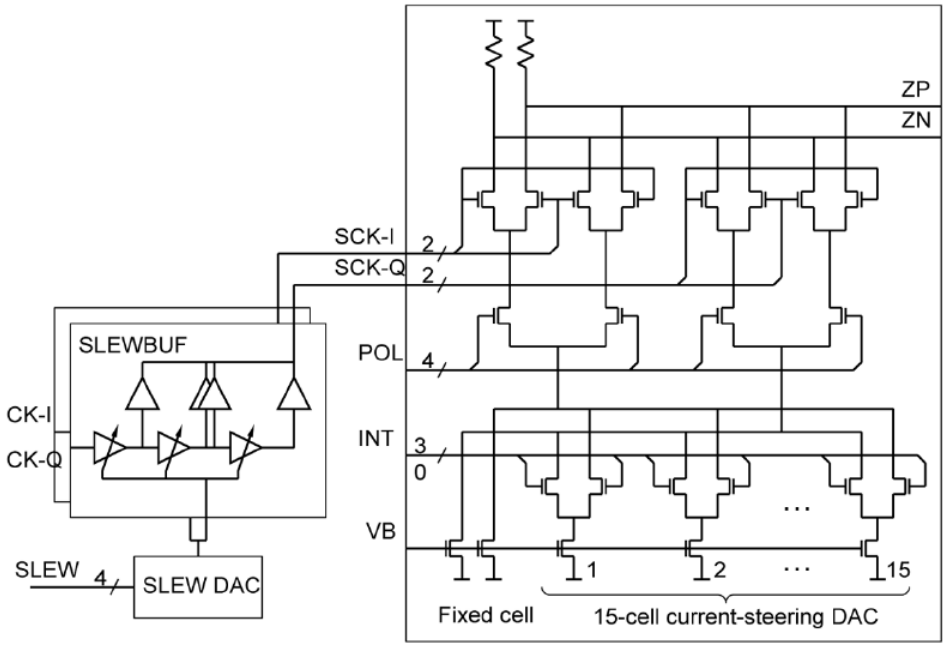
- Phase interpolators realize digital-to-phase conversion (DPC)
- Produce an output clock that is a weighted sum of two input clock phases
- Common circuit structures
  - Tail current summation interpolation
  - Voltage-mode interpolation
- Interpolator code mapping techniques
  - Sinusoidal
  - Linear




