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

Lecture 29: PLL Wrap-Up

Sam Palermo

Analog & Mixed-Signal Center

Texas A&M University

Announcements

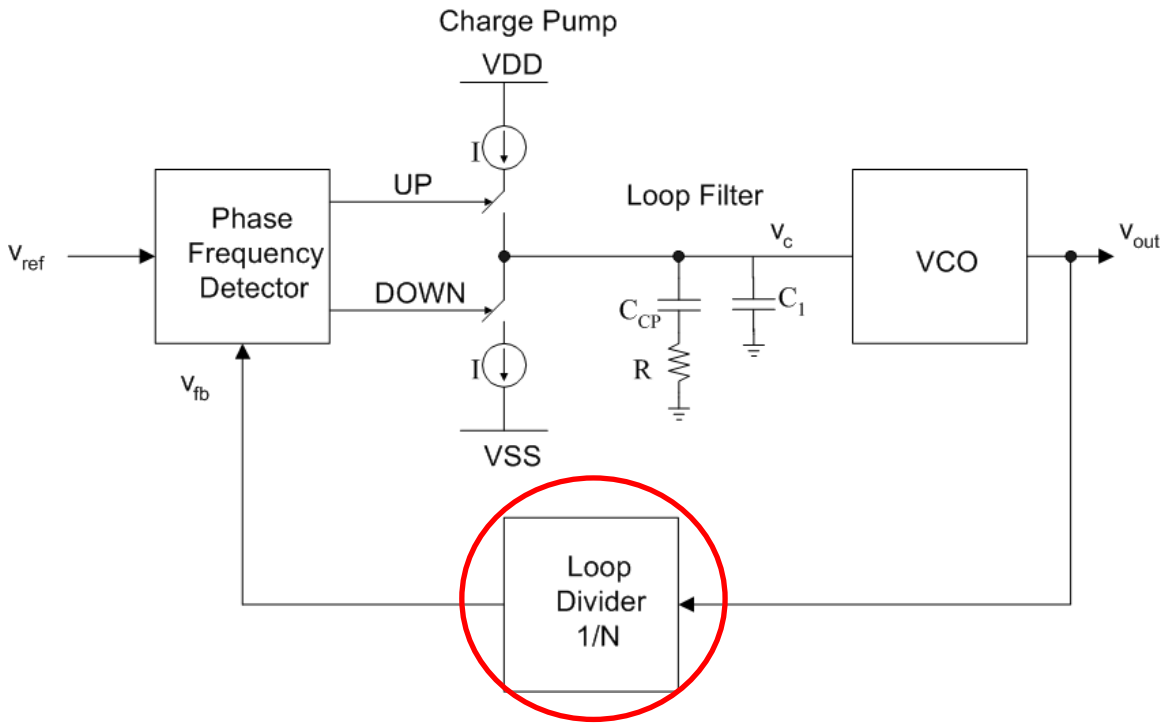
- Project Preliminary Report #2 due Monday April 26 in class
- Exam 2 is April 30
- Project feedback meetings
  - Wednesday 10:30-12
- Will post paper on Bandpass filtering of forwarded clocks

Agenda

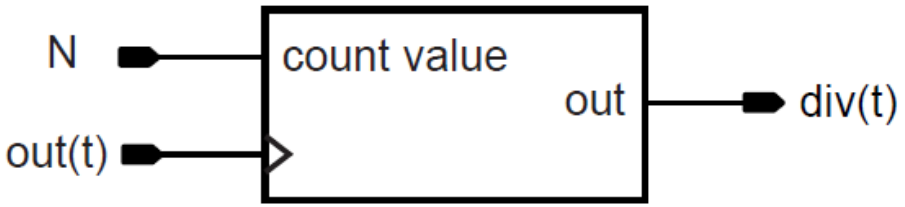
- Clock Dividers
- PLL Design Example
- CDR Introduction

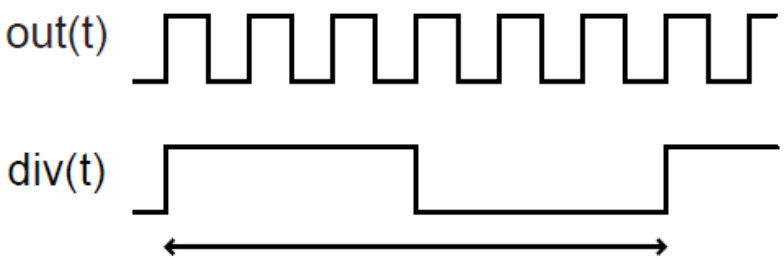
Charge-Pump PLL Circuits

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

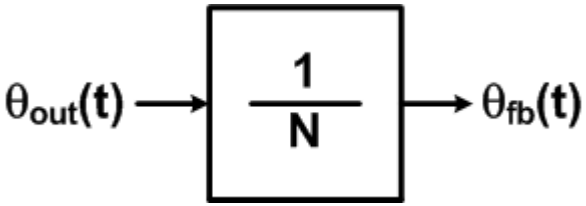


Loop Divider





[**Perrott**]

