UNLV | Howard R. Hughes College of ENGINEERING

# Use of an Over-damped PLL in place of DLL in SDRAM

Min Long Lan

ECG 721

11/23/2015

# Outline

- SDRAM Introduction
- Clock Synchronization
- Why is DLL used
- Why not use PLL
- Design of PLL for SDRAM
- Simulation
- Conclusion

# SDRAM Introduction

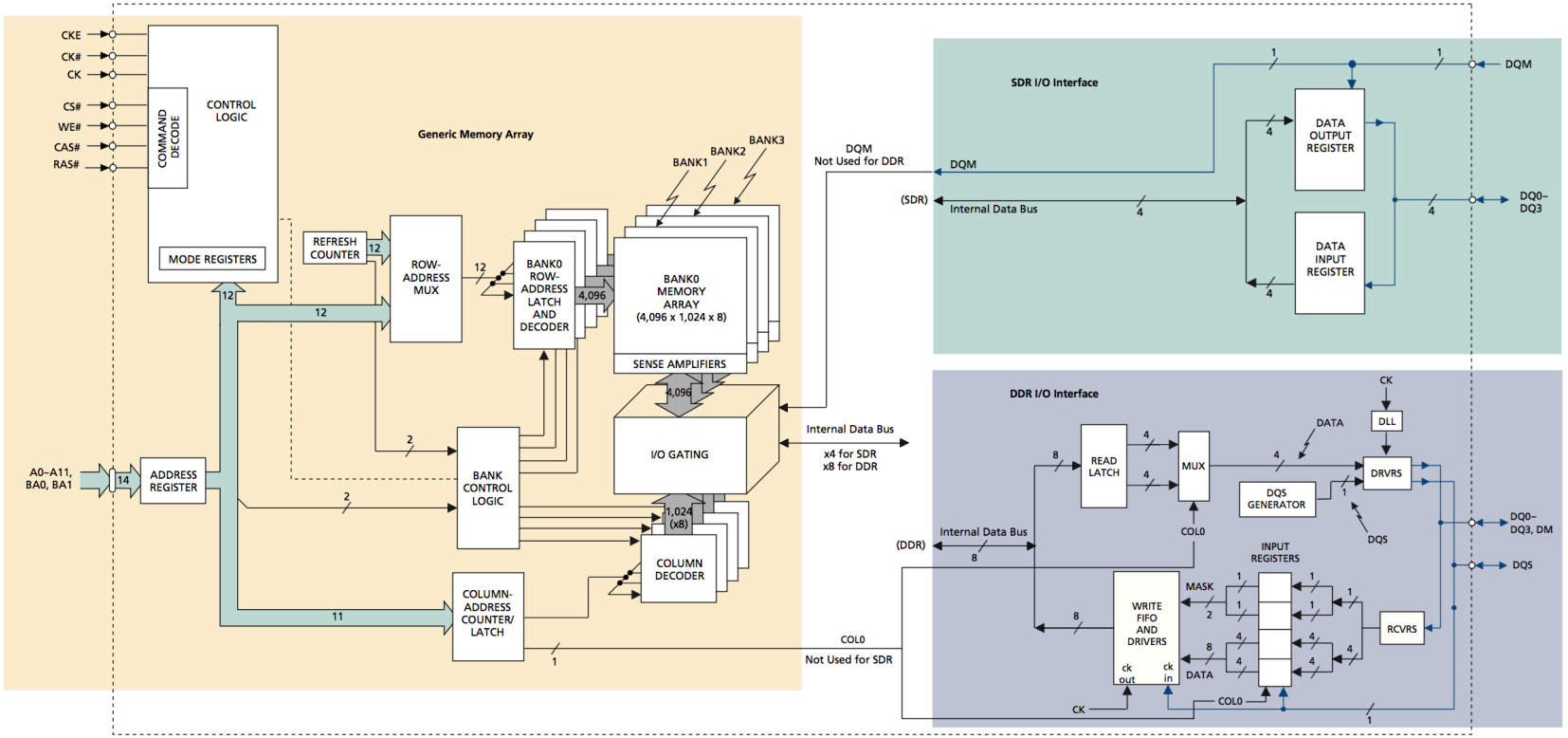



**Fig. 1** Functional Block Diagram
2 Meg x 4 Memory Array with SDR and DDR Interface [4]

# Synchronized DRAM vs DRAM

- Addition of a clock signal
- Commands vs Strobes
- DRAM: turn strobe signals on and off with timing
- SDRAM: execute commands in sequence

# Clock Synchronization

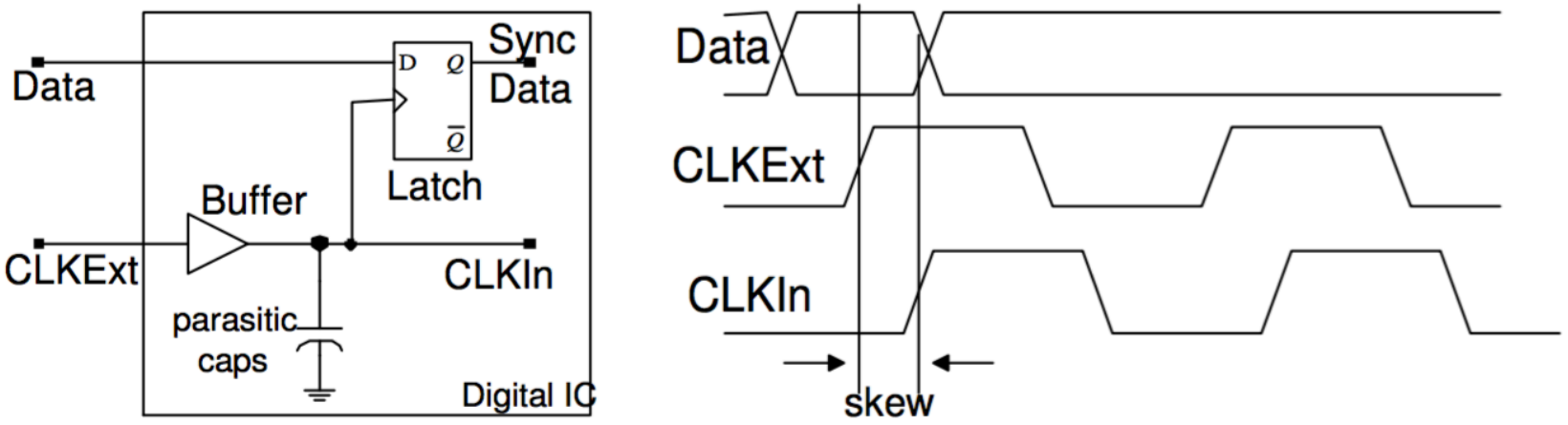



Figure 1.1 Clock skew in an integrated circuit [3]

# Why is DLL used

- Since external clock and internal (buffered) clock are same in frequency, simply add delay to align them in phase
- Add variable delay with small discrete unit delay element
- DLL is a 1st order system
  - Easier to design
  - Stable
- No feedback noise from phase error
  - Better jitter performance

# Delay Locked Loop

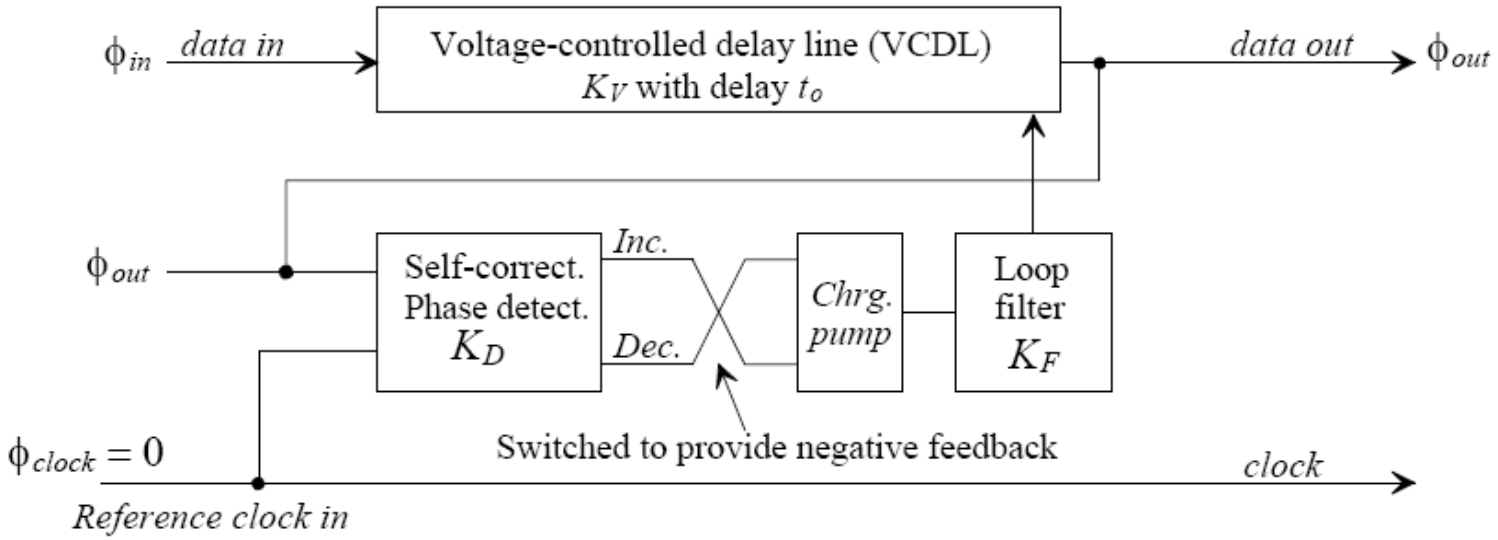



**Figure 19.53** Block diagram of a delay-locked loop. [1]

# PLL

- 2nd order system, increased design complexity
- Take very long time acquire lock, need to pull in the frequency first
- Need to balance between lock time and jitter
- Need to balance between gain (range) and jitter
- Higher damping factor, better stability, longer pull-in time

