ECEN689: Special Topics in High-Speed Links Circuits and Systems
Spring 2012

Lecture 11: Clocking Architectures & PLLs

Sam Palermo
Analog & Mixed-Signal Center
Texas A&M University
Agenda

• Clocking Architectures

• PLLs
  • PLL modeling
  • PLL noise transfer functions
  • PLL circuits
  • PLL design procedure
References

- High-speed link clocking tutorial paper, PLL analysis paper, and PLL thesis posted on website
- Posted PLL models in project section
- Website has additional links on PLL and jitter tutorials
- Majority of today’s PLL material comes from Fischette tutorial and M. Mansuri’s PhD thesis (UCLA)
High-Speed Electrical Link System
Clocking Terminology

Synchronous
- Every chip gets same frequency AND phase
- Used in low-speed busses

Mesochronous
- Same frequency, but unknown phase
- Requires phase recovery circuitry
  - Can do with or without full CDR
- Used in fast memories, internal system interfaces, MAC/Packet interfaces

Plesiochronous
- Almost the same frequency, resulting in slowly drifting phase
- Requires CDR
- Widely used in high-speed links

Asynchronous
- No clocks at all
- Request/acknowledge handshake procedure
- Used in embedded systems, Unix, Linux
I/O Clocking Architectures

- Three basic I/O architectures
  - Common Clock (Synchronous)
  - Forward Clock (Source Synchronous)
  - Embedded Clock (Clock Recovery)

- These I/O architectures are used for varying applications that require different levels of I/O bandwidth

- A processor may have one or all of these I/O types

- Often the same circuitry can be used to emulate different I/O schemes for design reuse
Common Clock I/O Architecture

- Common in original computer systems
- Synchronous system by design (no active deskew)
- Common bus clock controls chip-to-chip transfers
- Requires equal length routes to chips to minimize clock skew
- Data rates typically limited to ~100Mb/s
Common Clock I/O Cycle Time

Cycle time to meet setup time

$$\max(T_{clk-A} + T_{Aclk} + T_{drive} + T_{tof} + T_{receive} + T_{setup}) - \min(T_{Bclk} - T_{clk-B}) < T_{cycle}$$

Diagram of clock signals between Chip A and Chip B.
Common Clock I/O Limitations

• Difficult to control clock skew and propagation delay
• Need to have tight control of absolute delay to meet a given cycle time
• Sensitive to delay variations in on-chip circuits and board routes
• Hard to compensate for delay variations due to low correlation between on-chip and off-chip delays
• While commonly used in on-chip communication, offers limited speed in I/O applications
Forward Clock I/O Architecture

- Common high-speed reference clock is forwarded from TX chip to RX chip
  - Mesochronous system
- Used in processor-memory interfaces and multi-processor communication
  - Intel QPI
  - Hypertransport
- Requires one extra clock channel
- “Coherent” clocking allows low-to-high frequency jitter tracking
- Need good clock receive amplifier as the forwarded clock is attenuated by the channel
Forward Clock I/O Limitations

- Clock skew can limited forward clock I/O performance
  - Driver strength and loading mismatches
  - Interconnect length mismatches
- Low pass channel causes jitter amplification
- Duty-Cycle variations of forwarded clock
Forward Clock I/O De-Skew

- Per-channel de-skew allows for significant data rate increases
- Sample clock adjusted to center clock on the incoming data eye
- Implementations
  - Delay-Locked Loop and Phase Interpolators
  - Injection-Locked Oscillators
- Phase Acquisition can be
  - BER based – no additional input phase samplers
  - Phase detector based implemented with additional input phase samplers periodically powered on
Forward Clock I/O Circuits

- TX PLL
- TX Clock Distribution
- Replica TX Clock Driver
- Channel
- Forward Clock Amplifier
- RX Clock Distribution
- De-Skew Circuit
  - DLL/PI
  - Injection-Locked Oscillator
Embedded Clock I/O Architecture

- Can be used in mesochronous or plesiochronous systems
- Clock frequency and optimum phase position are extracted from incoming data stream
- Phase detection continuously running
- CDR Implementations
  - Per-channel PLL-based
  - Dual-loop w/ Global PLL &
    - Local DLL/PI
    - Local Phase-Rotator PLLs
Embedded Clock I/O Limitations

- Jitter tracking limited by CDR bandwidth
  - Technology scaling allows CDRs with higher bandwidths which can achieve higher frequency jitter tracking
- Generally more hardware than forward clock implementations
  - Extra input phase samplers
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
PLLs

- PLL modeling
- PLL noise transfer functions
- PLL circuits
- PLL design procedure
Introduction

• A phase-locked loop (PLL) is a negative feedback system where an oscillator-generated signal is phase AND frequency locked to a reference signal