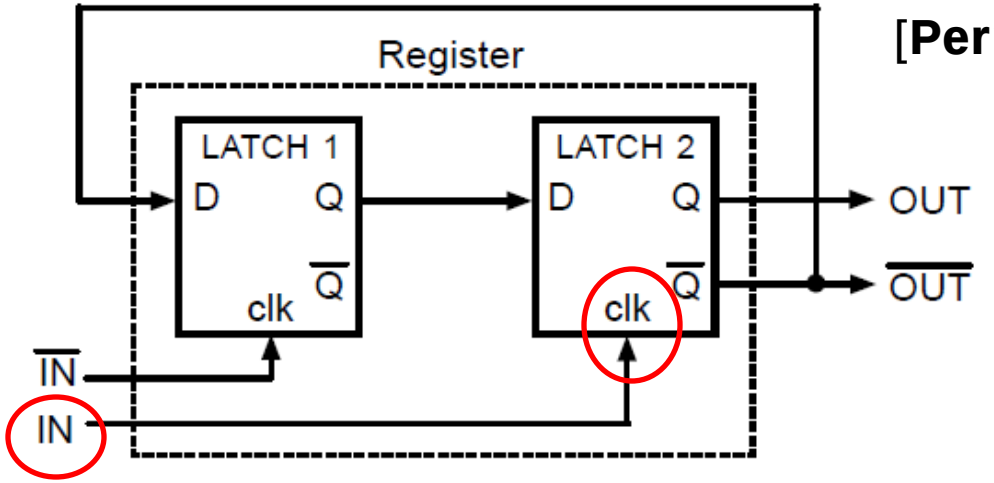


- Time-domain model

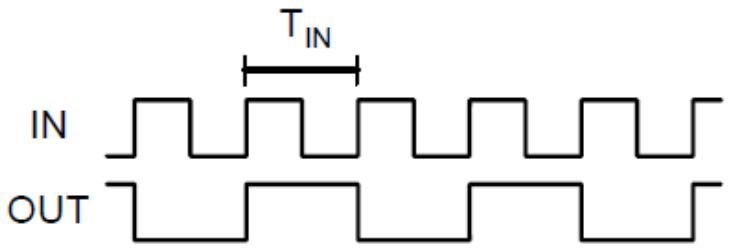
$$\omega_{fb}(t) = \frac{1}{N}\omega_{out}(t)$$

$$\theta_{fb}(t) = \int \frac{1}{N}\omega_{out}(t)\mathbf{dt} = \frac{1}{N}\theta_{out}(t)$$

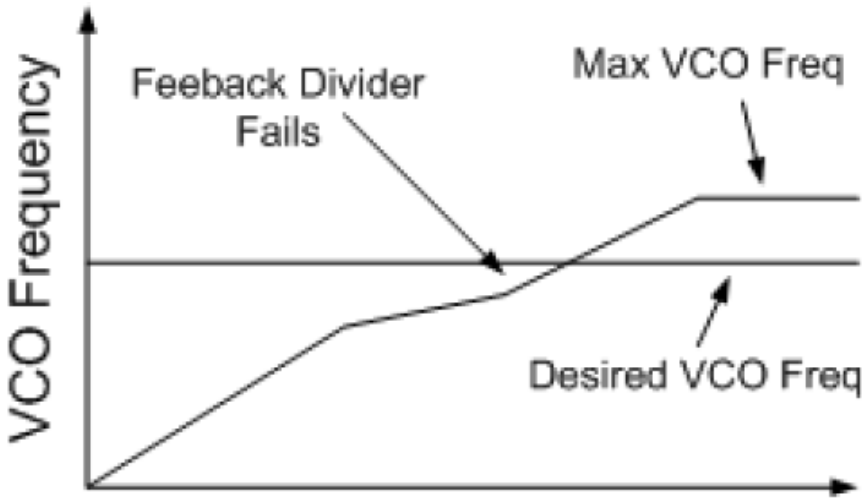
Basic Divide-by-2



[**Perrott**]



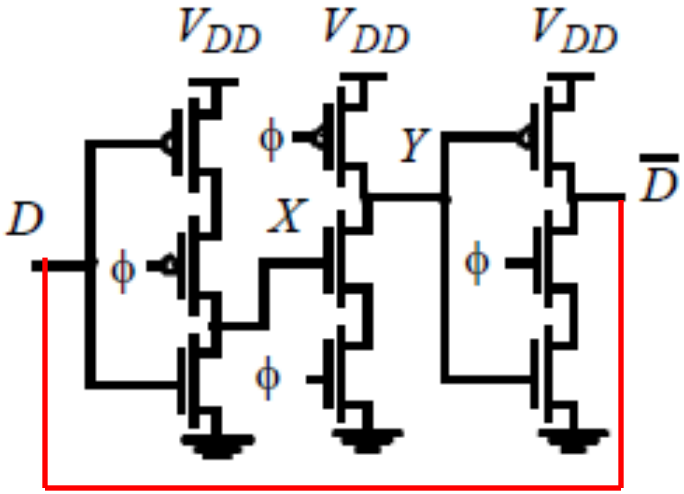
- Divide-by-2 can be realized by a flip-flip in “negative feedback”
- Divider should operate correctly up to the maximum output clock frequency of interest **PLUS** some margin



[**Fischette**]

