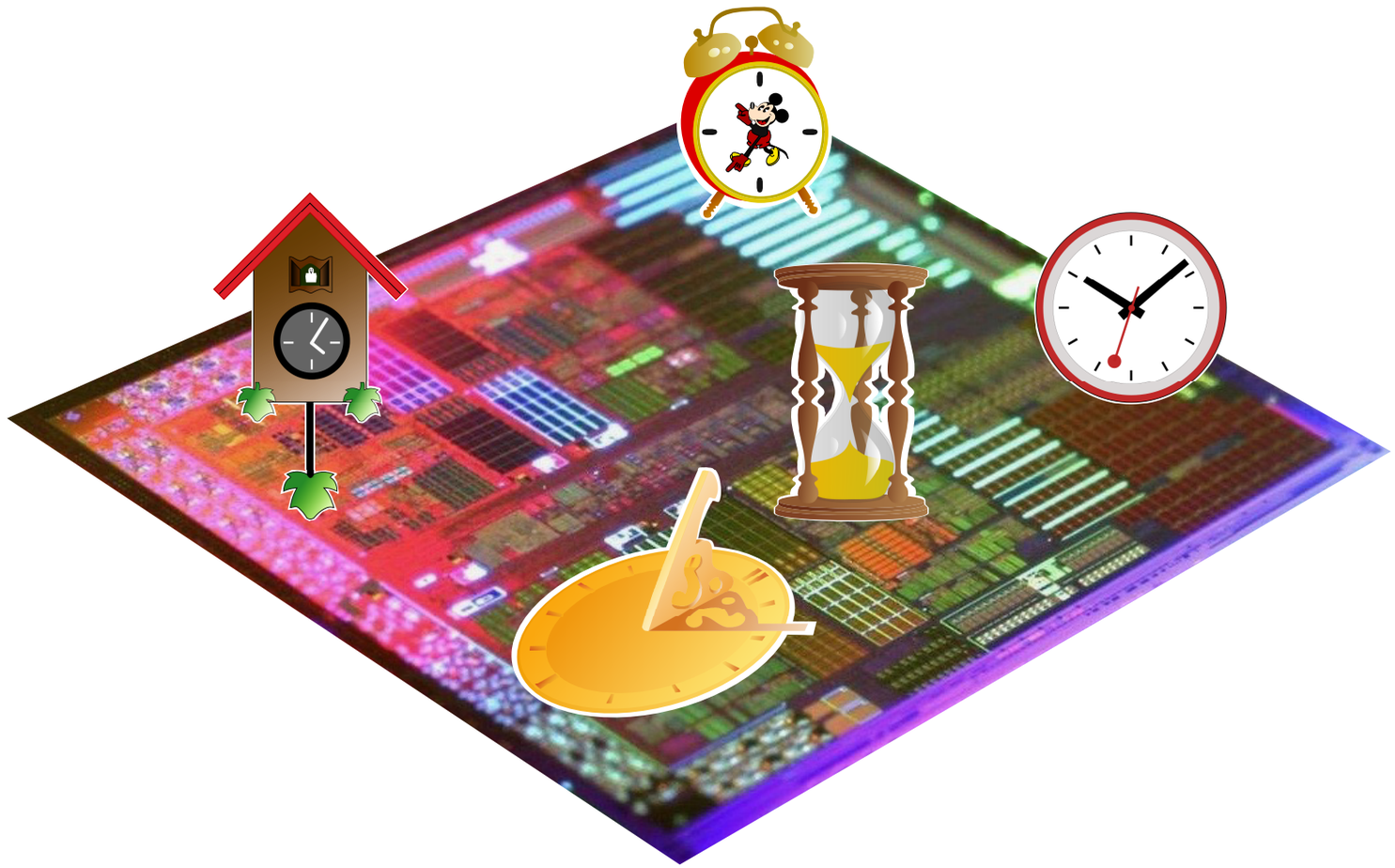
Phase Locked Loops, PLLs, for Clocking Chips



