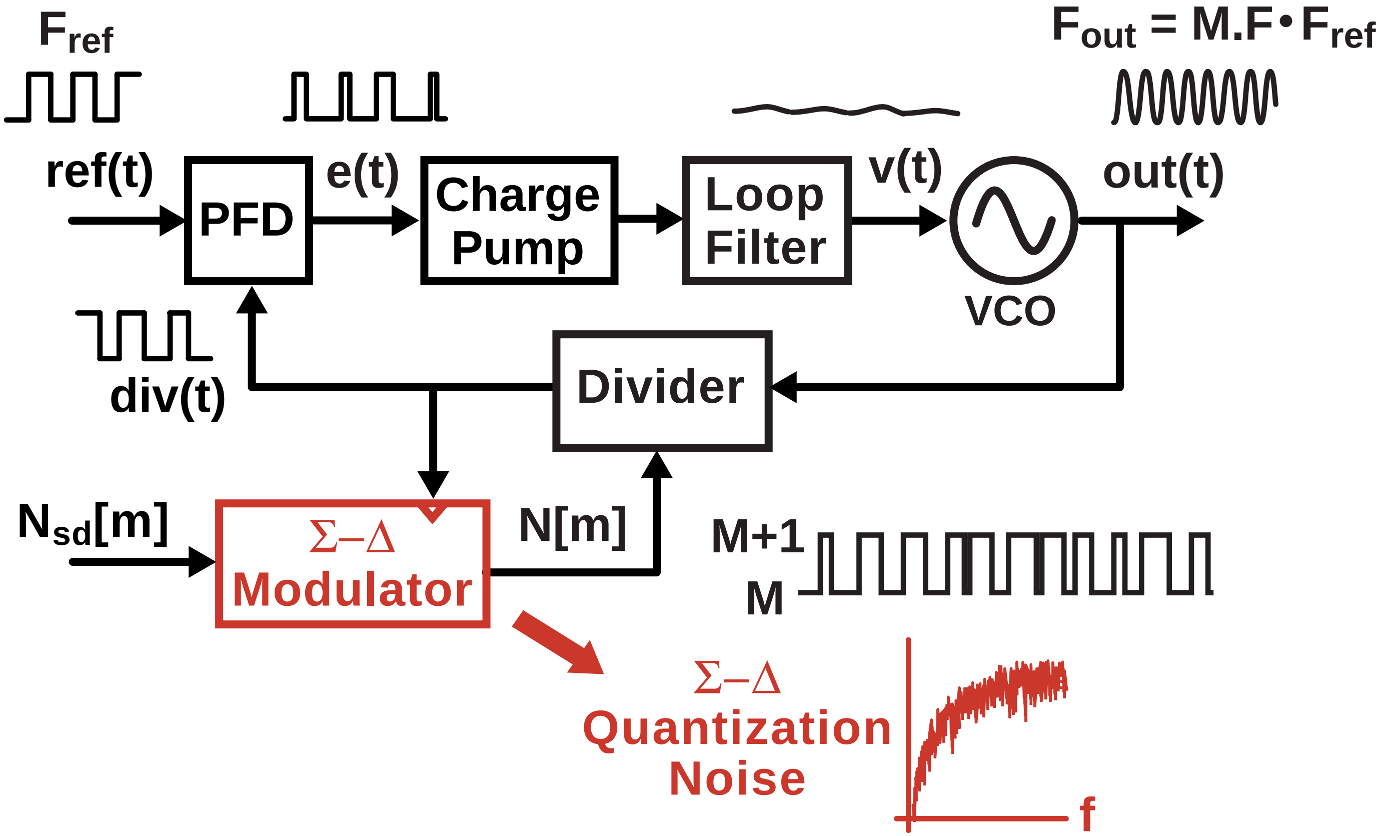
Short Course On Phase-Locked Loops and Their Applications Day 2, PM Lecture

Basic Building Blocks (Part II) High Speed Frequency Dividers, Phase Detectors, Charge Pumps, and Loop Filter Design

Michael Perrott

August 12, 2008

Fractional-N Frequency Synthesis



- Challenging building blocks
  - VCO
  - Divider
  - Charge Pump (and PFD)

Outline of Talk

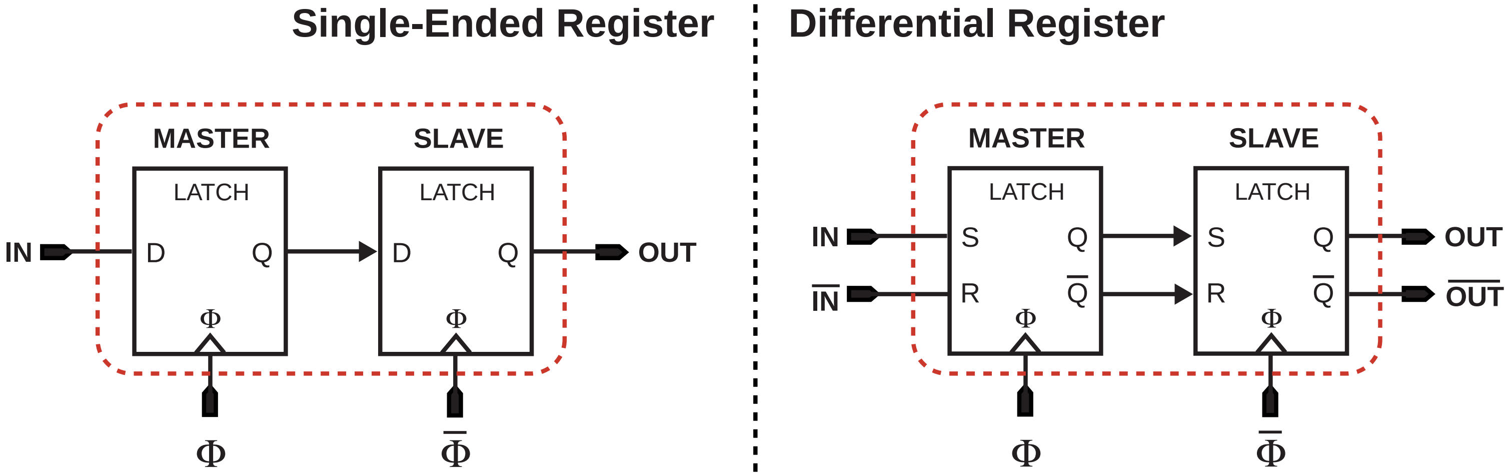
- **High speed frequency dividers**
  - **Background of key digital building blocks**

- **PFD and Charge Pumps**

- **Loop filter design**
  - **Closed loop PLL design using CAD**

Digital Background for Dividers

Edge-triggered Registers

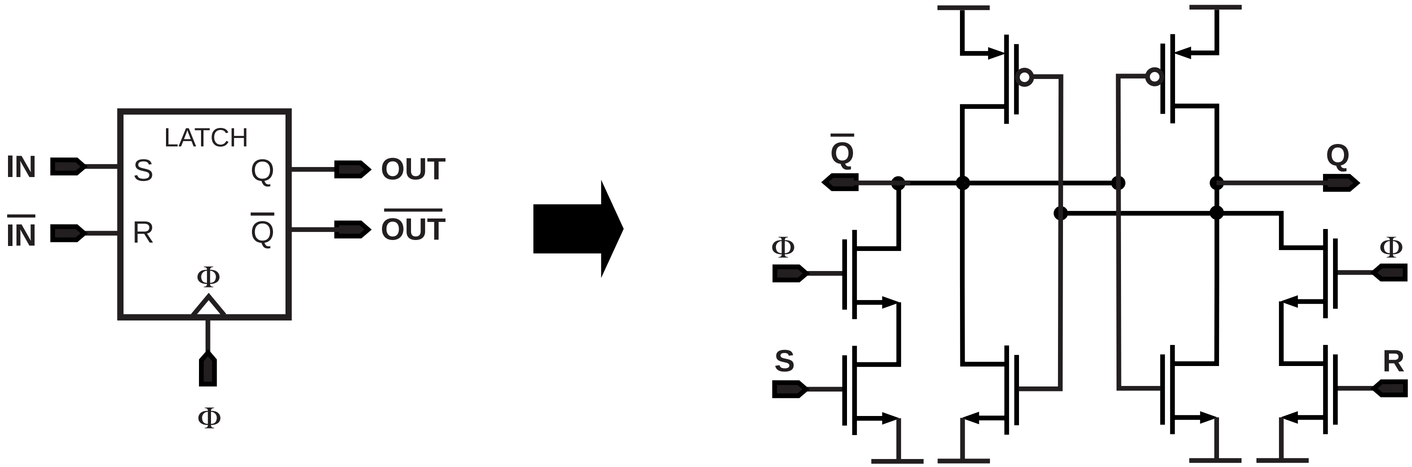


Single-Ended Register

Differential Register

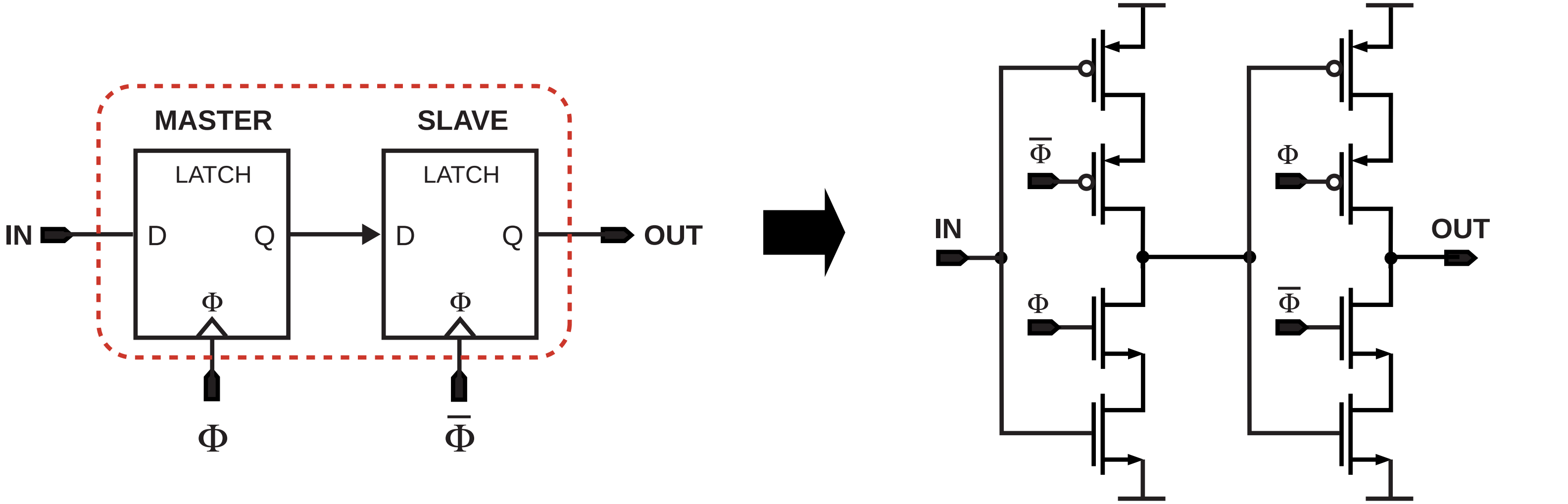
- **Achieved by cascading two latches that are transparent out of phase from one another**
- **Two general classes of latches**
  - **Static – employ positive feedback**
    - Robust
  - **Dynamic – store charge on parasitic capacitance**
    - Smaller, lower power in most cases
    - Negative: must be refreshed (due to leakage currents)

Static Latches



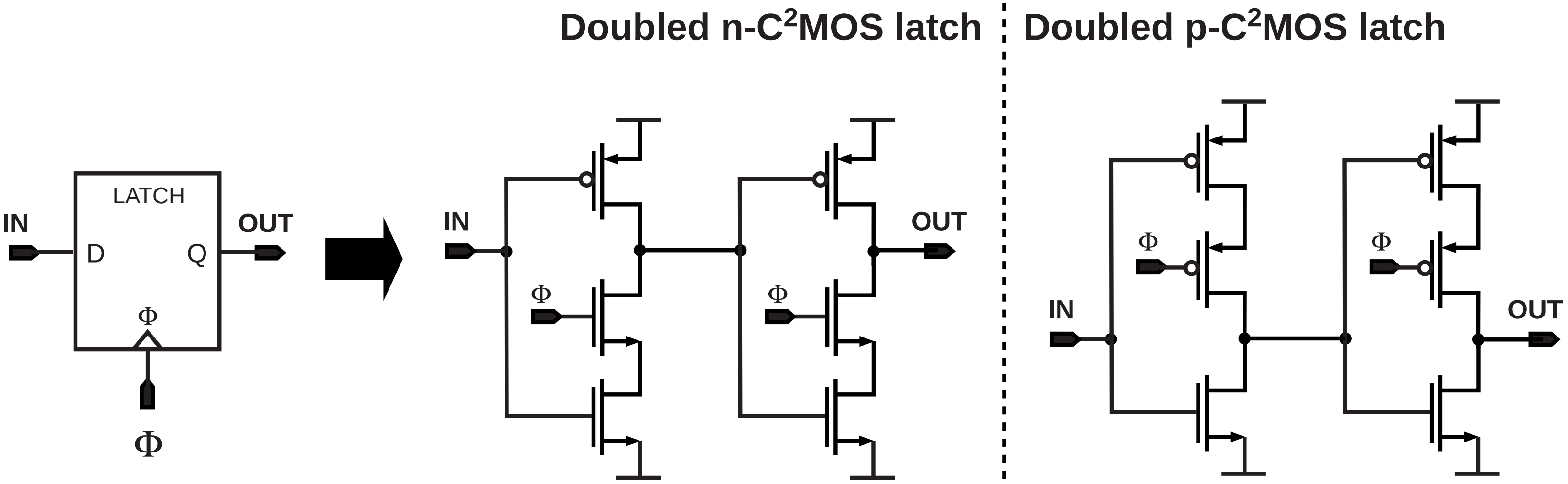
- **Classical case employs cross-coupled NAND/NOR gates to achieve positive feedback**
- **Above example uses cross-coupled inverters for positive feedback**
  - Set, reset, and clock transistors designed to have enough drive to overpower cross-coupled inverters
  - Relatively small number of transistors
  - Robust

Dynamic Latches



- **Leverage CMOS technology**
  - **High quality switches with small leakage available**
  - **Can switch in and store charge on parasitic capacitances quite reliability**
- **Achieves faster speed than full swing logic with fewer transistors**
- **Issues: higher sensitivity to noise, minimum refresh rate required due to charge leakage**

True Single Phase Clocked (TSPC) Latches



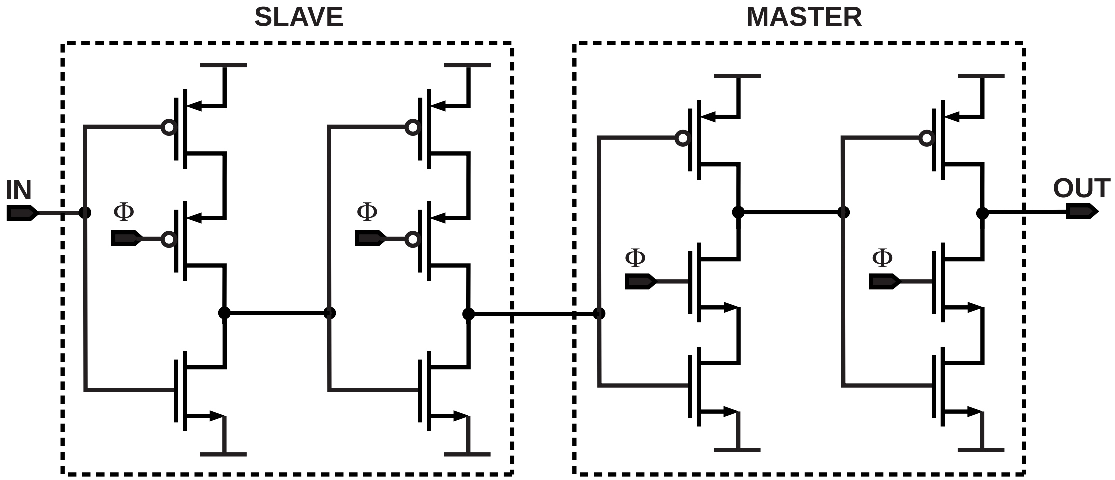
Doubled n-C²MOS latch

Doubled p-C²MOS latch

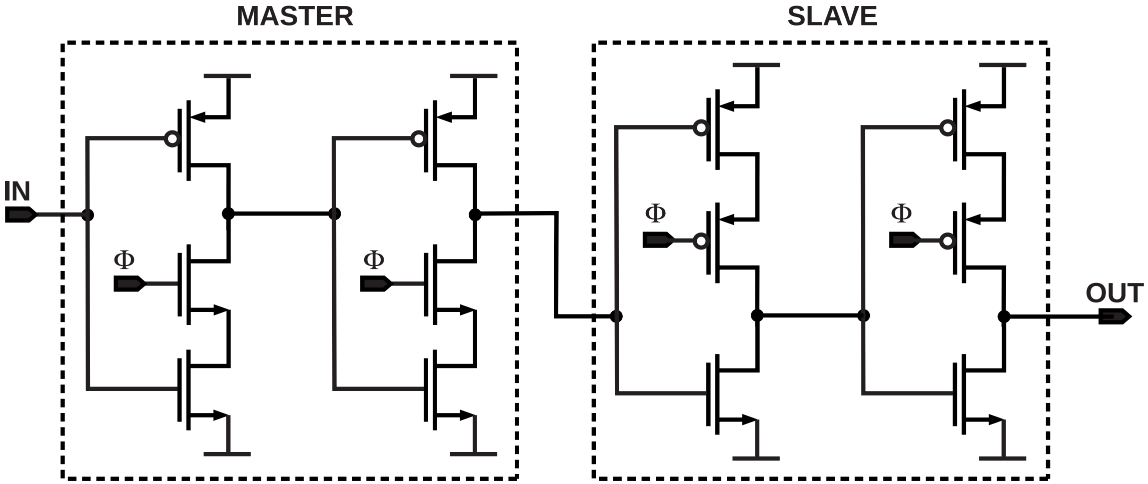
- **Allow register implementations with only one clock!**
  - **Latches made transparent at different portions of clock cycle by using appropriate latch “flavor” – n or p**
    - n latches are transparent only when Φ is 1
    - p latches are transparent only when Φ is 0
- **Benefits: simplified clock distribution, high speed**

Example TSPC Registers

- Positive edge-triggered version

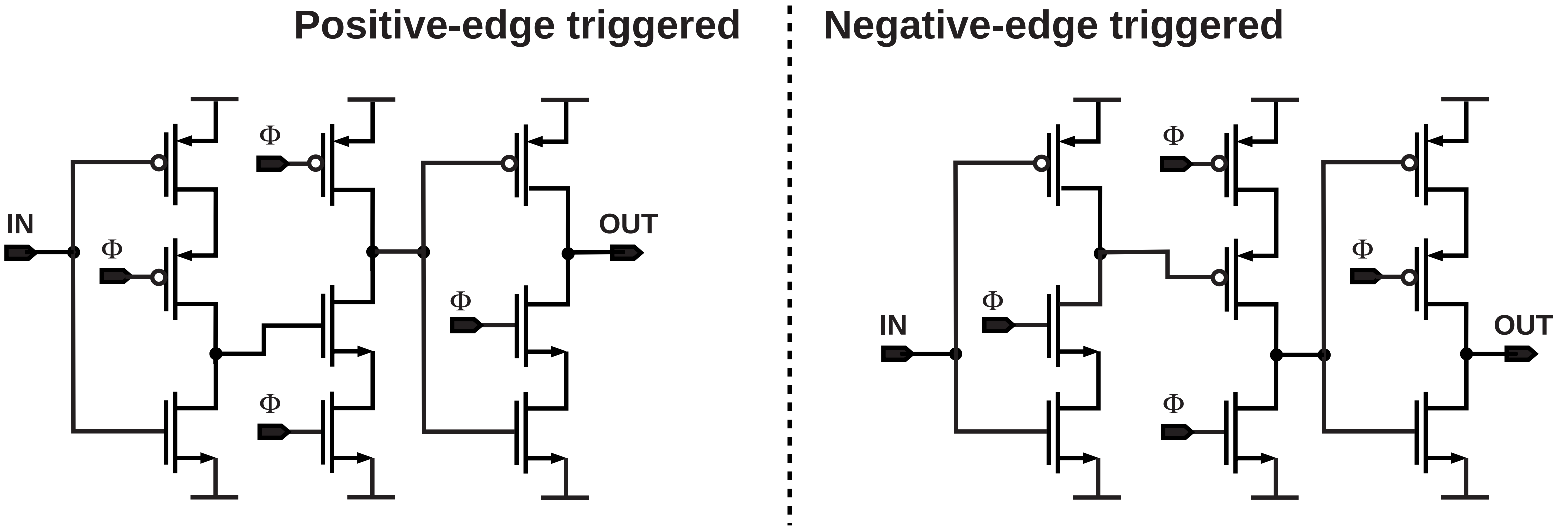


- Negative edge-triggered version



A Simplified Approach to TSPC Registers

- **Clever implementation of TSPC approach can be achieved with reduced transistor count**

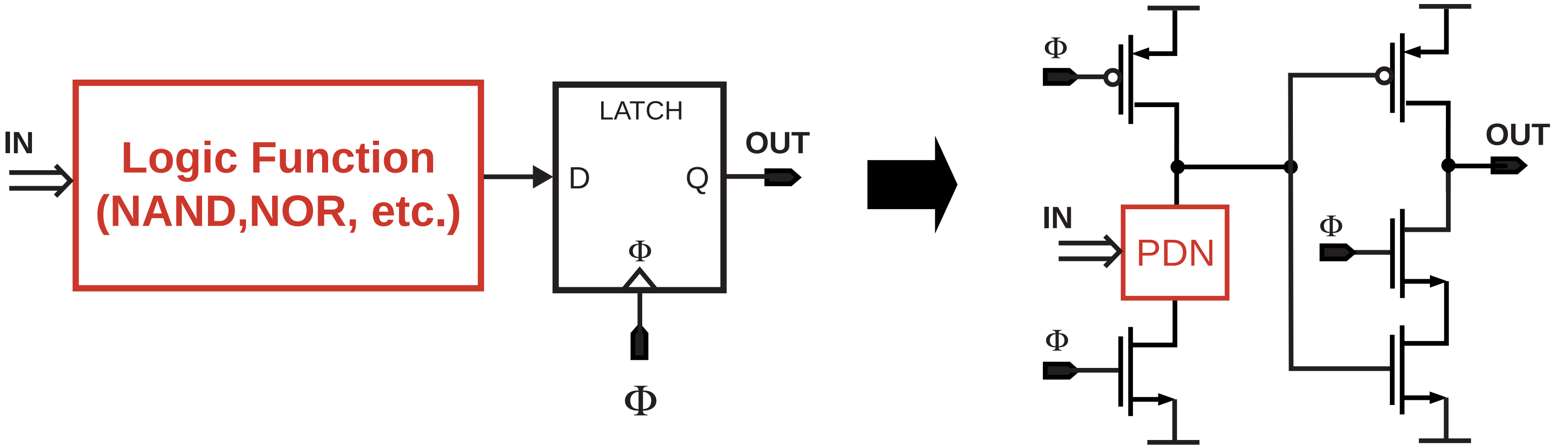


Positive-edge triggered

Negative-edge triggered

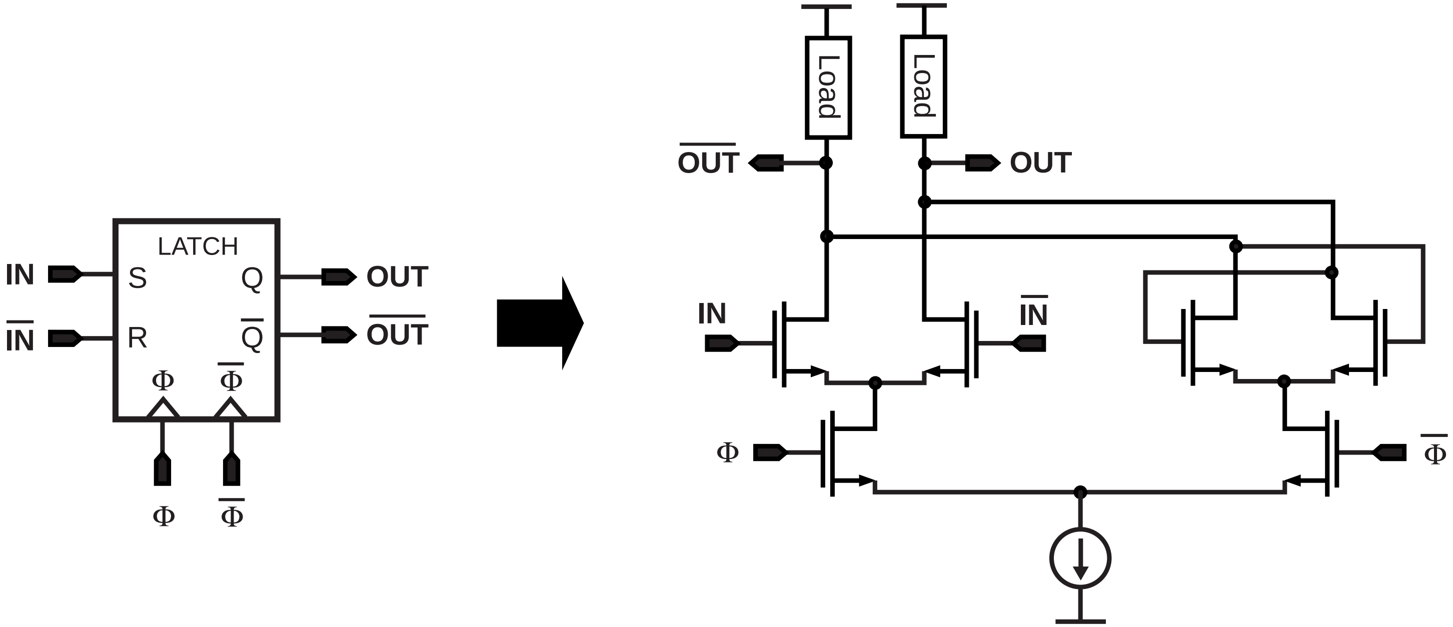
- **For more info on TSPC approach, see**
  - **J. Yuan and C. Svensson, “New Single-Clock CMOS Latches and Flipflops with Improved Speed and Power Savings”, JSSC, Jan 1997, pp 62-69**

Embedding of Logic within Latches



- We can often increase the speed of a logic function fed into a latch through embedding
  - Latch slowed down by extra transistors, but logic/latch combination is faster than direct cascade of the functions
- Method can be applied to both static and dynamic approaches
  - Dynamic approach shown above

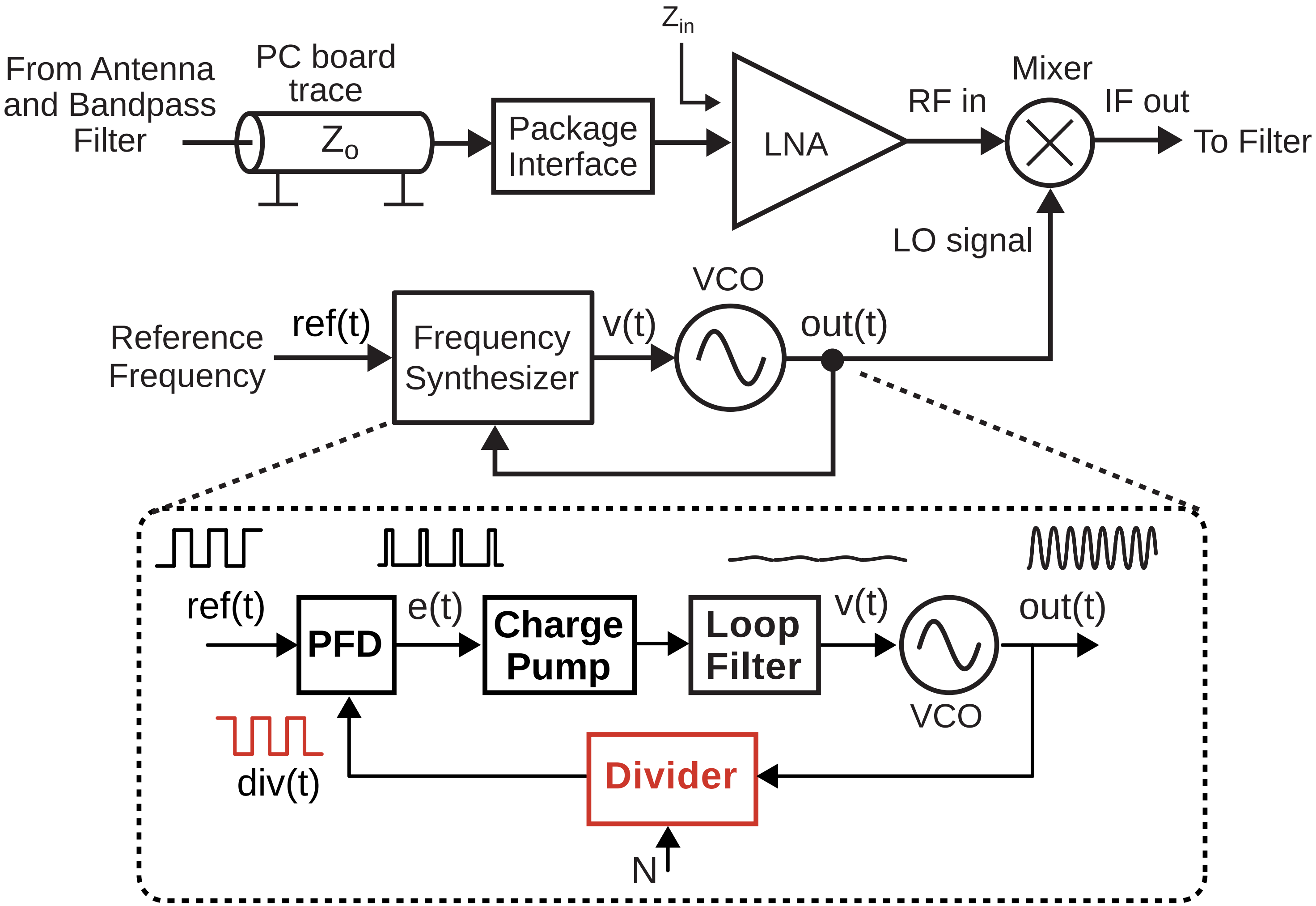
Highest Speed Achieved with Differential CML Latch