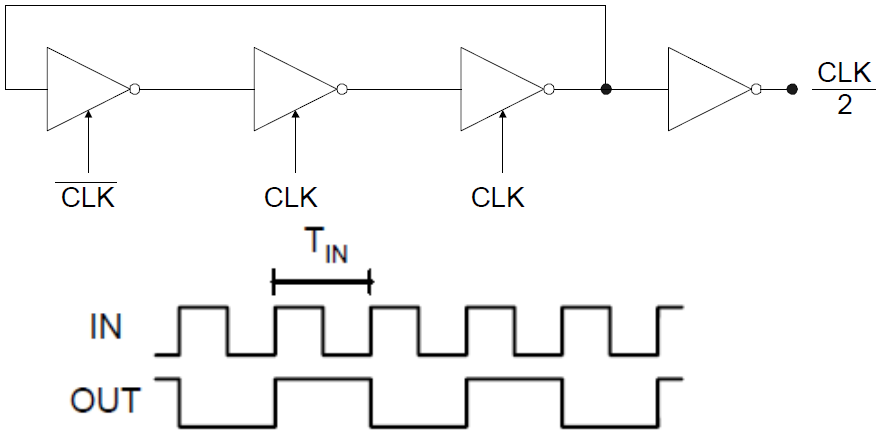
Divide-by-2 with TSPC FF

True Single Phase Clock Flip-Flop



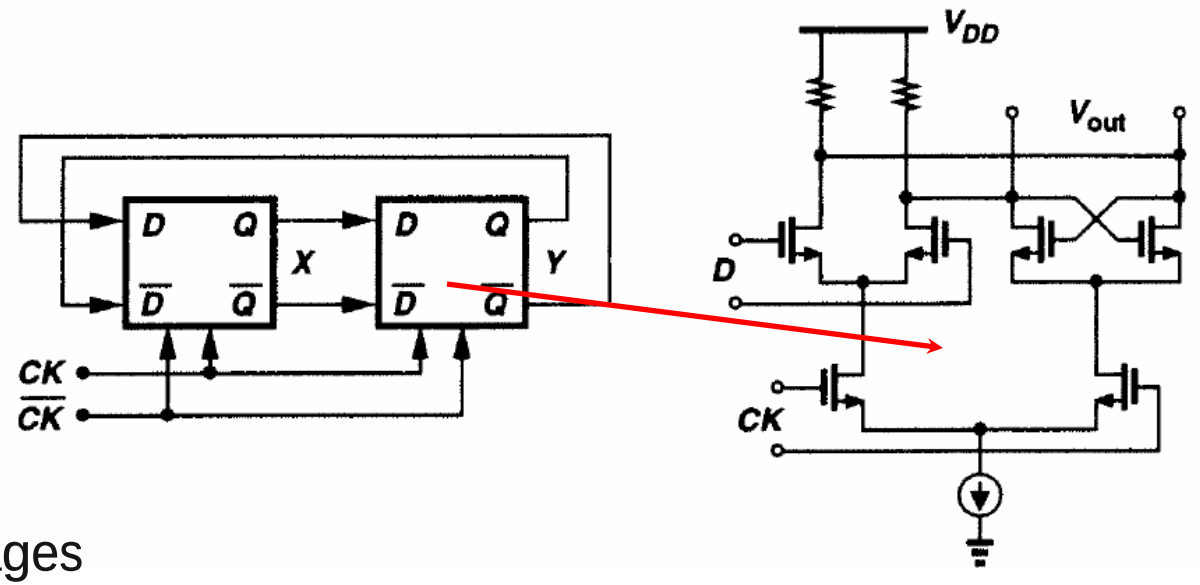
Divider Equivalent Circuit

Note: output inverter not in left schematic



- 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

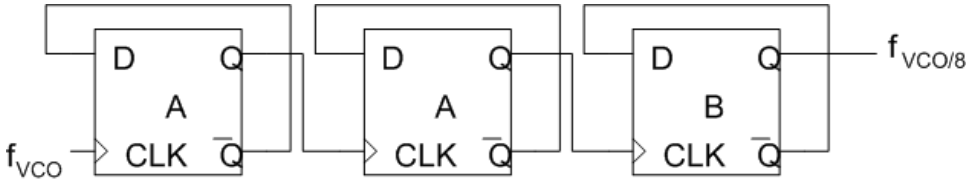


[**Razavi**]

- 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

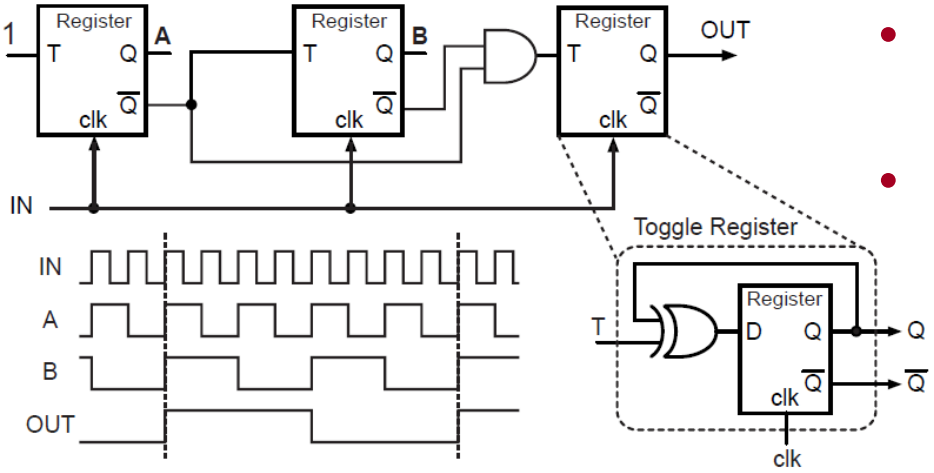
Binary Dividers: Asynchronous vs Synchronous

Asynchronous Divider



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

Synchronous Divider

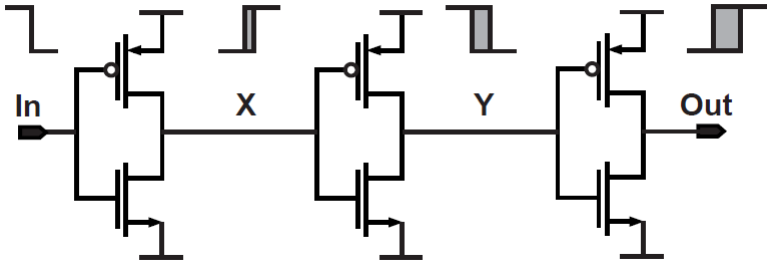


- Advantage
  - Reduced jitter
- Disadvantage
  - All flip-flops work at maximum frequency, resulting in high power
  - Large loading on high frequency clock

[**Perrott**]

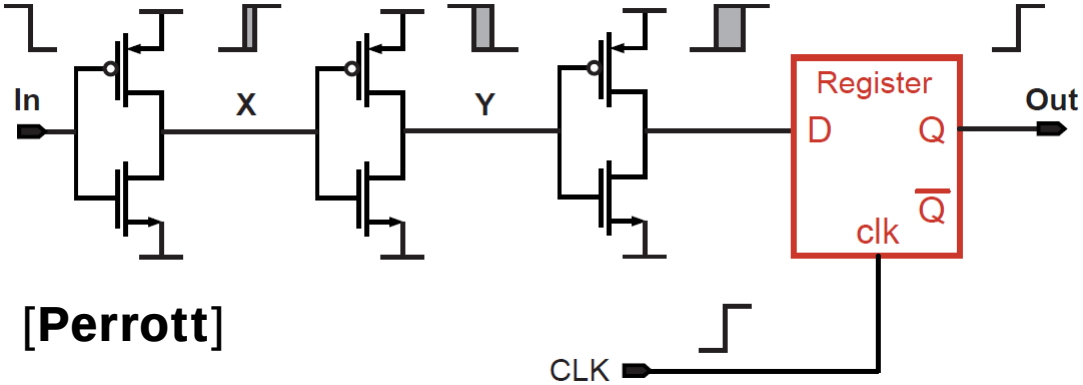
Jitter in Asynchronous vs Synchronous Dividers