Ian W. Jones
and
Felipe A. Kuentzer

ianjones@mpi-inf.mpg.de
kuentzer@ihp-microelectronics.com
December 2020

Outline

Part 1:

- Overview
- Intro to PLLs
- Clocking with PLLs
- Analog PLLs

Part 2:

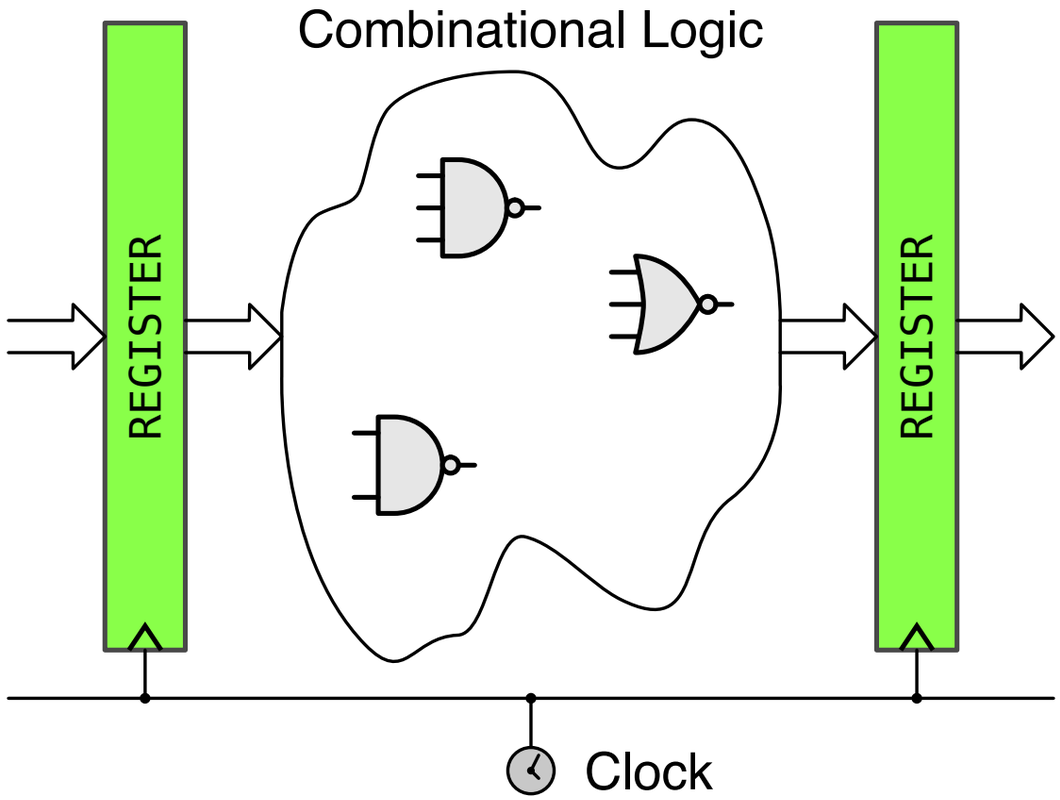
- Recap
- Digital PLLs
- Future challenges
- Summary

Overview

- Clocked chips => easier design.
- What is a PLL?
- PLLs: industry standard for clocking chips.
- Benefits of PLLs:
  - Accurate
  - Stable
  - Flexible