# Phase Locked Loop

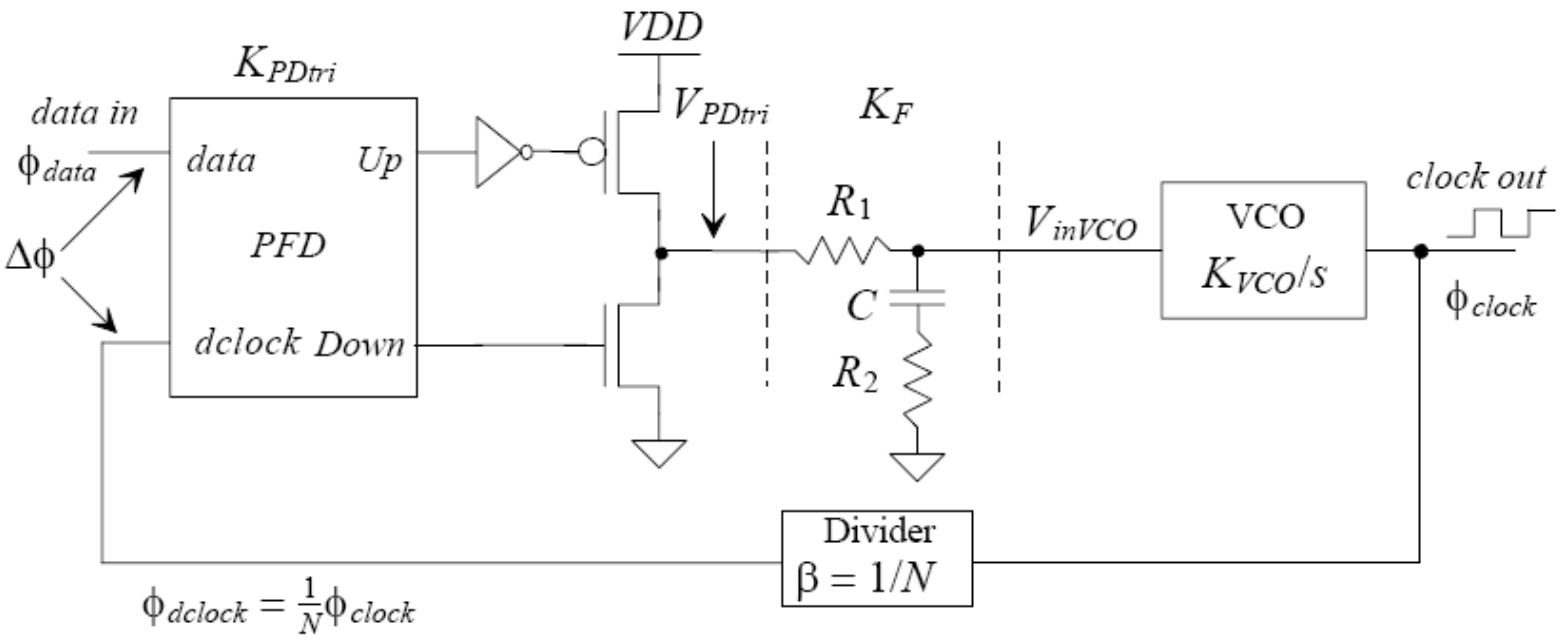



**Figure 19.32** Block diagram of a DPLL using a sequential phase detector (PFD). [1]

# DLL in SDRAM

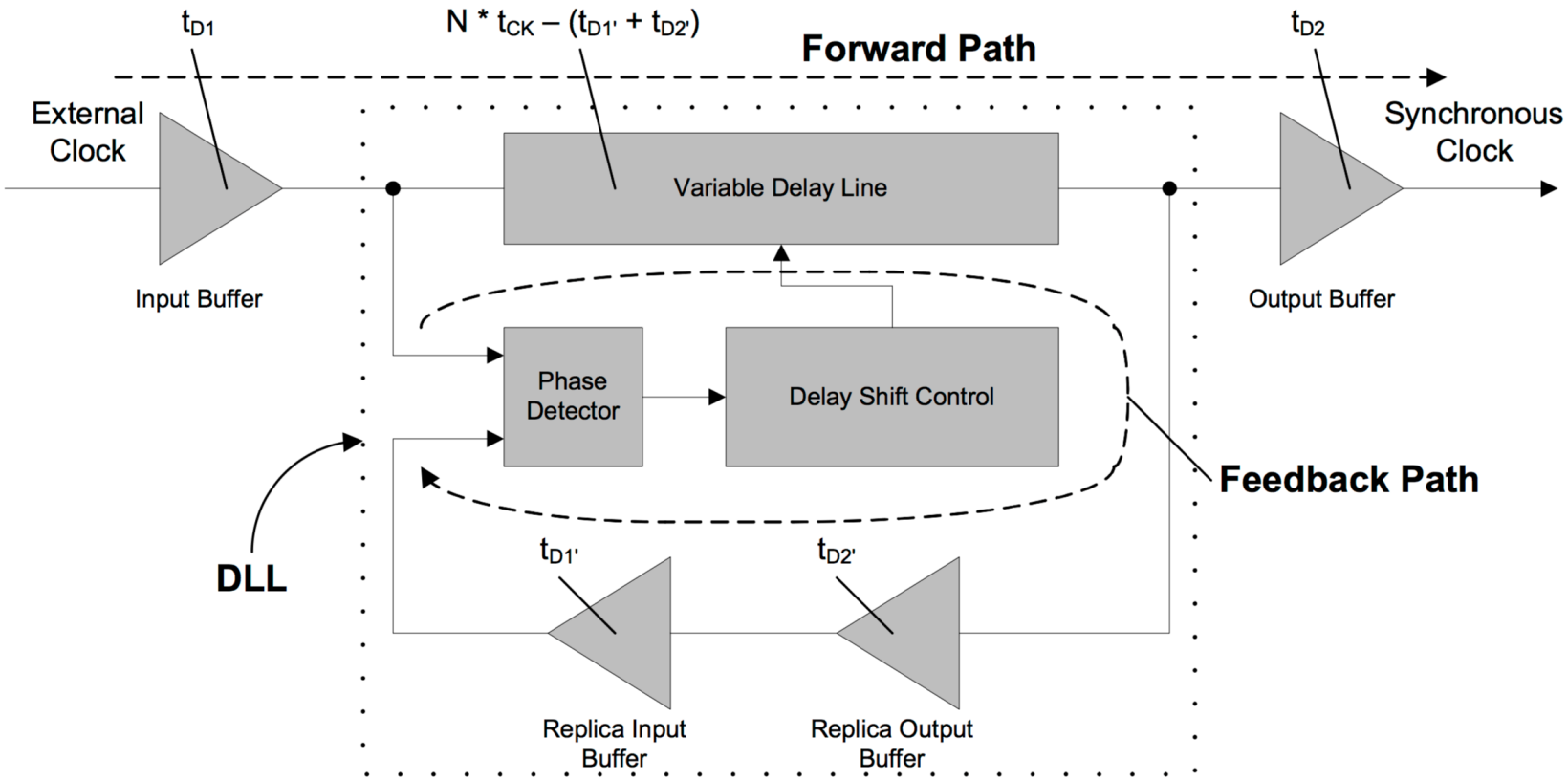
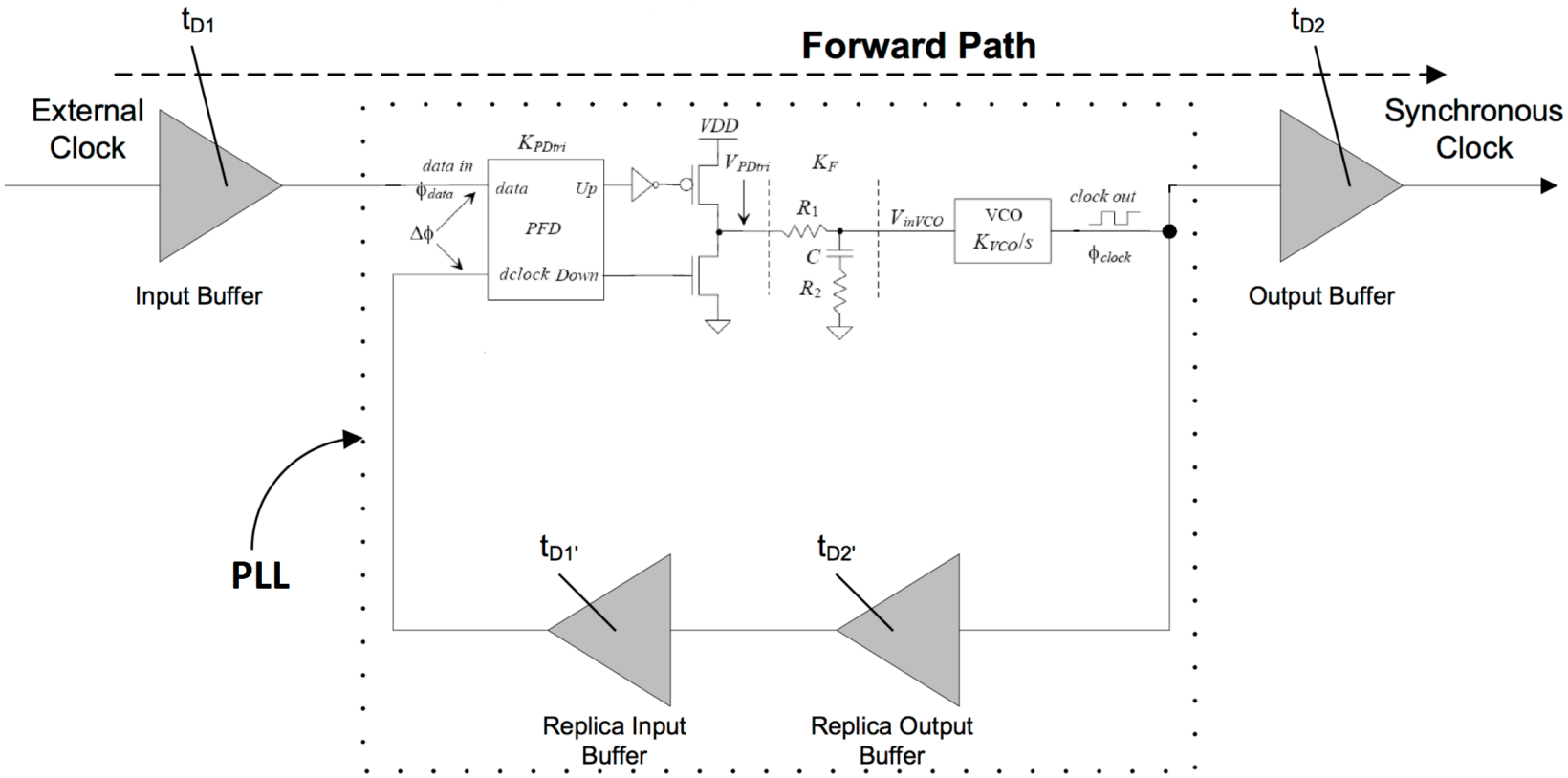