- **Employs positive feedback for memory**
  - **Realized with cross-coupled NMOS differential pair**
- **Method of operation**
  - **Follow mode: current directed through differential amplifier that passes input signal**
  - **Hold mode: current shifted to cross-coupled pair**

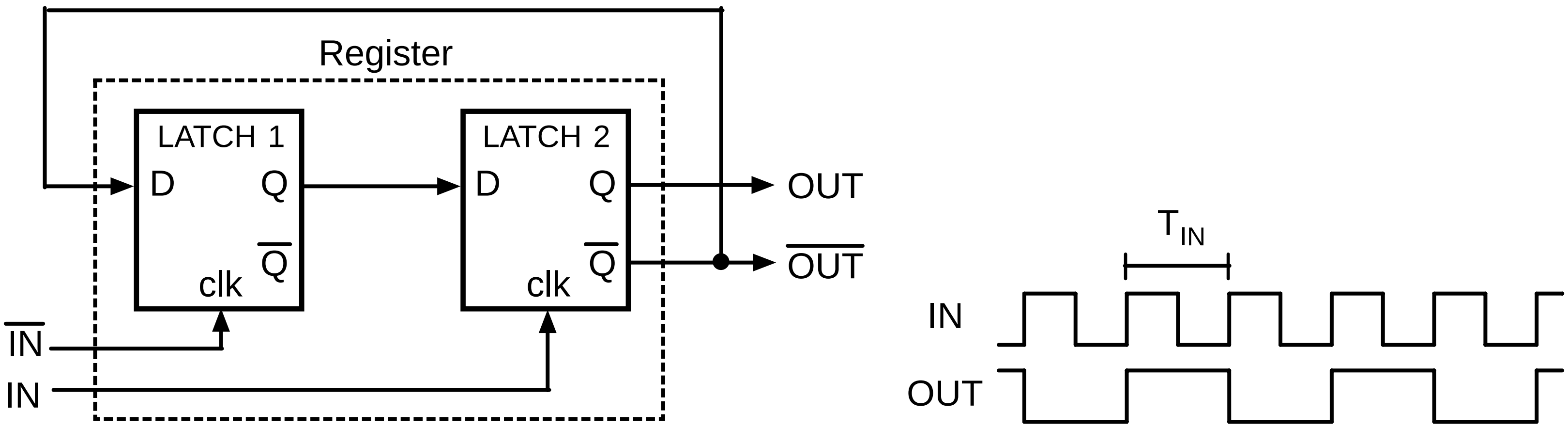
High Speed Frequency Dividers

High Speed Frequency Dividers in Wireless Systems



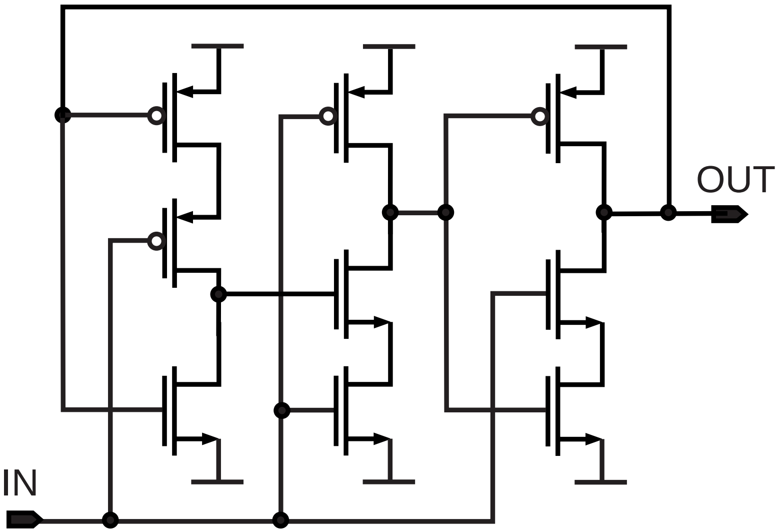
- **Design Issues: high speed, low power**

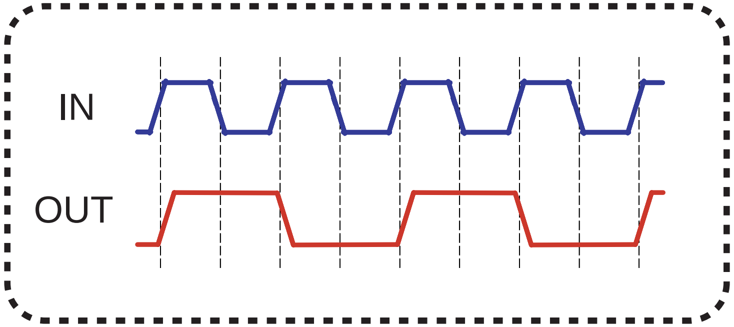
Divide-by-2 Circuit (Johnson Counter)



- **Achieves frequency division by clocking two latches (i.e., a register) in negative feedback**
- **Latches may be implemented in various ways according to speed/power requirements**

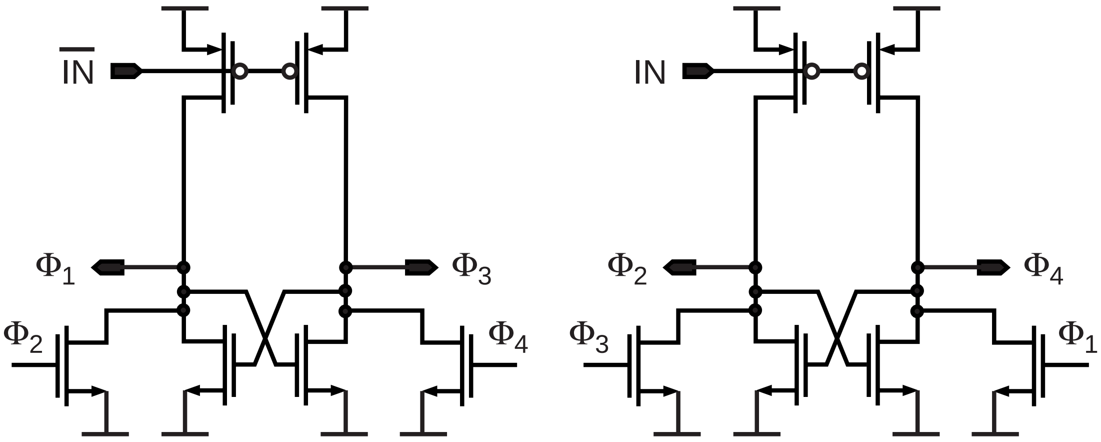
Divide-by-2 Using a TSPC register

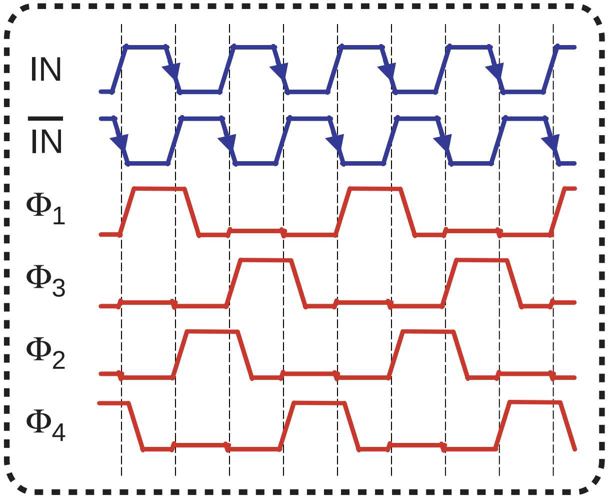




- **Advantages**
  - **Reasonably fast, compact size**
  - **No static power dissipation, differential clock not required**
- **Disadvantages**
  - **Slowed down by stacked PMOS, signals goes through three gates per cycle**
  - **Requires full swing input clock signal**

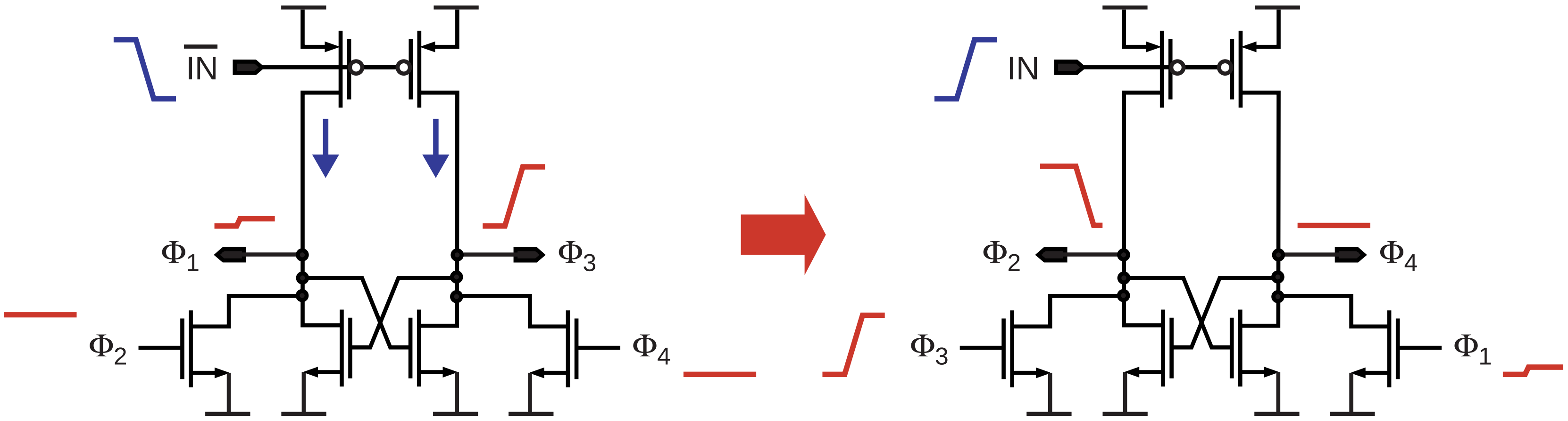
Divide-by-2 Using Razavi's Topology





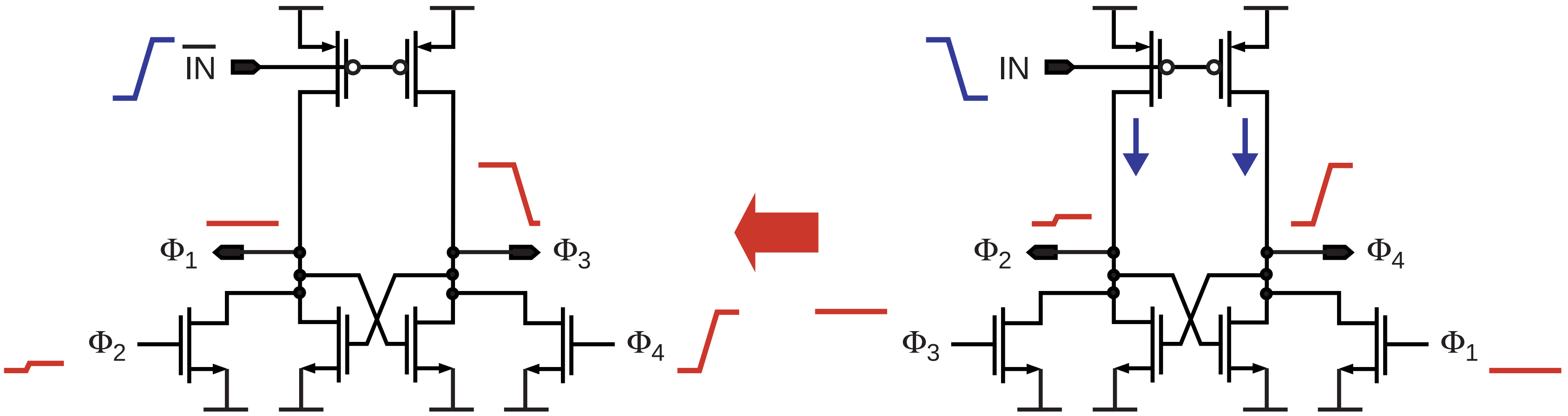
- **Faster topology than TSPC approach**
- **See B. Rezavi et. al., "Design of High Speed, Low Power Frequency Dividers and Phase-Locked Loops in Deep Submicron CMOS", JSSC, Feb 1995, pp 101-109**

Explanation of Razavi Divider Operation (Part 1)



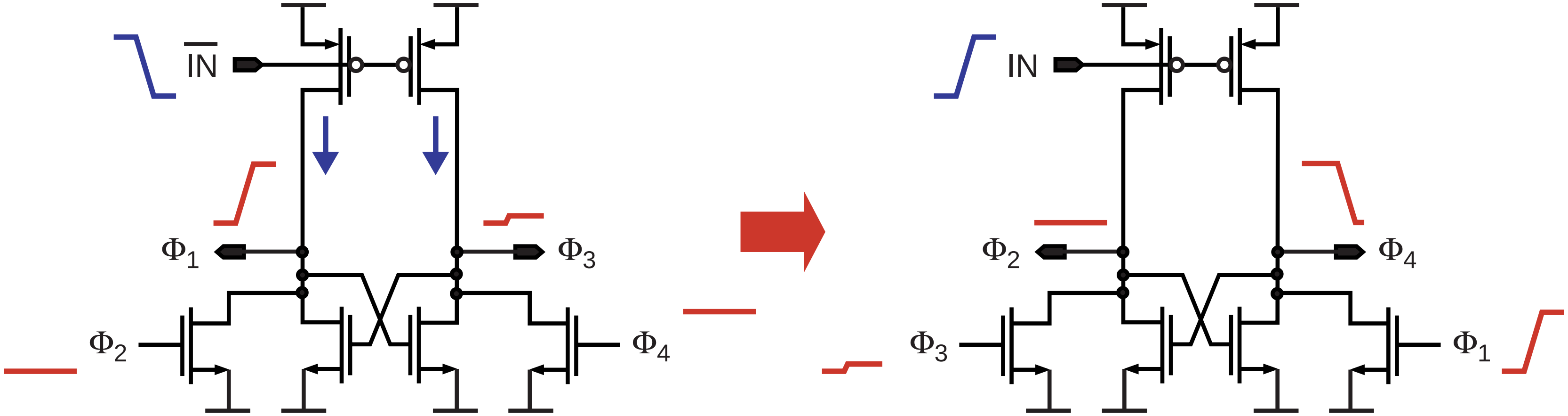
- **Left latch:**
  - **Clock drives current from PMOS devices of a given latch onto the NMOS cross-coupled pair**
  - **Latch output voltage rises asymmetrically according to voltage setting on gates of outside NMOS devices**
- **Right latch:**
  - **Outside NMOS devices discharge the latch output voltage as the left latch output voltage rises**

Explanation of Razavi Divider Operation (Part 2)



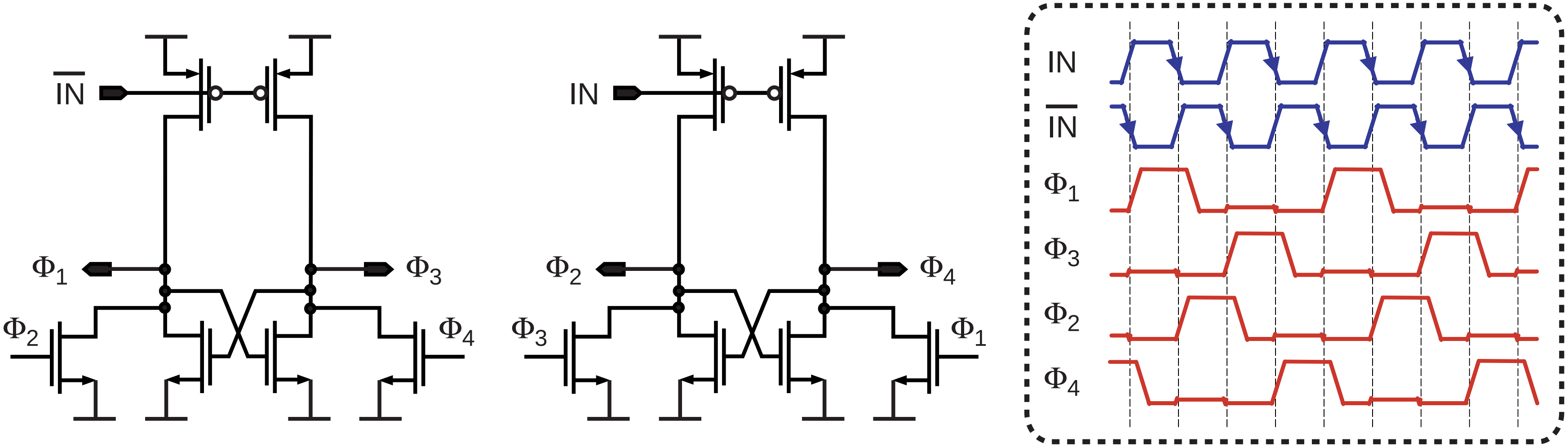
- **Right latch:**
  - **Clock drives current from PMOS devices of a given latch onto the NMOS cross-coupled pair**
  - **Latch output voltage rises asymmetrically according to voltage setting on gates of outside NMOS devices**
- **Left latch:**
  - **Outside NMOS devices discharge the latch output voltage as the left latch output voltage rises**

Explanation of Razavi Divider Operation (Part 3)



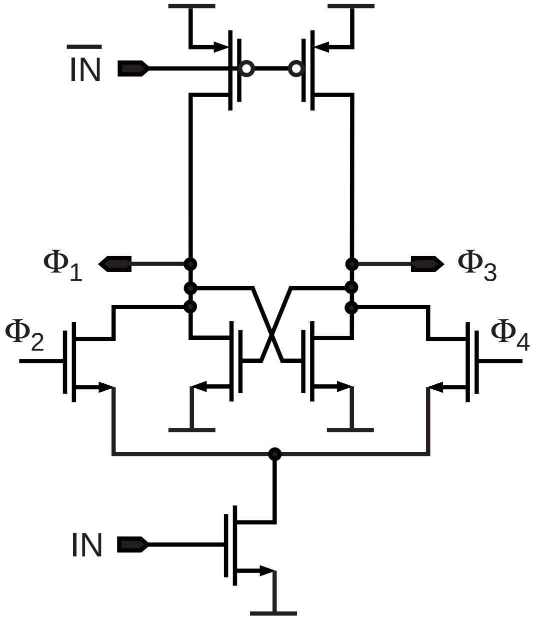
- **Process starts over again with current being driven into left latch**
  - **Voltage polarity at the output of the latch has now flipped**

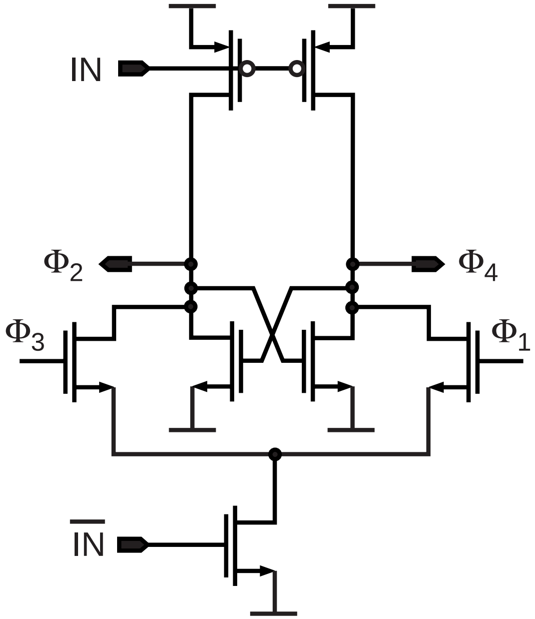
Advantages and Disadvantages of Razavi Topology

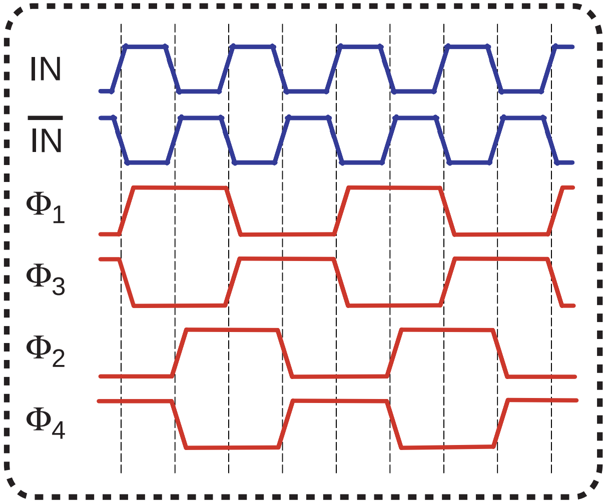


- **Advantages**
  - **Fast – no stacked PMOS, signal goes through only two gates per cycle**
- **Disadvantages**
  - **Static power**
  - **Full swing, differential input clock signal required**
- **Note: quarter period duty cycle can be turned into fifty percent duty cycle with OR gates after the divider**
  - **See my thesis at http://www-mtl.mit.edu/~perrott**

Divide-by-2 Using Wang Topology

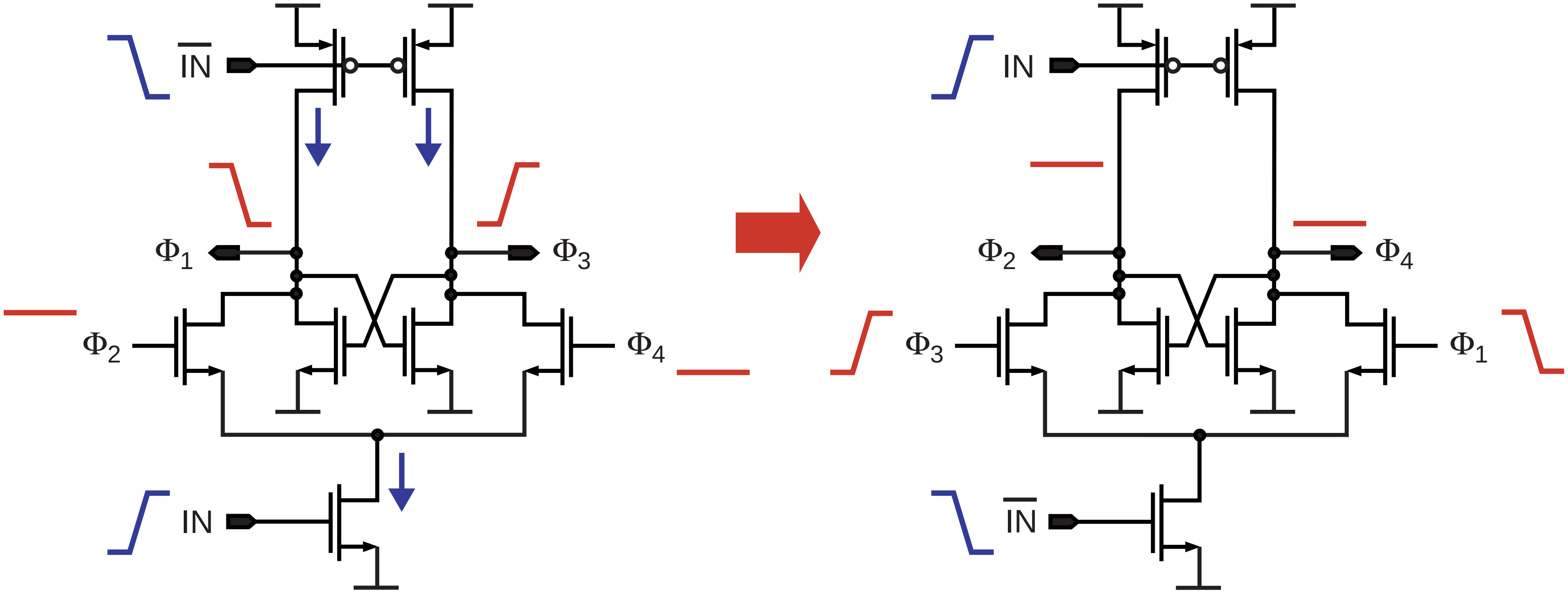






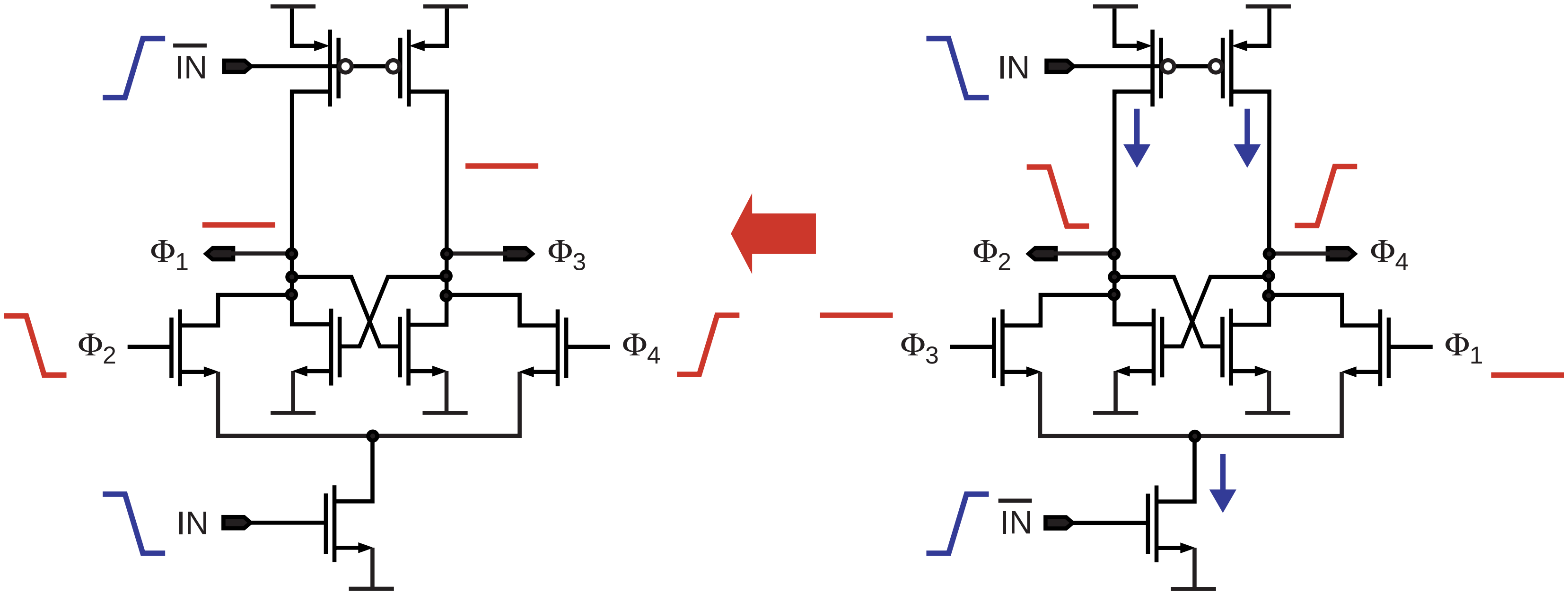
- **Claims to be faster than Razavi topology**
  - **Chief difference is addition of NMOS clock devices and different scaling of upper PMOS devices**
- **See HongMo Wang, “A 1.8 V 3 mW 16.8 GHz Frequency Divider in 0.25μm CMOS”, ISSCC 2000, pp 196-197**

Explanation of Wang Topology Operation (Part 1)



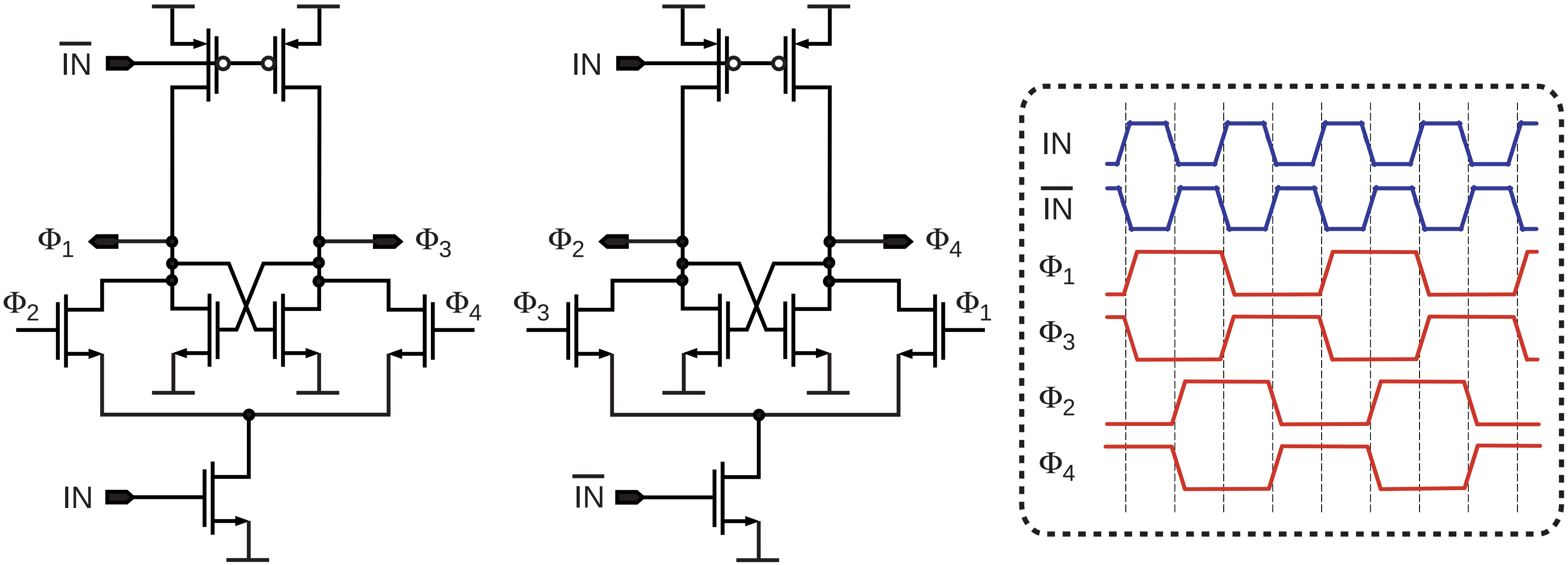
- **Left latch**
  - **Current driven into latch and output voltage responds similar to Razavi architecture**
- **Right latch**
  - **Different than Razavi architecture in that latch output voltage is *not* discharged due to presence of extra NMOS**

Explanation of Wang Topology Operation (Part 2)