Asynchronous



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

Synchronous

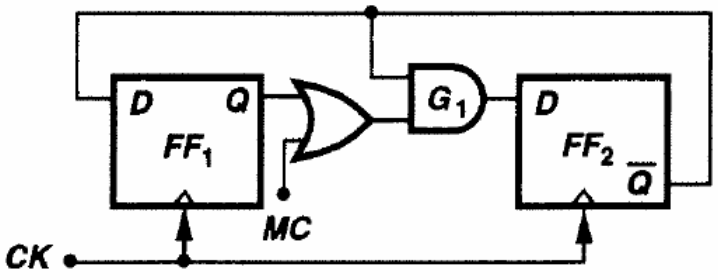


[**Perrott**]

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

Dual Modulus Prescalers

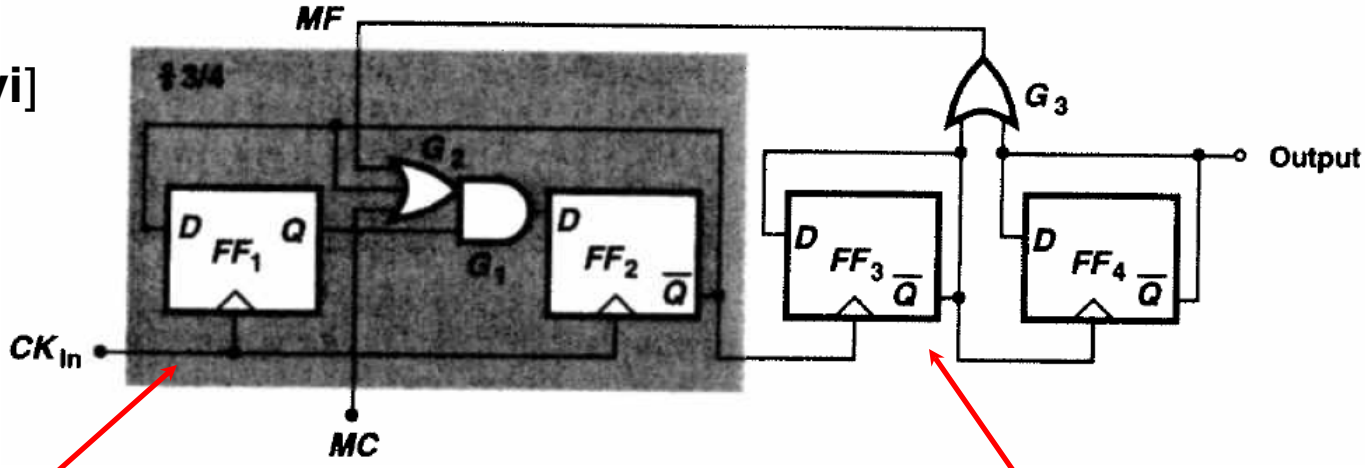
÷2/3



MC=0 → ÷3
MC=1 → ÷2

÷15/16

[**Razavi**]



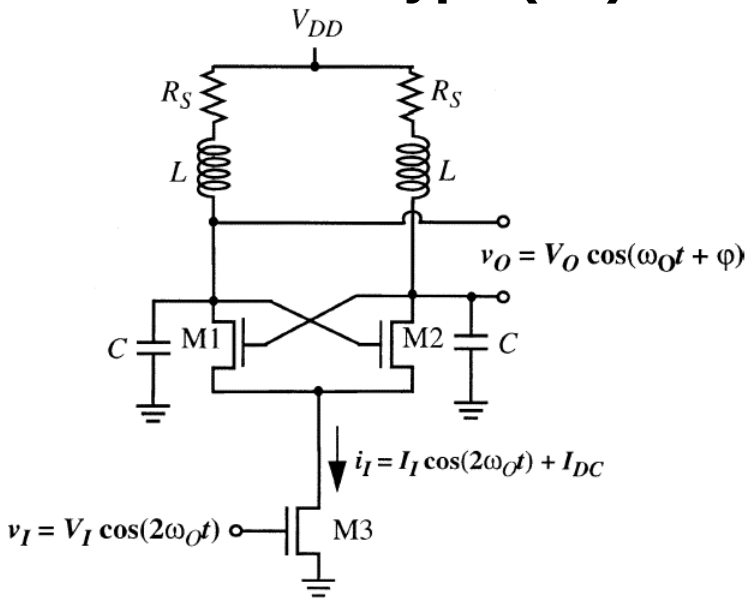
Synchronous ÷3/4

Asynchronous ÷4

- For /15, first prescaler circuit divides by 3 once and 4 three times during the 15 cycles

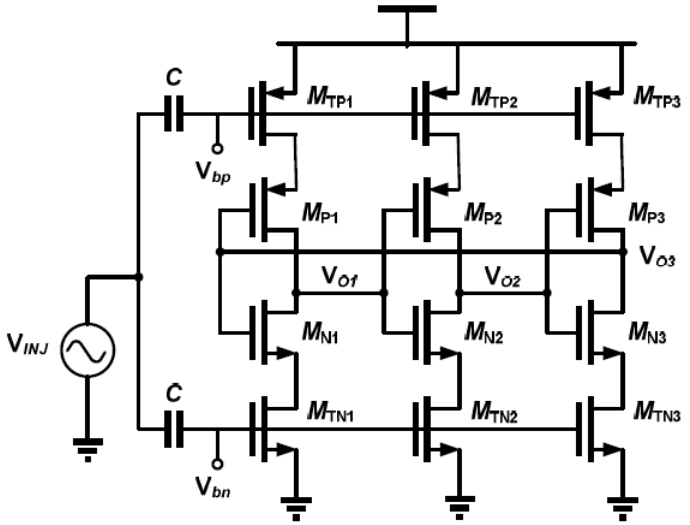
Injection-Locked Frequency Dividers

LC-oscillator type (/2)