• PLLs applications
  • Frequency synthesis
    • Multiplying a 100MHz reference clock to 10GHz
  • Skew cancellation
    • Phase aligning an internal clock to an I/O clock
  • Clock recovery
    • Extract from incoming data stream the clock frequency and optimum phase of high-speed sampling clocks
  • Modulation/De-modulation
    • Wireless systems
    • Spread-spectrum clocking
Forward Clock I/O Circuits

- TX PLL
- TX Clock Distribution
- Replica TX Clock Driver
- Channel
- Forward Clock Amplifier
- RX Clock Distribution
- De-Skew Circuit
  - DLL/PI
  - Injection-Locked Oscillator
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
PLL Block Diagram

\[ \phi_{\text{ref}}, \, C K_{\text{ref}} \rightarrow \text{Phase Detector} \rightarrow \text{error} \rightarrow \text{Low-Pass Filter} \rightarrow \text{Oscillator} \rightarrow \phi_{\text{out}}, \, C K_{\text{out}} \]

\[ \phi_{\text{feedback}}, \, C K_{\text{feedback}} \rightarrow \text{Frequency Divider} : N \]

[Mansuri]
Phase Detector

- Detects phase difference between feedback clock and reference clock
- The loop filter will filter the phase detector output, thus to characterize phase detector gain, extract average output voltage (or current for charge-pump PLLs)
Loop Filter

- Lowpass filter extracts average of phase detector error pulses
Voltage-Controlled Oscillator

\[ V_c(t) \rightarrow \text{out}(t) \]

- Time-domain phase relationship

\[ \phi_{out}(t) = \int \Delta \omega_{out}(t) dt = K_{VCO} \int v_c(t) dt \]

\[ \omega_{out}(t) = \omega_0 + \Delta \omega_{out}(t) = \omega_0 + K_{VCO} v_c(t) \]

Laplace Domain Model

\[ V_c(t) \rightarrow \frac{K_{VCO}}{s} \rightarrow \phi_{out}(t) \]
Loop Divider

• Time-domain model

\[ \omega_{fb}(t) = \frac{1}{N} \omega_{out}(t) \]

\[ \phi_{fb}(t) = \int \frac{1}{N} \omega_{out}(t) \, dt = \frac{1}{N} \phi_{out}(t) \]

[Perrott] N = 6
Understanding PLL Frequency Response

- Linear “small-signal” analysis is useful for understand PLL dynamics if
  - PLL is locked (or near lock)
  - Input phase deviation amplitude is small enough to maintain operation in lock range
- Frequency domain analysis can tell us how well the PLL tracks the input phase as it changes at a certain frequency
- PLL transfer function is different depending on which point in the loop the output is responding to

[Fischette]
Charge-Pump PLL Linear Model

Charge-pump supplies current to loop filter capacitor which integrates it to produce the VCO control voltage.

For stability, a zero is added with the resistor which gives a proportional gain term.
Open-Loop PLL Transfer Function

\[ H_{open}(s) = K_{PFD} \cdot \frac{I_{CP}}{2\pi} \cdot F(s) \cdot \frac{K_{VCO}}{s} \]

ignoring \( C_1 \): \[ H_{open}(s) = K_{PFD} \cdot \frac{I_{CP}}{2\pi \cdot C_{CP}} \cdot (1 + RC_{CP}s) \cdot \frac{K_{VCO}}{s^2} \] (2\textsuperscript{nd} order)

with \( C_1 \): \[ H_{open}(s) = K_{PFD} \cdot \frac{I_{CP}}{2\pi \cdot (C_{CP} + C_1)} \cdot (1 + RC_{CP}s) \cdot \frac{K_{VCO}}{s^2 \cdot [1 + R(C_{CP}||C_1)s]} \] (3\textsuperscript{rd} order)
Open-Loop PLL Transfer Function

\[ |H_{\text{open}}(s)| \cdot \frac{1}{N} \]

w/o C₁ (2\textsuperscript{nd} order)

\[ |H_{\text{open}}(s)| \cdot \frac{1}{N} \]

0dB \quad 20dB/\text{dec} \quad \omega_z \quad \omega_c

\[ \angle H_{\text{open}}(s) \]

-90\degree \quad -180\degree

with C₁ (3\textsuperscript{rd} order)

\[ |H_{\text{open}}(s)| \cdot \frac{1}{N} \]

0dB \quad 20dB/\text{dec} \quad \omega_z \quad \omega_c \quad \omega_{p3}

\[ \angle H_{\text{open}}(s) \]

-90\degree \quad -180\degree

[Mansuri]
Closed-Loop PLL Transfer Function

\[
\frac{\phi_{out}(s)}{\phi_{in}} = H_{closed}(s) = \frac{H_{open}(s)}{1 + H_{open}(s) \cdot 1/N}
\]

ignoring \( C_1 \):

\[
\frac{\phi_{out}(s)}{\phi_{in}} = \frac{K_{Loop} \cdot (1 + RC_{CP}s)}{s^2 + (K_{Loop}/N)RC_{CP}s + K_{Loop}/N}
\]

\[
K_{Loop} = K_{PFD} \cdot K_{VCO} \cdot I_{CP}/(2\pi C_{CP})
\]
PLL Natural Frequency and Damping Factor

\[
\frac{\phi_{out}(s)}{\phi_{in}} = \frac{K_{Loop} \cdot (1 + RC_{CPS}s)}{s^2 + (K_{Loop}/N)RC_{CPS}s + K_{Loop}/N}
\]