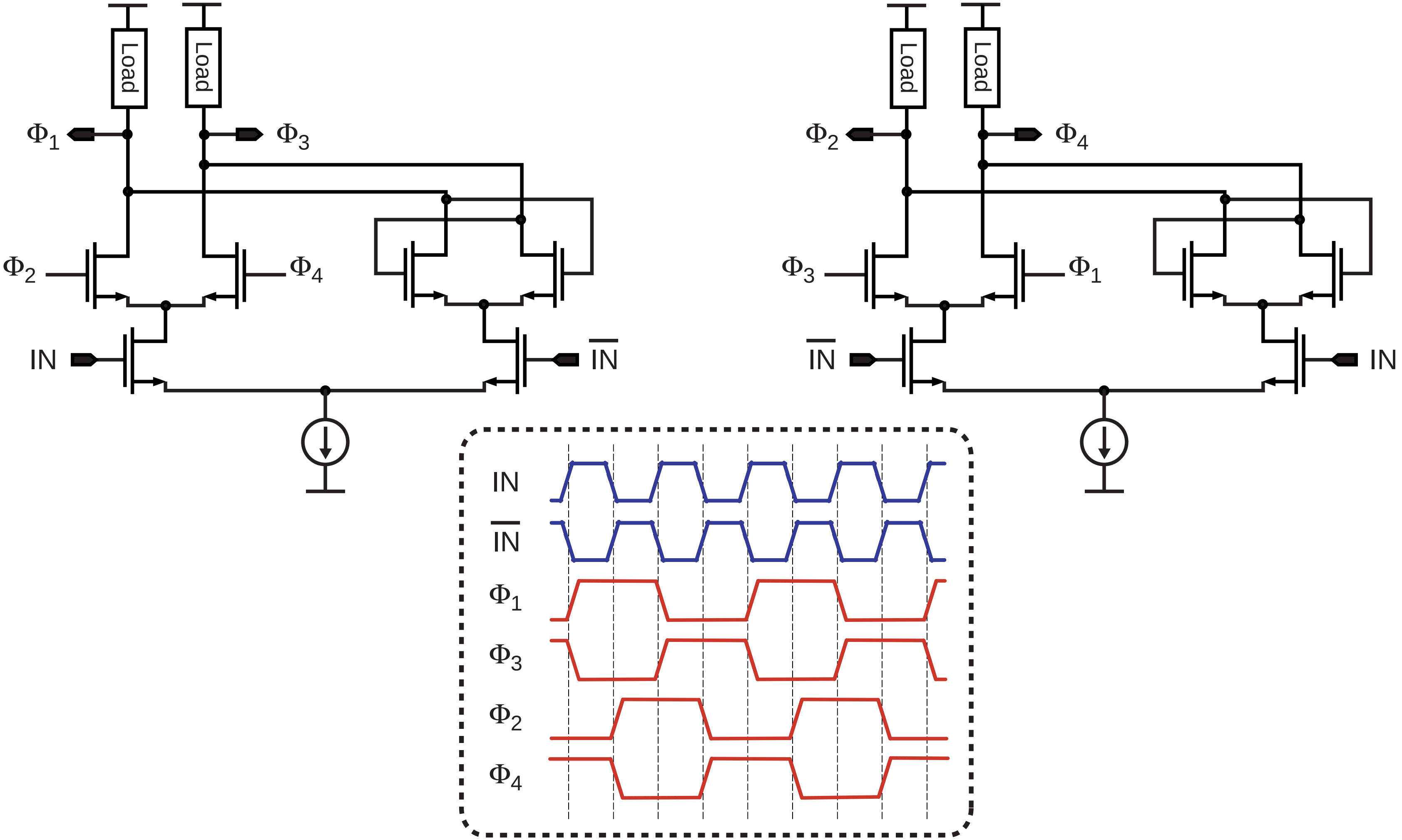
- **Same process repeats on the right side**
  - **The left side maintains its voltages due to presence of NMOS device**

Advantages and Disadvantages of Wang Topology



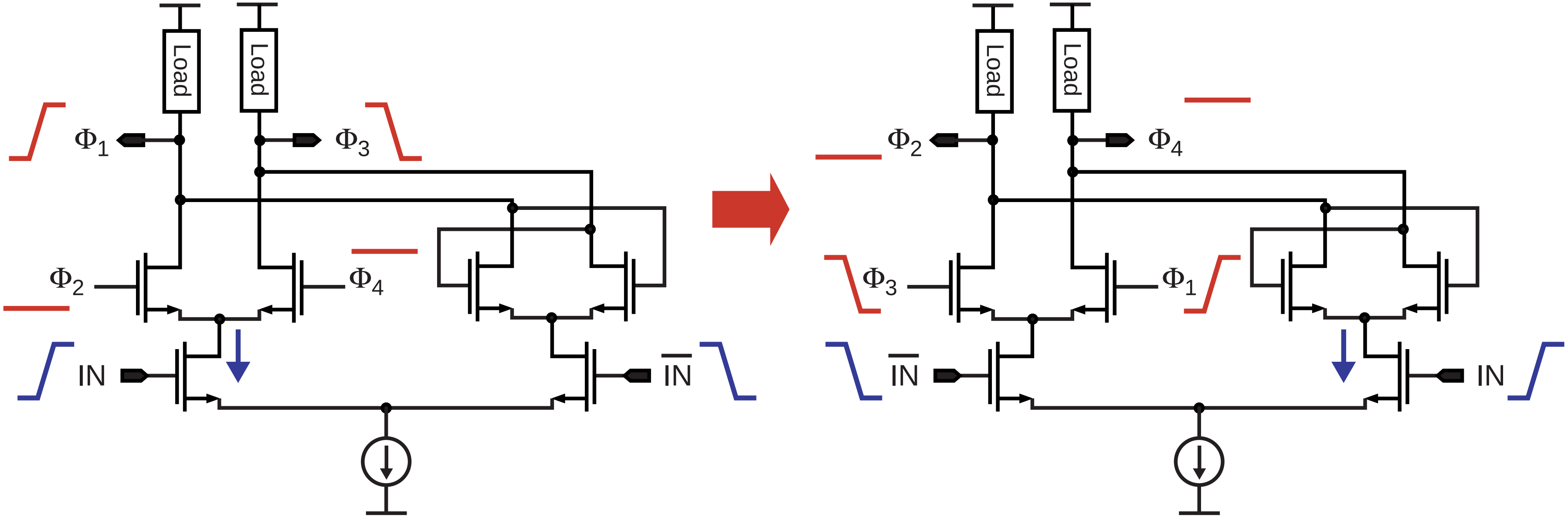
- **Advantages**
  - **Fast – no stacked PMOS, signal goes through only two gates per cycle**
- **Disadvantages**
  - **Static power**
  - **Full swing, differential input clock signal required**

Divide-by-2 Using CML Latches



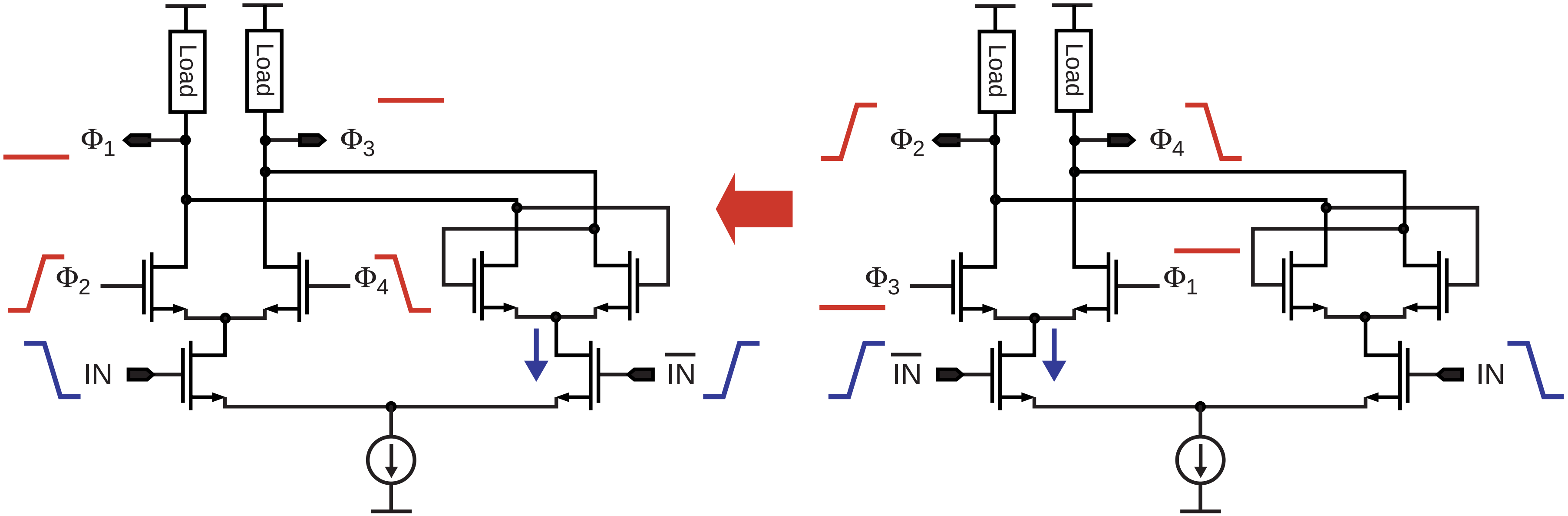
- **Fastest structure uses resistors for load**

Explanation of CML Topology Operation (Part 1)



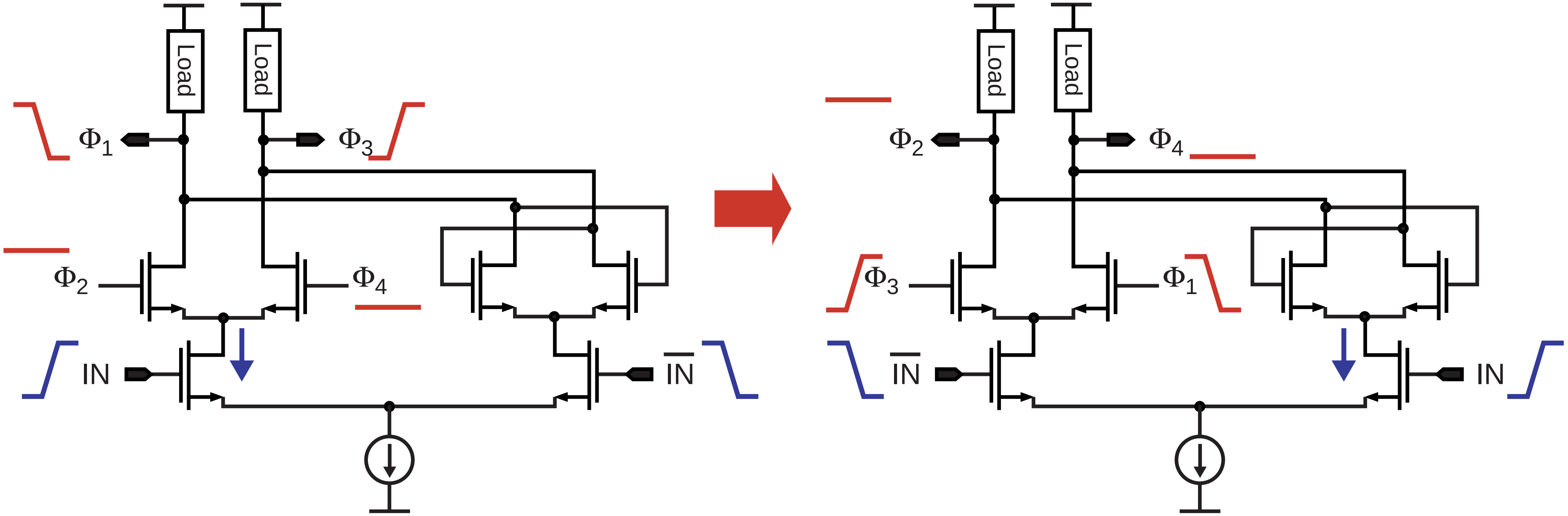
- **Left latch**
  - **Current directed into differential amp portion of latch**
    - Latch output follows input from right latch
- **Right latch**
  - **Current directed into cross-coupled pair portion of latch**
    - Latch output is held

Explanation of CML Topology Operation (Part 2)



- **Left latch**
  - **Current is directed into cross-coupled pair**
    - Latch output voltage retained
- **Right latch**
  - **Current is directed into differential amp**
    - Latch output voltage follows input from left latch

Explanation of CML Topology Operation (Part 3)



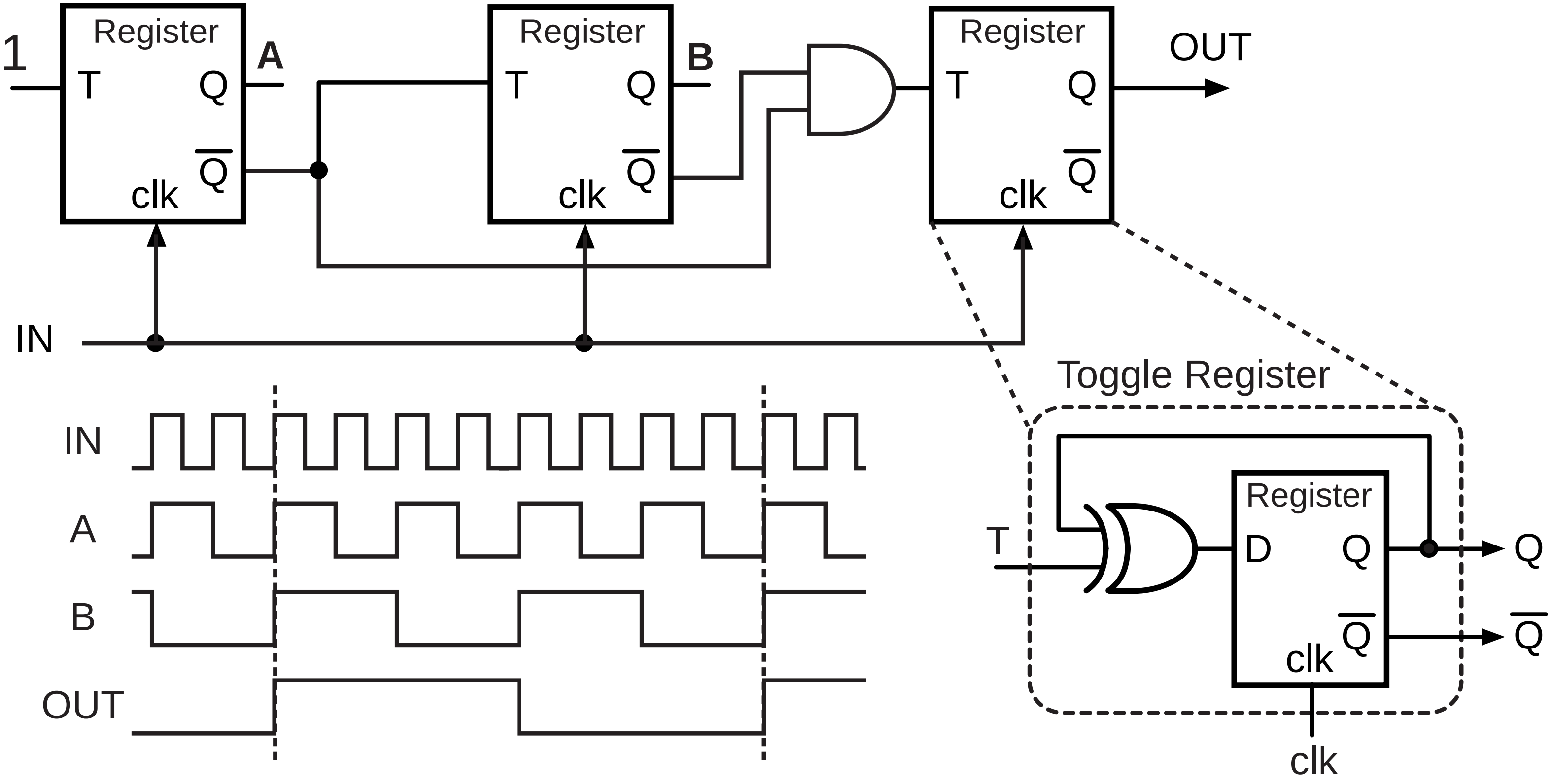
- **Same process repeats on left side**
  - **Voltage polarity is now flipped**

Advantages and Disadvantages of CML Topology

- Advantages
  - Very fast – no PMOS at all, signal goes through only two gates per cycle
  - Smaller input swing for input clock than previous approaches
    - Allows signal transitioning at higher frequencies
- Disadvantages
  - Static power
  - Differential signals required
  - Large area compared to previous approaches
  - Biasing sources required

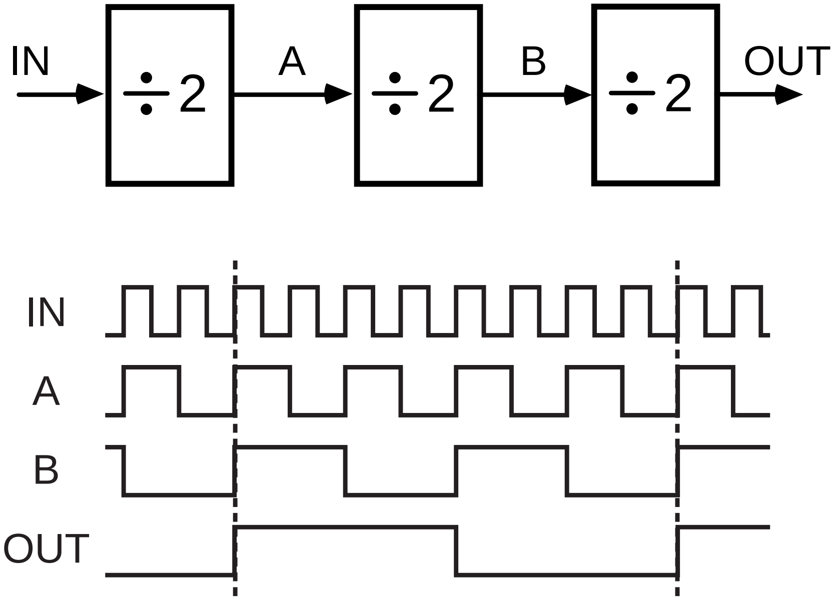
- Note: additional speedup can be obtained by using inductor peaking (i.e., place inductor in load)

Creating Higher Divide Values (Synchronous Approach)



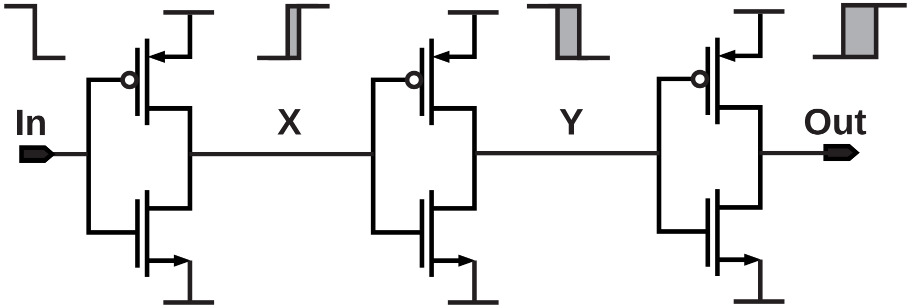
- Cascades toggle registers and logic to perform division
  - Advantage: low jitter (explained shortly)
  - Problems: high power (all registers run at high frequency), high loading on clock (IN signal drives *all* registers)

Creating Higher Divide Values (Asynchronous Approach)



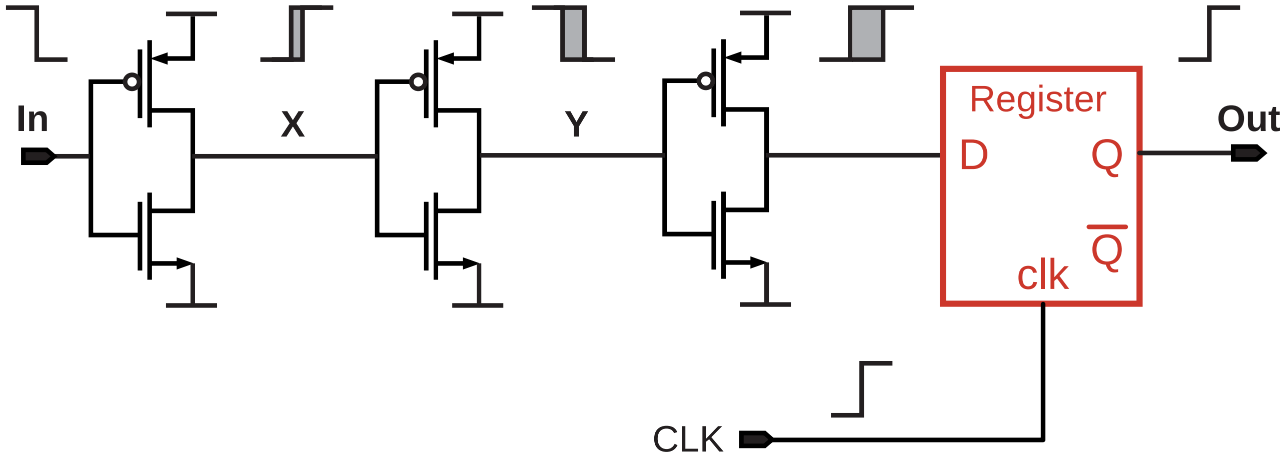
- Higher division achieved by simply cascading divide-by-2 stages
- Advantages over synchronous approach
  - Lower power: each stage runs at a lower frequency, allowing power to be correspondingly reduced
  - Less loading of input: IN signal only drives first stage
- Disadvantage: jitter is larger

Jitter in Asynchronous Designs



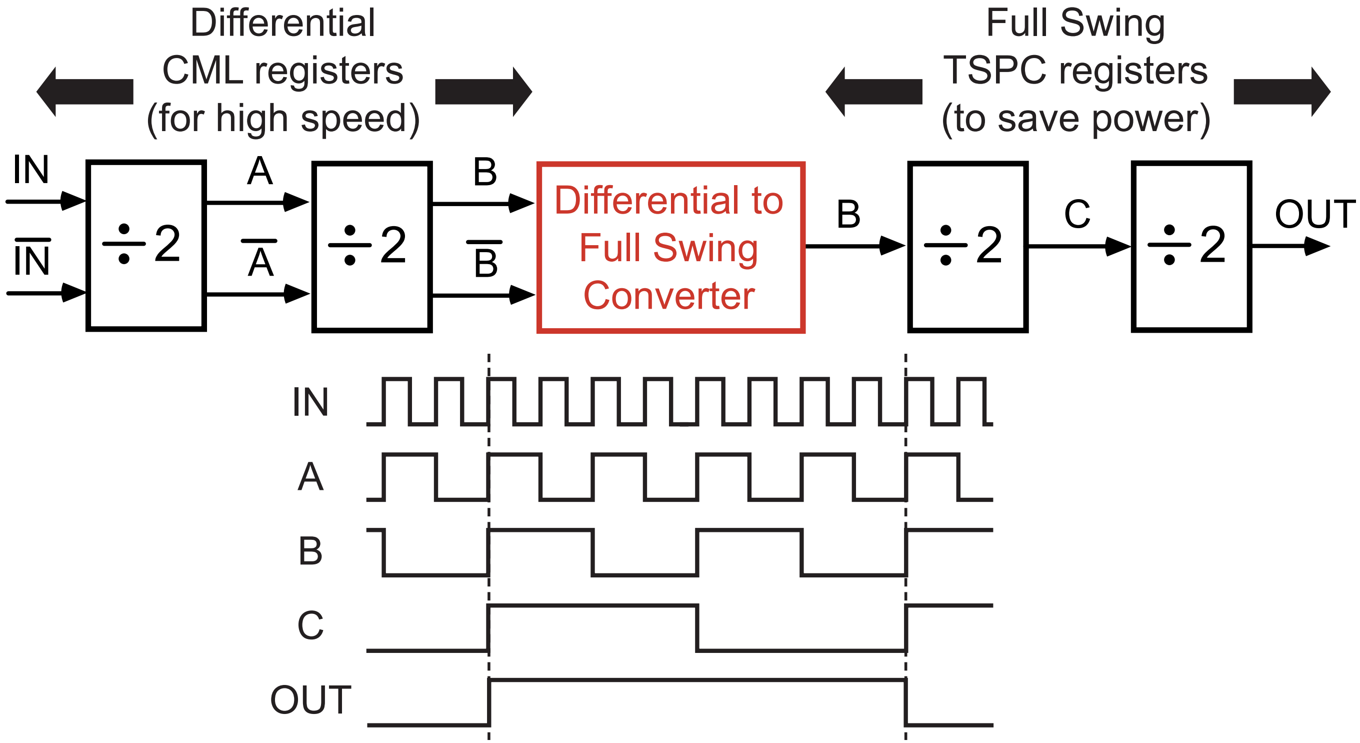
- Each logic stage adds jitter to its output
  - Jitter accumulates as it passes through more and more gates

Jitter in Synchronous Designs



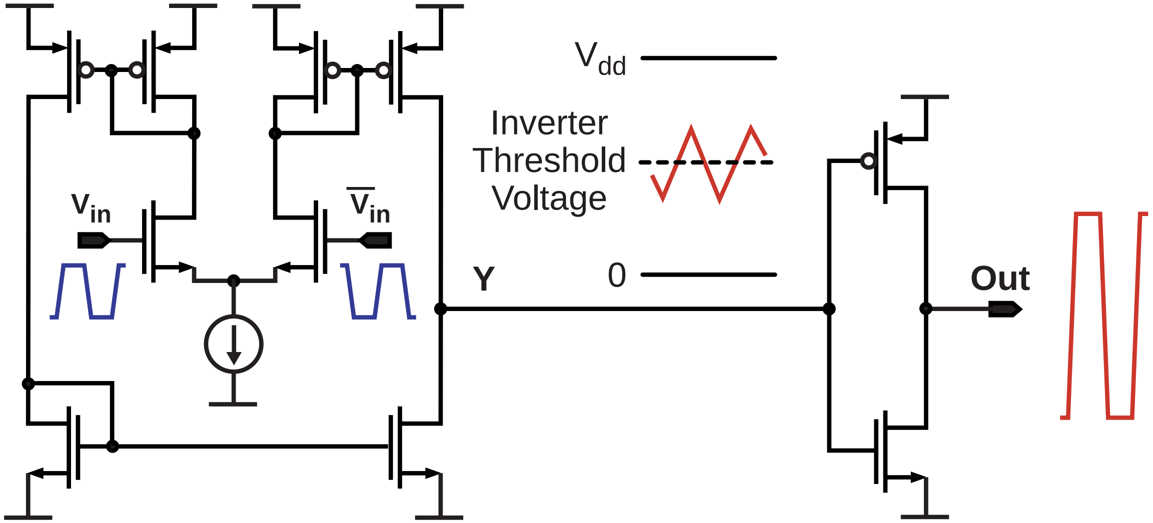
- Transition time of register output is set by the clock, not the incoming data input
  - Synchronous circuits have jitter performance corresponding to their clock
  - Jitter does not accumulate as signal travels through synchronous stages

High Speed, Low Power Asynchronous Dividers



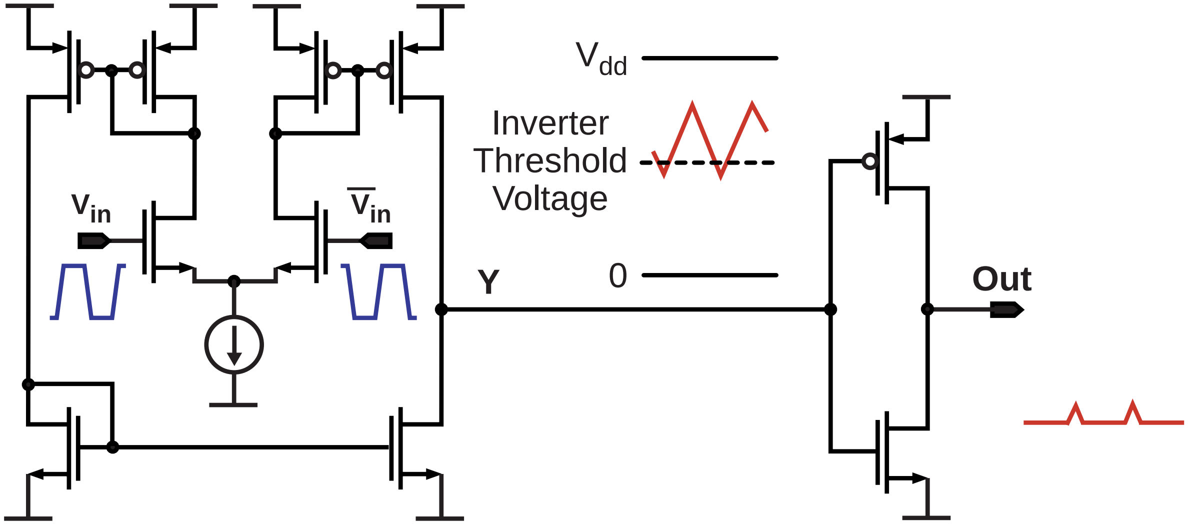
- **Highest speed achieved with differential CML registers**
  - **Static power consumption not an issue for high speed sections, but wasteful in low speed sections**
- **Lower power achieved by using full swing logic for low speed sections**

Differential to Full Swing Converter



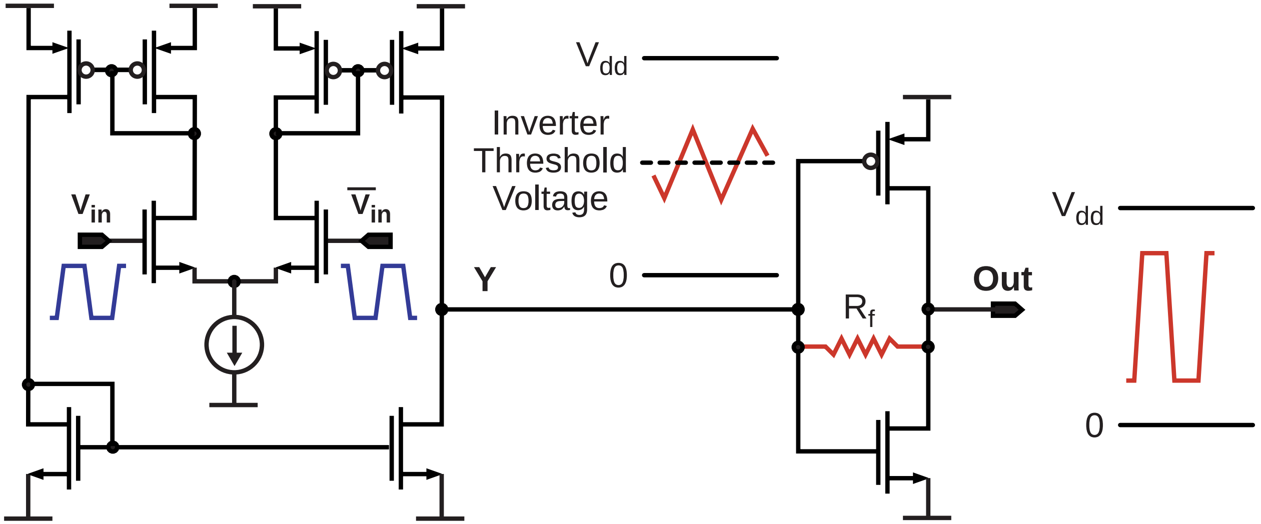
- **Use an opamp style circuit to translate differential input voltage to a single-ended output**
- **Use an inverter to amplify the single-ended output to full swing level**