[**Verma JSSC 2003, Rategh JSSC 1999**]

Ring-oscillator type (/3)



[**Lo CICC 2009**]

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

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

$$\textbf{VCO Tuning Range} = \textbf{100 - 300MHz *}$$

    - *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, ω_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**

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

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

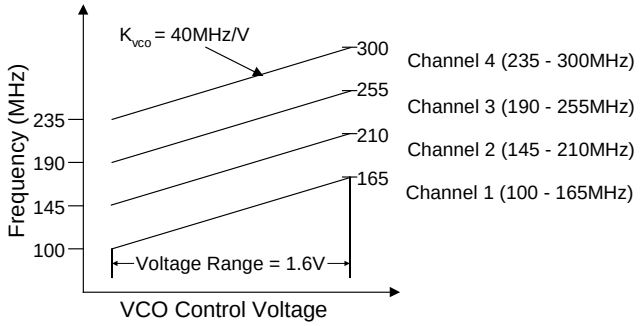
$$\omega_{3dB} = (0.75)(2\pi)312.5\text{kHz} = 1.47\,Mrad/s$$

  - For a damping factor of 0.707

$$\omega_n = \frac{\omega_{3dB}}{2.06} = \frac{1.47\,Mrad/s}{2.06} = 714\,krad/s$$

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



$$K_{VCO} = \frac{(2\pi)65\text{MHz}}{1.6\text{V}} = 255\,Mrad/sV$$

- Step 6 – Determine Charge Pump Current & Filter Cap

$$\text{Set } I = 25\mu A$$

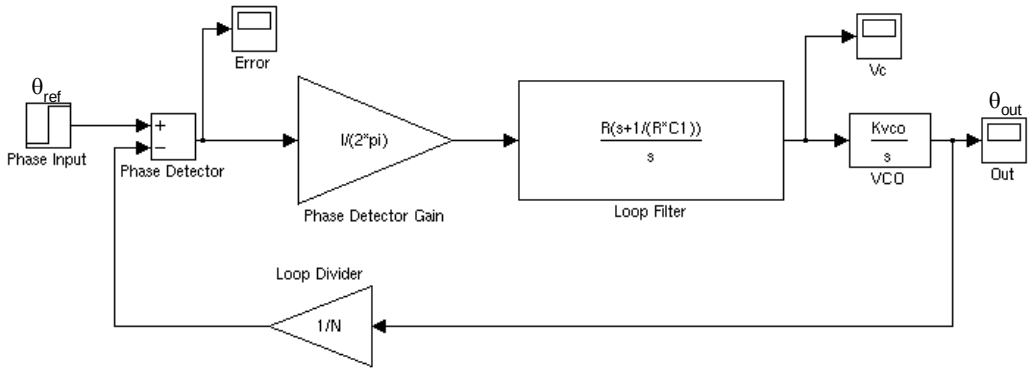
$$C_1 = \frac{(25\mu A)\left(255\frac{Mrad}{sV}\right)}{2\pi(32)\left(714\frac{krad}{s}\right)^2} = 62.2\,pF$$

- Step 7 – Determine Filter R and Secondary Cap

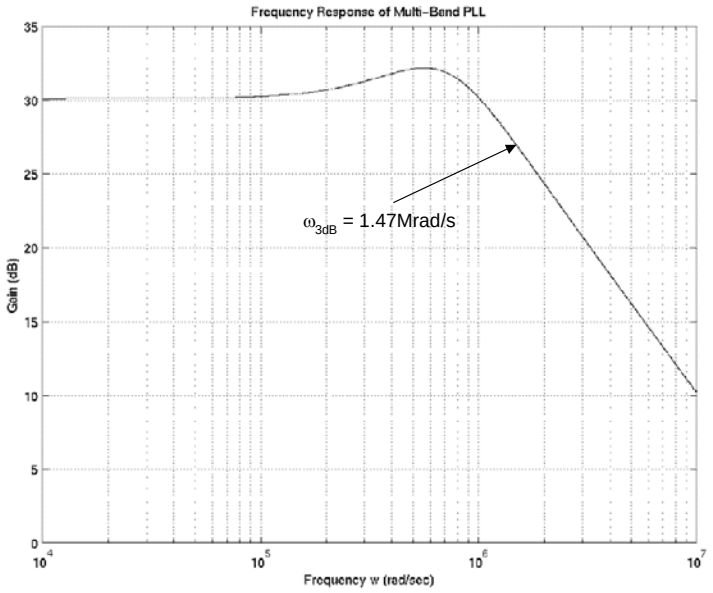
$$R = \frac{2\zeta}{\omega_n C_1} = \frac{2(0.707)}{\left(714\frac{krad}{s}\right)(62.2\,pF)} = 31.8k\Omega$$