Standard 2\textsuperscript{nd}-order denominator: \(s^2 + 2\zeta\omega_n s + \omega_n^2\)

Natural Frequency: \(\omega_n = \sqrt{\frac{K_{Loop}}{N}}\)

Damping Factor: \(\zeta = \frac{\omega_n}{2 \cdot \omega_z}\)

Loop Bandwidth: \(\omega_{3dB} = \omega_n \left(a + \sqrt{a^2 + 1}\right)^{1/2}\)

\[
a = 2\zeta^2 + 1 - \frac{\omega_n N}{K_{PD}K_{VCO}} \left(4\zeta - \frac{\omega_n N}{K_{PD}K_{VCO}}\right)
\]
Damping Factor Impact

- If damping factor is too low, frequency peaking occurs
  - Damping factor ~1 is usually preferred
- Excessively high damping also causes peaking
  - Need 3rd order model to observe this
Damping Factor Impact

- Peaking in frequency domain leads to ringing in the time domain

\[ \theta_{ref}(s) = \frac{\Delta\omega}{s^2} \]

- \( K_{PD} = \frac{25 \mu A}{2\pi} \)
- \( K_{VCO} = 2\pi 40MHz/V \)
- \( N = 32 \)
- \( \omega_n = 1 \) (normalized)
PLL Noise Transfer Function

[Diagram of PLL Noise Transfer Function]

- Input Clock
- PD
- VCO
- Clock Buffer
- $V_{n_{in}}$
- $K_{PD}$
- $I_{CP}$
- $R$
- $C$
- $V_{n_{VCO}}$
- $V_{n_{buf}}$
- $\phi_{out}$
- $\phi_{n_{in}}$
- $\phi_{n_{VCO}}$
- $\phi_{n_{buf}}$

[Mansuri]
Input Noise Transfer Function

Input Phase Noise:

\[
H_{n_{IN}}(s) = \frac{\phi_{out}}{\phi_{n_{IN}}} = \frac{K_{Loop}(RCs + 1)}{s^2 + \left(\frac{K_{Loop}}{N}\right) RCs + \frac{K_{Loop}}{N}} = \frac{N(2\zeta\omega_n s + \omega_n^2)}{s^2 + 2\zeta\omega_n s + \omega_n^2}
\]

Voltage Noise on Input Clock Source:

\[
T_{n_{IN}}(s) = \frac{\phi_{out}}{v_{n_{IN}}} = \left(\frac{\phi_{out}}{\phi_{n_{IN}}}\right)\left(\frac{K_o}{s}\right) = \frac{K_oK_{Loop}(RCs + 1)}{s\left(s^2 + \left(\frac{K_{Loop}}{N}\right) RCs + \frac{K_{Loop}}{N}\right)}
\]

[Image: Diagram of the input noise transfer function with components labeled and equations derived from Mansuri.]
VCO Noise Transfer Function

\[ H_{n_{\text{vco}}} (s) = \frac{\phi_{\text{out}}}{\phi_{n_{\text{vco}}}} = \frac{s^2}{s^2 + \left( \frac{K_{\text{Loop}}}{N} \right) RCs + \frac{K_{\text{Loop}}}{N}} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \]

VCO Phase Noise:

Voltage Noise on VCO Inputs:

\[ T_{n_{\text{vco}}} (s) = \frac{\phi_{\text{out}}}{v_{n_{\text{vco}}}} = \left( \frac{\phi_{\text{out}}}{\phi_{n_{\text{vco}}}} \right) \left( \frac{K_{\text{vco}}}{s} \right) = \frac{K_{\text{vco}} s}{s^2 + \left( \frac{K_{\text{Loop}}}{N} \right) RCs + \frac{K_{\text{Loop}}}{N}} \]
Clock Buffer Noise Transfer Function

Output Phase Noise: \( H_{n_{\text{buf}}} (s) = \frac{\phi_{\text{out}}}{\phi_{n_{\text{buf}}}} = \frac{s^2}{s^2 + \left( \frac{K_{\text{Loop}}}{N} \right) RCs + \frac{K_{\text{Loop}}}{N}} \approx \frac{s^2}{s^2 + \left( \frac{K_{\text{Loop}}}{N} \right) RCs + \frac{K_{\text{Loop}}}{N}} \)