Issue: Architecture Very Sensitive to DC Offset



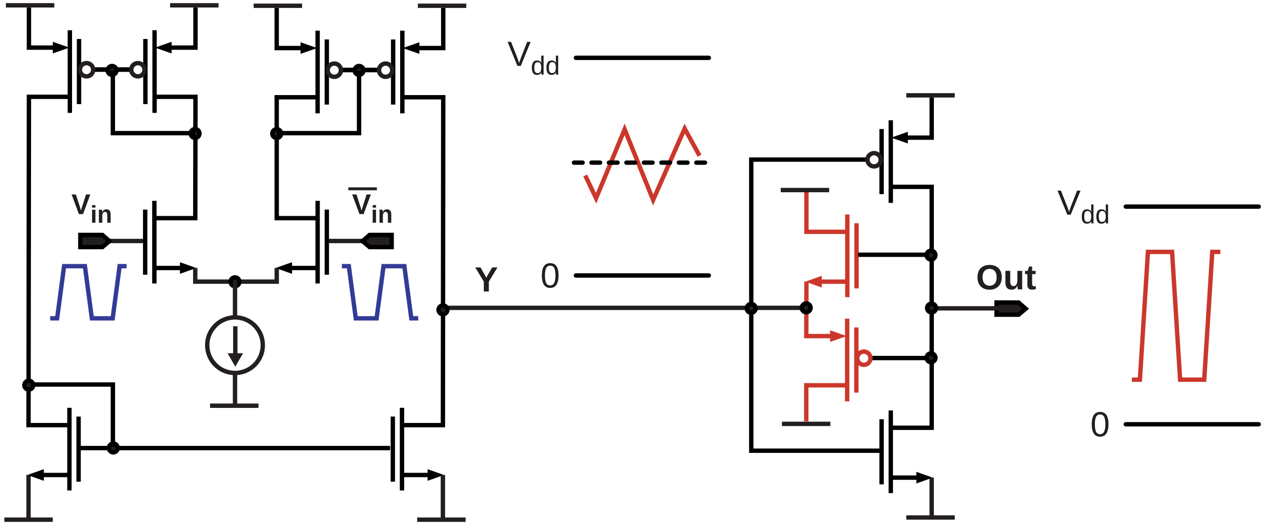
- **Opamp style circuit has very high DC gain from V_{in} to node Y**
- **DC offset will cause signal to rise above or fall below inverter threshold**
  - **Output signal rails rather than pulsing**

Use Resistor Feedback to Reduce DC Gain



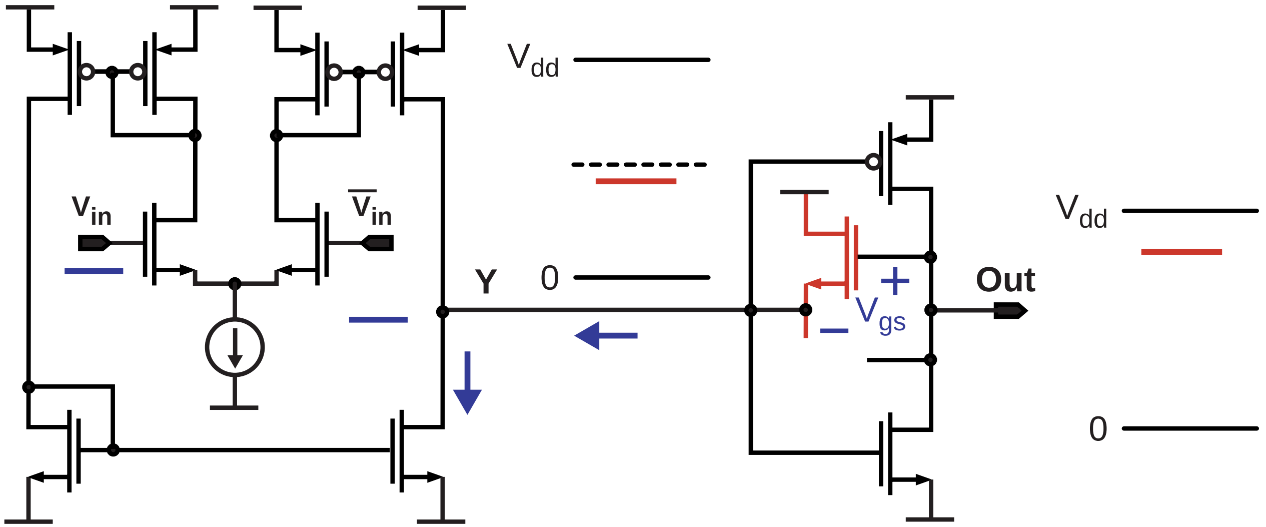
- Idea: create transresistance amplifier rather than voltage amplifier out of inverter by using feedback resistor
  - Presents a low impedance to node Y
  - Current from opamp style circuit is shunted through resistor
  - DC offset at input shifts output waveform slightly, but not node Y (to first order)
- Circuit is robust against DC offset!

Alternate Implementation of Inverter Feedback



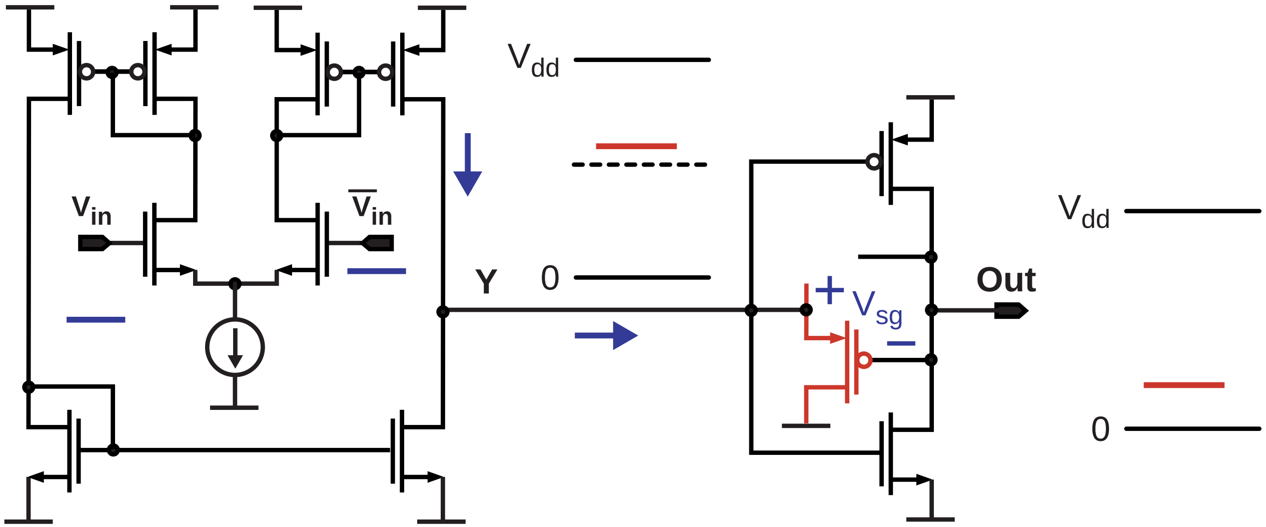
- **Nonlinear feedback using MOS devices can be used in place of resistor**
  - **Smaller area than resistor implementation**
- **Analysis done by examining impact of feedback when output is high or low**

Impact of Nonlinear Feedback When Output is High



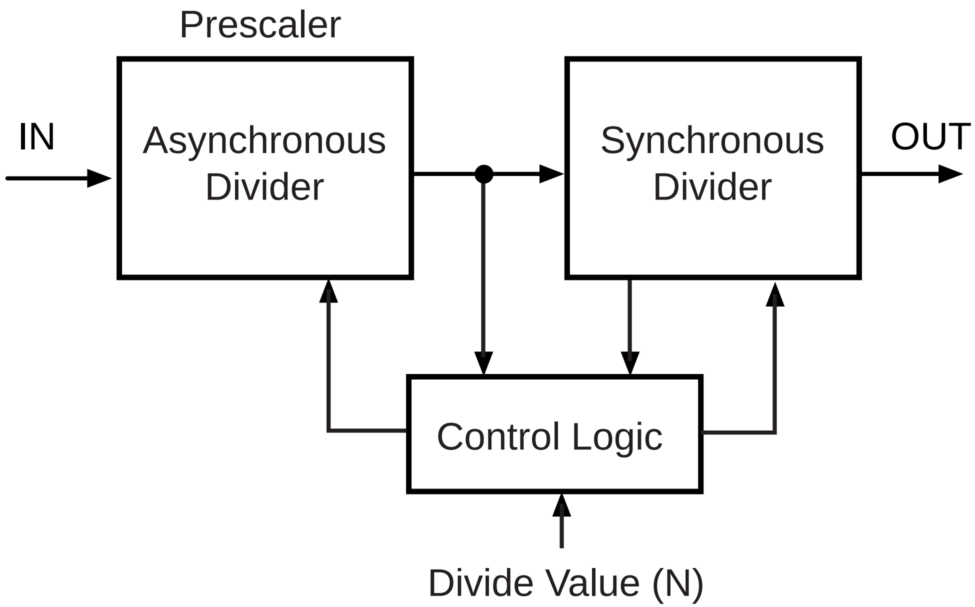
- **Corresponds to case where current flows into node Y**
  - **NMOS device acts like source follower**
  - **PMOS device is shut off**
- **Output is approximately set to V_{gs} of NMOS feedback device away from inverter threshold voltage**
  - **Inverter input is set to a value that yields that output voltage**
    - High DC gain of inverter insures it is close to inverter threshold

Impact of Nonlinear Feedback When Output is Low



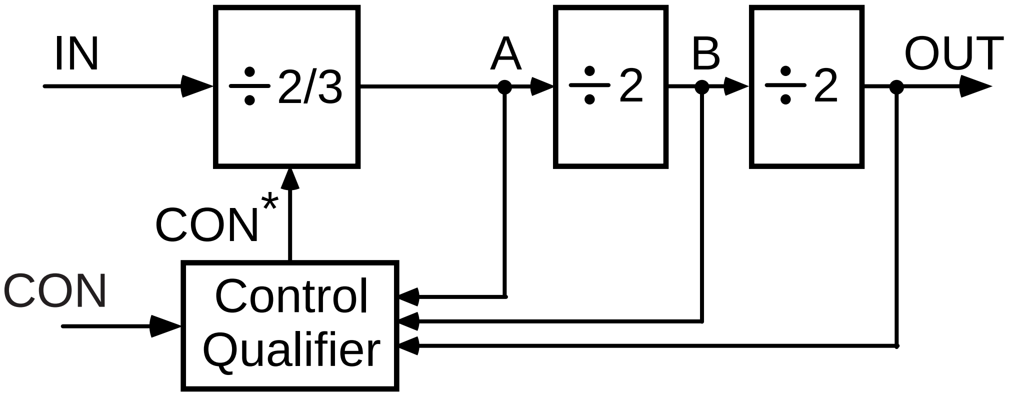
- Corresponds to case where current flows out of node Y
  - NMOS device is shut off
  - PMOS device acts like source follower
- Output is approximately set to Vgs of PMOS feedback device away from inverter threshold voltage
  - Inverter input is set to a value that yields that output voltage
    - High DC gain of inverter insures it is close to inverter threshold

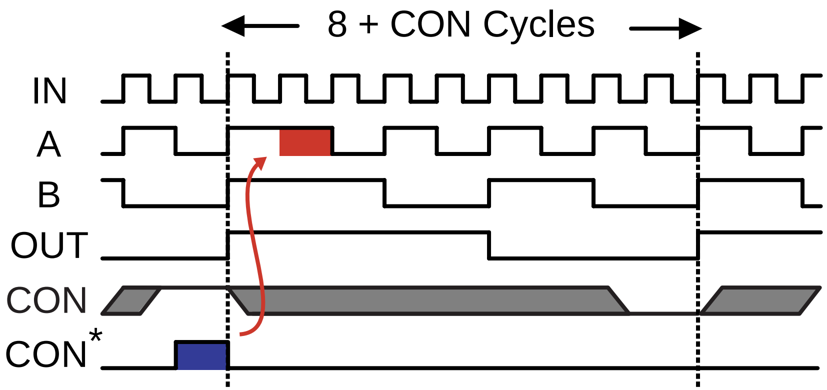
Variable Frequency Division



- Classical design partitions variable divider into two sections
  - Asynchronous section (called a prescaler) is fast
    - Often supports a limited range of divide values
  - Synchronous section has no jitter accumulation and a wide range of divide values
  - Control logic coordinates sections to produce a wide range of divide values

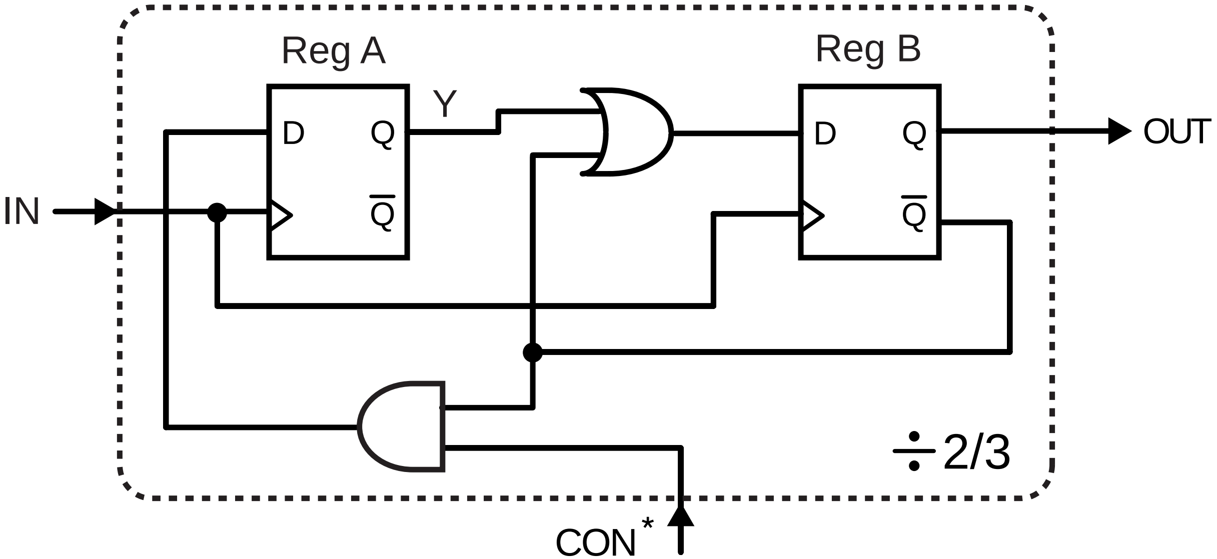
Dual Modulus Prescalers





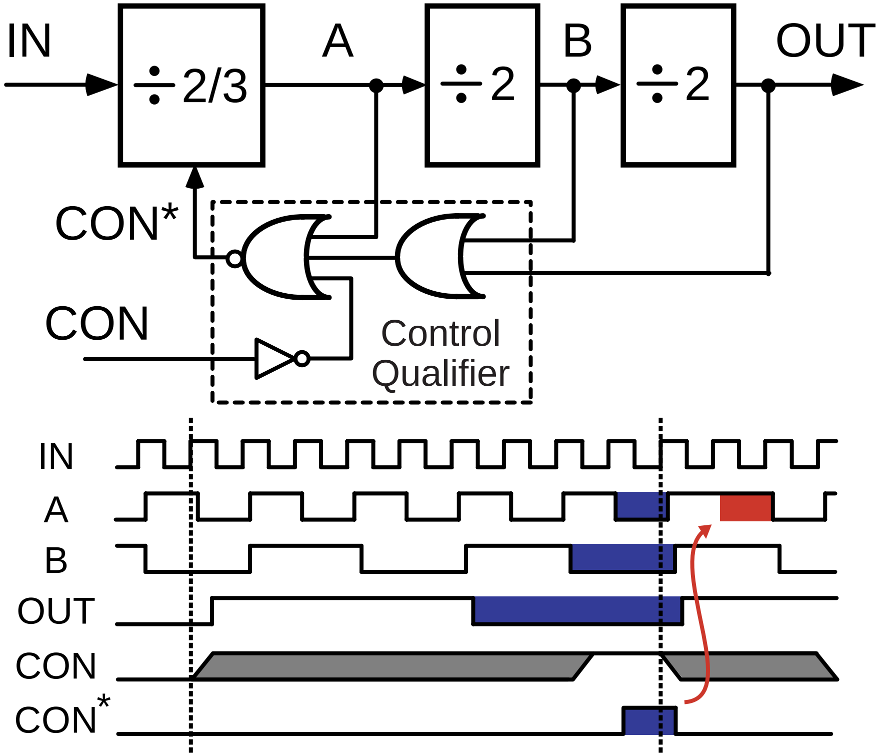
- Dual modulus design supports two divide values
  - In this case, divide-by-8 or 9 according to CON signal
- One cycle resolution achieved with front-end “2/3” divider

Divide-by-2/3 Design (Classical Approach)



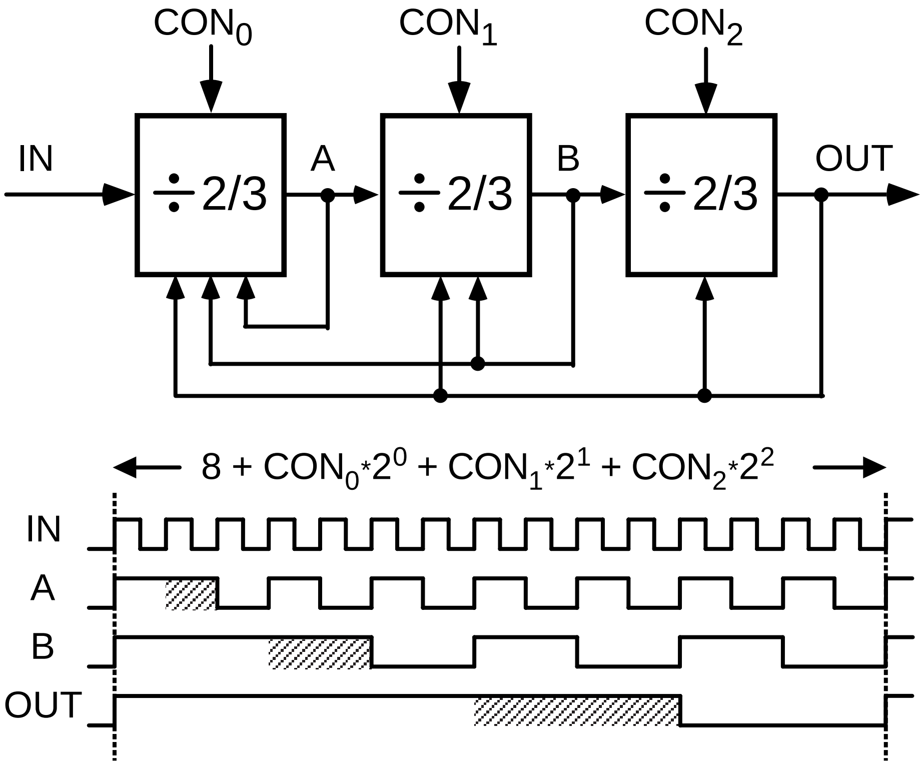
- **Normal mode of operation: CON* = 0) Y = 0**
  - **Register B acts as divide-by-2 circuit**
- **Divide-by-3 operation: CON* = 1) Y = 1**
  - **Reg B remains high for an extra cycle**
    - Causes Y to be set back to 0) Reg B toggles again
    - CON* must be set back to 0 before Reg B toggles to prevent extra pulses from being swallowed

Control Qualifier Design (Classical Approach)



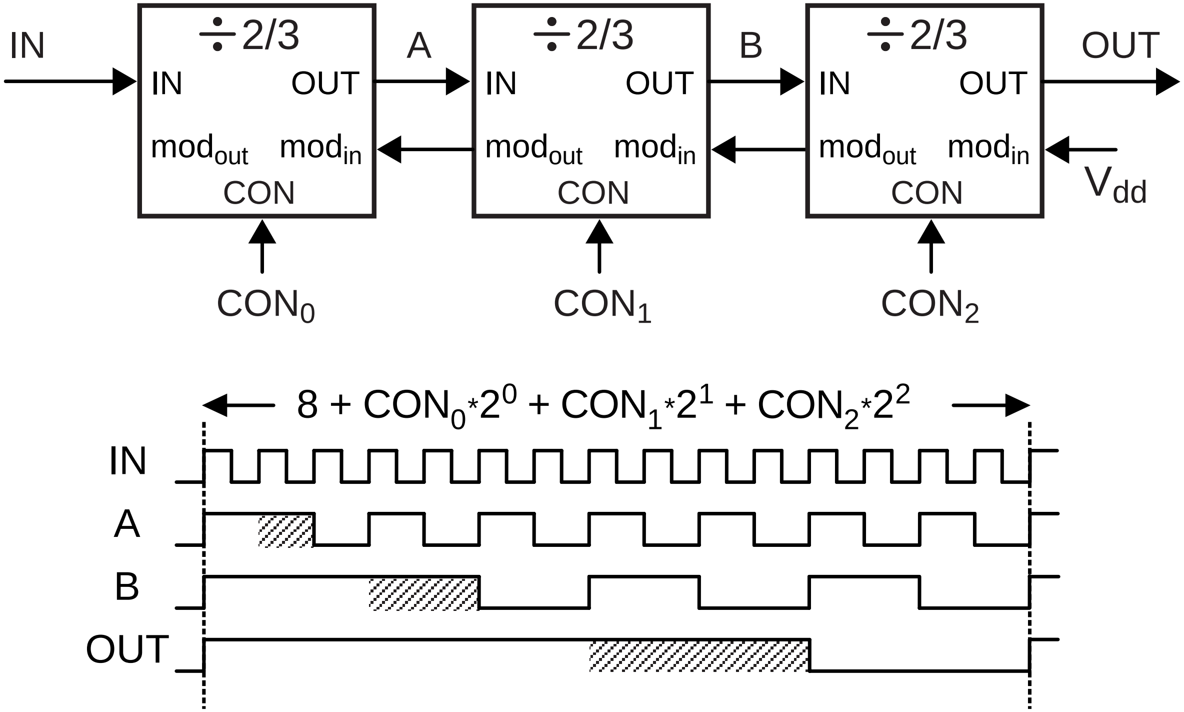
- **Must align CON signal to first “2/3” divider stage**
  - **CON signal is based on logic clocked by divider output**
    - There will be skew between “2/3” divider timing and CON
- **Classical approach cleverly utilizes outputs from each section to “gate” the CON signal to “2/3” divider**

Multi-Modulus Prescalers



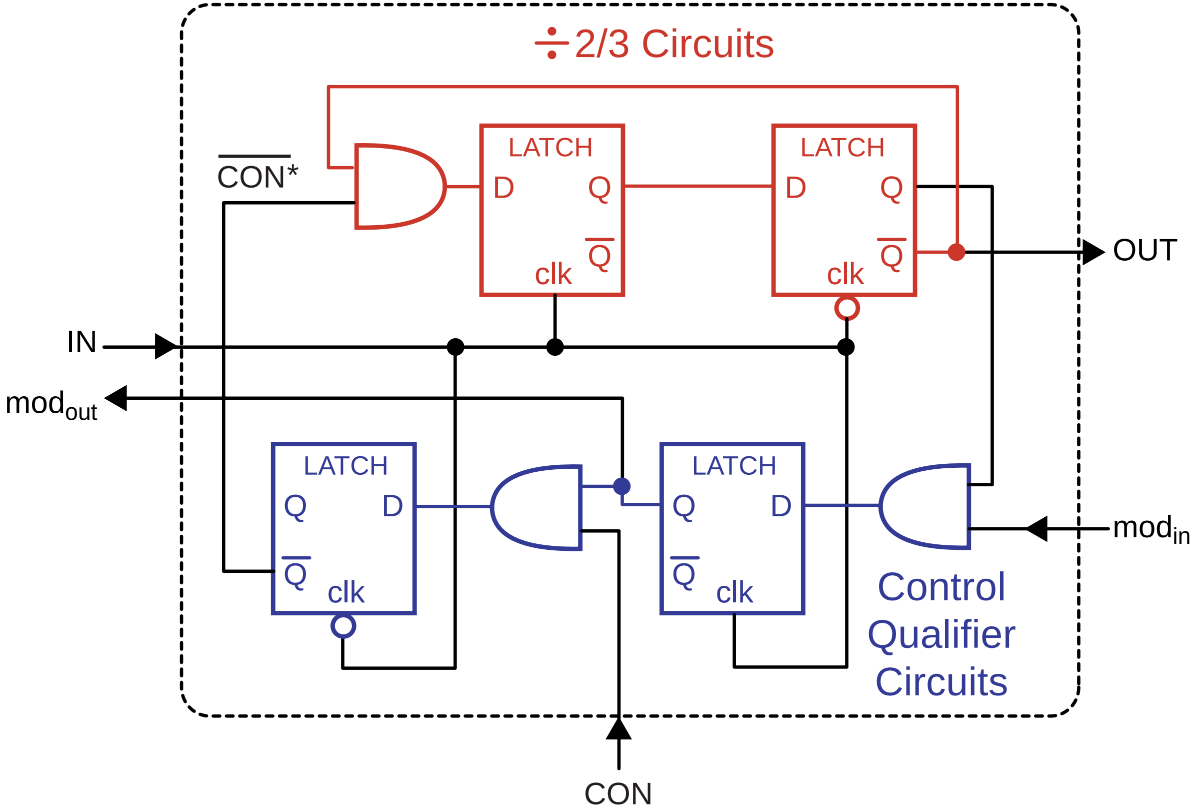
- **Cascaded 2/3 sections achieves a range of 2^n to 2^{n+1}-1**
  - **Above example is 8/ ··· /15 divider**
- **Asynchronous design allows high speed and low power operation to be achieved**
  - **Only negative is jitter accumulation**

A More Modular Design



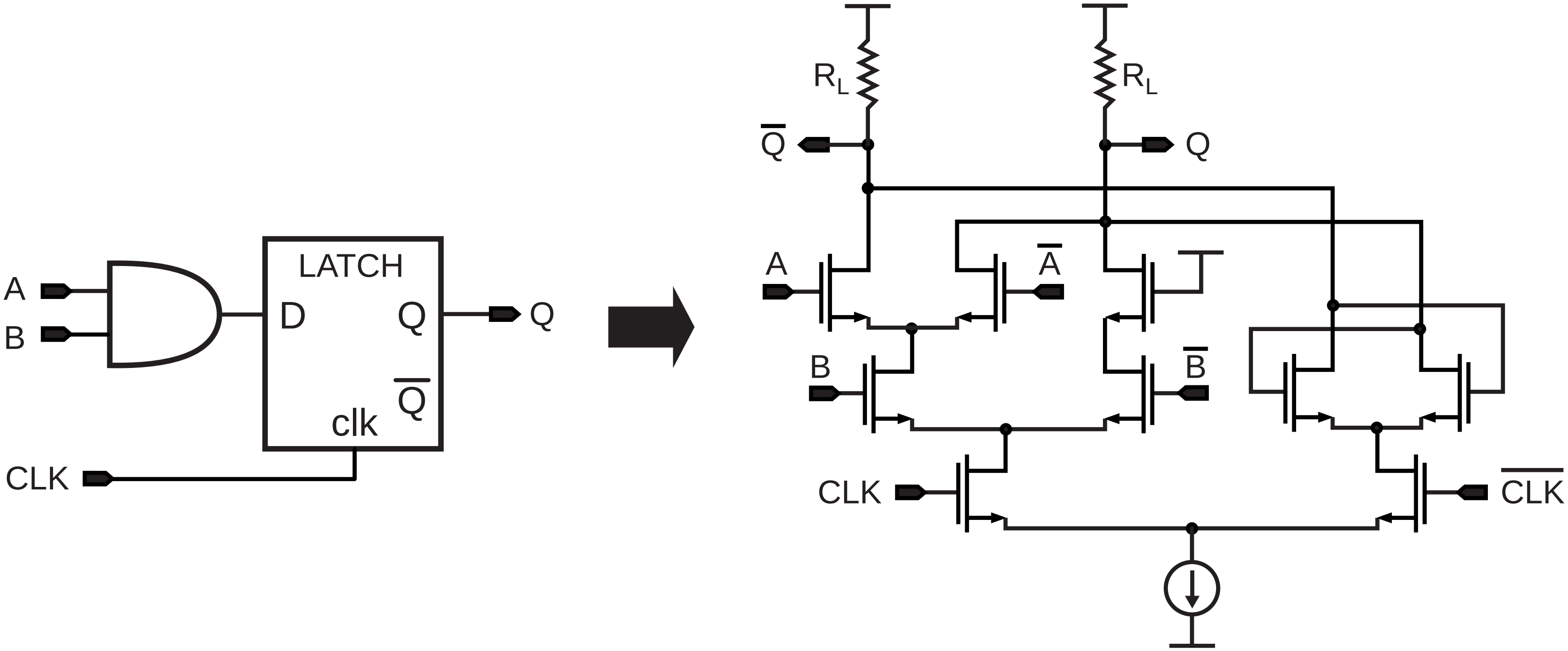
- Perform control qualification by synchronizing within each stage before passing to previous one
  - Compare to previous slide in which all outputs required for qualification of first 2/3 stage
- See Vaucher et. al., “A Family of Low-Power Truly Modular Programmable Dividers …”, JSSC, July 2000

Implementation of 2/3 Sections in Modular Approach



- **Approach has similar complexity to classical design**
  - **Consists of two registers with accompanying logic gates**
- **Cleverly utilizes "gating" register to pass synchronized control qualifying signal to the previous stage**

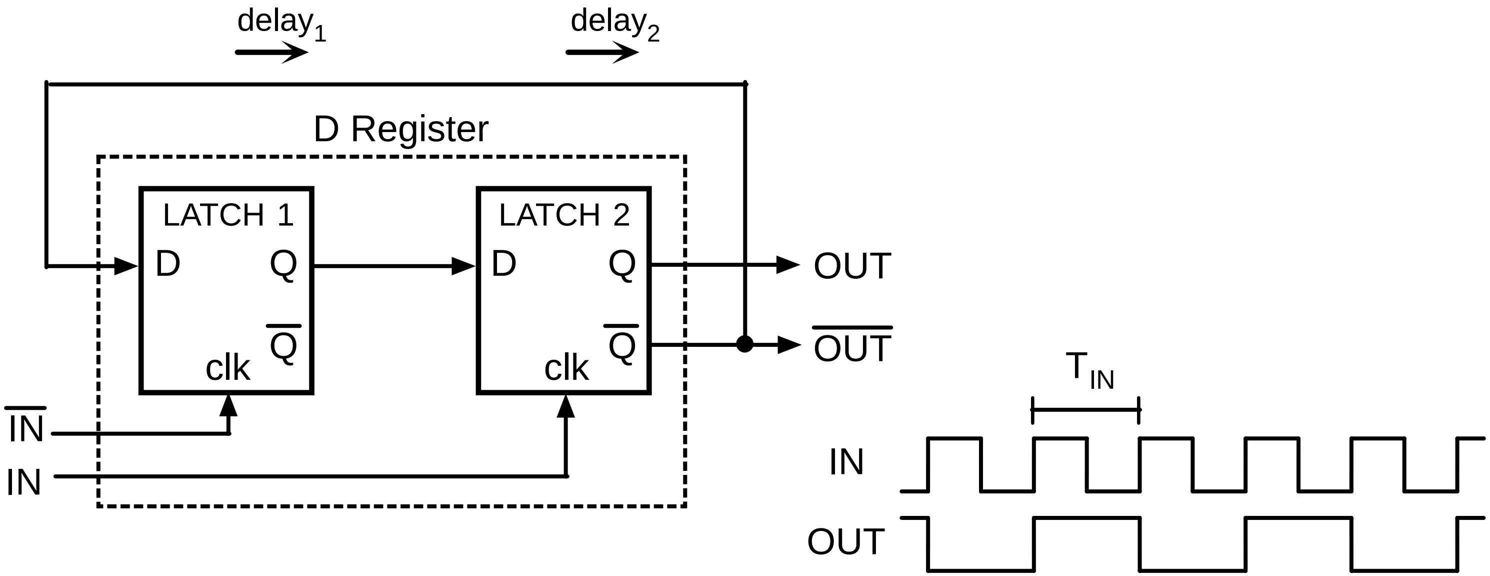
Implementation of Latch and And Gate in 2/3 Section



- **Combine AND gate and latch for faster speed and lower power dissipation**
- **Note that all primitives in 2/3 Section on previous slide consist of this combination or just a straight latch**

Can We Go Even Faster?