$$C_2 < \frac{C_1}{10} = 6.22\,pF \Rightarrow C_2 = 6\,pF$$

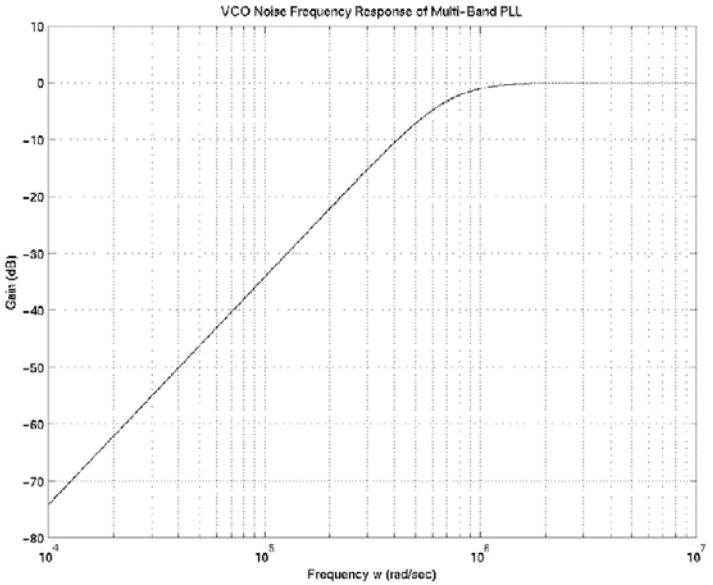
PLL Linear Phase Model



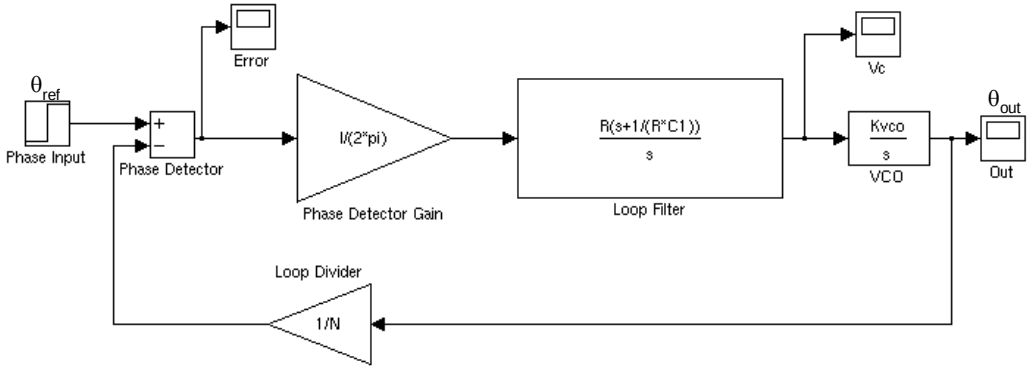
$20\log_{10}\left|\frac{\theta_{out}(s)}{\theta_{ref}(s)}\right|$