**Figure 1.2** DLL Block Diagram [2]

# Use of PLL in SDRAM



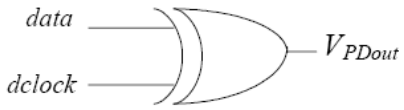
# Use of PLL in SDRAM

Design consideration

- Goal is to align external clock with output clock
- Does not depend on data transitions
  - Clock_in (data_in) are always oscillating.

# Phase Detector

- XOR Phase Detector




- Phase Frequency Detector

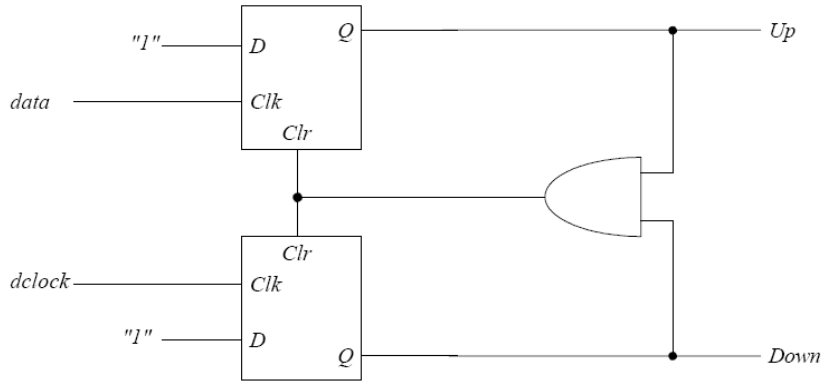
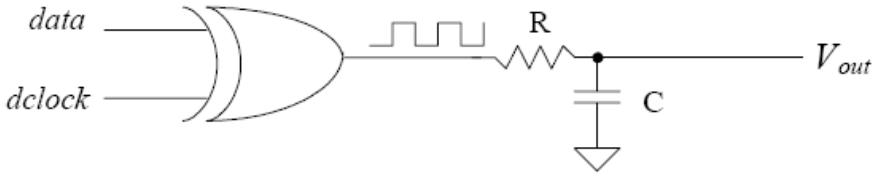


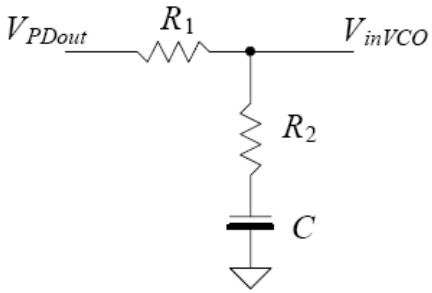

**Figure 19.10** Phase frequency detector (PFD). [1]

# Loop Filter (XOR PD)

- Simple RC Filter




- Passive Lag Filter




- Active PI Filter

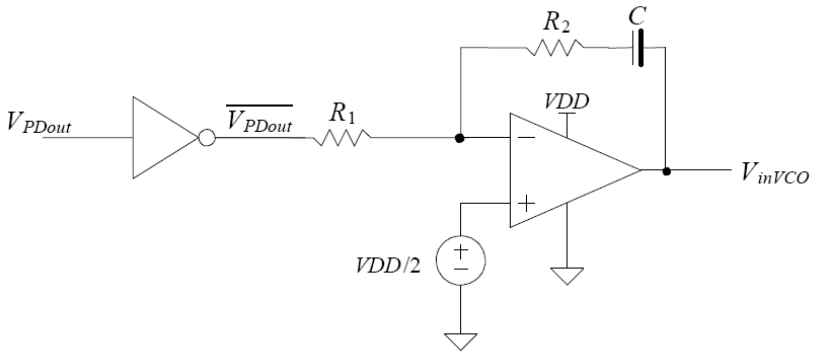



**Figure 19.30** Active-PI loop filter. [1]

# Loop Filter (PFD)

- Tri-State Output
- Charge Pump Output

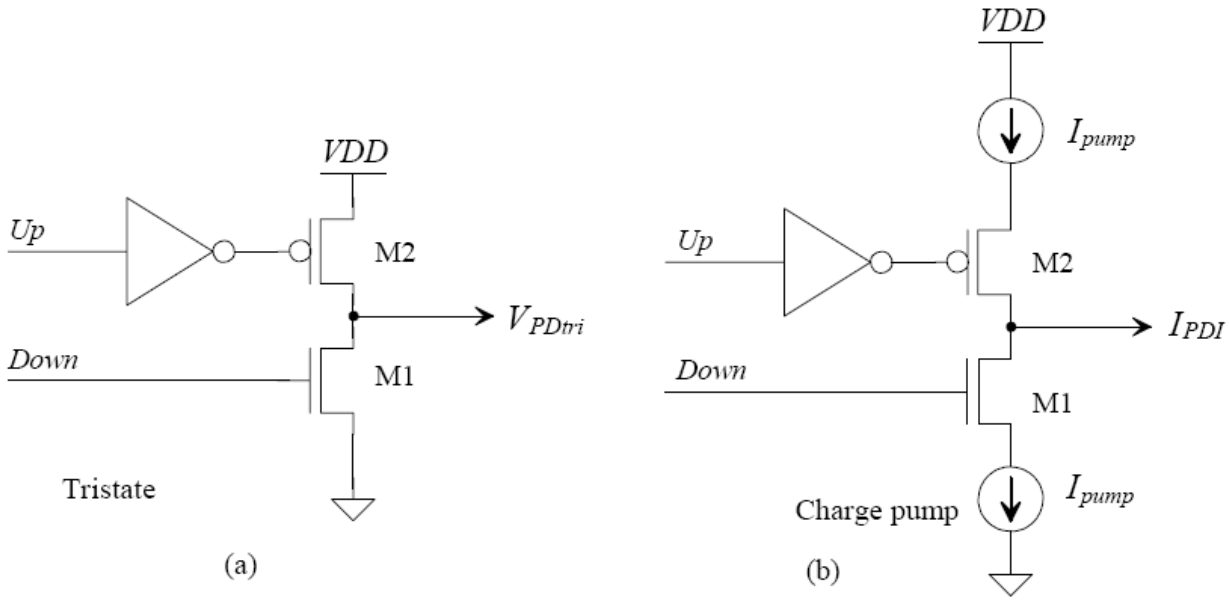



**Figure 19.12** (a) Tri-state and (b) charge pump outputs of the PFD. [1]

# Voltage Controlled Oscillator

- Current-Starved VCO
  - Ring oscillator whose invertors are current limited by voltage controlled current sources.
- Source-Coupled VCO
  - Similar to astable multivibrator, but voltage controlled