Voltage Noise on Buffer Inputs:

\[
T_{n_{\text{buf}}} (s) = \frac{\phi_{\text{out}}}{V_{n_{\text{buf}}}} = \left( \frac{\phi_{\text{out}}}{\phi_{n_{\text{buf}}}} \right) \left( \frac{K_{\text{delay}} \omega_{\text{VCO}}}{s + \frac{\omega_{\text{buf}}}{\omega_{\text{buf}}}} + 1 \right) = \left( \frac{K_{\text{delay}} \omega_{\text{VCO}}}{s + \frac{\omega_{\text{buf}}}{\omega_{\text{buf}}}} + 1 \right) \frac{s^2}{s^2 + \left( \frac{K_{\text{Loop}}}{N} \right) RCs + \frac{K_{\text{Loop}}}{N}} \approx \frac{K_{\text{delay}} \omega_{\text{VCO}} s^2}{s^2 + \left( \frac{K_{\text{Loop}}}{N} \right) RCs + \frac{K_{\text{Loop}}}{N}}
\]
Noise Transfer Functions

Input phase noise (N=1) (low-pass)

VCO input voltage noise (band-pass)

Buffer phase or voltage noise (high-pass)

Clock buffer noise

Input clock noise

Mansuri
Charge-Pump PLL Circuits

- Phase Detector
- Charge-Pump
- Loop Filter
- VCO
- Divider
Phase Detector

- Detects phase difference between feedback clock and reference clock
- The loop filter will filter the phase detector output, thus to characterize phase detector gain, extract average output voltage (or current for charge-pump PLLs)

\[
\phi_{\text{ref}} \rightarrow \Sigma \rightarrow \phi_e \rightarrow K_{PD} \rightarrow v_e
\]

\[\text{avg}\{Ve(t)\} = K_{PD} \Delta \phi\]
Analog Multiplier Phase Detector

- If $\omega_1 = \omega_2$ and filtering out high-frequency term

$$
\bar{y}(t) = \frac{\alpha A_1 A_2}{2} \cos[(\omega_1 + \omega_2)t + \Delta \phi] + \frac{\alpha A_1 A_2}{2} \cos[(\omega_1 - \omega_2)t - \Delta \phi]
$$

- $\alpha$ is mixer gain

- Near $\Delta \phi$ lock region of $\pi/2$:

$$
\bar{y}(t) \approx \frac{\alpha A_1 A_2}{2} \left( \frac{\pi}{2} - \Delta \phi \right)
$$

$K_{PD} = -\frac{\alpha A_1 A_2}{2}$

[Razavi]
XOR Phase Detector

- Sensitive to clock duty cycle
XOR Phase Detector

Width is same for both leading and lagging phase difference!

$$W = -\left(\frac{\Phi_{\text{ref}} - \Phi_{\text{div}}}{\pi}\right)T/2$$

$$W = \left(\frac{\Phi_{\text{ref}} - \Phi_{\text{div}}}{\pi}\right)T/2$$

[Perrott]
Cycle Slipping

- If there is a frequency difference between the input reference and PLL feedback signals, the phase detector can jump between regions of different gain.
  - PLL is no longer acting as a linear system.

\[ \text{avg}\{e(t)\} \]

\[ \Phi_{\text{ref}} - \Phi_{\text{div}} \]

- Gain = \(-2/\pi\)
- Gain = \(2/\pi\)

Phase detector range = \(\pi\)

(positive feedback operation)  (negative feedback operation)
Cycle Slipping

- If frequency difference is too large the PLL may not lock
Phase Frequency Detector (PFD)

- Phase Frequency Detector allows for wide frequency locking range, potentially entire VCO tuning range
- 3-stage operation with UP and DOWN outputs
- Edge-triggered results in duty cycle insensitivity
PFD Transfer Characteristic

UP=1 & DN=-1

- Constant slope and polarity asymmetry about zero phase allows for wide frequency range operation

\[ \text{gain} = \frac{1}{2\pi} \]

\[ \phi_{\text{ref}} - \phi_{\text{div}} \]

Phase detector range = $4\pi$
PFD Deadzone

- If phase error is small, then short output pulses are produced by PFD
- Cannot effectively propagate these pulses to switch charge pump
- Results in phase detector “dead zone” which causes low loop gain and increased jitter
- Solution is to add delay in PFD reset path to force a minimum UP and DOWN pulse length
PFD Operation

Ref

Cycle Slip

FbClk

Min. Pulse Width

GoFaster

Vctl

GoSlower
Charge-Pump PLL Circuits

- Phase Detector
- Charge-Pump
- Loop Filter
- VCO
- Divider
Charge Pump

- Converts PFD output signals to charge
- Charge is proportional to PFD pulse widths

**PFD-CP Gain:** \( \left( \frac{1}{2\pi} \right) I_{CP} \)
Simple Charge Pump

- Issues
  - Switch resistance can impact UP/DN current matching as a function of $V_{ctrl}$
  - Clock feedthrough and charge injection from switches onto $V_{ctrl}$
  - Charge sharing between current source drain nodes’ capacitance and $V_{ctrl}$

[Razavi]
Charge Pump Mismatch

- PLL will lock with static phase error
- Extra “ripple” on Vctrl
  - Results in frequency domain spurs at the reference clock frequency offset from the carrier