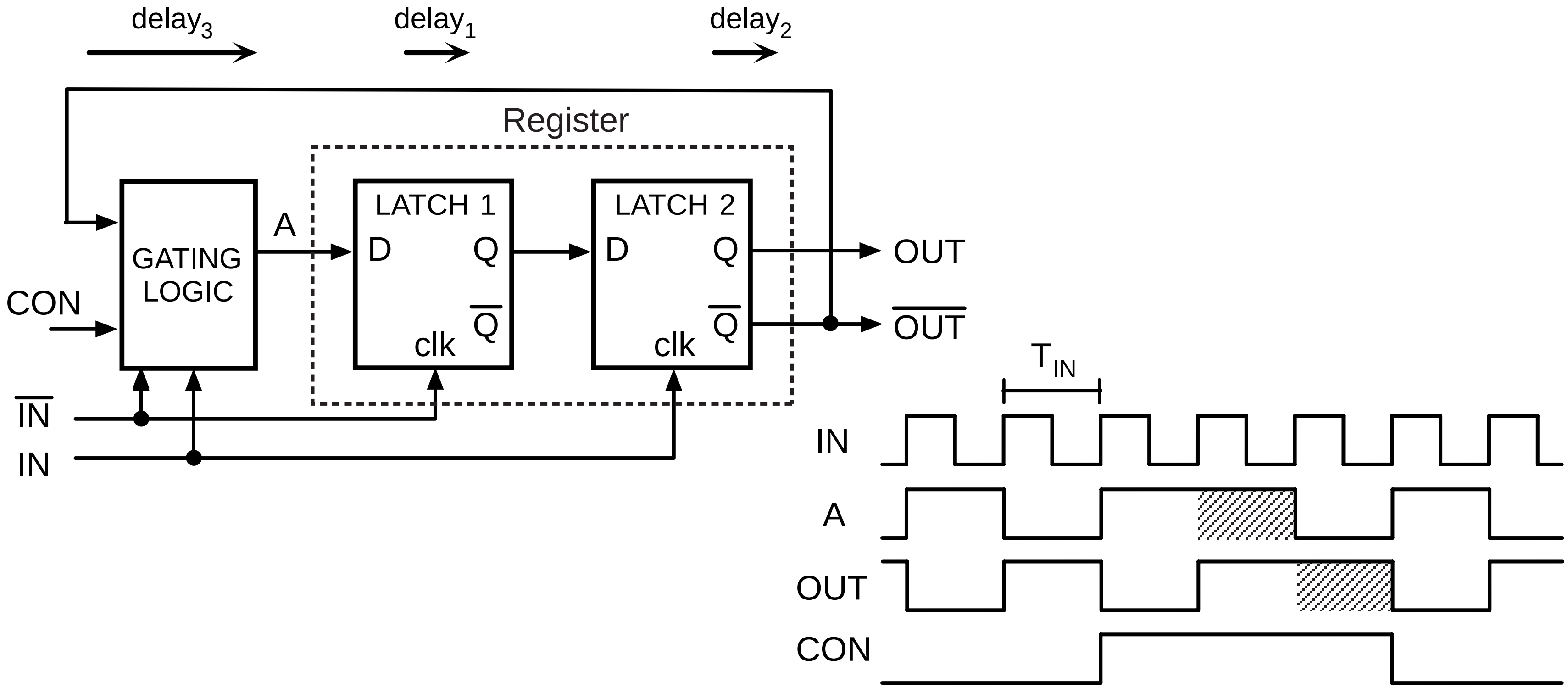
Speed Limitations of Divide-by-2 Circuit



- **Maximum speed limited only by propagation delay (delay$_1$, delay$_2$) of latches and setup time of latches (T$_s$)**

$$\frac{T_{IN}}{2} > delay_1 + T_s, \quad \frac{T_{IN}}{2} > delay_2 + T_s$$

Speed Limitations of Gated Divide-by-2/3 Circuit

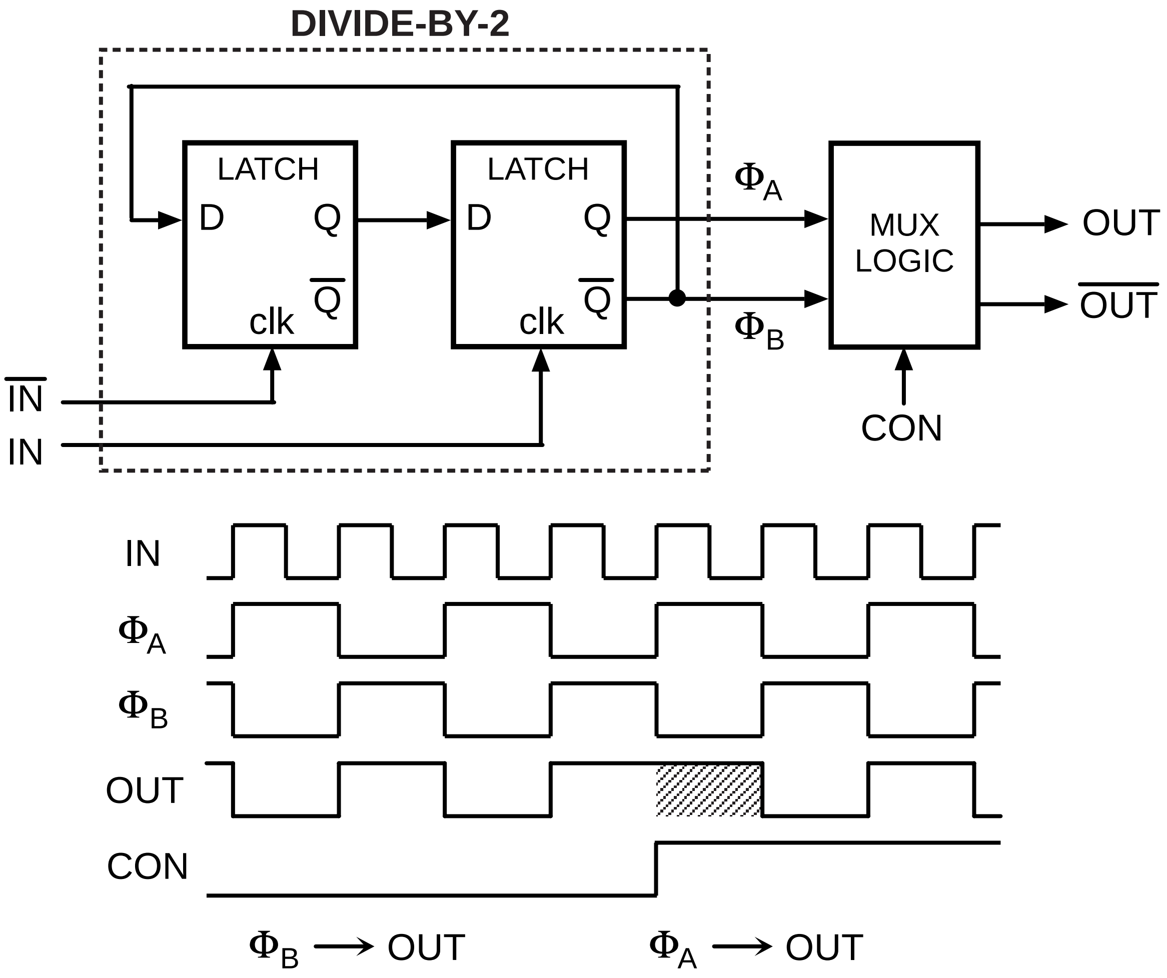


- **Maximum speed limited by latch *plus* gating logic**

$$\frac{T_{IN}}{2} > delay_2 + delay_3 + T_s$$

- **Gated divide-by-2/3 *fundamentally* slower than divide-by-2**

Divide-by-2/3 Using Phase Shifting



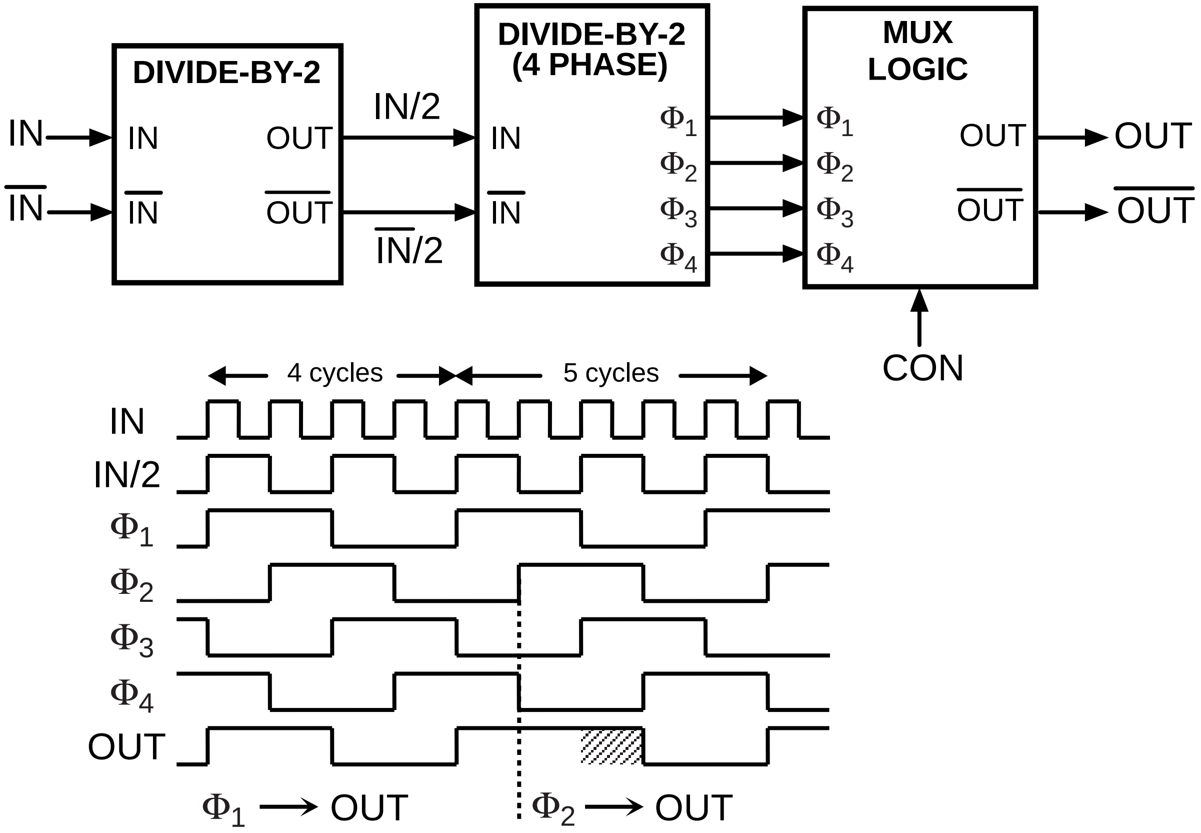
- **Achieves speed of divide-by-2 circuits!**
  - **MUX logic runs at half the input clock speed**

Implementation Challenges to Phase Shifting

- **Avoiding glitches**
  - **By assumption of sine wave characteristics**
    - Craninckx et. al., “A 1.75 GHz/3 V Dual-Modulus Divide-by-128/129 Prescaler …”, JSSC, July 1996
  - **By make-before-break switching**
    - My thesis: http://www-mtl.mit.edu/~perrott/
  - **Through re-timed multiplexor**
    - Krishnapura et. al, “A 5.3 GHz Programmable Divider for HiPerLan in 0.25μm CMOS”, JSSC, July 2000

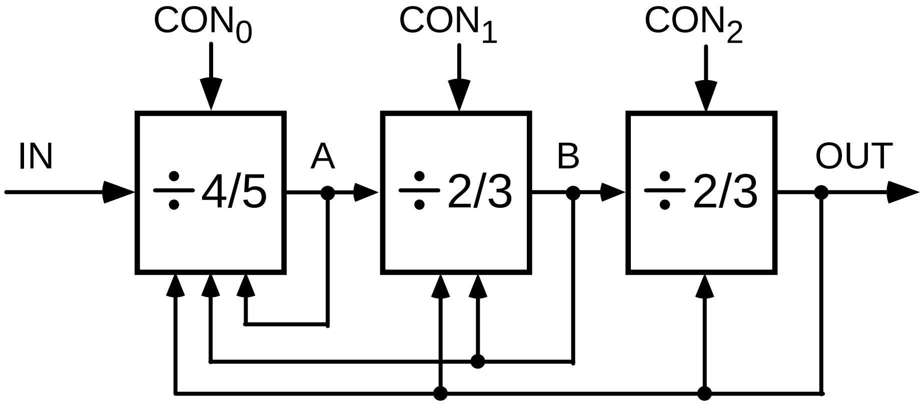
- **Avoiding jitter due to mismatch in phases**
  - **Through calibration**
    - Park et. al., “A 1.8-GHz Self-Calibrated Phase-Locked Loop with Precise I/Q Matching”, JSSC, May 2001

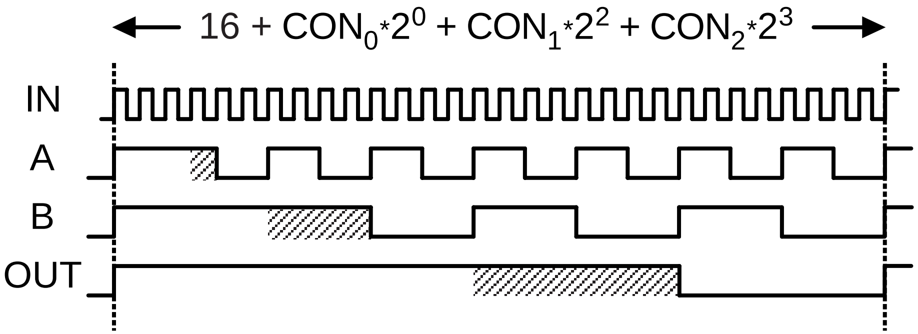
Further Reduction of MUX Operating Frequency



- **Leverage the fact that divide-by-2 circuit has 4 phases**
  - **Create divide-by-4/5 by cascading *two* divide-by-2 circuits**
    - Note that single cycle pulse swallowing still achieved
  - **Mux operates at one fourth the input frequency!**

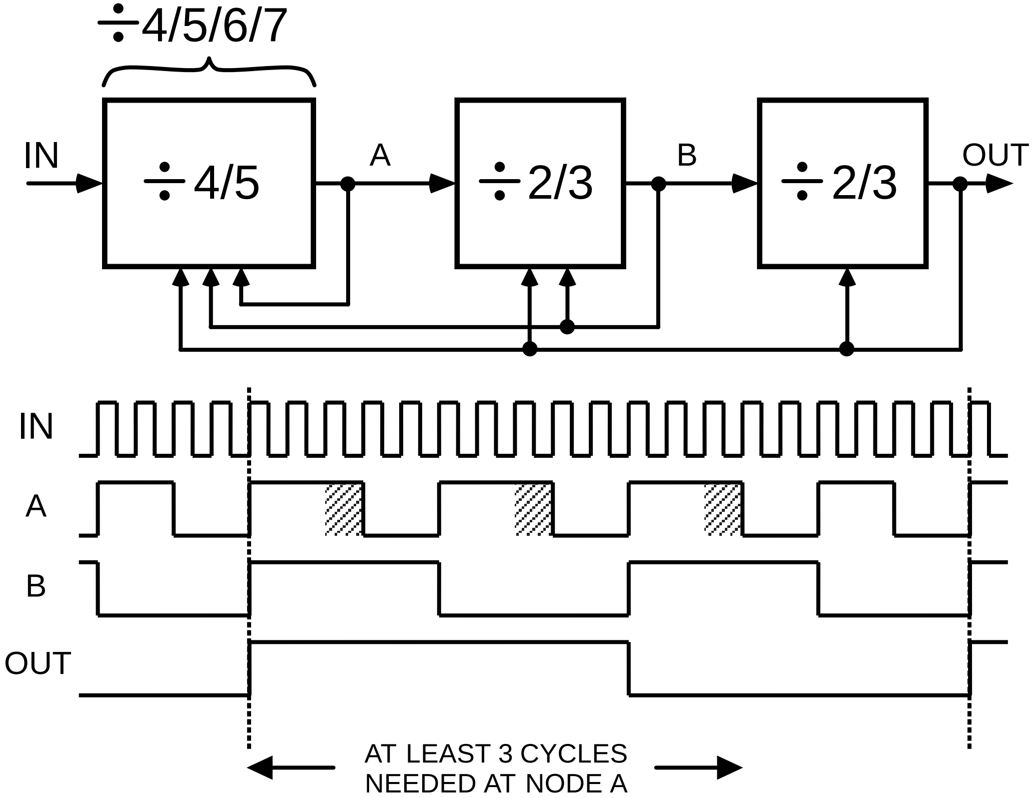
Impact of Divide-by-4/5 in Multi-Modulus Prescaler





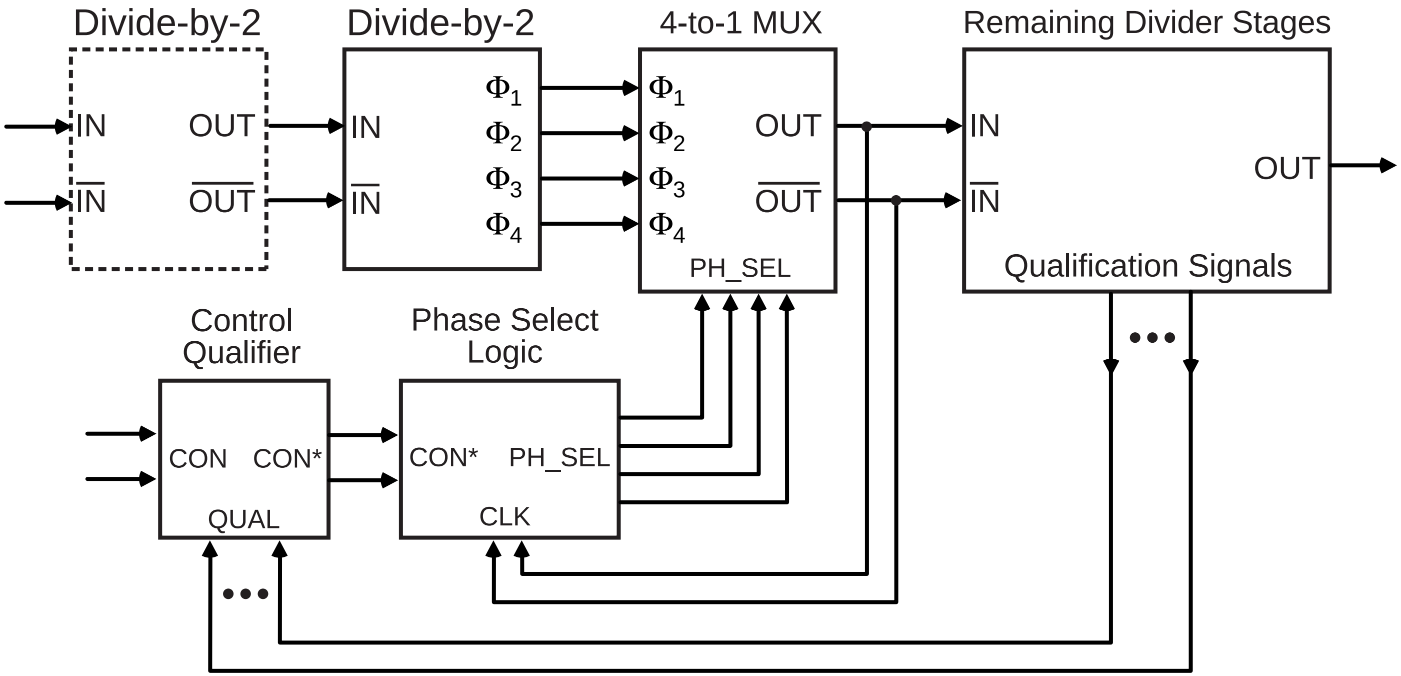
- Issue – gaps are created in divide value range
  - Divide-by-4/5 lowers swallowing resolution of following stage

Method to "Fill In" Divide Value Range



- Allow divide-by-4/5 to swallow more than one input cycle per OUT period
  - Divide-by-4/5 changed to Divide-by-4/5/6/7
- Note: at least two divide-by-2/3 sections must follow

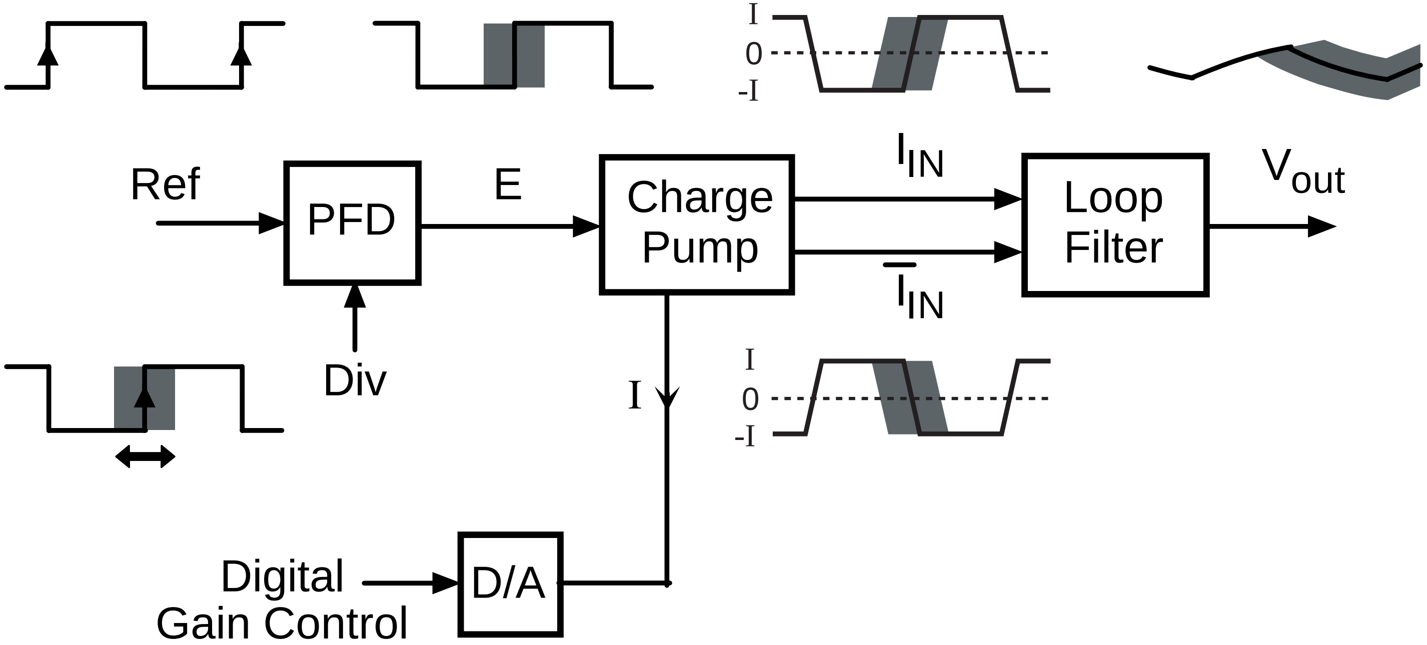
Example Architecture for a Phase-Shifted Divider



- **Phase shifting in first divide-by-4/5/6/7 stage to achieve high speed**
- **Remaining stages correspond to gated divide-by-2/3 cells**
- **For details, see my thesis**
  - **http://www-mtl.mit.edu/~perrott/**

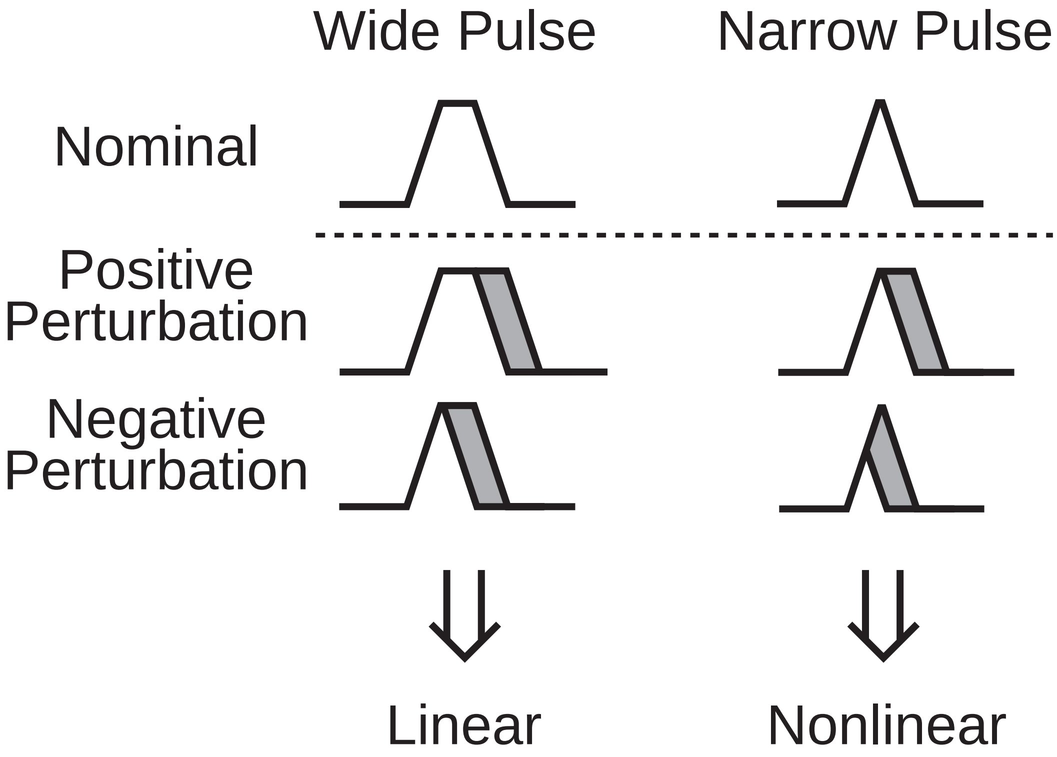
PFD/Charge Pump

Analog Phase Comparison Path



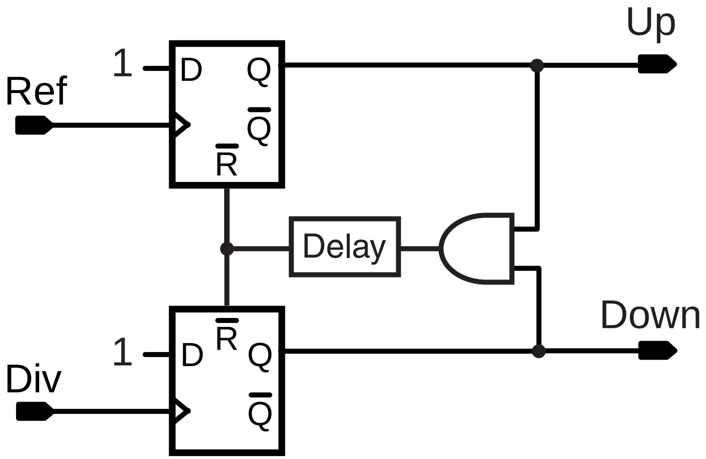
- Performs measurement of Ref and Div phase difference
- Sets PLL bandwidth and determines PLL stability
  - Note: Digital control of charge pump current allows tuning
- Key performance issues
  - Linearity, noise, power, area