# Current-Starved VCO

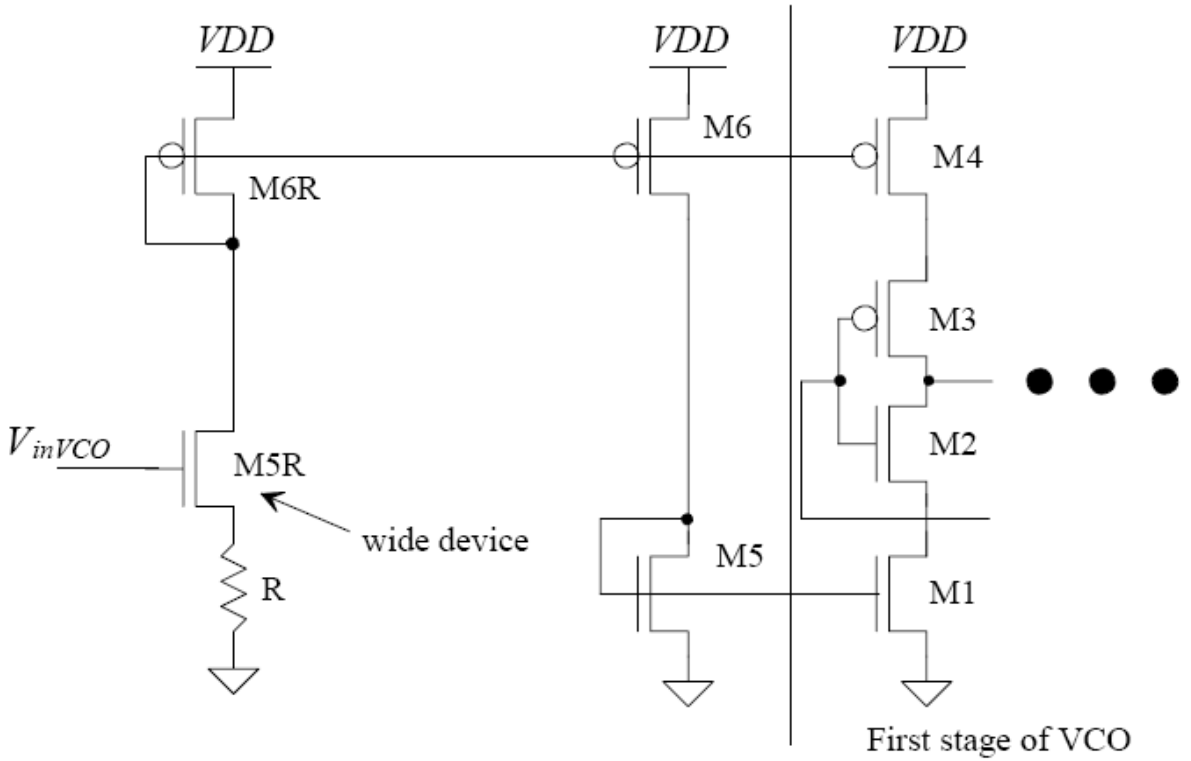



**Figure 19.17** Linearizing the current in a current-starved VCO. [1]

# Current-Starved VCO

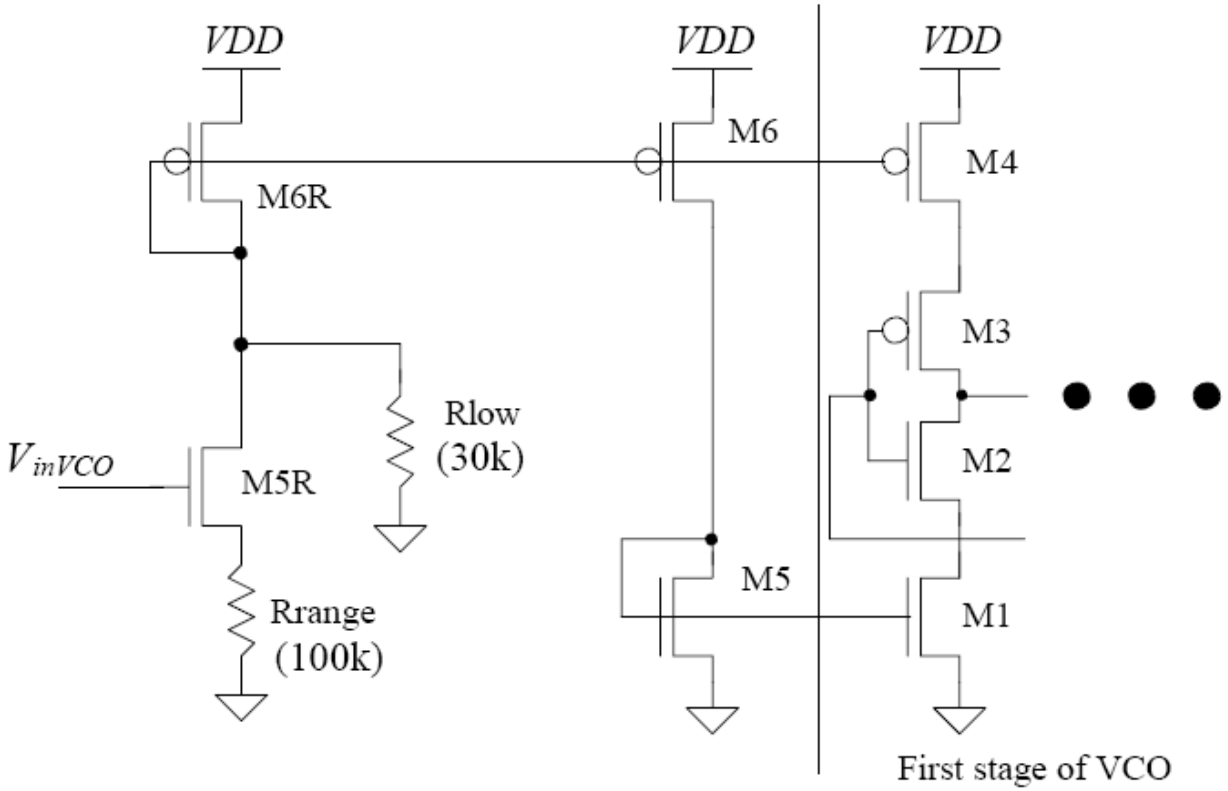



**Figure 19.25** Limiting the current in a current-starved VCO. [1]

# Current-Starved VCO

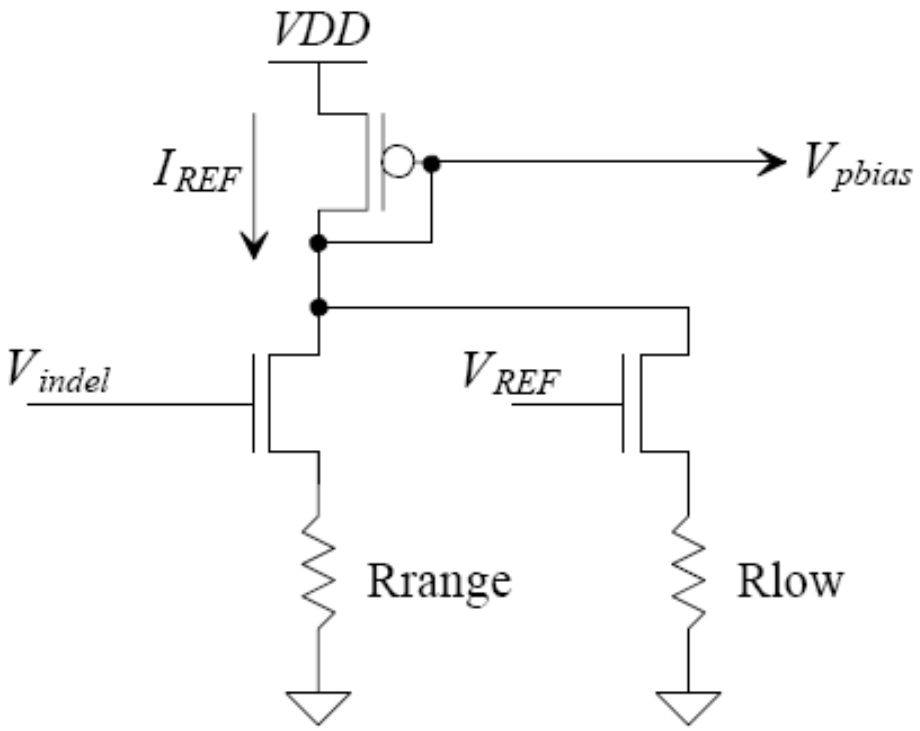



**Figure 19.60** Lowering the current range in the bias circuit. [1]

# Source-Coupled VCO

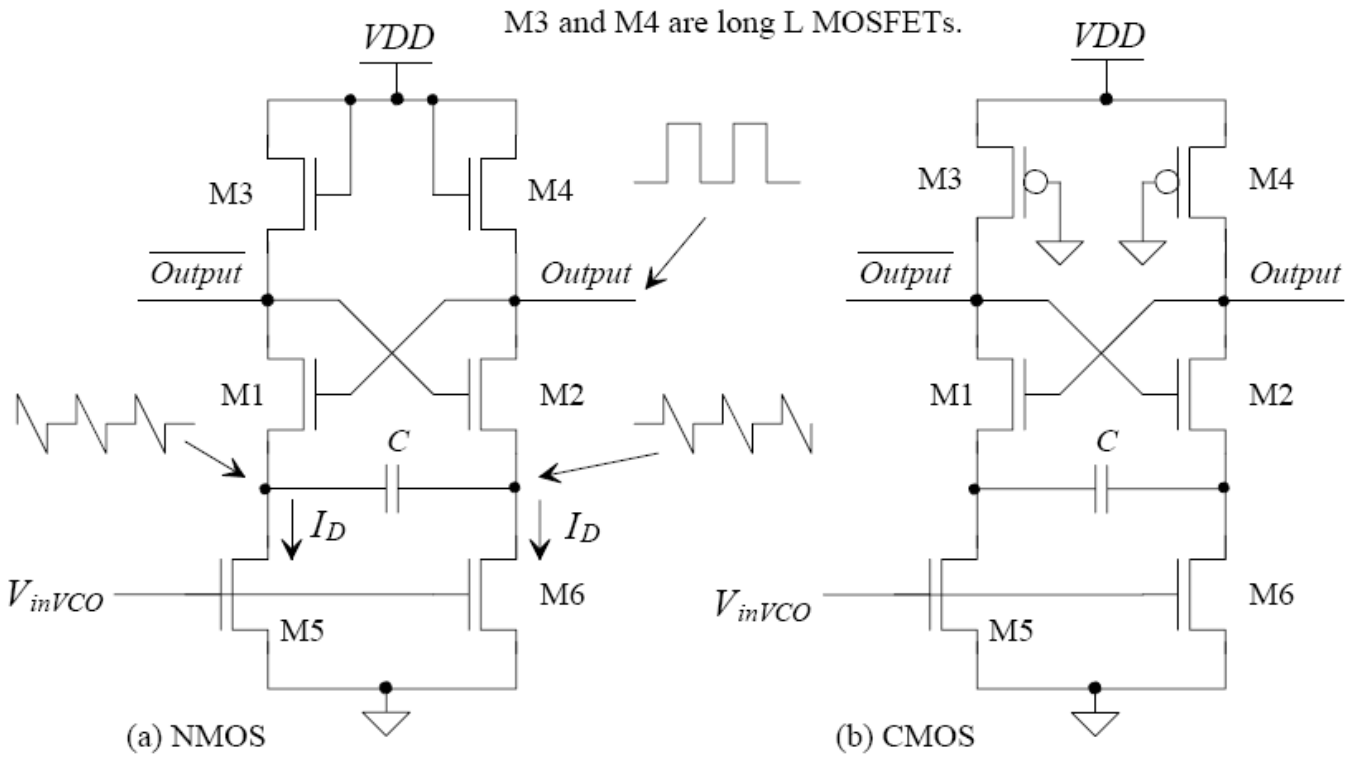



**Figure 19.19** Source coupled voltage-controlled oscillators (also known as source coupled multivibrators). [1]

# Source-Coupled VCO

- Advantages