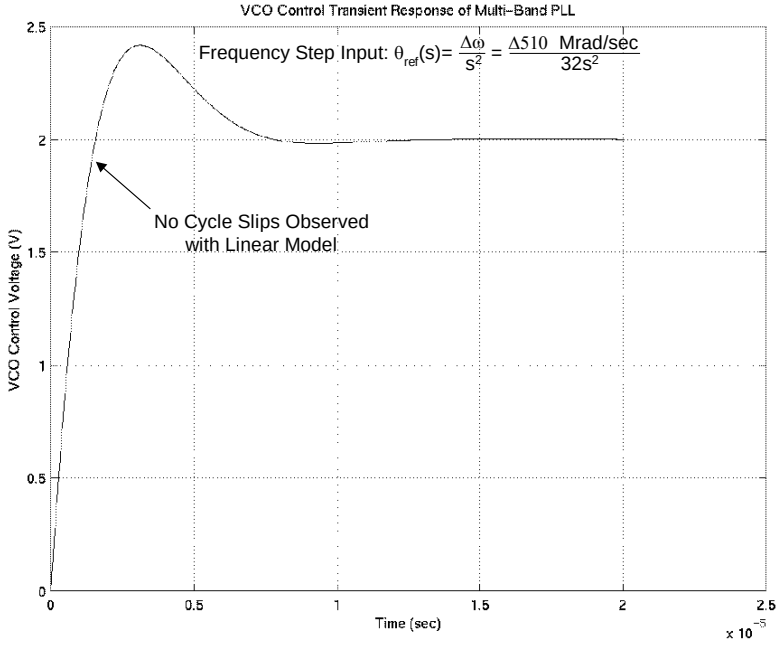


$20\log_{10}\left|\frac{\theta_{out}(s)}{\theta_{vcon}(s)}\right|$



PLL Linear Phase Model: Frequency Step Response





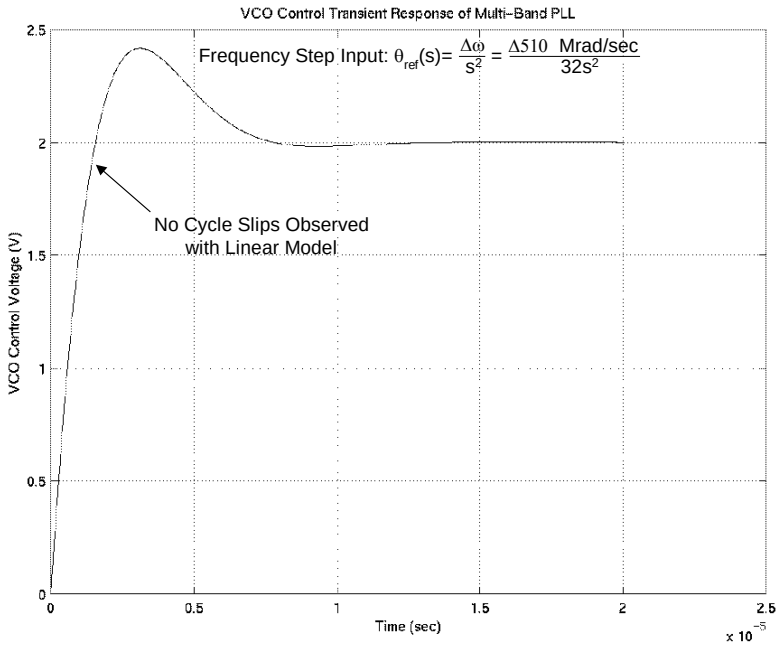
PLL Behavioral Model

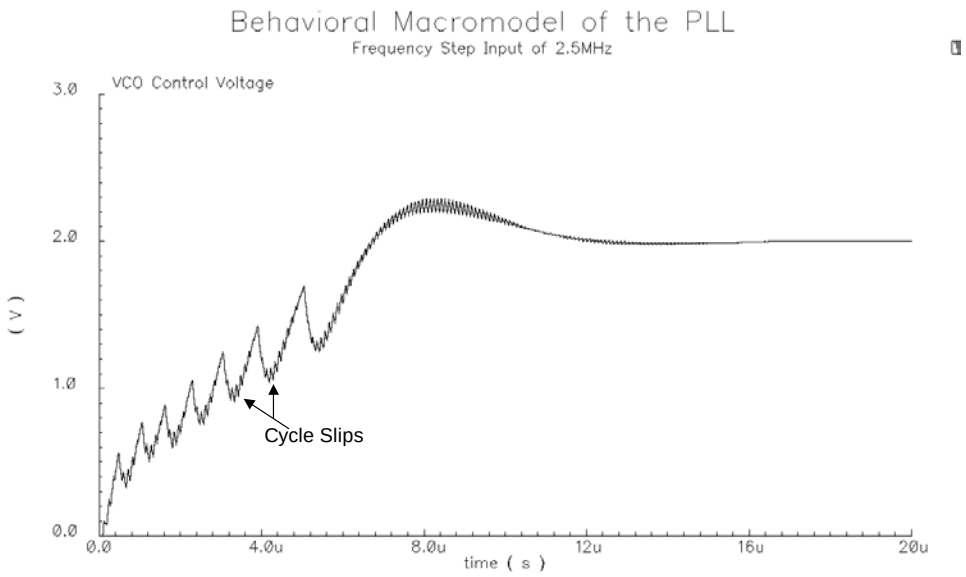
- From my MSc thesis: http://www.ece.tamu.edu/~spalermo/docs/msc_thesis_sam_palermo.pdf
- Written in SpectreHDL
- Also look at CppSim: http://www.cppsim.com/

```
// 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 0 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=10
// Voltage Controlled Oscillator
//xvco vd out vco (gain=40e6 fc=256e6)
xvco 0 vd out nv_temp vd_gnd
+ switch_vco (u=0.8 d=-0.8 gain=40e6 fc=256e6)
// Divider
xdivider 0 out fvco buffer n_temp divider (divisor=32)
op dc
```

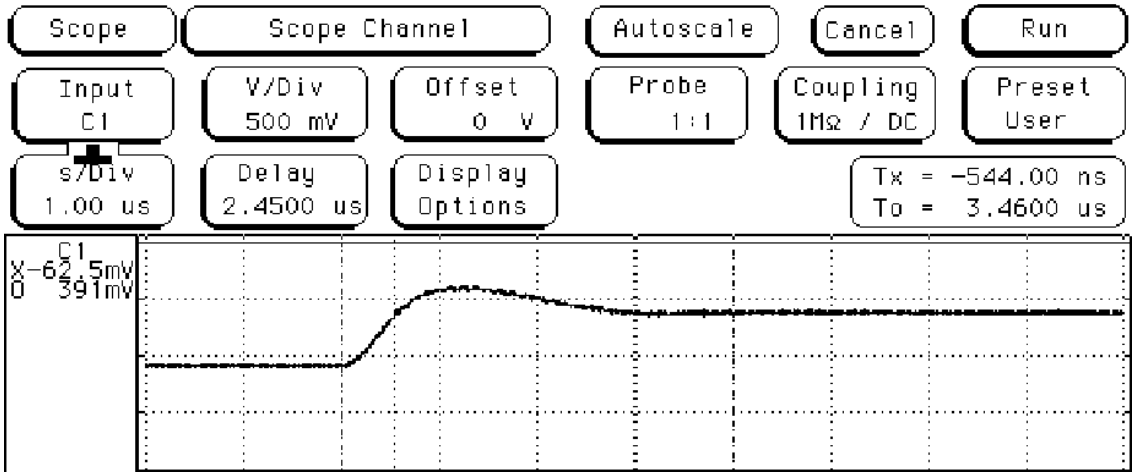
```
timedom tran stop=20u step=20p ic=all maxstep=20p skipdc=yes relref=alllocal
simulator lang=spice
.ic vd=0
save vd control fref fvco nv_temp vd_gnd
.OPTIONS rawfmt=psfbin save=selected diagnose=yes vabstol=.01
+ reltol=.99
********************************************************************************
// 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 r dff
xdffdown gnd dd fvco down downbar r dff
xnand gnd up down r nand
ends dig_pfd
********************************************************************************
// 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;
```

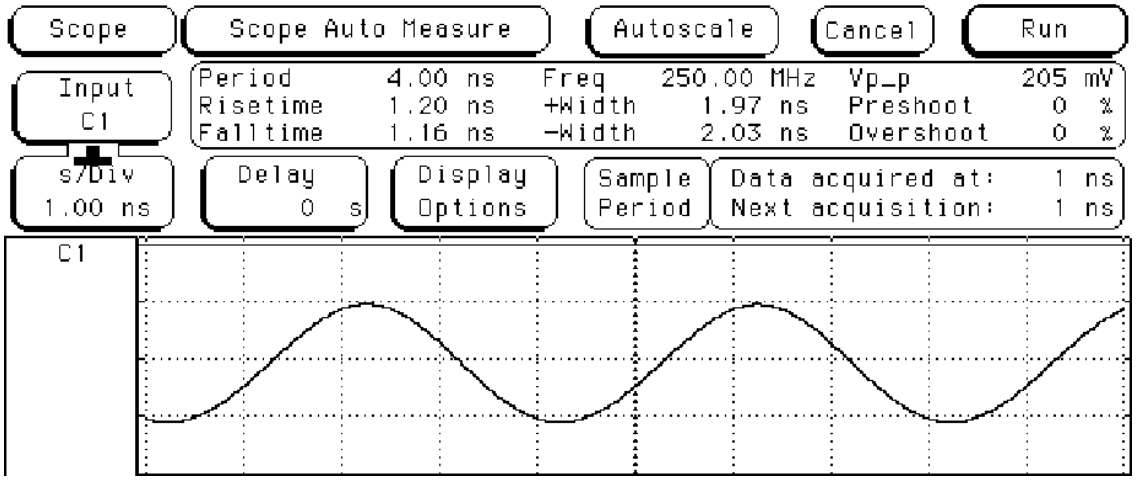
PLL Frequency Step Response: Linear vs Behavioral Model