**[Bulzacchelli]**

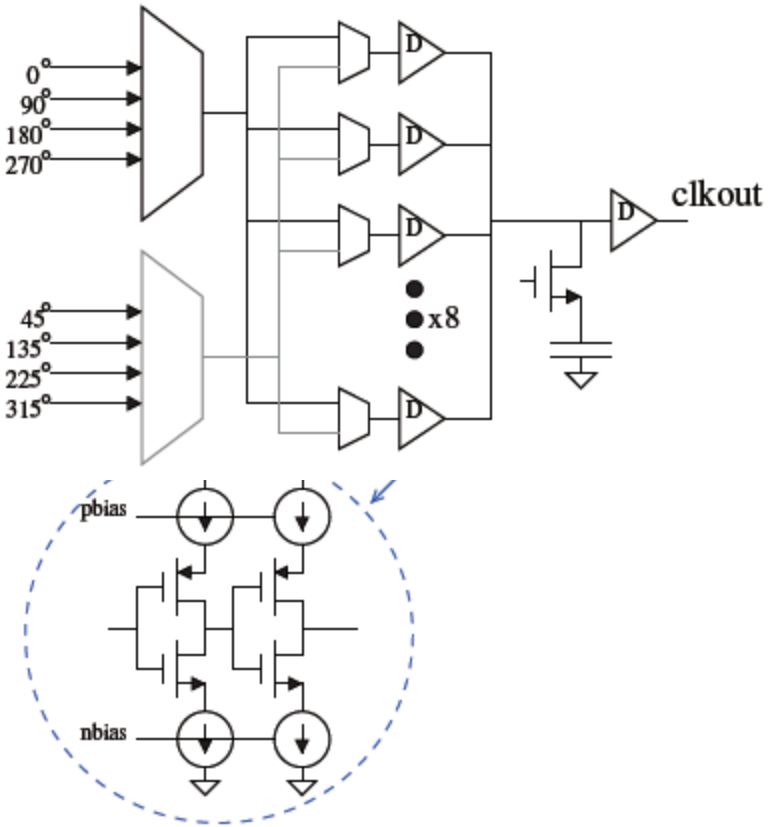
# Phase Interpolator Examples

**Tail-Current Summation**



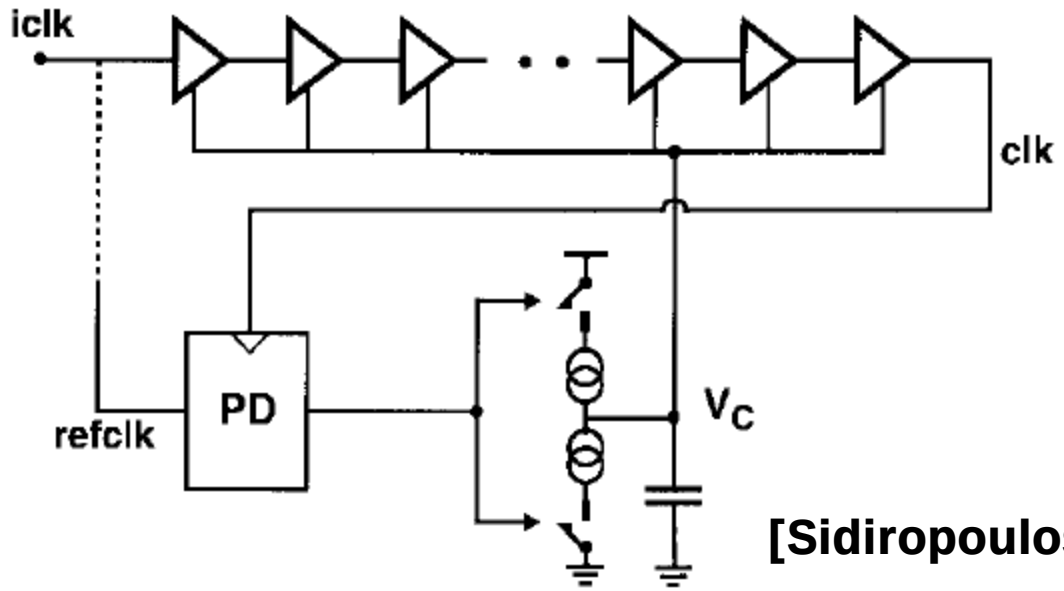

**[Bulzacchelli JSSC 2006]**

**Voltage-Mode Summation**




**[Joshi VLSI Symp 2009]**

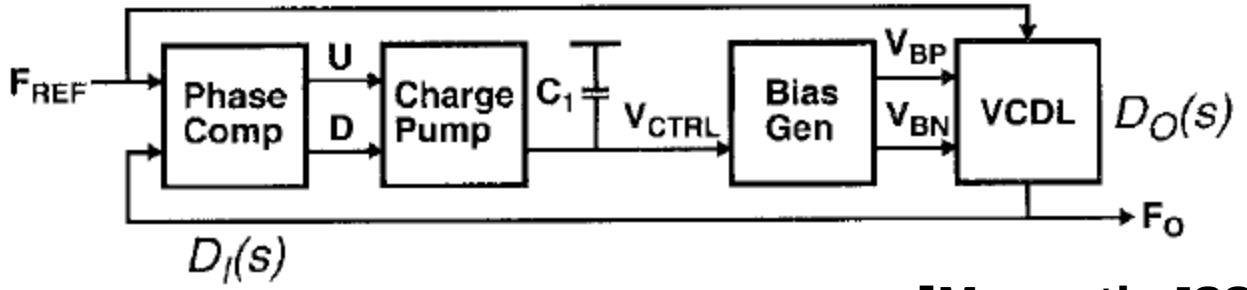
# Delay-Locked Loop (DLL)




**[Sidiropoulos JSSC 1997]**

- DLLs lock delay of a voltage-controlled delay line (VCDL)
- Typically lock the delay to 1 or ½ input clock cycles
  - If locking to ½ clock cycle the DLL is sensitive to clock duty cycle
- DLL does not self-generate the output clock, only delays the input clock

# Delay-Locked Loop (DLL)




**[Maneatis JSSC 1996]**

$$\frac{D_O(s)}{D_I(s)} = \frac{1}{1 + s/\omega_N}$$

$$\omega_N = I_{CH} \cdot K_{DL} \cdot F_{\mathrm{REF}} \cdot \frac{1}{C_1}$$

- First-order loop as delay line doesn't introduce a pole
- VCDL doesn't accumulate jitter like a VCO
- DLL doesn't filter input jitter

# Next Time

- CDR Wrap-Up
  - PI
  - DLL
  - Jitter Properties
- Injection-Locked Oscillator De-Skew