  - Lower Layout without ring oscillator
  - Able to set center frequency by external (off-chip) capacitor
- Disadvantages
  - Need a capacitor
  - Need to regenerate full logic level

# PLL Design (500 MHz)

- Over-damped: $\zeta = 2$
- Using PFD with charge pump output
- Takes 500 MHz input clock (data)
- Synchronizes clock, not data
- Divider not needed: N=1
- Lock range: 100 MHz

# VCO Design

- Using Current-Starved VCO
- Center frequency: 500 MHz
- Frequency Range: 200 MHz

# VCO Design

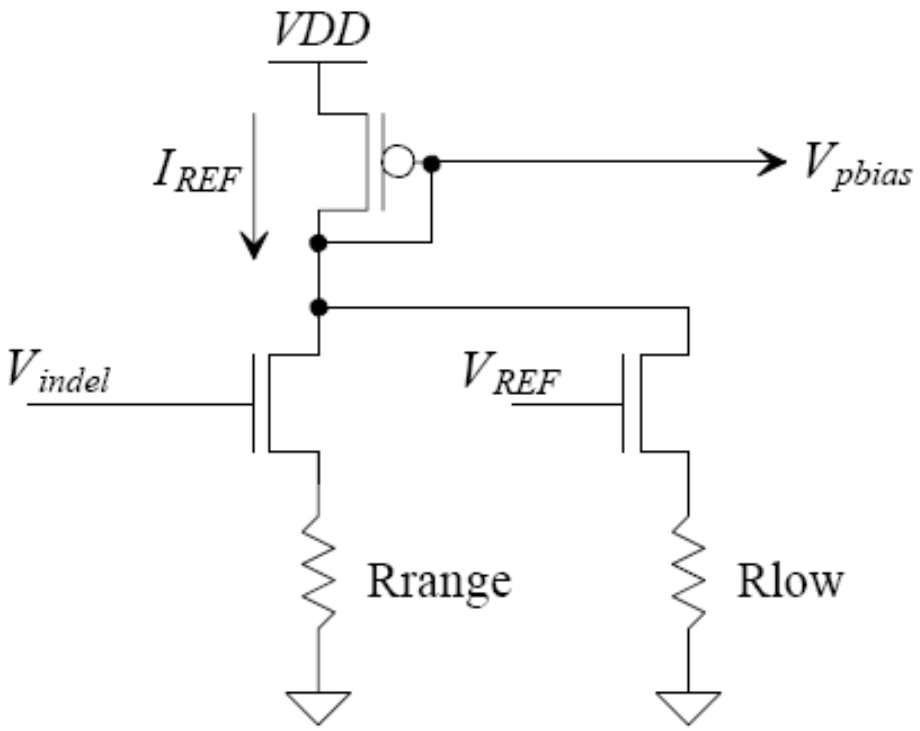
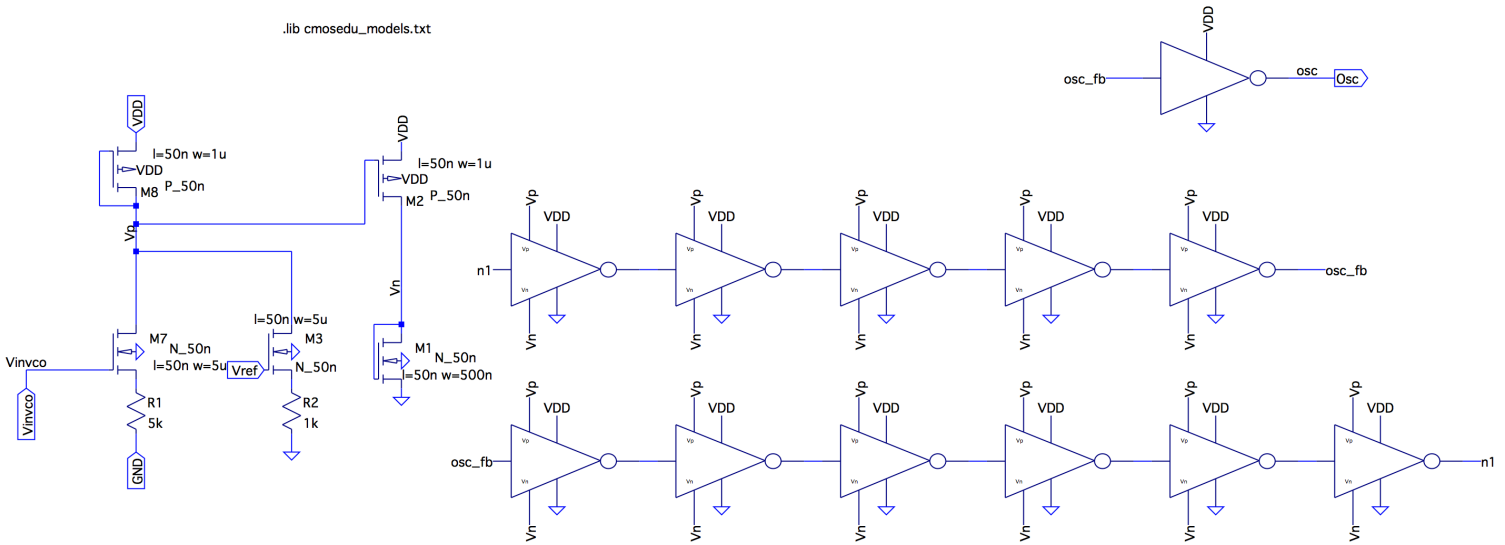