[Razavi]
Charge Pump w/ Improved Matching

- Amplifier keeps current source Vds voltages constant resulting in reduced transient current mismatch

[Young J SSC 1992]
Charge Pump w/ Reversed Switches

- Swapping switches reduces charge injection
  - MOS caps (Md1-4) provide extra charge injection cancellation

- Helper transistors Mx and My quickly turn-off current source

- Dummy brand helps to match PFD loading

[Ingino J SSC 2001]
Charge-Pump PLL Circuits

- Phase Detector
- Charge-Pump
- Loop Filter
- VCO
- Divider
Loop Filter

- Lowpass filter extracts average of phase detector error pulses
Loop Filter Transfer Function

- Neglecting secondary capacitor, $C_2$

\[ F(s) = \frac{V_c(s)}{I_e(s)} = \frac{R \left( s + \frac{1}{RC_1} \right)}{s} \]
Loop Filter Transfer Function

- With secondary capacitor, $C_2$

$$Z(s) = \frac{1}{C_2} \left( \frac{1}{s + \frac{1}{RC_1}} \right)$$
Why have C2?

- Secondary capacitor smoothes control voltage ripple
- Can’t make too big or loop will go unstable
  - $C_2 < C_1/10$ for stability
  - $C_2 > C_1/50$ for low jitter

PLL Synthesizing a 380MHz Signal
Filter Capacitors

• To minimize area, we would like to use highest density caps

• Thin oxide MOS cap gate leakage can be an issue
  • Similar to adding a non-linear parallel resistor to the capacitor
  • Leakage is voltage and temperature dependent
  • Will result in excess phase noise and spurs

• Metal caps or thick oxide caps are a better choice
  • Trade-off is area

• Metal cap density can be < 1/10 thin oxide caps

• Filter cap frequency response can be relatively low, as PLL loop bandwidths are typically 1-50MHz
Charge-Pump PLL Circuits

- Phase Detector
- Charge-Pump
- Loop Filter
- VCO
- Divider
Voltage-Controlled Oscillator

- Time-domain phase relationship

\[ \phi_{out}(t) = \int \Delta \omega_{out}(t) \, dt = K_{VCO} \int v_c(t) \, dt \]

\[ \omega_{out}(t) = \omega_0 + \Delta \omega_{out}(t) = \omega_0 + K_{VCO} v_c(t) \]
Voltage-Controlled Oscillators (VCO)

• Ring Oscillator
  • Easy to integrate
  • Wide tuning range (5x)
  • Higher phase noise

• LC Oscillator
  • Large area
  • Narrow tuning range (20-30%)
  • Lower phase noise
Barkhausen’s Oscillation Criteria

Closed-loop transfer function: \[ \frac{H(j\omega)}{1 - H(j\omega)} \]

• Sustained oscillation occurs if \( H(j\omega) = 1 \)

• 2 conditions:
  • Gain = 1 at oscillation frequency \( \omega_0 \)
  • Total phase shift around loop is \( n360^\circ \) at oscillation frequency \( \omega_0 \)
Ring Oscillator Example

Three-stage ring oscillator

\[ H(s) = -\frac{A_0^3}{\left(1 + s/\omega_0\right)^3} \]

\[ \omega_{osc} = \sqrt{3}\omega_0 \]

\[ \tan^{-1}\frac{\omega_{osc}}{\omega_o} = 60^\circ \]

\[ \frac{V_{out}(s)}{V_{in}(s)} = \frac{-A_0^3}{\left(1 + s/\omega_0\right)^3} = \frac{-A_0^3}{\left(1 + s/\omega_0\right)^3 + A_0^3} \]

\[ \sqrt{1 + \left(\frac{\omega_{osc}}{\omega_0}\right)^2} = 1 \]

\[ A_0 = 2 \]
Ring Oscillator Example

- 4-stage oscillator
  - $A_0 = \sqrt{2}$
  - Phase shift = 45

- Easier to make a larger-stage oscillator oscillate, as it requires less gain and phase shift per stage

$$H(s) = -\frac{A_0^3}{3} \left(1 + \frac{s}{\omega_0}\right)^3$$

$$\omega_{osc} = \sqrt{3} \omega_0$$

$$\tan^{-1} \frac{\omega_{osc}}{\omega_0} = 60^\circ$$

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{-A_0^3}{1 + A_0^3} \left(1 + \frac{s}{\omega_0}\right)^3 + A_0^3$$

$$A_0 = 2$$
LC Oscillator Example

- Oscillation phase shift condition satisfied at the frequency when the LC (and R) tank load displays a purely real impedance, i.e. 0° phase shift

**LC tank impedance**

\[ Z_{eq}(s) = \frac{R_S + L_1 s}{1 + L_1 C_1 s^2 + R_S C_1 s} \]

\[ |Z_{eq}(s = j\omega)|^2 = \frac{R_S^2 + L_1^2 \omega^2}{\left(1 - L_1 C_1 \omega^2\right)^2 + R_S^2 C_1^2 \omega^2} \]
LC Oscillator Example