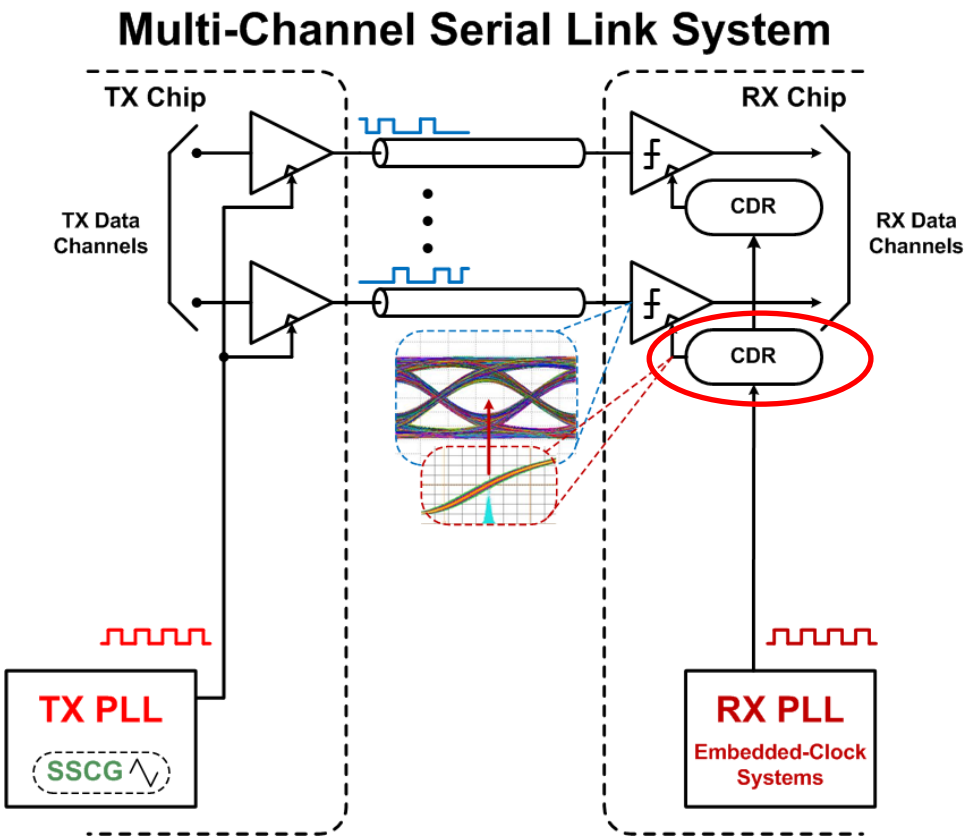


PLL Lab Results



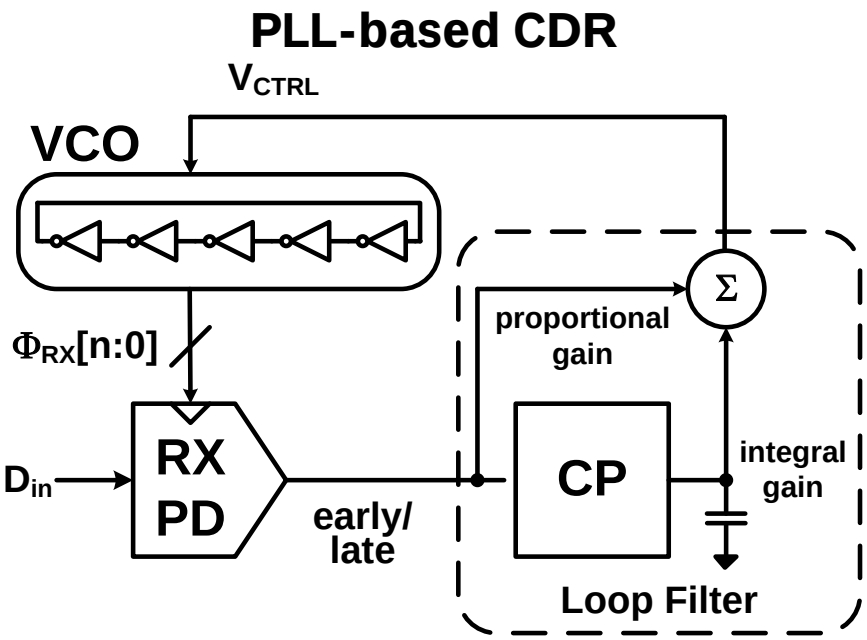


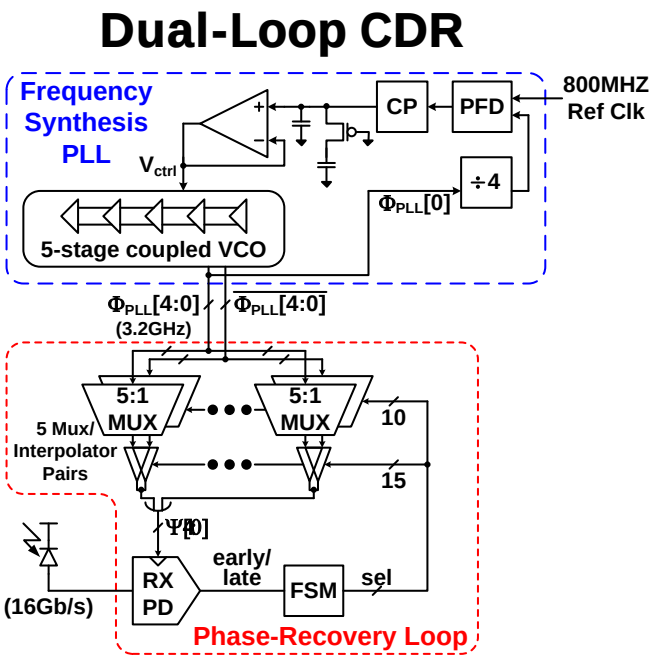
Embedded Clock I/O Circuits



- 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)





- 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

Next Time

- CDRs