**Figure 19.60** Lowering the current range in the bias circuit. [1]

# VCO Design

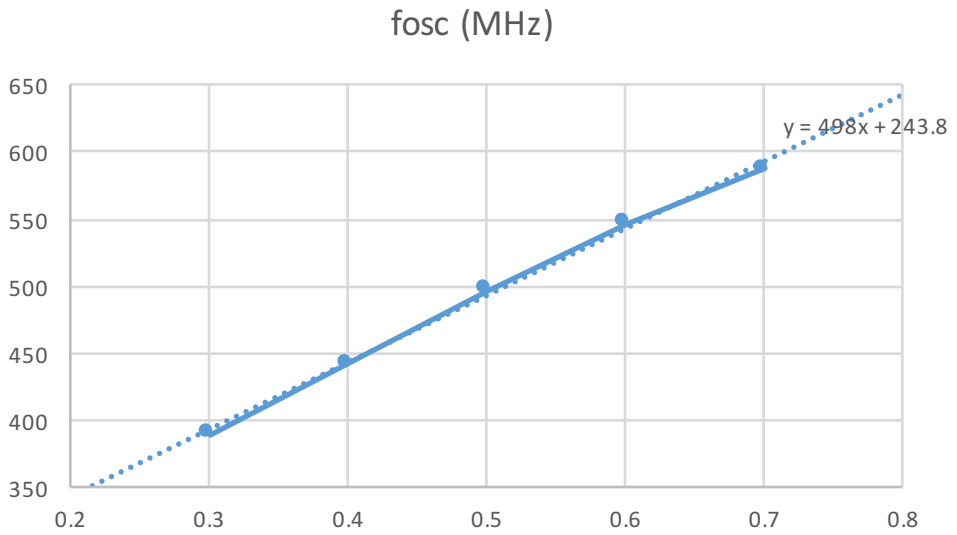
- $I_D = \frac{V_{inVCO} - V_{THN}}{R_{range}} + \frac{V_{REF} - V_{THN}}{R_{low}}$
- $f_{osc} = \frac{1}{N \cdot C_{tot} \cdot VDD}$ (Eq. 19.23) [1]

# VCO Simulation



# VCO Design

| VinVCO | *fosc* (MHz) |
|---|---|
| 0.3 | 390 |
| 0.4 | 443 |
| 0.5 | 497 |
| 0.6 | 547 |
| 0.7 | 587 |




$$K_{VCO} = 2\pi \cdot 498\times10^{6}$$
$$= 3.129\times10^{9}\ radians/V \cdot s$$