- Transforming the series loss resistor of the inductor to an equivalent parallel resistance

\[
L_P = L_1 \left(1 + \frac{R_S^2}{L_1^2 \omega^2}\right), \quad C_P = C_1, \quad R_P \approx \frac{L_1^2 \omega^2}{R_S}
\]

\[
\omega_1 = \frac{1}{\sqrt{L_P C_P}}
\]

[Razavi]
• Phase condition satisfied at

\[ \omega_1 = \frac{1}{\sqrt{L_p C_p}} \]

• Gain condition satisfied when \((g_m R_p)^2 \geq 1\)

• Can also view this circuit as a parallel combination of a tank with loss resistance \(2R_p\) and negative resistance of \(2/g_m\)

• Oscillation is satisfied when

\[ \frac{1}{g_m} \leq R_p \]
Supply-Tuned Ring Oscillator

\[ T_{VCO} = 2nT_D \approx \frac{2nC_{stage}}{\beta(V_c - V_{th})} \]

\[ K_{VCO} = \frac{\partial f_{VCO}}{\partial V_c} = \frac{\beta}{2nC_{stage}} \]
Current-Starved Ring Oscillator

Current-starved VCO.
Capacitive-Tuned Ring Oscillator

\[ C_{\text{eff}} = \frac{C}{1 + sCR} \]
Symmetric Load Ring Oscillator

- Symmetric load provides frequency tuning at excellent supply noise rejection
- See Maneatis papers for self-biased techniques to obtain constant damping factor and loop bandwidth (% of ref clk)
LC Oscillator

- A variable capacitor (varactor) is often used to adjust oscillation frequency.

- Total capacitance includes both tuning capacitance and fixed capacitances which reduce the tuning range.

\[ \omega_{osc} = \frac{1}{\sqrt{L_p C_p}} = \frac{1}{\sqrt{L_p \left( C_{tune} + C_{fixed} \right)}} \]
Varactors

- pn junction varactor
  - Avoid forward bias region to prevent oscillator nonlinearity

- MOS varactor
  - Accumulation-mode devices have better Q than inversion-mode
Oscillator Noise

Jitter

PHASE NOISE

[McNeill]
Oscillator Phase Noise Model

Leeson’s Model:

\[ L(\Delta f) = 10 \log \left( \frac{2FkT}{P_{\text{sig}}} \left( \frac{1}{2Q} \frac{f_o}{\Delta f} \right)^2 \left( 1 + \frac{\Delta f_{1/3}}{\Delta f} \right) \right) \text{ (dBc/Hz)} \]

For improved model see Hajimiri papers
• Measure distribution of clock threshold crossings
• Plot $\sigma$ as a function of delay $\Delta T$
Open-Loop VCO Jitter

- Jitter $\sigma$ is proportional to $\sqrt{\Delta T}$
- $\kappa$ is VCO time domain figure of merit

$\sigma_{\Delta T(OL)}(\Delta T) \approx \kappa \sqrt{\Delta T}$
VCO in Closed-Loop PLL Jitter

- PLL limits $\sigma$ for delays longer than loop bandwidth $\tau_L$

$$\tau_L = \frac{1}{2\pi f_L}$$
Converting Phase Noise to Jitter

- RMS jitter for $\Delta T$ accumulation
  
  $$\sigma_{\Delta T}^2 = \frac{8}{\omega_o^2} \int_{0}^{\infty} S_\phi(f) \sin^2(\pi f \Delta T) df$$

- As $\Delta T$ goes to $\infty$
  
  $$\sigma_T^2 = \frac{2}{\omega_o^2} R_\phi(0) = \frac{4}{\omega_o^2} \int_{0}^{\infty} S_\phi(f) df$$

- Integration range depends on application bandwidth
  
  - $f_{\text{min}}$ set by standard
    - Ex. Assumed CDR tracking bandwidth
  - Usually stop integration at $f_o/2$ or $f_o$ due to measurement limitations and aliasing components

---

[Mansuri]
Charge-Pump PLL Circuits

- Phase Detector
- Charge-Pump
- Loop Filter
- VCO
- Divider
Loop Divider

• Time-domain model

\[ \omega_{fb}(t) = \frac{1}{N} \omega_{out}(t) \]

\[ \phi_{fb}(t) = \int \frac{1}{N} \omega_{out}(t) dt = \frac{1}{N} \phi_{out}(t) \]
Basic Divide-by-2

- Divide-by-2 can be realized by a flip-flop in “negative feedback”
- Divider should operate correctly up to the maximum output clock frequency of interest **PLUS** some margin

[Perrott]

[Graph showing VCO Frequency over time with labels: Feedback Divider Fails, Max VCO Freq, Desired VCO Freq]
Divide-by-2 with TSPC FF