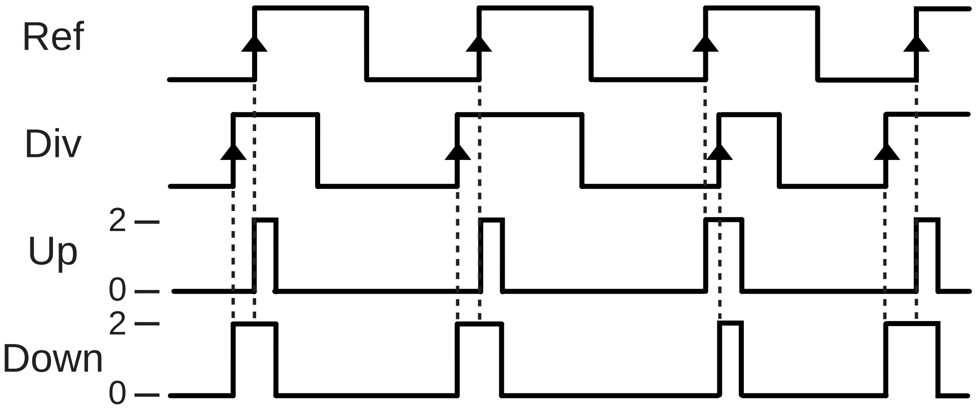
Achieving Linear Operation of PFD/Charge Pump



- **Key issue: pulses need to fully settle to avoid nonlinearity in PFD/charge pump**
  - **Impact of nonlinearity is noise folding/spurs**
  - **Runt pulses cause "dead zone" in PFD characteristic**

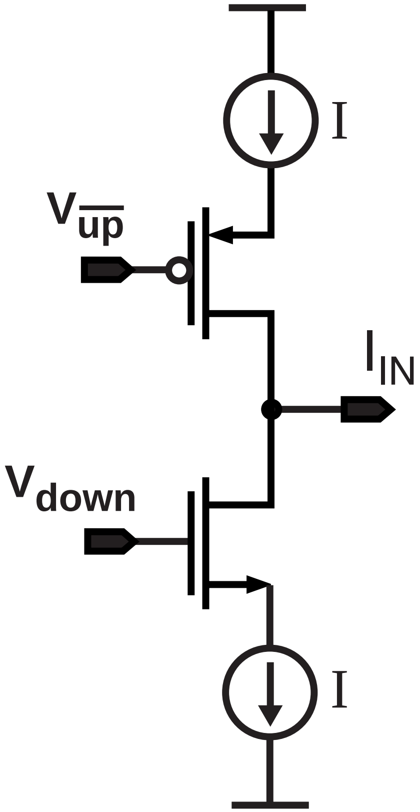
Add Delay in Reset Path to Prevent Dead Zone





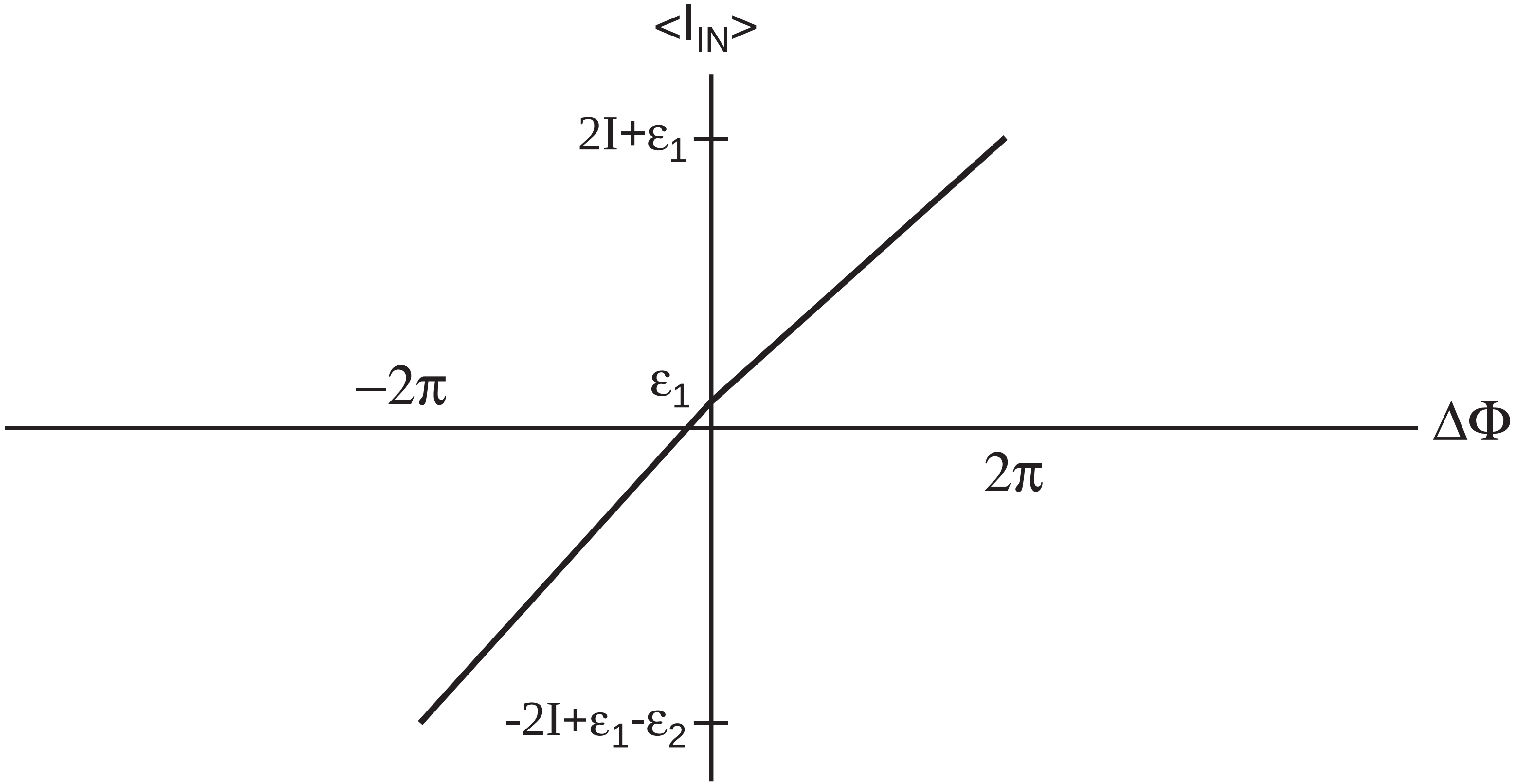
- **Minimum pulse width set by length of delay**
  - **Avoids incomplete settling at the expense of higher noise**

Simple View of Charge Pump



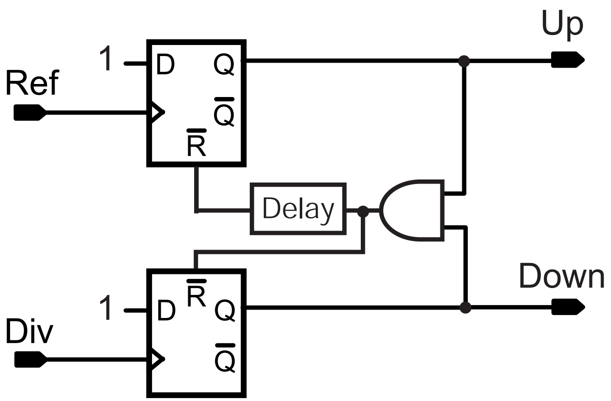
- Current sources implemented by current mirror circuits
  - Variation puts switches at supply/gnd rails
- Key issue: hard to achieve precise matching of up and down currents

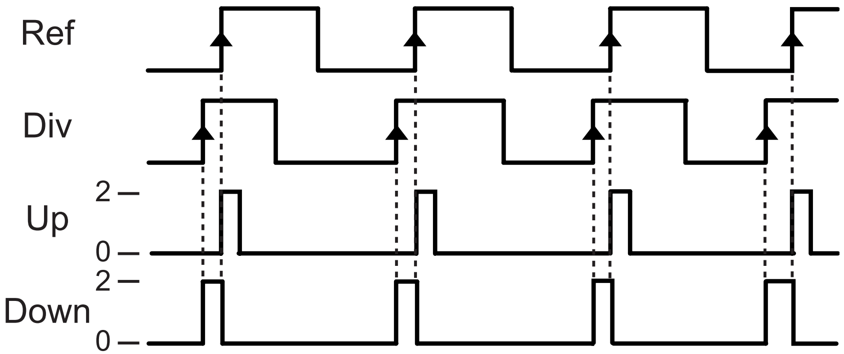
Practical Charge Pump Characteristic

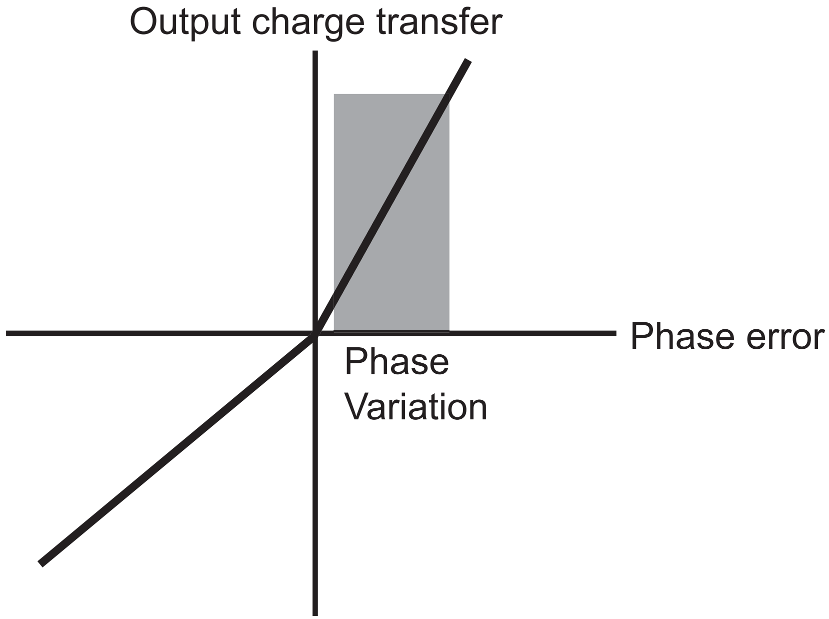


- **Gain slope changes due to up/down current mismatch**
  - **Causes nonlinearity in phase comparison operation**
- **Offset in phase due to timing mismatch between Up and Down PFD paths**

Offset PFD for High Linearity



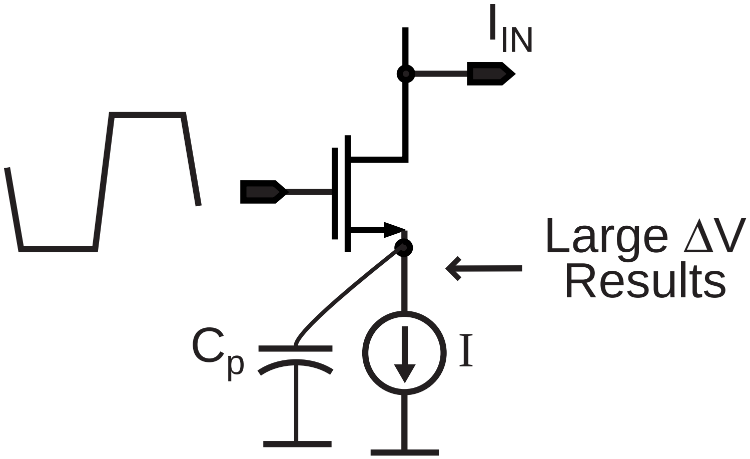




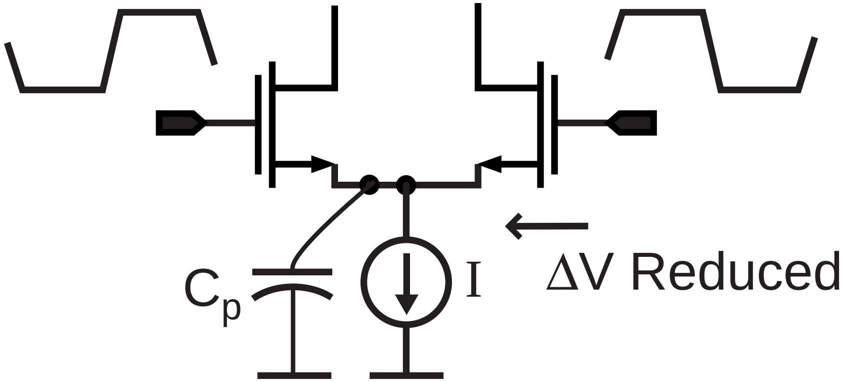
- **Change delay path such that Up pulse is always constant and follows Down pulse**
  - **Only Down pulse varies in width**
- **Confines phase variation to one side of PFD characteristic**

Single-Ended Versus Differential Charge Pumps

Single-Ended

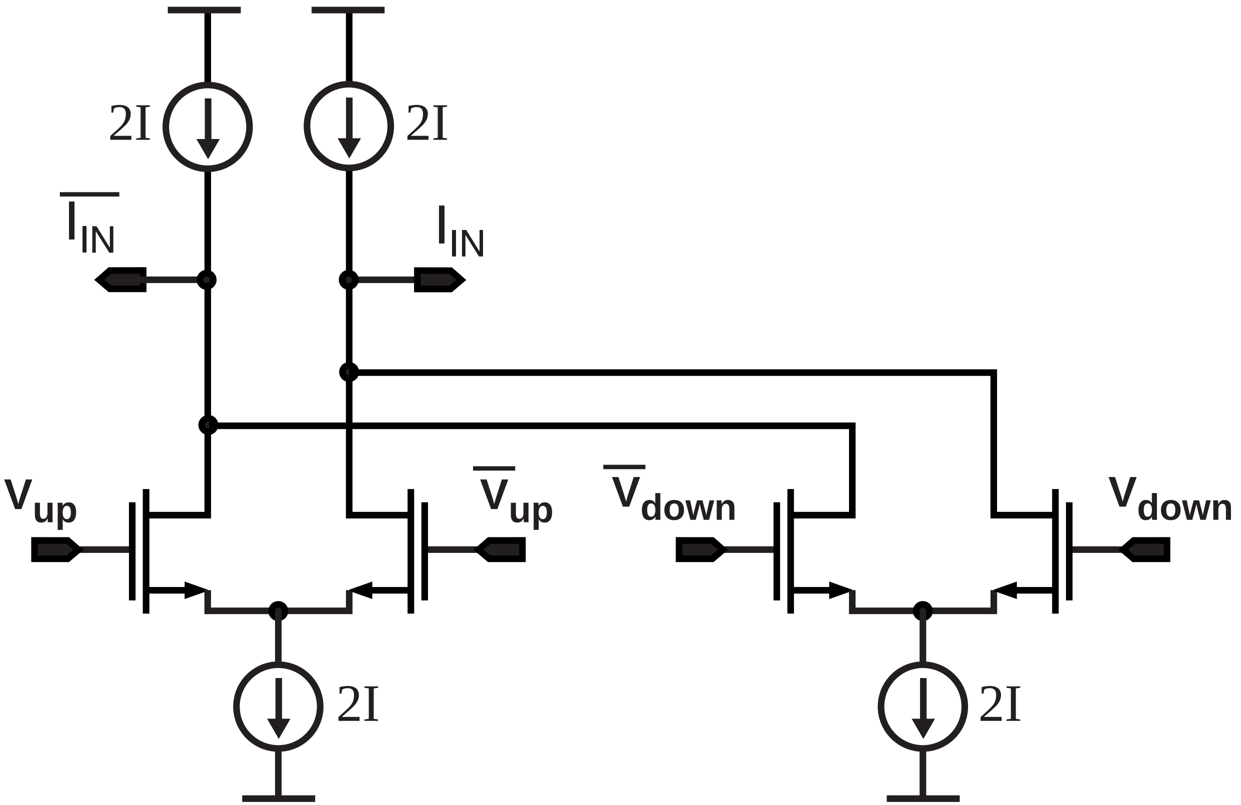


Differential



- Stray capacitance causes increased transient times for charge pump
  - Increases the minimum required on-time of PFD pulses
- Differential structure substantially reduces impact of stray capacitance
  - Reduces voltage variation on cap due to switching action

Example Differential Charge Pump Structure



- Zero output current is achieved by canceling Up and Down currents
- In practice, this structure is rarely used due to high noise it produces

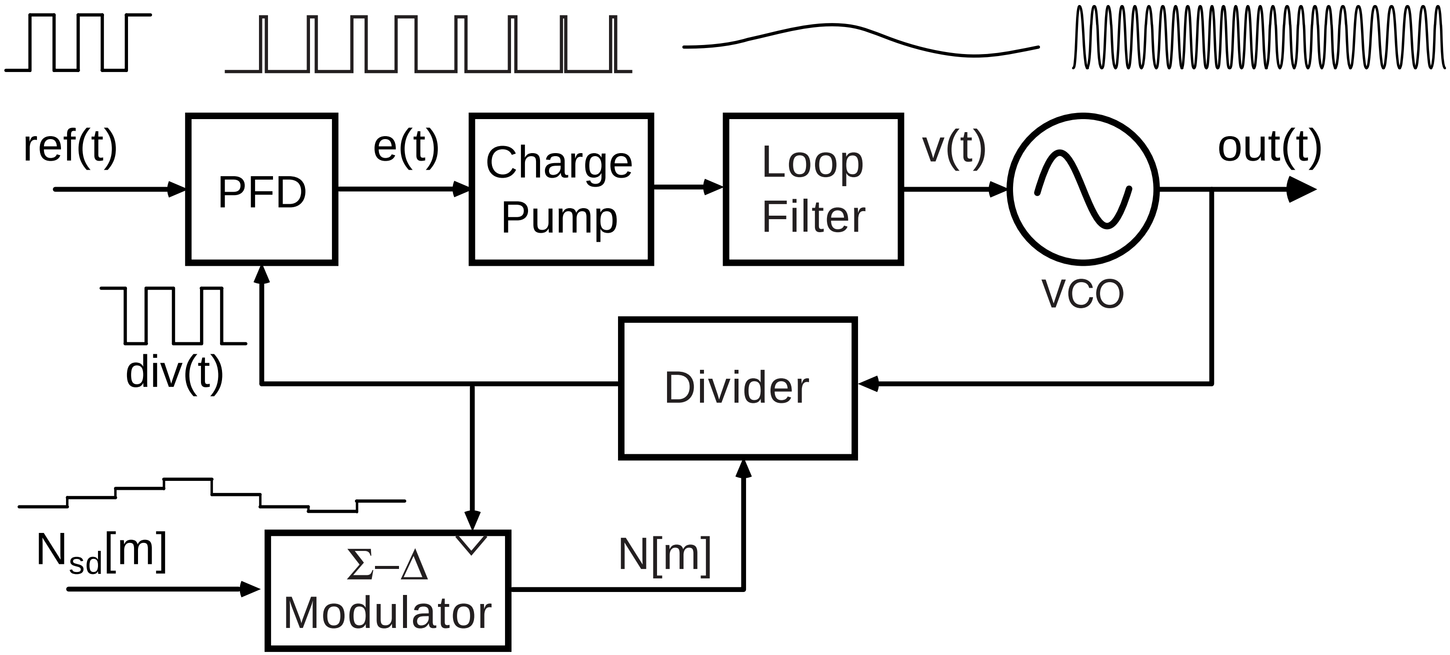
Loop Filter Design

Outline

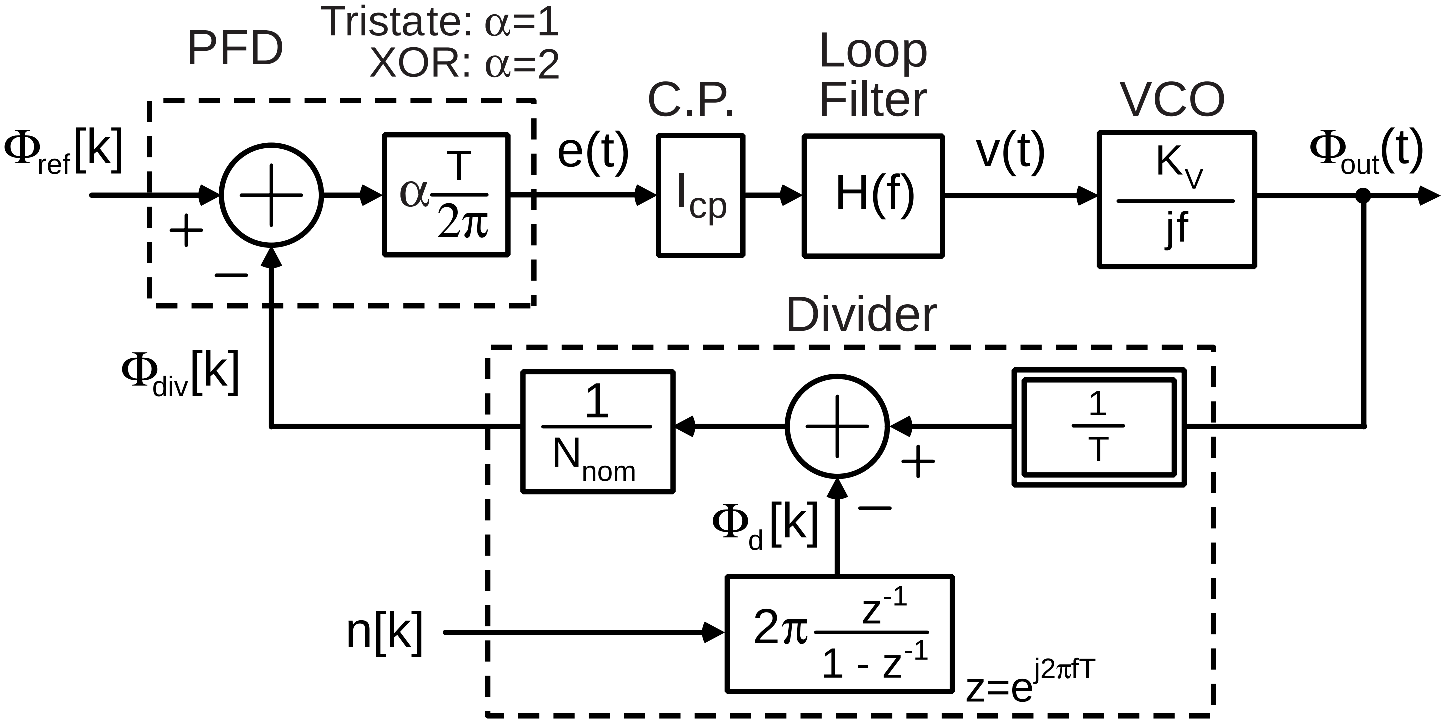
- Closed-Loop Design of Frequency Synthesizers
  - Introduction
  - Background on Classical Open Loop Design Approach
  - Closed Loop Design Approach
  - Example and Verification
  - Conclusion

Σ–Δ *Fractional-N Frequency Synthesizer*

- **Focus on this architecture since it is essentially a "super set" of other synthesizers, including integer-N and fractional-N**
  - **If we can design and simulate this structure, we can also do so for classical integer-N designs**



Frequency-domain Model



See Perrott et. al. *JSSC*, Aug. 2002 for details

- **Closed loop dynamics parameterized by**

$$\boxed{G(f) = \frac{A(f)}{1 + A(f)}} \quad \text{where} \quad A(f) = \frac{\alpha I_{cp} H(f) K_V}{N_{nom} 2\pi j f}$$

Review of Classical Design Approach

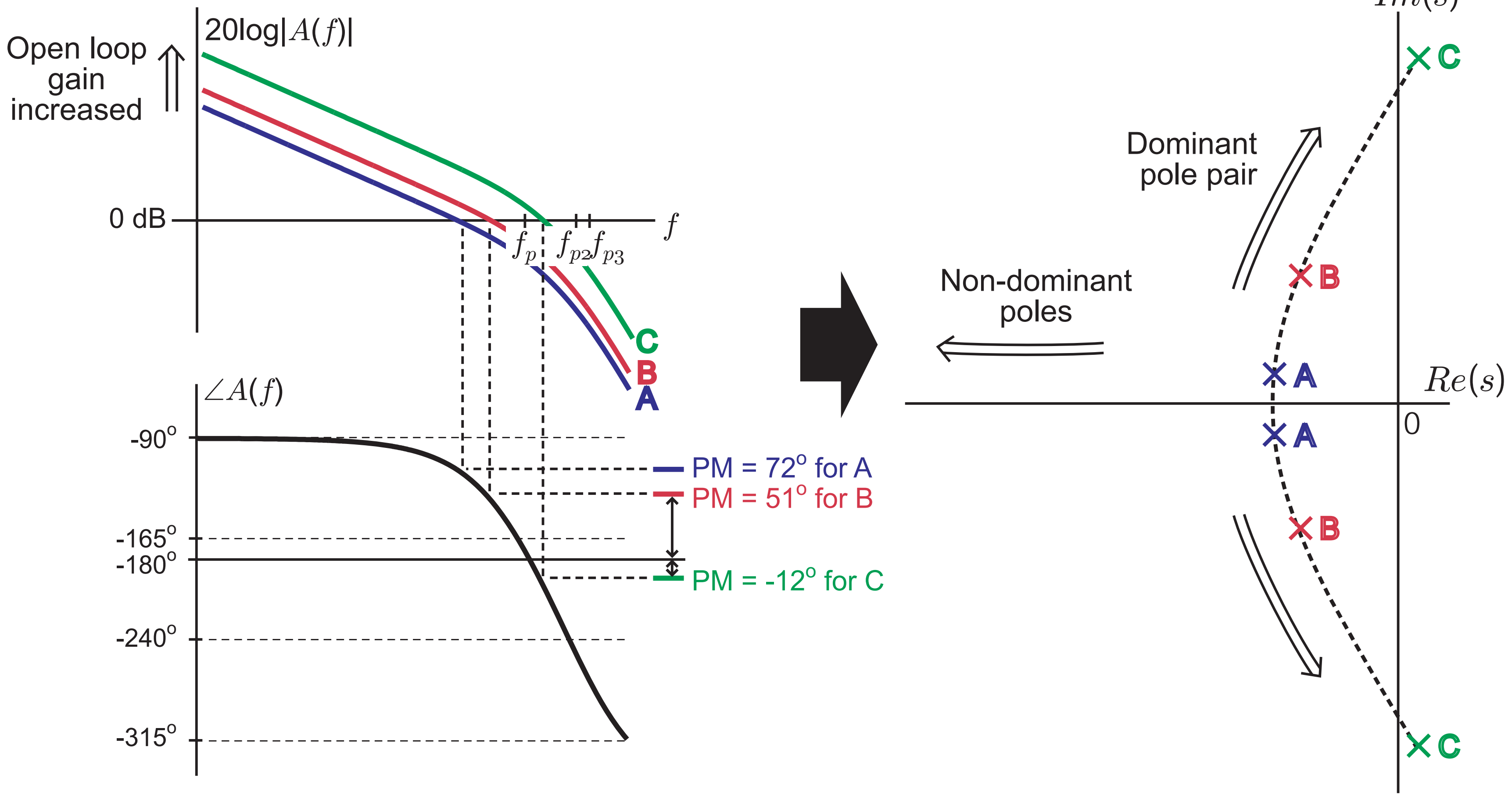
Given the desired closed-loop bandwidth, order, and system type:

1. **Choose an appropriate topology for H(f)**
   - **Depends on order, type**
2. **Choose pole/zero values for H(f) as appropriate for the required bandwidth**
3. **Adjust the open-loop gain to achieve the required bandwidth while maintaining stability**
   - **Plot gain and phase bode plots of A(f)**
   - **Use phase (or gain) margin criterion to infer stability**

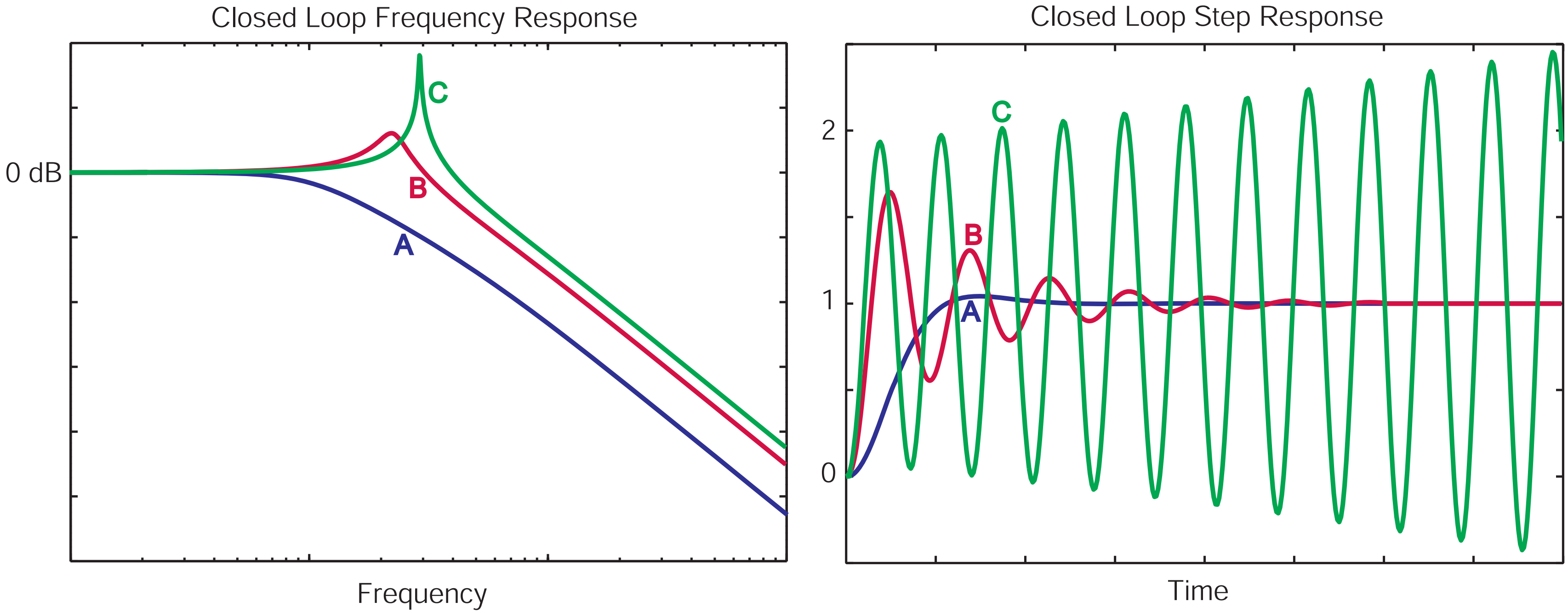
Example: First Order, Type I with Parasitic Poles