# PLL using PFD with CP output

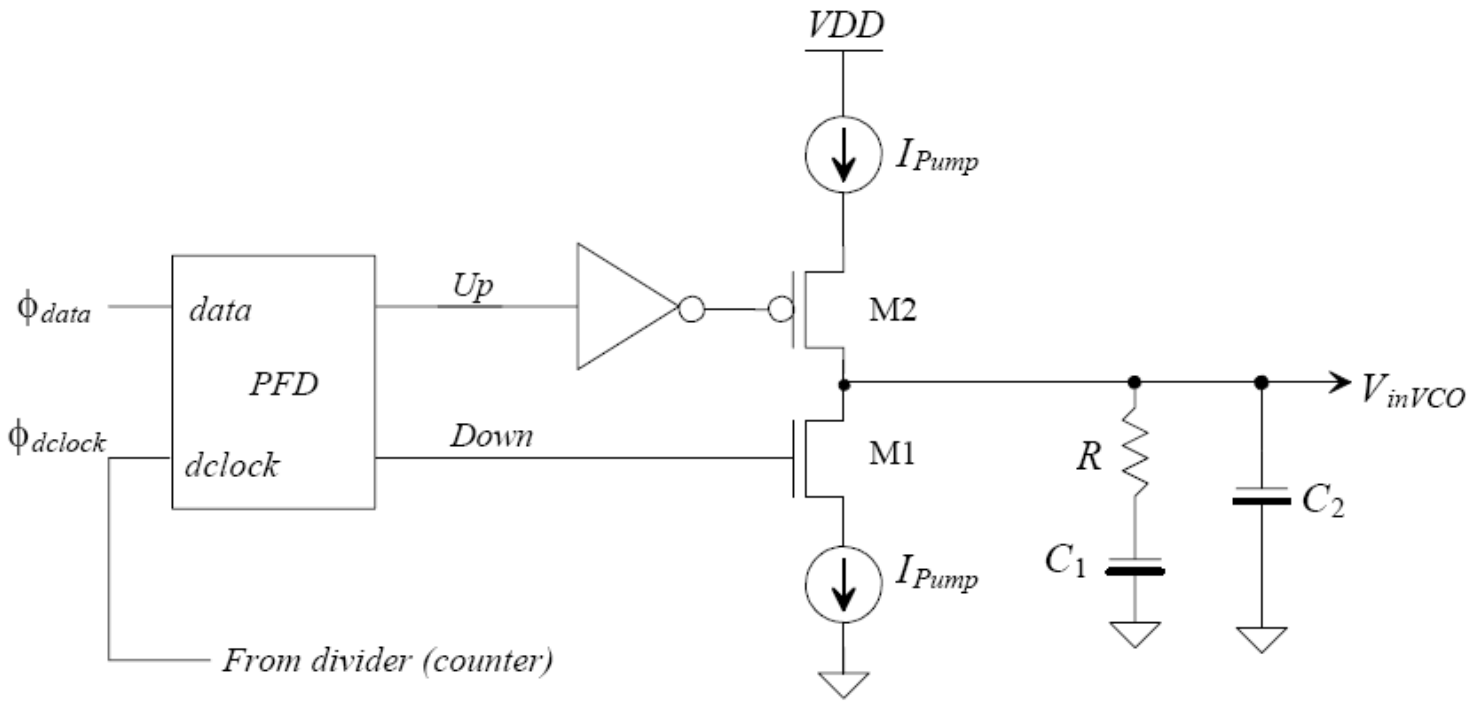



**Figure 19.36** PFD using the charge pump. [1]

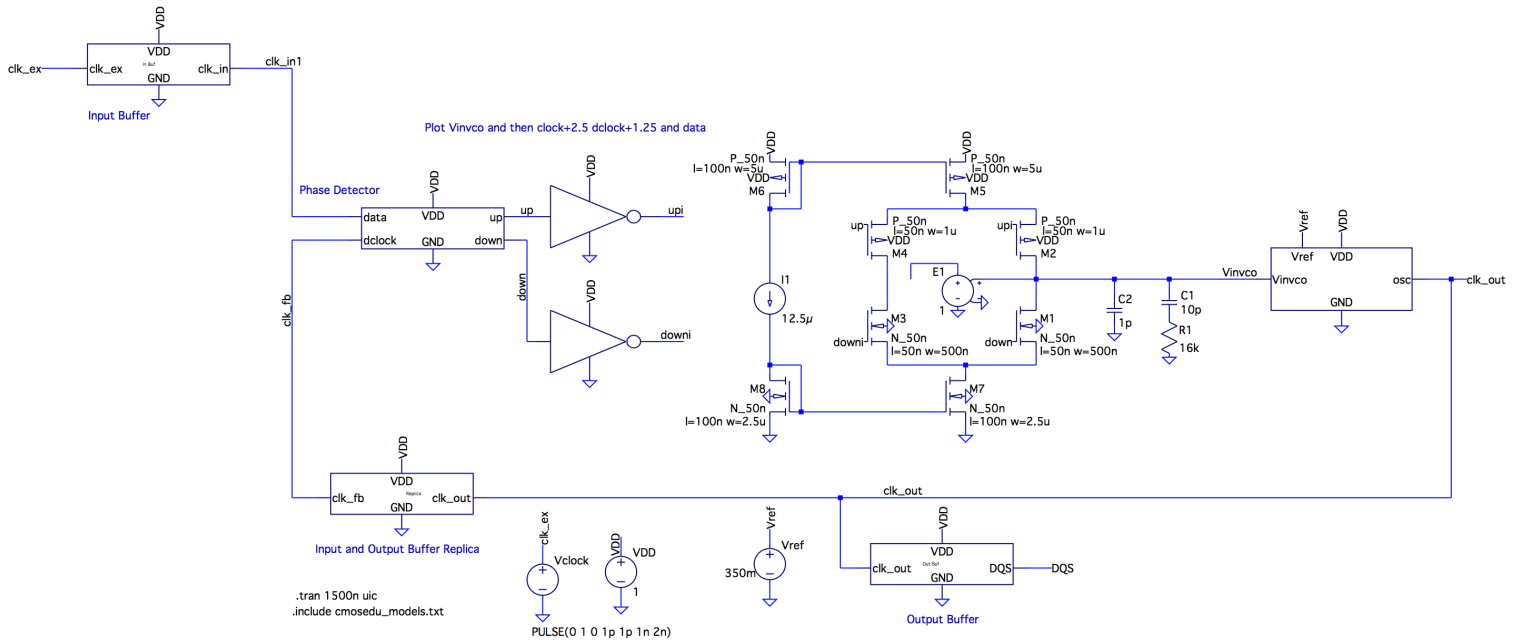
# PLL using PFD with CP output

- Select $I_{pump}$ for Lock range $\Delta\omega_L = 2\pi \cdot 100\text{MHz}$
- $\Delta\omega_L = 4\pi\zeta\omega_n$ (Eq. 19.53) [1]
- $\omega_n = \sqrt{\frac{K_{PDI}K_{VCO}}{NC_1}}$ (Eq. 19.57) [1]
- $K_{PDI} = \frac{I_{pump}}{2\pi}$ [1]
- $\zeta = \frac{\omega_n}{2} \cdot RC_1$ (Eq. 19.58) [1]

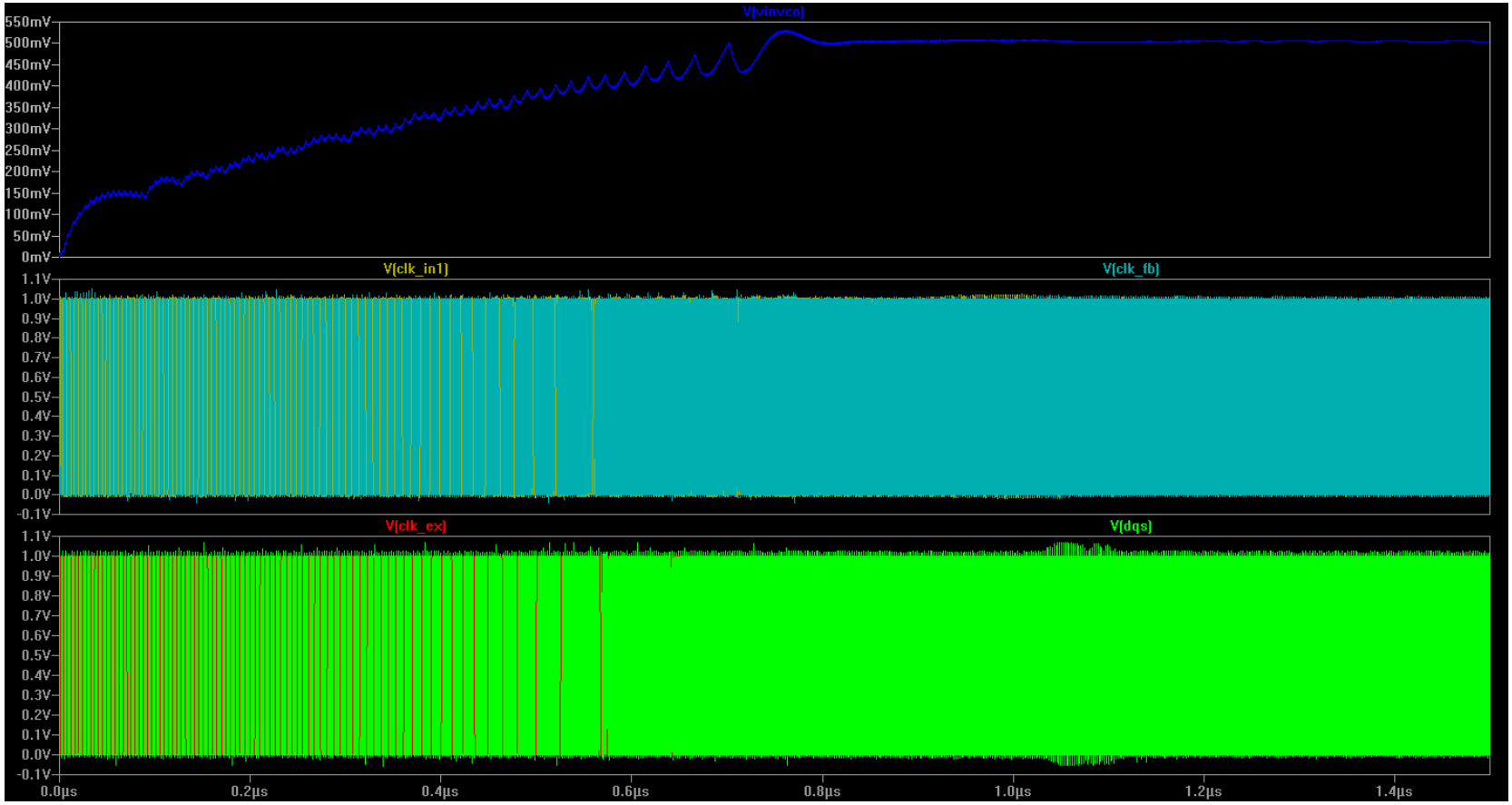
# PLL using PFD with CP output

- $\omega_n = \frac{\Delta\omega_L}{4\pi\zeta} = \frac{2\pi \cdot 100\times10^6}{4\pi \cdot 2} = 25\times10^6\ radians/V \cdot s$
- $25\times10^6 = \sqrt{\frac{I_{pump} \cdot 3.129\times10^9}{2\pi \cdot 1 \cdot 10\times10^{-12}}} \rightarrow I_{pump} \approx 12.5\ \mu A$
- Select $C_1 = 10$ pF and $C_2 = \frac{C_1}{10} = 1$ pF
- $R = \frac{2\zeta}{\omega_n C_1} = \frac{2 \cdot 2}{25\times10^6 \cdot 10\times10^{-12}} = 16\ k\Omega$