True Single Phase Clock Flip-Flop

- **Advantages**
  - Reasonably fast, compact size, and no static power
  - Requires only one phase of the clock

- **Disadvantages**
  - Signal needs to propagate through three gates per input cycle
  - Need full swing CMOS inputs
  - Dynamic flip-flop may have issues at very low frequency operation (test mode) depending on process leakage
Divide-by-2 with CML FF

- **Advantages**
  - Signal only propagates through two CML gates per input cycle
  - Accepts CML input levels
- **Disadvantages**
  - Larger size and dissipates static power
  - Requires differential input
  - Need tail current biasing
- Additional speedup (>50%) can be achieved with shunt peaking inductors

[Razavi]
Binary Dividers: Asynchronous vs Synchronous

**Asynchronous Divider**

- **Advantages**
  - Each stage runs at lower frequency, resulting in reduced power
  - Reduced high frequency clock loading

- **Disadvantage**
  - Jitter accumulation

**Synchronous Divider**

- **Advantage**
  - Reduced jitter

- **Disadvantage**
  - All flip-flops work at maximum frequency, resulting in high power
  - Large loading on high frequency clock

[Perrott]
Jitter in Asynchronous vs Synchronous Dividers

Asynchronous

- Jitter accumulates with the clock-to-Q delays through the divider
- Extra divider delay can also degrade PLL phase margin

Synchronous

- Divider output is “sampled” with high frequency clock
- Jitter on divider clock is similar to VCO output
- Minimal divider delay

[Perrott]
Dual Modulus Prescalers

\[ \div 2/3 \]
\[ \div 15/16 \]

MC=0 \rightarrow \div 3
MC=1 \rightarrow \div 2

Synchronous \div 3/4

Asynchronous \div 4

• For /15, first prescaler circuit divides by 3 once and 4 three times during the 15 cycles
Injection-Locked Frequency Dividers

**LC-oscillator type (/ 2)**

\[ v_O = V_O \cos(\omega_0 t + \phi) \]

\[ i_I = I_I \cos(2\omega_0 t) + I_{DC} \]

**Ring-oscillator type (/ 3)**

\[ v_{INJ} \]

- Superharmonic injection-locked oscillators (ILOs) can realize frequency dividers
- Faster and lower power than flip-flop based dividers
- Injection locking range can be limited

[Verma J SSC 2003, Rategh J SSC 1999] [Lo CI CC 2009]
Example PLL Design Procedure

- Design procedure for a 100-300MHz frequency synthesizer

- **Step 1 – Determine VCO Tuning Range**
  - Needs to be at least the output frequency range plus some margin (10-20%) dependent on PVT tolerance
  
  \[
  \text{VCO Tuning Range} = 100 - 300\text{MHz} \ * \\
  \]
  - *Note if you want the frequency extremes (100 or 300MHz) you probably want to add some margin here

- **Step 2 – Determine Loop Division Ratio, N**
  - This is a function of what reference clocks you have access to, loop bandwidth, dominant noise sources
  
  \[
  N = 32 \\
  \]

- **Step 3 – Determine Damping Factor**
  - Damping factors between 0.5 and 2 are reasonable, with 0.7 or 1 commonly chosen
  
  \[
  \zeta = \frac{1}{\sqrt{2}} \approx 0.707 \\
  \]
Example PLL Design Procedure

• Step 4 – Determine natural frequency, $\omega_n$
  • This is a function of the desired loop bandwidth and also the damping factor
  • **Maximum loop bandwidth should be less than 1/10th the input reference clock for the loop to act as a continuous-time system**

  Lowest Input Reference Frequency $= \frac{100\text{MHz}}{32} = 3.125\text{MHz}$

  • Set the loop bandwidth with some margin - 75% of max value

  $\omega_{3dB} = (0.75)(2\pi)312.5\text{kHz} = 1.47 \frac{M\text{rad}}{s}$

  • For a damping factor of 0.707

  $\omega_n = \frac{\omega_{3dB}}{2.06} = \frac{1.47 \frac{M\text{rad}}{s}}{2.06} = 714 \frac{k\text{rad}}{s}$
Example PLL Design Procedure

- Step 5 – Determine $K_{VCO}$
  - This is a function of the VCO and charge pump operating voltage range
  - Here I use a combination of discrete tuning caps, resulting in multiple frequency bands over the total frequency range

- Step 6 – Determine Charge Pump Current & Filter Cap

  Set $I = 25 \mu A$

  \[
  C_1 = \frac{(25 \mu A)(255 \text{ Mrad/sV})}{2\pi(32)^2\left(714 \text{ krad/s}\right)^2} = 62.2 \text{ pF}
  \]