Evaluation of Phase Margin

Closed Loop Pole Locations of $G(f)$



First Order, Type I: Frequency and Step Responses



Limitations of Open Loop Design Approach

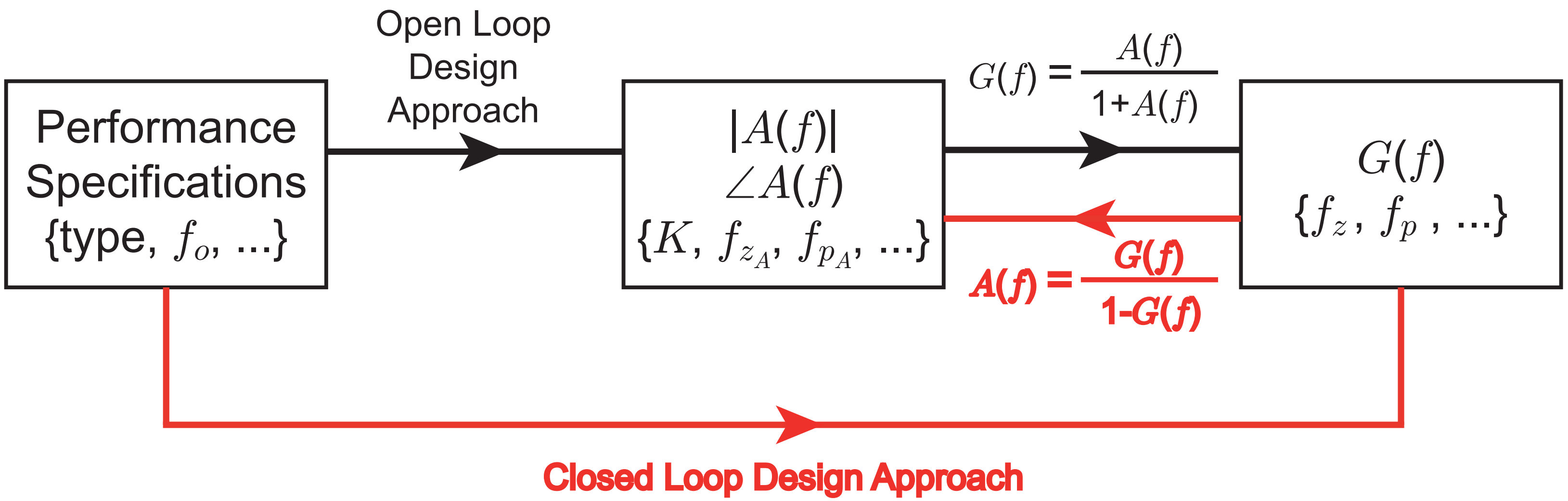
- **Constrained for applications which require precise filter response**
- **Complicated once parasitic poles are taken into account**
- **Poor control over filter shape**
- **Inadequate for systems with third order rolloff**
  - **Phase margin criterion based on second order systems**

Closed loop design approach:

Directly design G(f) by specifying dominant pole and zero locations on the s-plane (pole-zero diagram)

Closed Loop Design Approach: Overview

- G(f) completely describes the closed loop dynamics
  - Design of this function is the ultimate goal

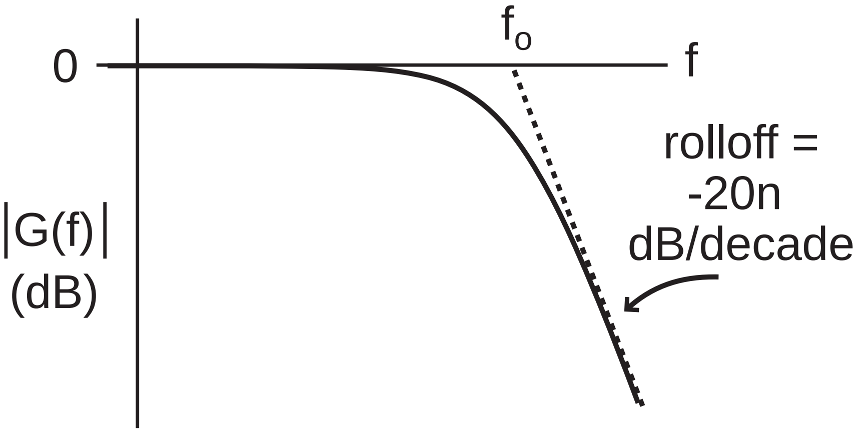

Open Loop
Design
Approach
Performance
Specifications
{type, f_o, ...}
$|A(f)|$
$\angle A(f)$
$\{K,\ f_{z_A},\ f_{p_A},\ ...\}$
$G(f) = \frac{A(f)}{1+A(f)}$
$A(f) = \frac{G(f)}{1-G(f)}$
$G(f)$
$\{f_z,\ f_p,\ ...\}$
Closed Loop Design Approach

- Instead of indirectly designing G(f) using plots of A(f), solve for G(f) directly as a function of specification parameters
- Solve for A(f) that will achieve desired G(f)
- Account for the impact of parasitic poles/zeros

Closed Loop Design Approach: Implementation

- **Download PLL Design Assistant Software**
  - **Part of CppSim package at http://www.cppsim.com**
- **Read accompanying manual**
- **Algorithm described by C.Y. Lau et. al. in “Fractional-N Frequency Synthesizer Design at the Transfer Function Level Using a Direct Closed Loop Realization Algorithm”, Design Automation Conference, 2003**

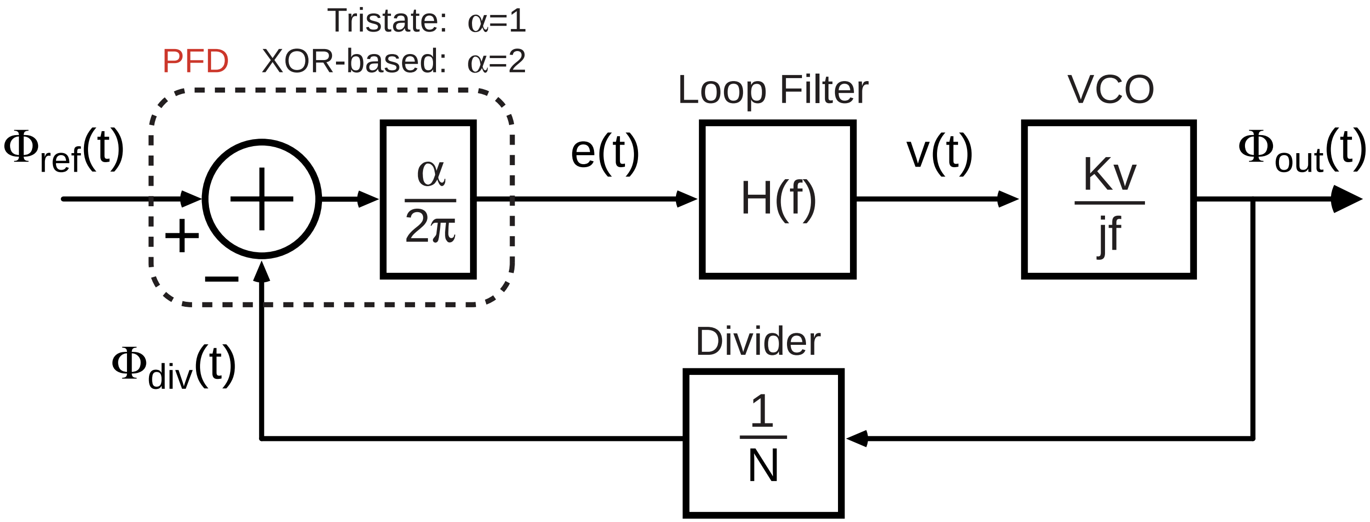
Definition of Bandwidth, Order, and Shape for G(f)



- **Bandwidth – f_o**
  - **Defined in asymptotic manner as shown**
- **Order – n**
  - **Defined according to the rolloff characteristic of G(f)**
- **Shape**
  - **Defined according to standard filter design methodologies**
    - Butterworth, Bessel, Chebyshev, etc.

Definition of Type

- **Type I: one integrator in PLL open loop transfer function**
  - **VCO adds on integrator**
  - **Loop filter, H(f), has no integrators**
- **Type II: two integrators in PLL open loop transfer function**
  - **Loop filter, H(f), has one integrator**



Loop Filter Transfer Function Vs Type and Order of G(f)

H(s) Topology For Different Type and Orders of G(f)

	Type I	Type II
Order 1	K_{LP}	$K_{LP}\dfrac{1+s/w_z}{s}$
Order 2	$\dfrac{K_{LP}}{1+s/w_p}$	$K_{LP}\dfrac{1+s/w_z}{s(1+s/w_p)}$
Order 3	$\dfrac{K_{LP}}{1+s/(w_pQ_p)+(s/w_p)^2}$	$\dfrac{K_{LP}(1+s/w_z)}{s(1+s/(w_pQ_p)+(s/w_p)^2)}$

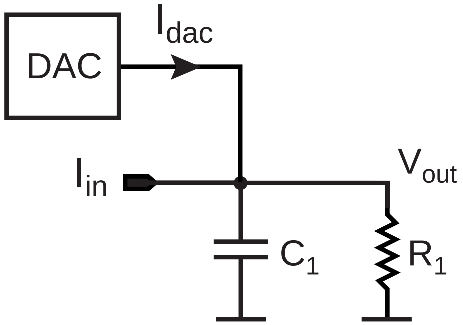
$$\text{where } K_{LP} = K\frac{N_{nom}}{K_v I_{cp}\alpha}$$

Calculated from software

- Practical PLL implementations limited to above
  - Prohibitive analog complexity for higher order, type
- Open loop gain, K, will be calculated by algorithm
  - Loop filter gain related to open loop gain as shown above

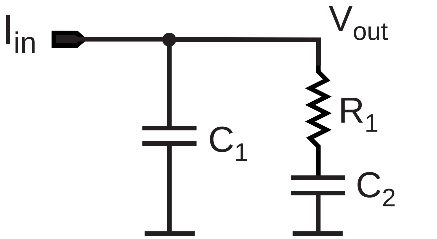
Passive Topologies to Realize a Second Order PLL

Type I, Order 2



$$\frac{V_{out}}{I_{in}} = \frac{R_1}{1+sR_1C_1}$$

Type II, Order 2

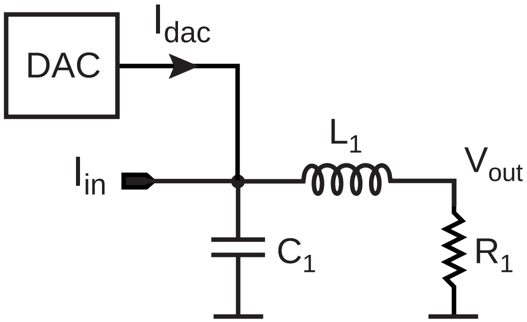


$$\frac{V_{out}}{I_{in}} = \frac{1}{s(C_1+C_2)}\,\frac{1+sR_1C_2}{1+sR_1C_{||}}$$

- **DAC is used for Type I implementation to coarsely tune VCO**
  - **Allows full range of VCO to be achieved**

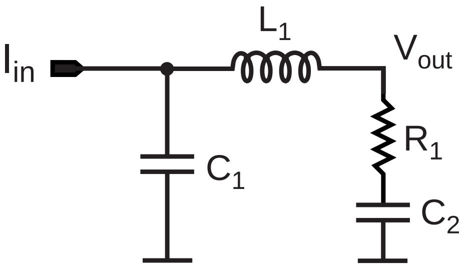
Passive Topologies to Realize a Third Order PLL

Type I, Order 3



$$\frac{V_{out}}{I_{in}} = \frac{R_1}{1+sR_1C_1+s^2L_1C_1}$$

Type II, Order 3



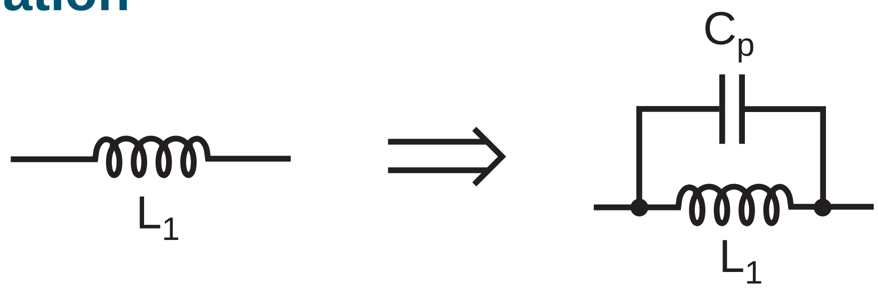
$$\frac{V_{out}}{I_{in}} = \frac{1}{s(C_1+C_2)}\,\frac{1+sR_1C_2}{1+sR_1C_{||}+s^2L_1C_{||}}$$

where $C_{||} = C_1C_2/(C_1+C_2)$

- **Inductor is necessary to create a complex pole pair**
  - **Must be implemented off-chip due to its large value**

Problem with Passive Loop Filter Implementations

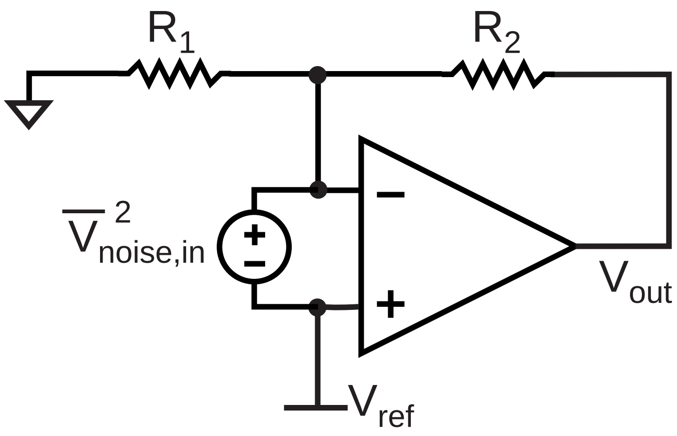
- Large voltage swing required at charge pump output
  - Must support full range of VCO input
- Non-ideal behavior of inductors (for third order G(f) implementations)
  - Hard to realize large inductor values
  - Self resonance of inductor reduces high frequency attenuation



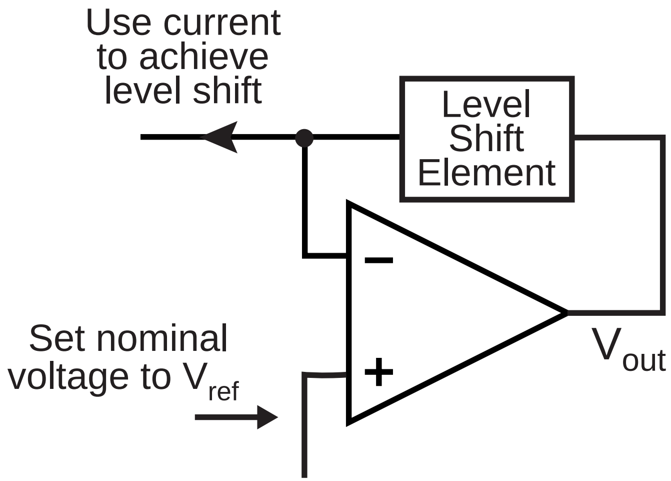
Alternative: active loop filter implementation

Guidelines for Active Loop Filter Design

- **Use topologies with unity gain feedback in the opamp**
  - **Minimizes influence of opamp noise**



- **Perform level shifting in feedback of opamp**
  - **Fixes voltage at charge pump output**

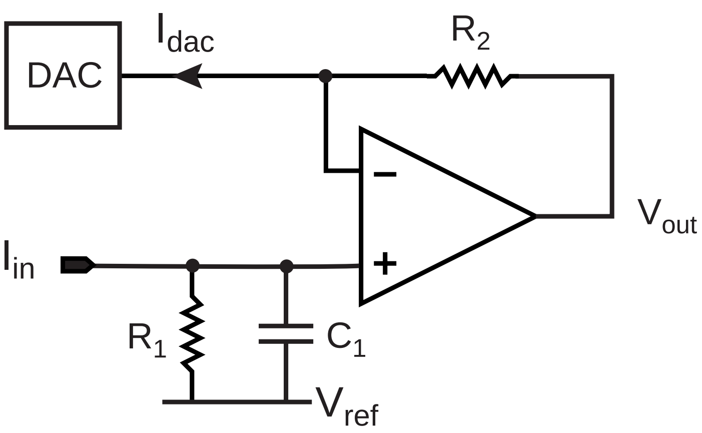


Use current to achieve level shift
Level Shift Element
−
+
V_{out}
Set nominal voltage to V_{ref}

- **Prevent fast edges from directly reaching opamp inputs**
  - **Will otherwise cause opamp to be driven into nonlinear region of operation**

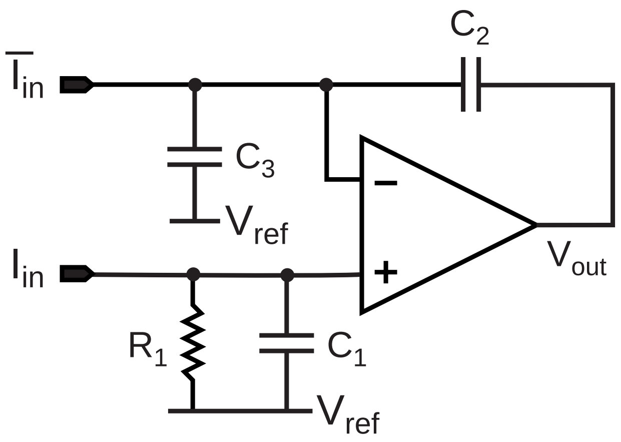
Active Topologies To Realize a Second Order PLL

Type I, Order 2



$$\frac{V_{out}}{I_{in}} = \frac{R_1}{1+sR_1C_1}$$

Type II, Order 2

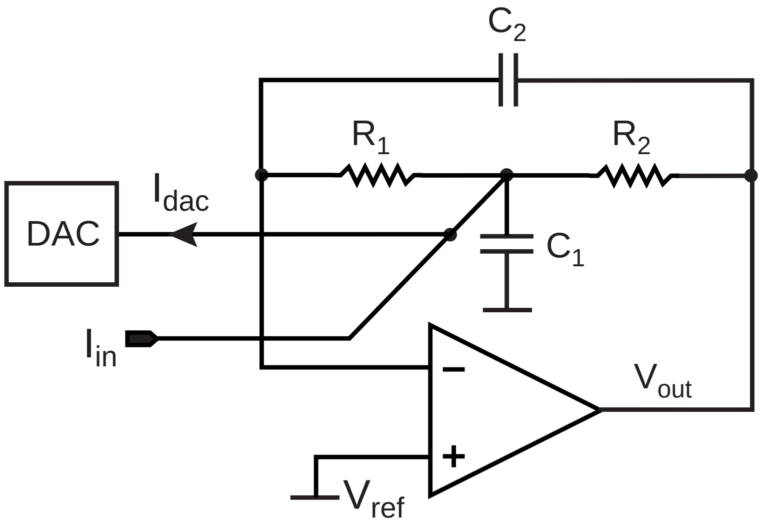


$$\frac{V_{out}}{I_{in}} = \frac{1+sR_1(C_1+C_2+C_3)}{sC_2(1+sR_1C_1)}$$

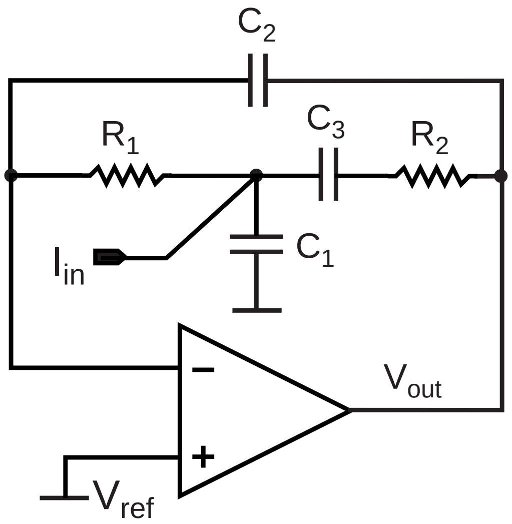
- **Follows guidelines from previous slide**
- **Charge pump output is terminated directly with a high Q capacitor**
  - **Smooths fast edges from charge pump before they reach the opamp input(s)**

Active Topologies To Realize a Third Order PLL

Type I, Order 3



Type II, Order 3



$$\frac{V_{out}}{I_{in}} = \frac{-R_2}{1+s(R_1+R_2)C_2+s^2R_1R_2C_1C_2}$$

$$\frac{V_{out}}{I_{in}} = \frac{-1}{s(C_1+C_2)}\,\frac{1+sR_2C_3}{1+sC_{||}(R_1(1+C_1/C_3)+R_2)+s^2R_1R_2C_1C_{||}}$$

where $C_{||}= C_2C_3/(C_2+C_3)$

- **Follows active implementation guidelines from a few slides ago**

Example Design

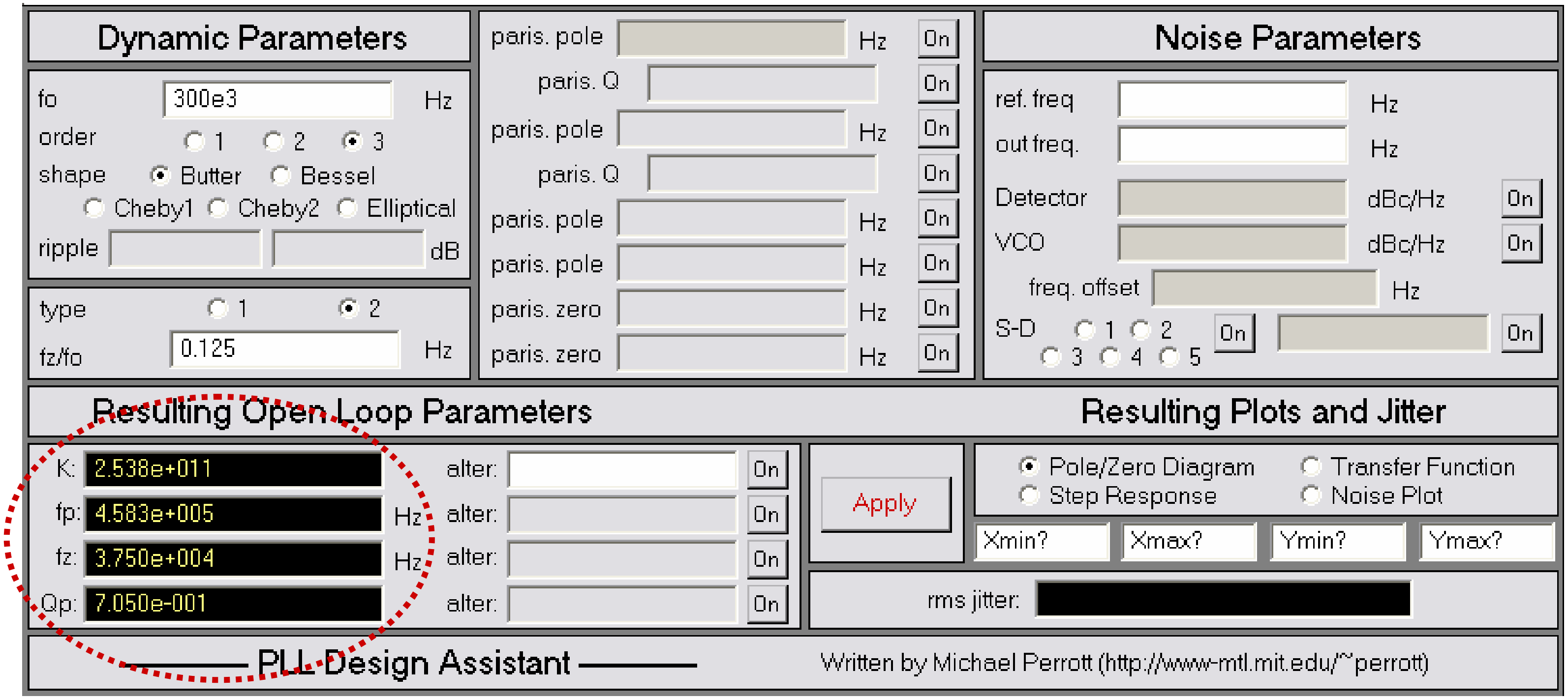
- Type II, 3rd order, Butterworth, f_o = 300kHz, f_z/f_o = 0.125
  - No parasitic poles
- Required loop filter transfer function can be found from table:

$$\Rightarrow H(s) = \frac{K_{LF}\left(1+\frac{s}{w_z}\right)}{s\left(1+\frac{s}{w_pQ_p}+\left(\frac{s}{w_p}\right)^2\right)} \quad \text{where}$$

$$K_{LF} = \frac{N_{nom}K}{\alpha I_{cp}K_v}$$

Use PLL Design Assistant to Calculate Parameters

$$H(s) = \frac{K_{LF}\left(1+\frac{s}{w_z}\right)}{s\left(1+\frac{s}{w_pQ_p}+\left(\frac{s}{w_p}\right)^2\right)} \quad \text{where} \quad K_{LF} = \frac{N_{nom}K}{\alpha I_{cp}K_v}$$



Dynamic Parameters

fo 300e3 Hz

order ○ 1 ○ 2 ● 3

shape ● Butter ○ Bessel ○ Cheby1 ○ Cheby2 ○ Elliptical

ripple dB

type ○ 1 ● 2

fz/fo 0.125 Hz

Resulting Open Loop Parameters

K: 2.538e+011

fp: 4.583e+005 Hz

fz: 3.750e+004 Hz

Qp: 7.050e-001

Resulting Plots and Jitter

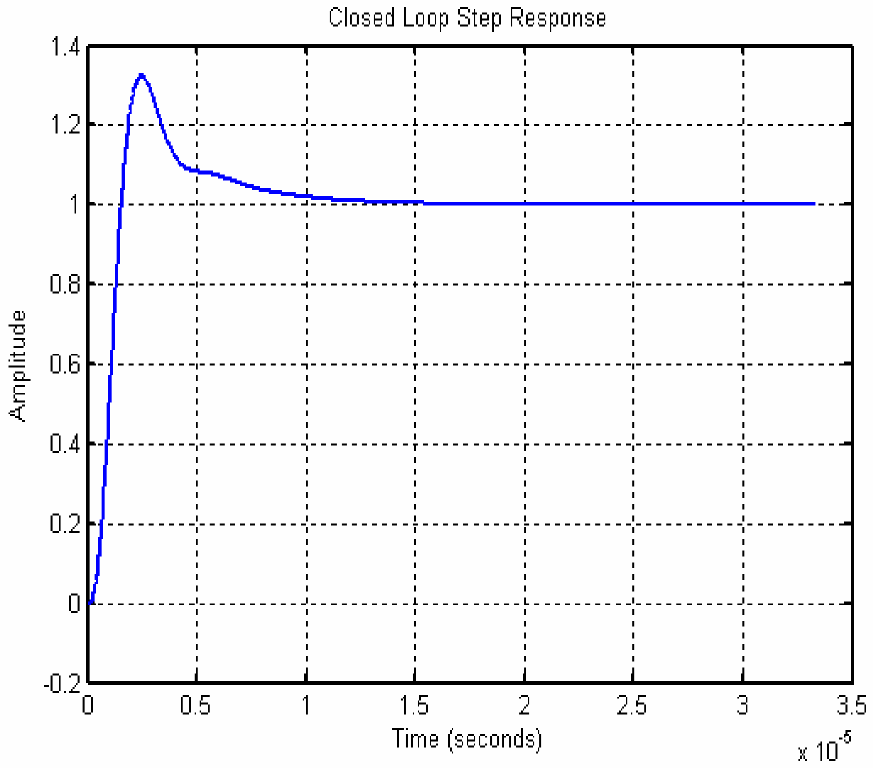
● Pole/Zero Diagram ○ Transfer Function ○ Step Response ○ Noise Plot

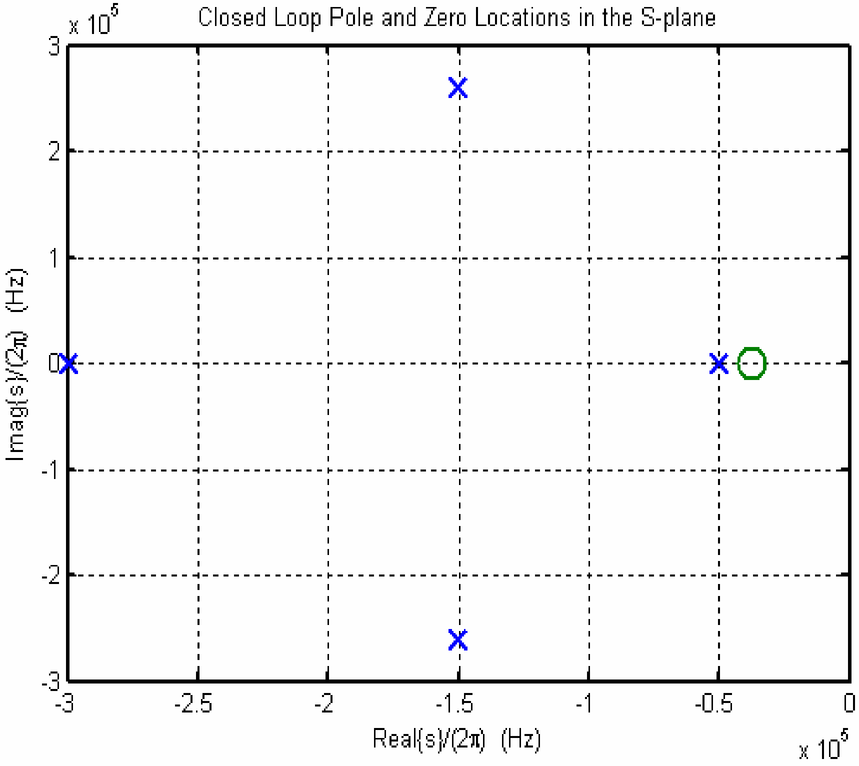
rms jitter:

PLL Design Assistant

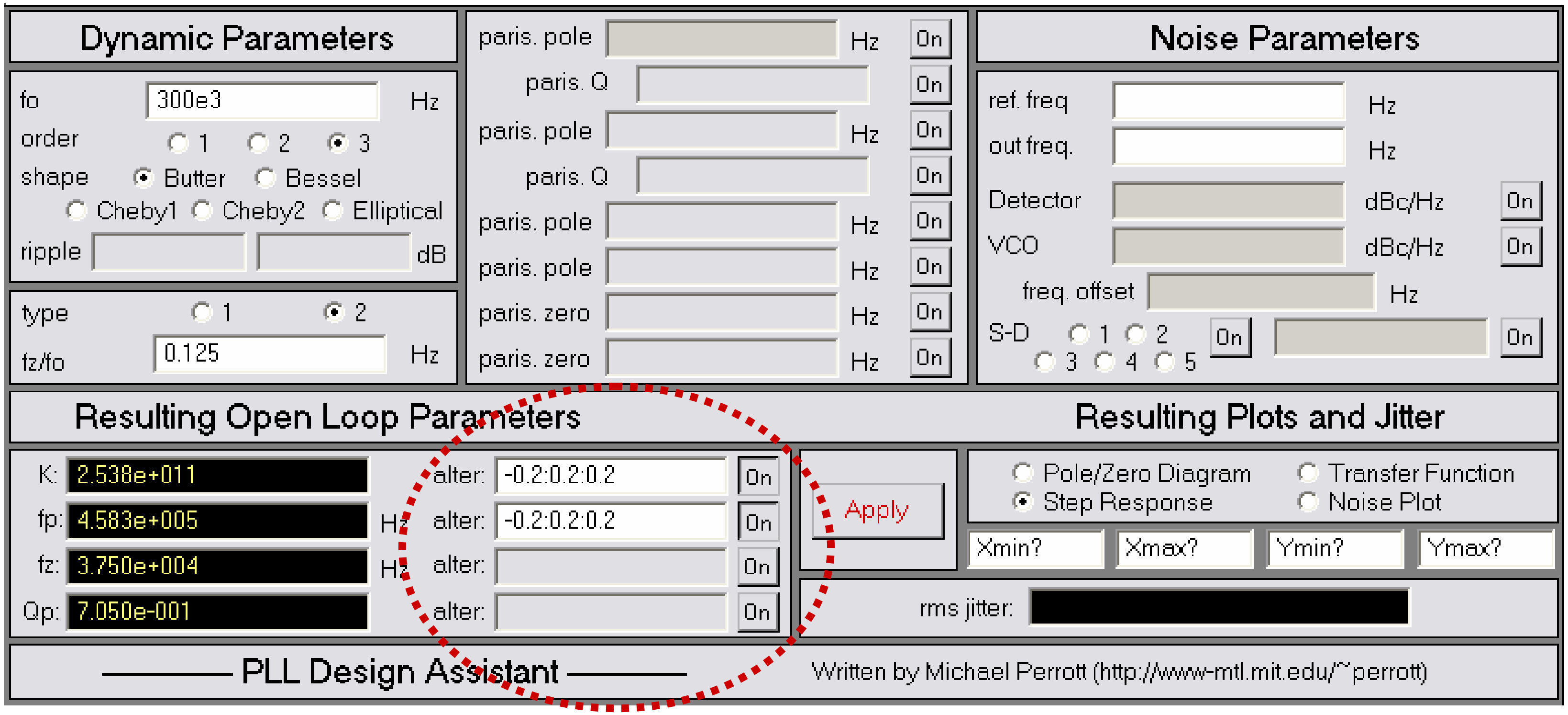
Written by Michael Perrott (http://www-mtl.mit.edu/~perrott)

Resulting Step Response and Pole/Zero Diagram





Impact of Open Loop Parameter Variations



Dynamic Parameters

fo: 300e3 Hz

order: ○ 1 ○ 2 ● 3

shape: ● Butter ○ Bessel ○ Cheby1 ○ Cheby2 ○ Elliptical

ripple: dB

type: ○ 1 ● 2

fz/fo: 0.125 Hz

Resulting Open Loop Parameters

K: 2.538e+011

fp: 4.583e+005 Hz

fz: 3.750e+004 Hz

Qp: 7.050e-001

alter: -0.2:0.2:0.2 On

alter: -0.2:0.2:0.2 On

alter: On

alter: On

Resulting Plots and Jitter

○ Pole/Zero Diagram ○ Transfer Function
● Step Response ○ Noise Plot

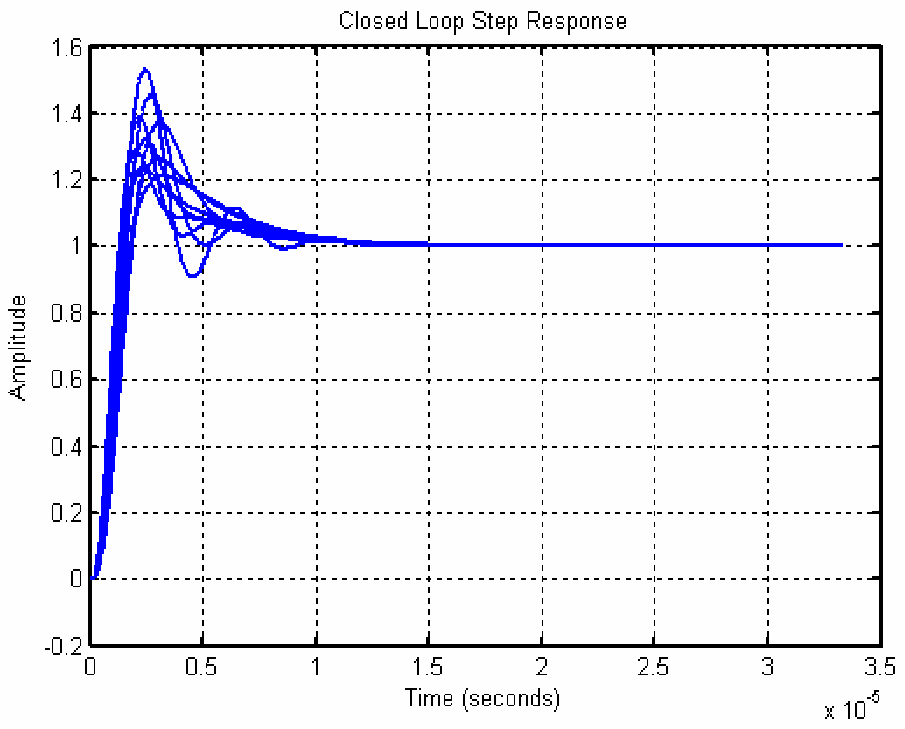
rms jitter:

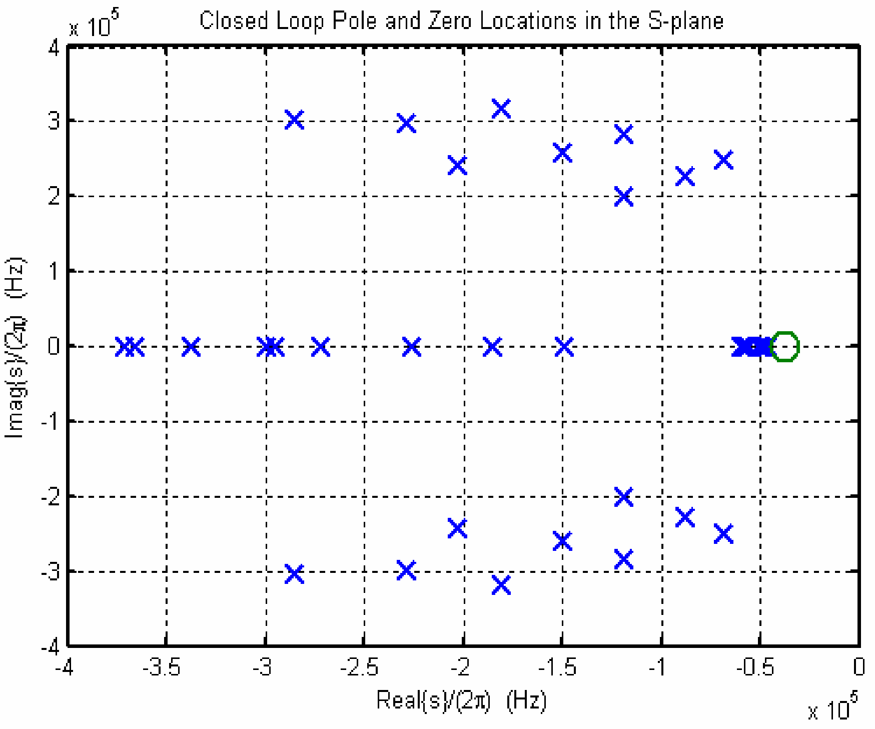
—— PLL Design Assistant ——

Written by Michael Perrott (http://www-mtl.mit.edu/~perrott)

- **Open loop parameter variations can be directly entered into tool**

Resulting Step Responses and Pole/Zero Diagrams



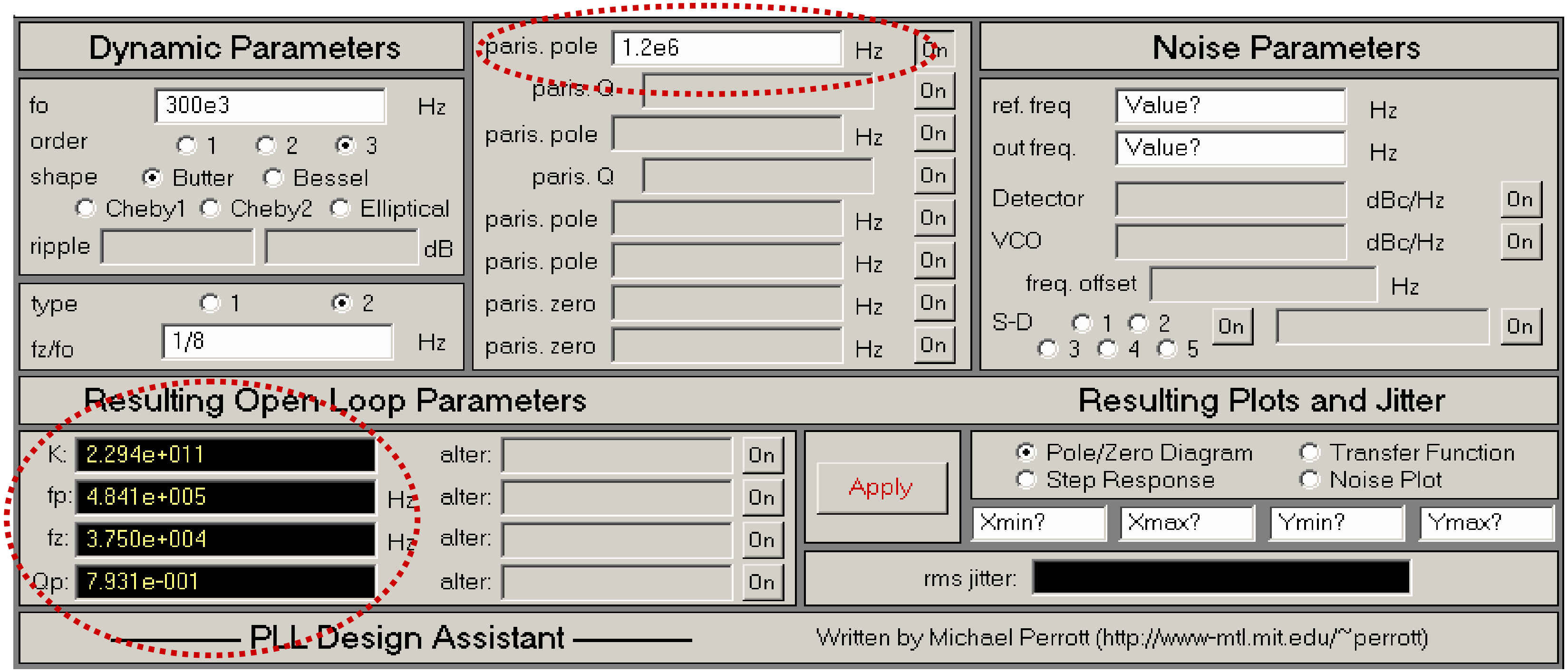


- Impact of variations on the loop dynamics can be visualized instantly and taken into account at early stage of design

Design with Parasitic Pole

- Include a parasitic pole at nominal value f_{p1} = 1.2MHz

$$H(s) = \frac{K_{LF}\left(1+\frac{s}{w_z}\right)}{s\left(1+\frac{s}{w_pQ_p}+\left(\frac{s}{w_p}\right)^2\right)\left(1+\frac{s}{w_{p1}}\right)}$$



Dynamic Parameters

fo: 300e3 Hz

order: ○ 1 ○ 2 ● 3

shape: ● Butter ○ Bessel ○ Cheby1 ○ Cheby2 ○ Elliptical

type: ○ 1 ● 2

fz/fo: 1/8 Hz

paris. pole: 1.2e6 Hz

Noise Parameters

ref. freq: Value? Hz

out freq.: Value? Hz

Resulting Open Loop Parameters

K: 2.294e+011

fp: 4.841e+005 Hz

fz: 3.750e+004 Hz

Qp: 7.931e-001

Resulting Plots and Jitter

● Pole/Zero Diagram ○ Transfer Function ○ Step Response ○ Noise Plot

Xmin? Xmax? Ymin? Ymax?

rms jitter:

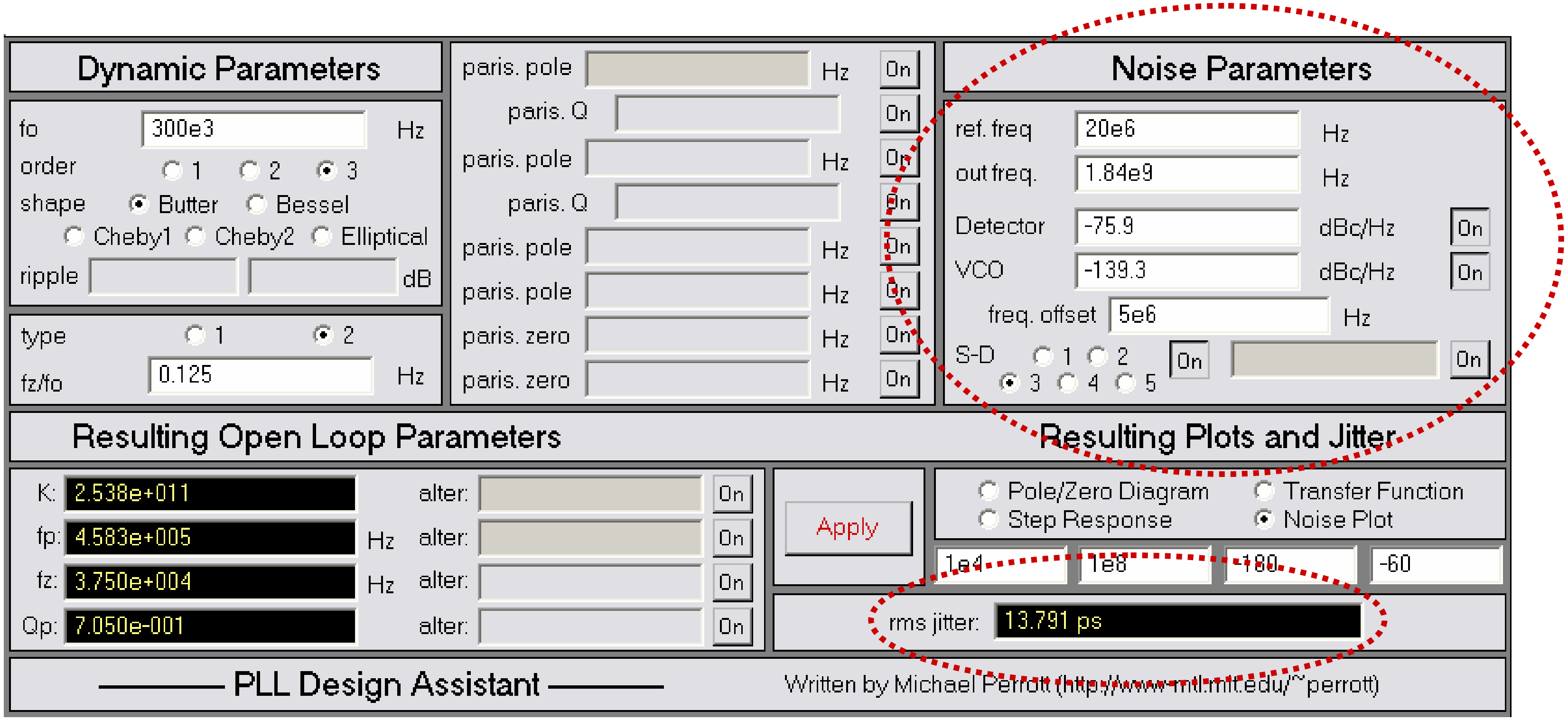
PLL Design Assistant

Written by Michael Perrott (http://www-mtl.mit.edu/~perrott)

⇨ K, f_p and Q_p are adjusted to obtain the same dominant pole locations

Noise Estimation

- Phase noise plots can be easily obtained
  - Jitter calculated by integrating over frequency range



Dynamic Parameters

fo 300e3 Hz

order ○ 1 ○ 2 ● 3

shape ● Butter ○ Bessel

○ Cheby1 ○ Cheby2 ○ Elliptical

ripple dB

type ○ 1 ● 2

fz/fo 0.125 Hz

paris. pole Hz On

paris. Q On

paris. pole Hz On

paris. Q On

paris. pole Hz On

paris. pole Hz On

paris. zero Hz On

paris. zero Hz On

Noise Parameters

ref. freq 20e6 Hz

out freq. 1.84e9 Hz

Detector -75.9 dBc/Hz On

VCO -139.3 dBc/Hz On

freq. offset 5e6 Hz

S-D ○ 1 ○ 2 ● 3 ○ 4 ○ 5 On On

Resulting Open Loop Parameters

K: 2.538e+011 alter: On

fp: 4.583e+005 Hz alter: On

fz: 3.750e+004 Hz alter: On

Qp: 7.050e-001 alter: On

Apply

Resulting Plots and Jitter

○ Pole/Zero Diagram ○ Transfer Function

○ Step Response ● Noise Plot

1e4 1e8 -180 -60

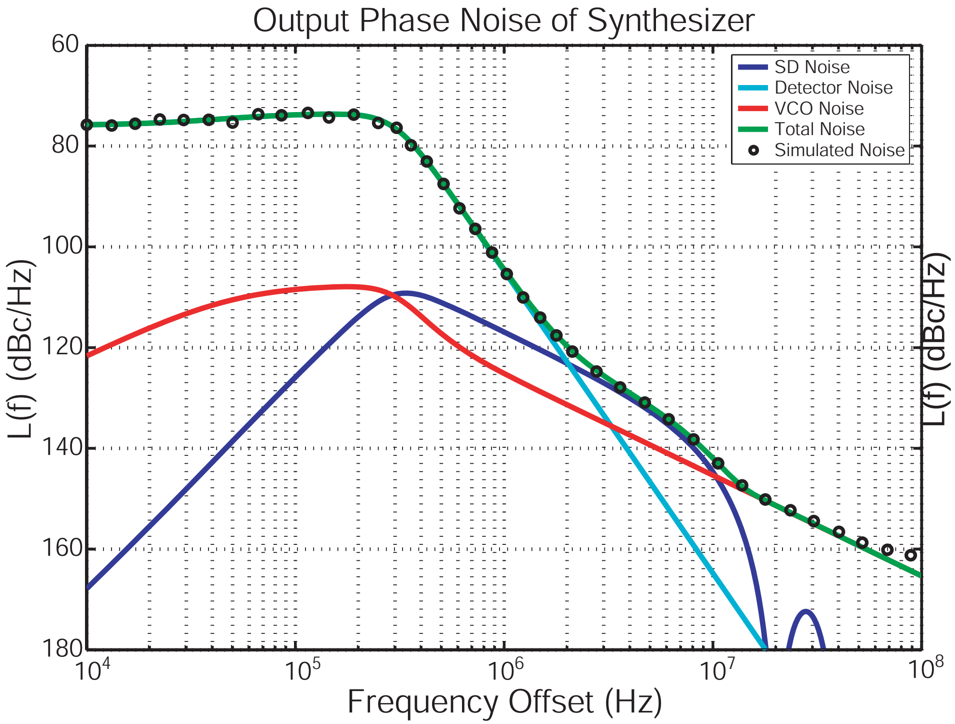
rms jitter: 13.791 ps

PLL Design Assistant

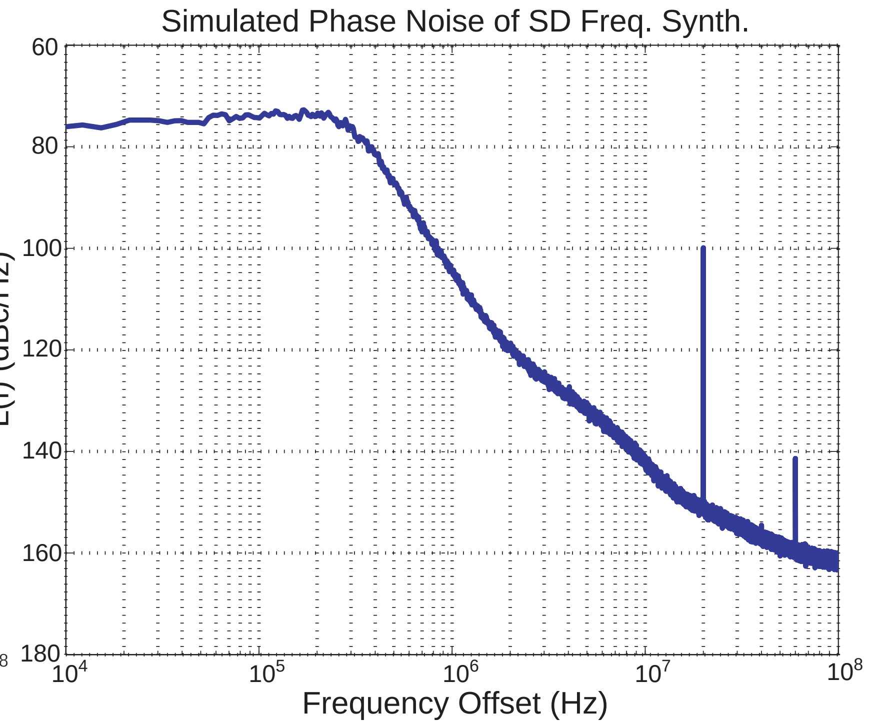
Written by Michael Perrott (http://www-mtl.mit.edu/~perrott)

Calculated Versus Simulated Phase Noise Spectrum

Without parasitic pole:

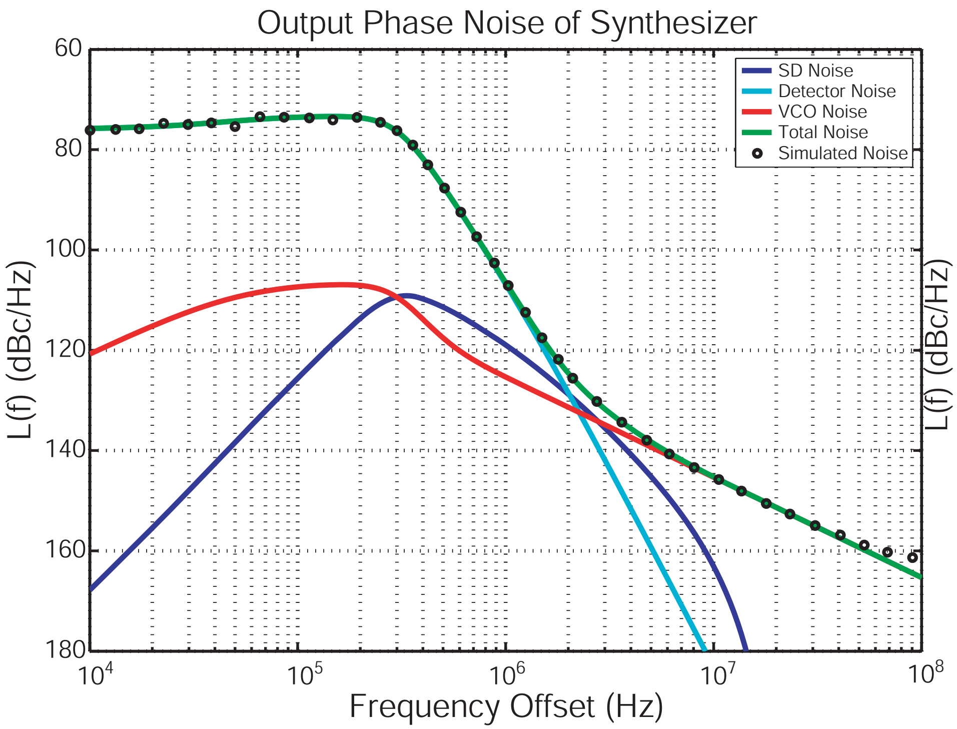

Output Phase Noise of Synthesizer

RMS jitter = 13.791ps

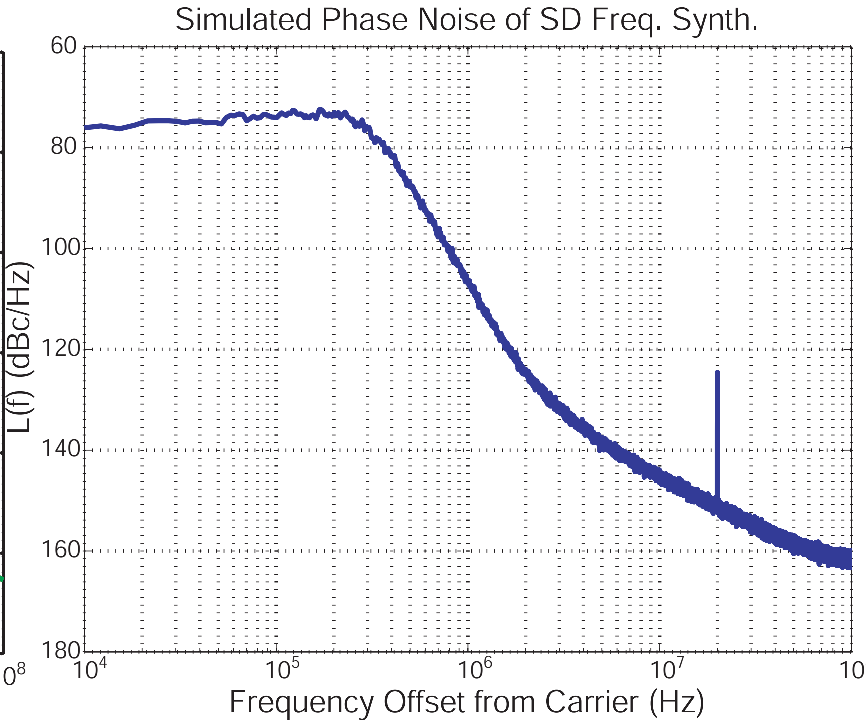

Simulated Phase Noise of SD Freq. Synth.

Calculated Versus Simulated Phase Noise Spectrum

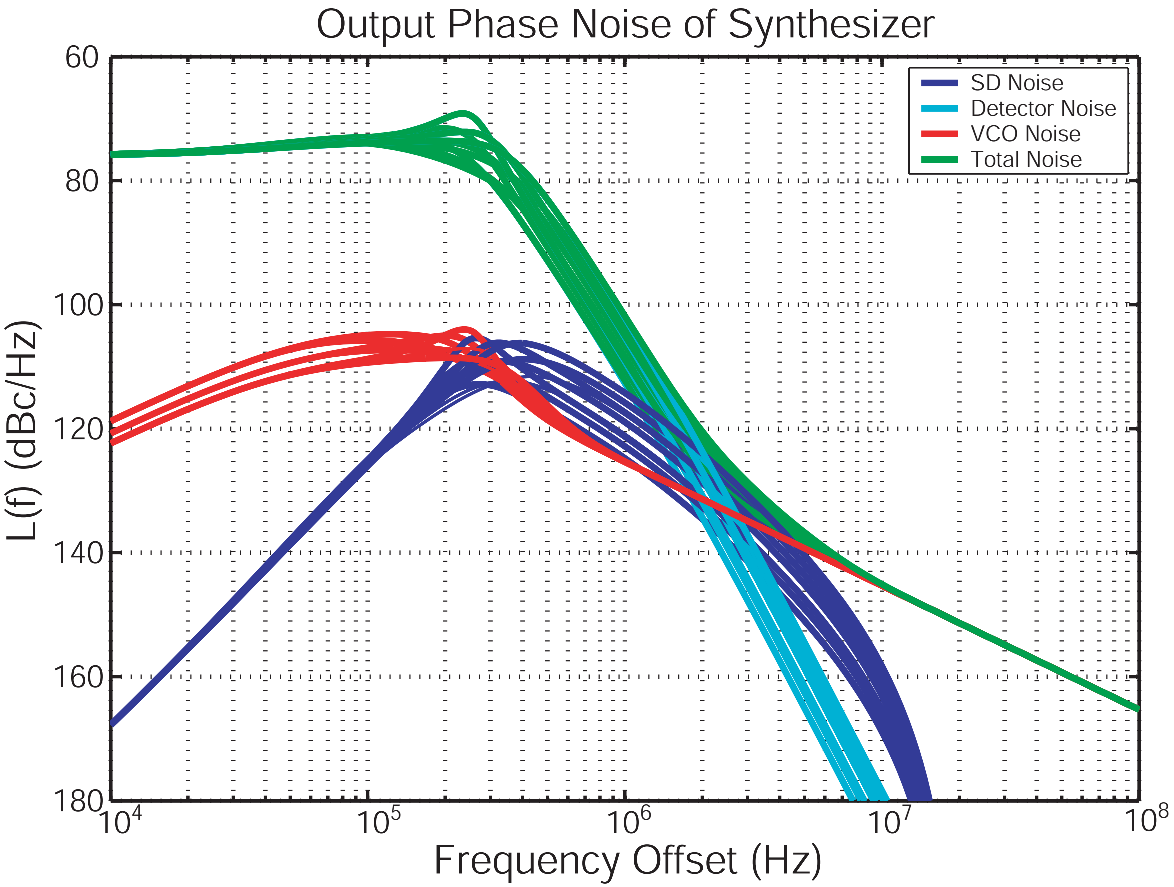
With parasitic pole at 1.2 MHz:



RMS jitter = 14.057ps



Noise under Open Loop Parameter Variations



RMS jitter = 11.678ps (min), 18.211ps (max)

- **Impact of open loop parameter variations on phase noise and jitter can be visualized immediately**

Conclusion

- **CAD-based closed loop design approach facilitates:**
  - **Accurate control of closed loop dynamics**
    - Bandwidth, Order, Shape, Type
  - **Straightforward design of higher order PLL’s**
  - **Direct assessment of impact of parasitic poles/zeros**
- **Techniques implemented in a GUI-based CAD tool**

- **Beginners can quickly come up to speed in designing PLL’s**
- **Experienced designers can quickly evaluate the performance of different PLL configurations**