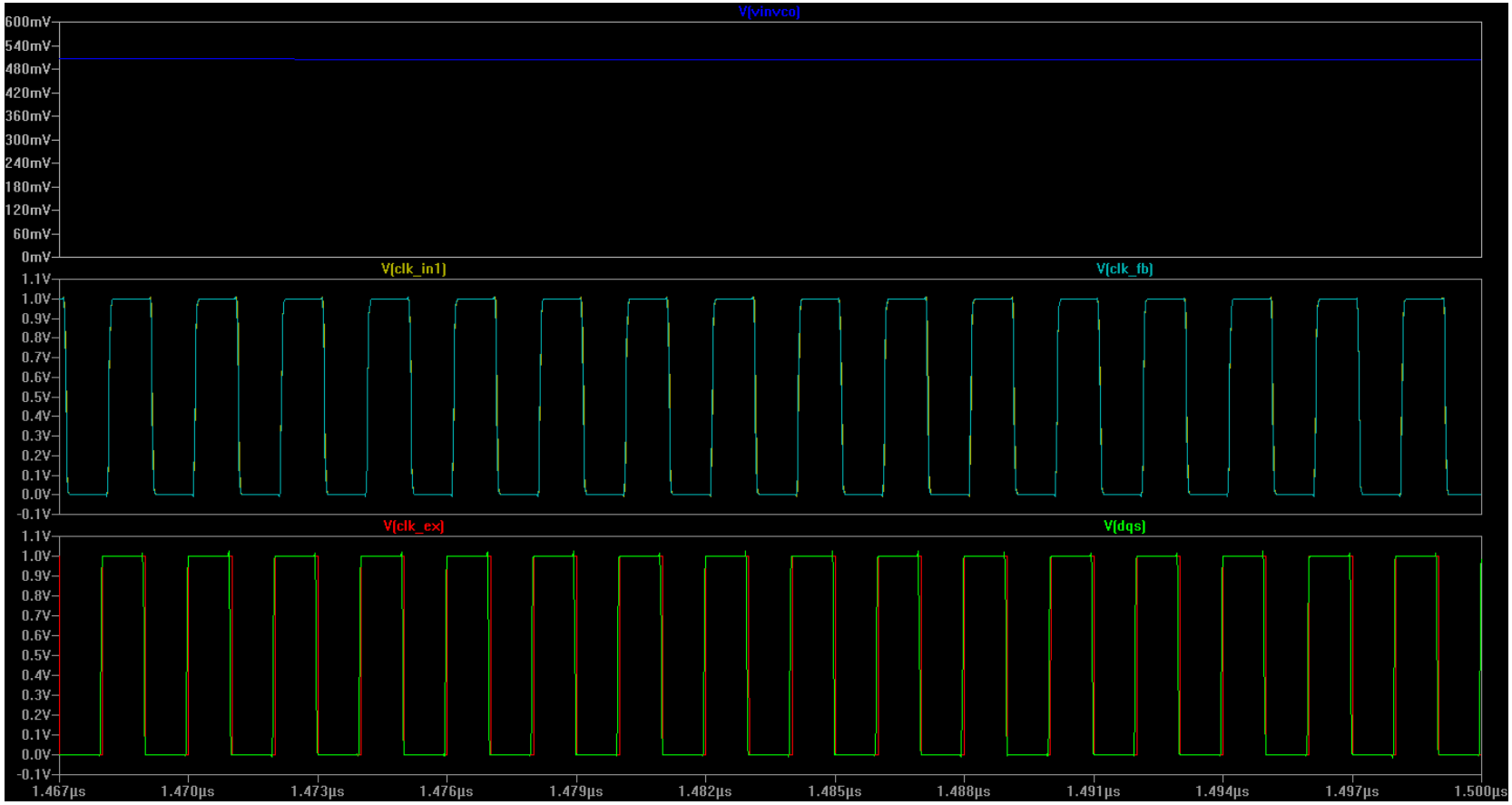
# Simulation


VDD
clk_ex
clk_in
clk_in1
GND
Input Buffer
Plot Vinvco and then clock+2.5 dclock+1.25 and data
Phase Detector
data
dclock
up
down
upi
downi
clk_fb
P_50n
l=100n w=5u
M6
M5
M4
M2
M3
M1
M7
M8
N_50n
l=50n w=500n
l=100n w=2.5u
I1
12.5µ
E1
C2
1p
C1
10p
R1
16k
Vinvco
Vref
osc
clk_out
Input and Output Buffer Replica
Vclock
PULSE(0 1 0 1p 1p 1n 2n)
350m
DQS
Output Buffer
.tran 1500n uic
.include cmosedu_models.txt

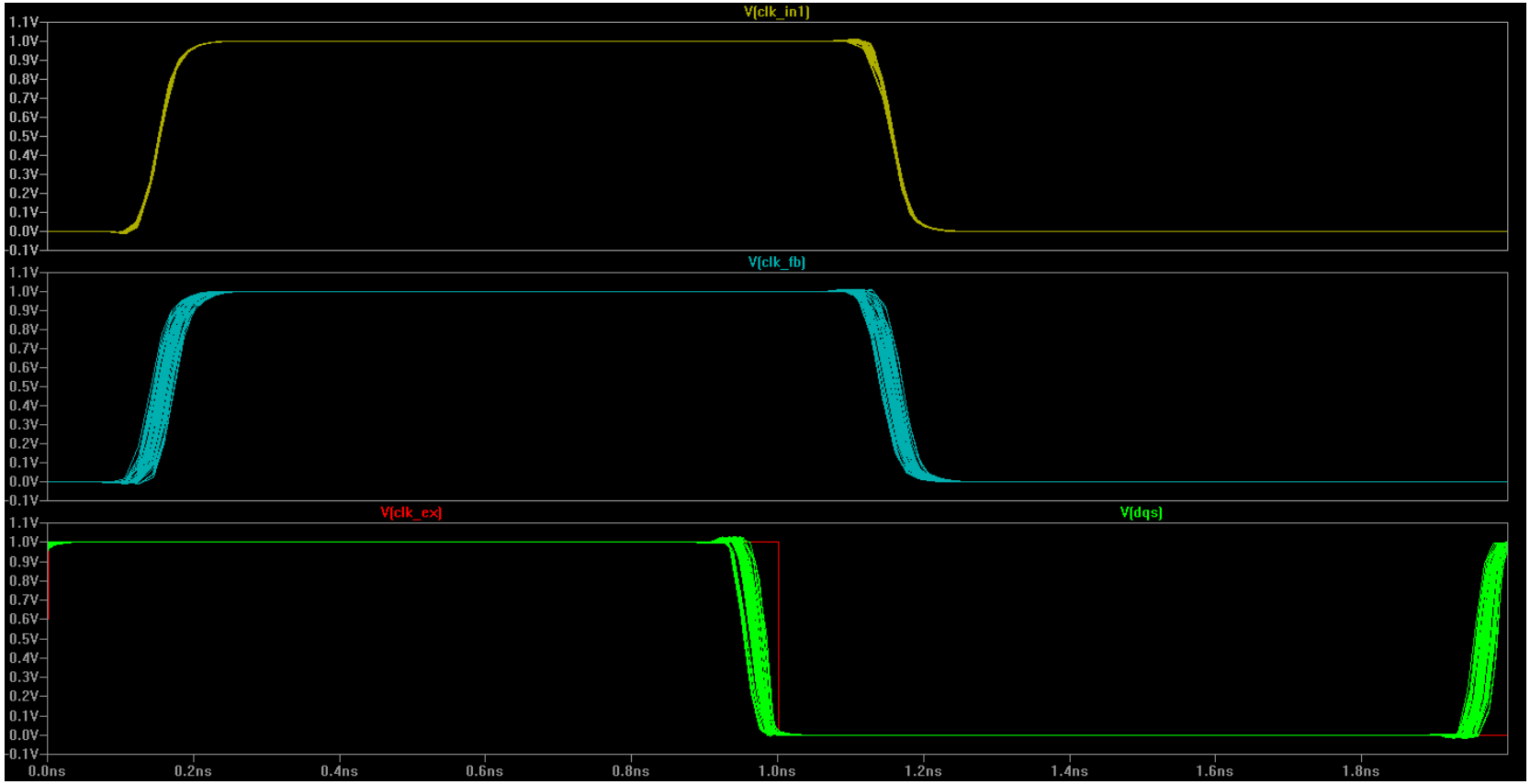
# Simulation


V(vinvco)
550mV
500mV
450mV
400mV
350mV
300mV
250mV
200mV
150mV
100mV
50mV
0mV
V(clk_in1)
V(clk_fb)
1.1V
1.0V
0.9V
0.8V
0.7V
0.6V
0.5V
0.4V
0.3V
0.2V
0.1V
0.0V
-0.1V
V(clk_ex)
V(dqs)
0.0µs
0.2µs
0.4µs
0.6µs
0.8µs
1.0µs
1.2µs
1.4µs

# Simulation


V(vinvco)
600mV
540mV
480mV
420mV
360mV
300mV
240mV
180mV
120mV
60mV
0mV
V(clk_in1)
V(clk_fb)
1.1V
1.0V
0.9V
0.8V
0.7V
0.6V
0.5V
0.4V
0.3V
0.2V
0.1V
0.0V
-0.1V
V(clk_ex)
V(dqs)
1.467µs
1.470µs
1.473µs
1.476µs
1.479µs
1.482µs
1.485µs
1.488µs
1.491µs
1.494µs
1.497µs
1.500µs

# Jitter


V(clk_in1)
V(clk_fb)
V(clk_ex)
V(dqs)
1.1V
1.0V
0.9V
0.8V
0.7V
0.6V
0.5V
0.4V
0.3V
0.2V
0.1V
0.0V
-0.1V
0.0ns
0.2ns
0.4ns
0.6ns
0.8ns
1.0ns
1.2ns
1.4ns
1.6ns
1.8ns

# Ways to Improve

- Use coarse and fine loops to help shorten pull-in time and reduce jitter.
- Use differential delay element to reject power supply and ground noises.

# Conclusion

- Using PLL in place of DLL in SDRAM is possible
  - But, impractical
- Benefits
  - Filters input noise
- Drawbacks
  - Design complexity with 2nd order system
  - Need to detect and generate frequency
    - Take very long time to pull in the frequency (deal breaker)
  - Feedback phase noise

# References


- [1] Baker, R.J., "CMOS: Circuit Design, Layout, and Simulation," 3rd Ed. , Wiley-IEEE, 2010.
- [2] Becker, E.A., "*Design of an Integrated Half-Cycle Delay Line Duty Cycle Corrector Delay-Locked Loop*", 2008
- [3] Lin, F., "Research and Design of Low Jitter, Wide Locking-Range All-Digital Phase-Locked and Delay-Locked Loops", 2000.
- [4] General DDR SDRAM Functionality, TN-46-05, Micron Technology, 2001

# Questions