- Step 7 – Determine Filter R and Secondary Cap

  \[
  R = \frac{2\zeta}{\omega_n C_1} = \frac{2(0.707)}{\left(714 \text{ krad/s}\right)(62.2 \text{ pF})} = 31.8 \text{ k}\Omega
  \]

  \[
  C_2 = \frac{C_1}{10} = 6.22 \text{ pF} \Rightarrow C_2 = 6 \text{ pF}
  \]

\[
K_{VCO} = \frac{(2\pi)65\text{MHz}}{1.6\text{V}} = 255 \frac{\text{Mrad/s}}{\text{sV}}
\]
PLL Linear Phase Model

\[ 20\log_{10} \frac{\theta_{out}(s)}{\theta_{ref}(s)} = 1.47 \text{Mrad/s} \]
PLL Linear Phase Model: Frequency Step Response

\[ \theta_{\text{ref}}(s) = \text{Frequency Step Input: } \Delta \omega_{\text{ref}} \]

\[ \theta_{\text{out}}(s) = \frac{K_{\text{vco}}(s+1)(s+2)}{s} \]

No Cycle Slips Observed with Linear Model
PLL Behavioral Model

- Written in SpectreHDL
- Also look at CppSim: http://www.cppsim.com/

```spectre
// Multi-Band Phase Locked Loop Frequency Synthesizer Macromodel
// Main Spectre File
// Samuel Palermo
simulator lang=spectre
include "/home/samuel/research/pll/macromodels/pd/dig_pfd/dig_pfd.def"
include "/home/samuel/research/pll/macromodels/lpf/lpf.def"
ahdl_include "/home/samuel/research/pll/macromodels/vco/vco.def"
ahdl_include "/home/samuel/research/pll/macromodels/vco/switch_vco.def"
ahdl_include "+ "/home/samuel/research/pll/macromodels/divider/divider.def"
include "/home/samuel/research/pll/macromodels/vco/reference.def"

// Power Supply
vdd dd 0 vsource dc=1
// Reference Signal
xref 0 control fref reference
vcontrol control 0 vsource type=pwl wave=[0 0.64 1u 0.64]
// Digital Tri-State Phase/Frequency Comparator
xdig_pfd 0 dd fref fvco up upbar down downbar dig_pfd
// Charge Pump
iup dd 1 isource dc=25u
idown 2 0 isource dc=25u
gup 1 vd up 0 relay vt1=0 vt2=1 ropen=100M rclosed=1m
gupbar 1 upbar 0 relay vt1=0 vt2=1 ropen=100M rclosed=1m
gdown vd 2 down 0 relay vt1=0 vt2=1 ropen=100M rclosed=1m
gdownbar dd 2 downbar 0 relay vt1=0 vt2=1 ropen=100M rclosed=1m
// Loop Filter
xfilter 0 vd lpf
gvdgnd vd 0 vd_gnd 0 relay vt1=0 vt2=1 ropen=100M rclosed=1m
// Voltage Controlled Oscillator
//xvco vd out vco (gain=40e6 fc=256e6)
xvco 0 vd out ev_temp vd_gnd
+ switch_vco (u=0.8 d=-0.8 gain=40e6 fc=256e6)
// Divider
xdistributor 0 out fvco buffer n_temp divider (divisor=32)
```

```spice
timedem tran stop=20u step=20p ic=all maxstep=20p skipdc=yes relref=alllocal
simulator lang=spice
.ic vd=0
save vd control fref fvco nv_temp vd_gnd
.OPTIONS rawfmt=pstbin save=selected diagnose=yes vabstol=.01
+ reltol=.99
*******************************************************************************
// Digital Phase Frequency Detector Macromodel
// Samuel Palermo
subckt dig_pfd (gnd dd fref fvco up upbar down downbar)
ahdl_include "/home/samuel/research/pll/macromodels/pd/dig_pfd/dff.def"
ahdl_include "+ "/home/samuel/research/pll/macromodels/pd/dig_pfd/nand.def"
xdffup gnd dd fref up upbar down downbar dff
xdffdown gnd dd fvco down downbar r dff
xndand gnd up down r and
*******************************************************************************
// D Flip Flop Macromodel
// Samuel Palermo
module dff(gnd, D, CLK, Q, QBAR, R) ()
node [V, I] gnd, D, CLK, Q, QBAR, R ;
{ real Q_temp;
  real QBAR_temp;
  initial {
    Q_temp=0;
    QBAR_temp=1;
  }
  analog {
    if ($threshold (V(CLK, gnd)=1, 1)) {
      if (V(D,gnd)==1) {
        Q_temp=1;
        QBAR_temp=0;
      } else {
        Q_temp=0;
        QBAR_temp=1;
      }
    }
    if (V(R, gnd)==0) {
      Q_temp=0;
      QBAR_temp=1;
    }
  }
}
PLL Frequency Step Response: Linear vs Behavioral Model

Frequency Step Input: $\theta_{\text{ref}}(s) = \frac{\Delta \omega}{s^2}$ = 510 Mrad/sec

No Cycle Slips Observed with Linear Model

Behavioral Macromodel of the PLL

Frequency Step Input of 2.5MHz

VCO Control Voltage

Cycle Slips
Next Time

• CDRs