Overview

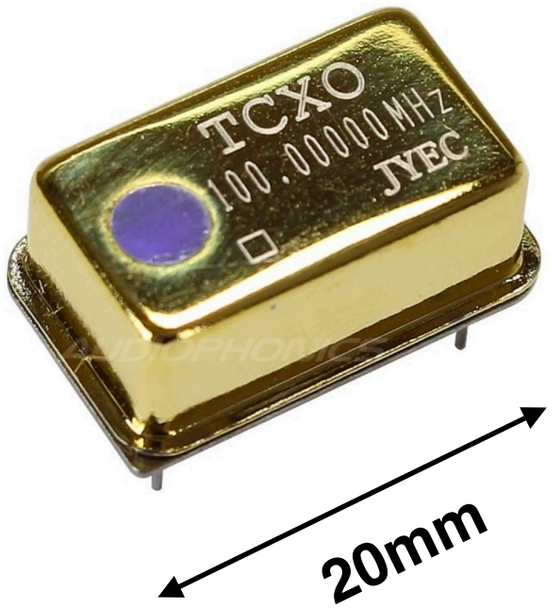
Clocking chips makes their design easier.



Clocked registers separate the data of successive computations.

Clocking a Chip

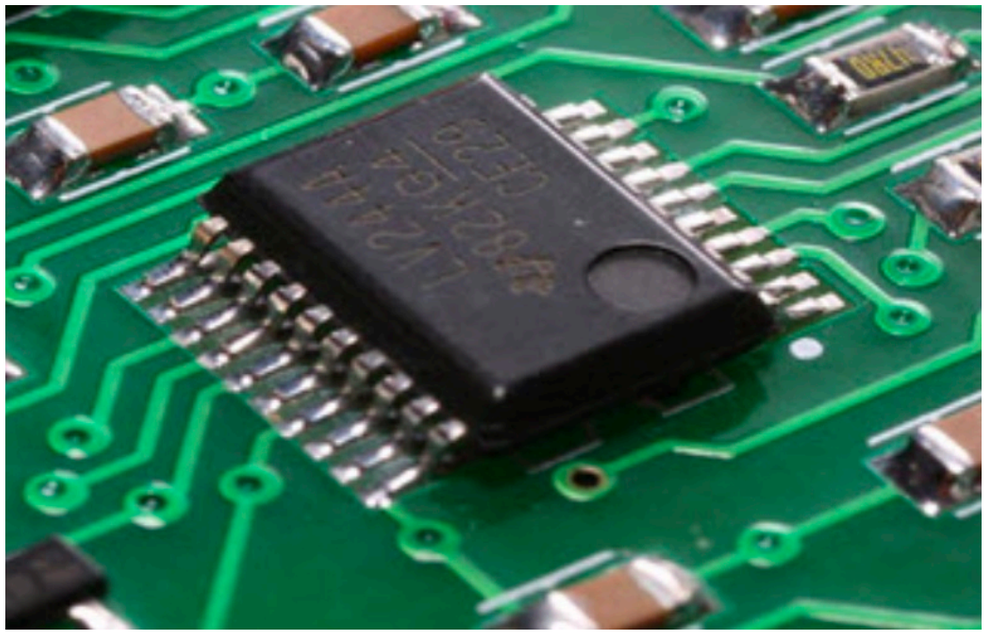
Quartz crystal oscillator:



– Accurate
– Stable
– Cheap

Clocking a Chip

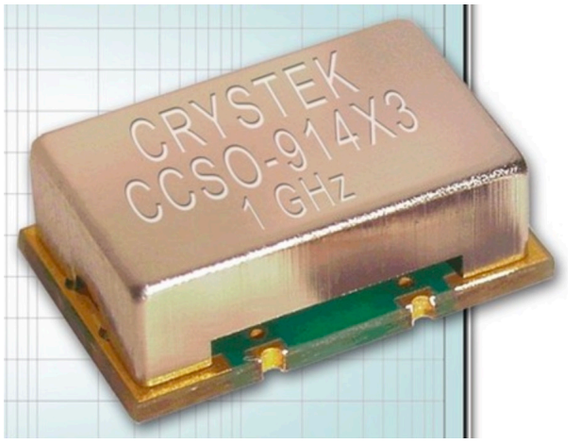
Frequency below a few 100 MHz:

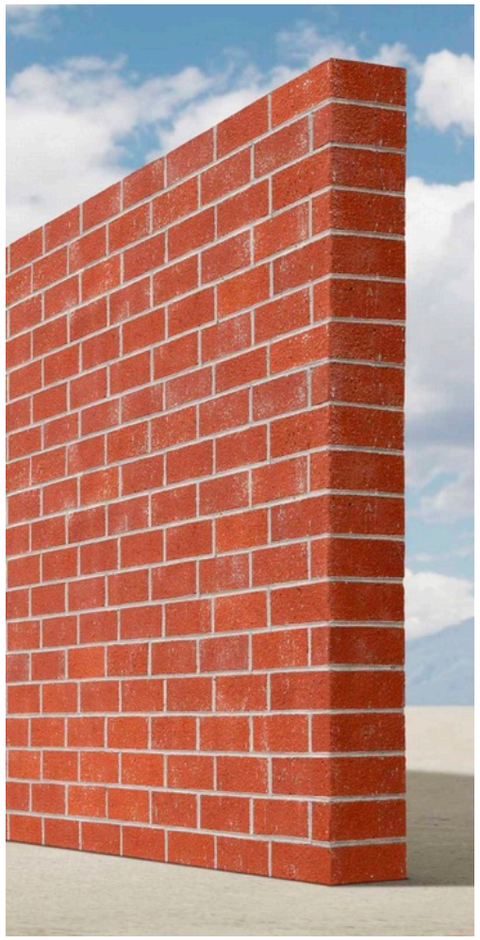


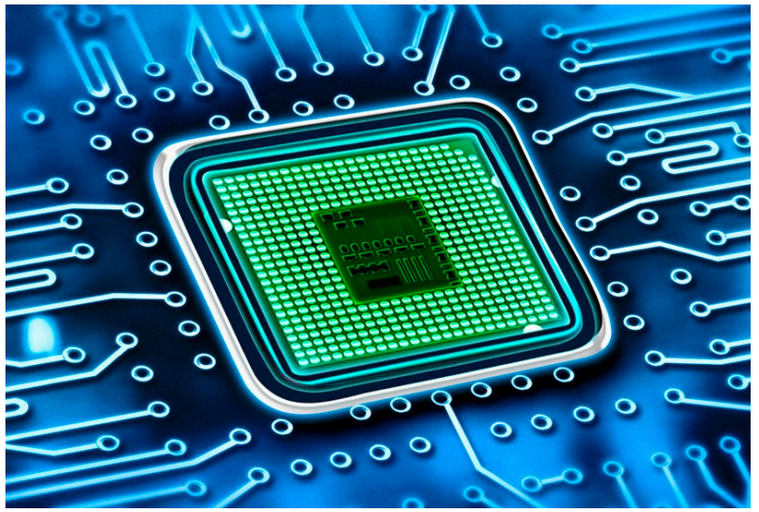
chip is clocked directly by off-chip oscillator

Clocking a Chip

Above a few 100 MHz:



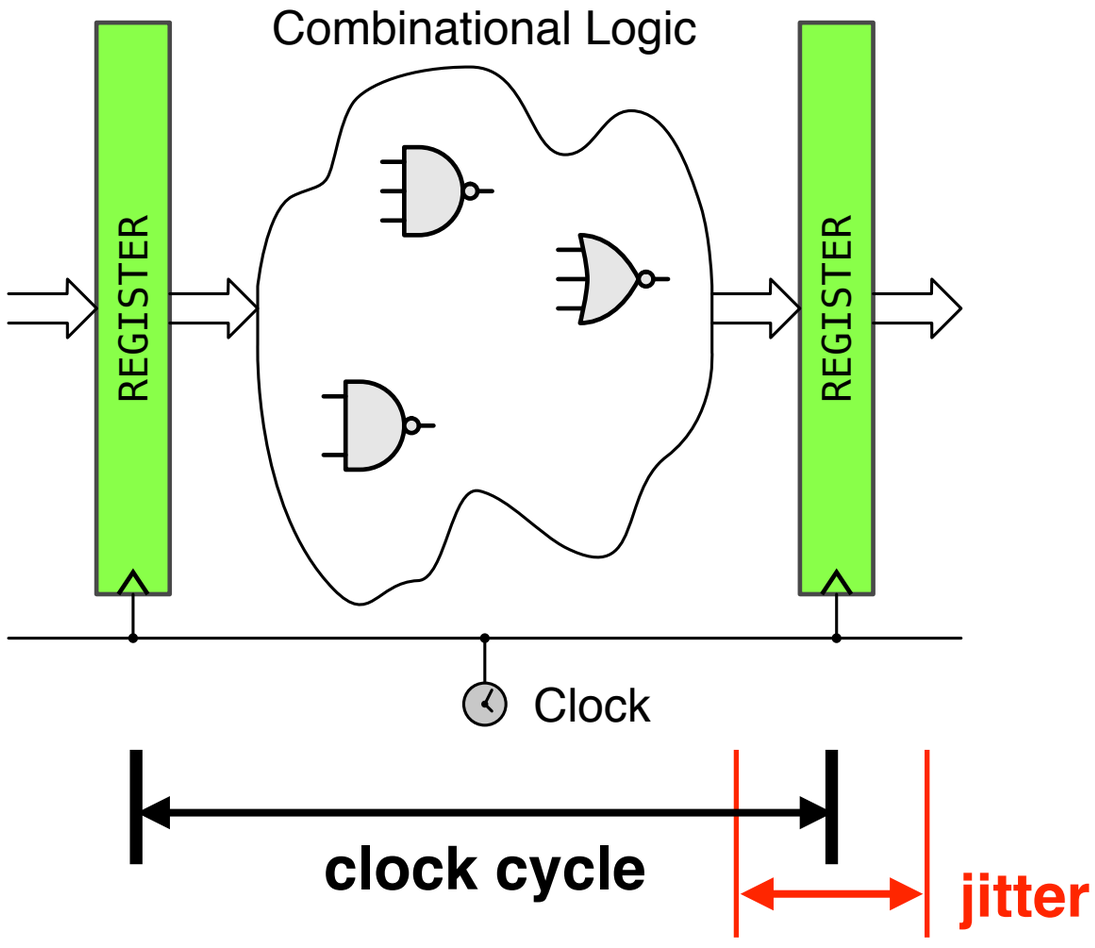




hard to deliver off-chip clock signal

Clocking a Chip

Clock Jitter:

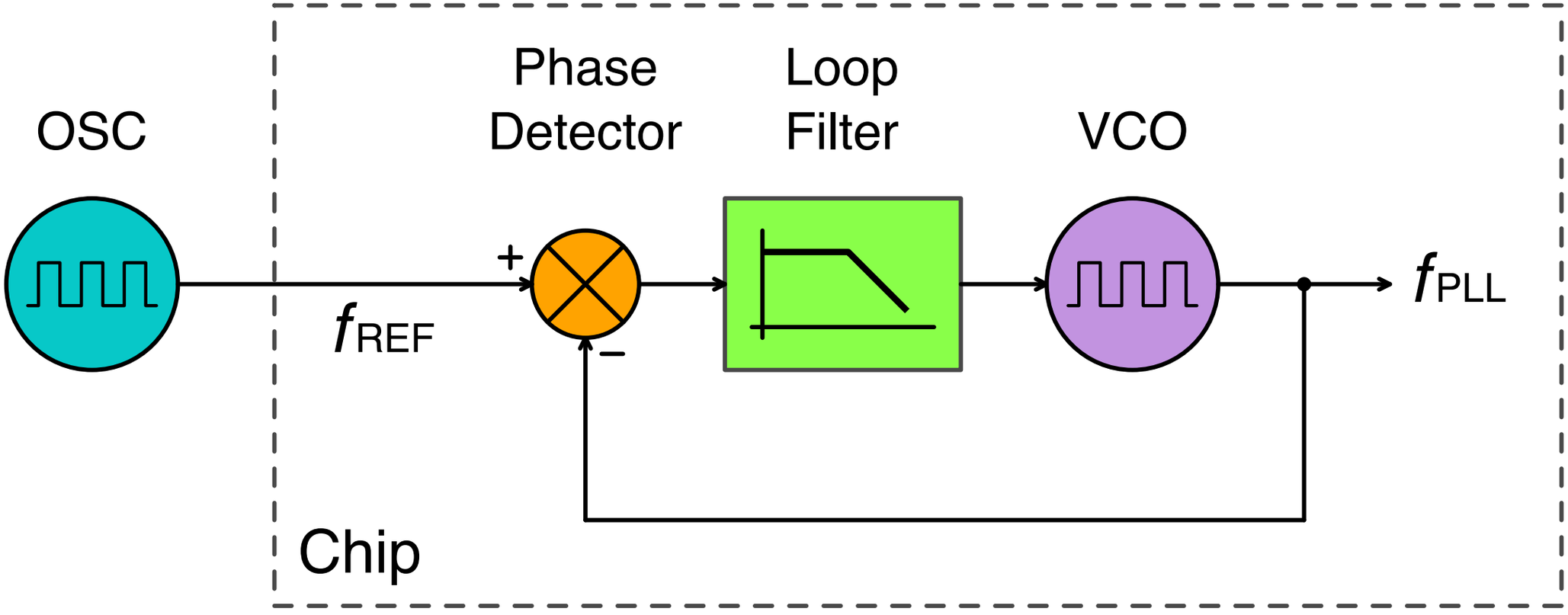


Frequency: **100MHz**
Clock cycle: **10,000 ps**
Jitter: **0.1ps**

Jitter subtracts from useful compute time.

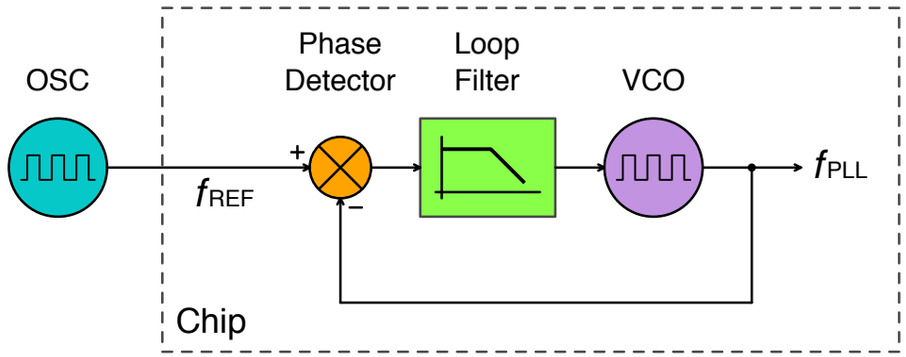
Basic Phase-Locked Loop, PLL

Make an on-chip copy of the external clock:



Control loop locks **phase** of f_{PLL} to f_{REF}

Basic PLL



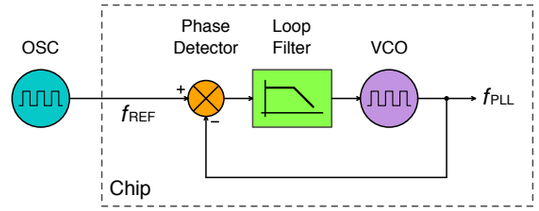
OSC – External clock reference, e.g., a quartz crystal oscillator

VCO – On-chip Voltage Controlled Oscillator, generates f_{PLL}

Phase Detector – Generates output signal that is proportional to the phase difference between f_{PLL} and f_{REF}

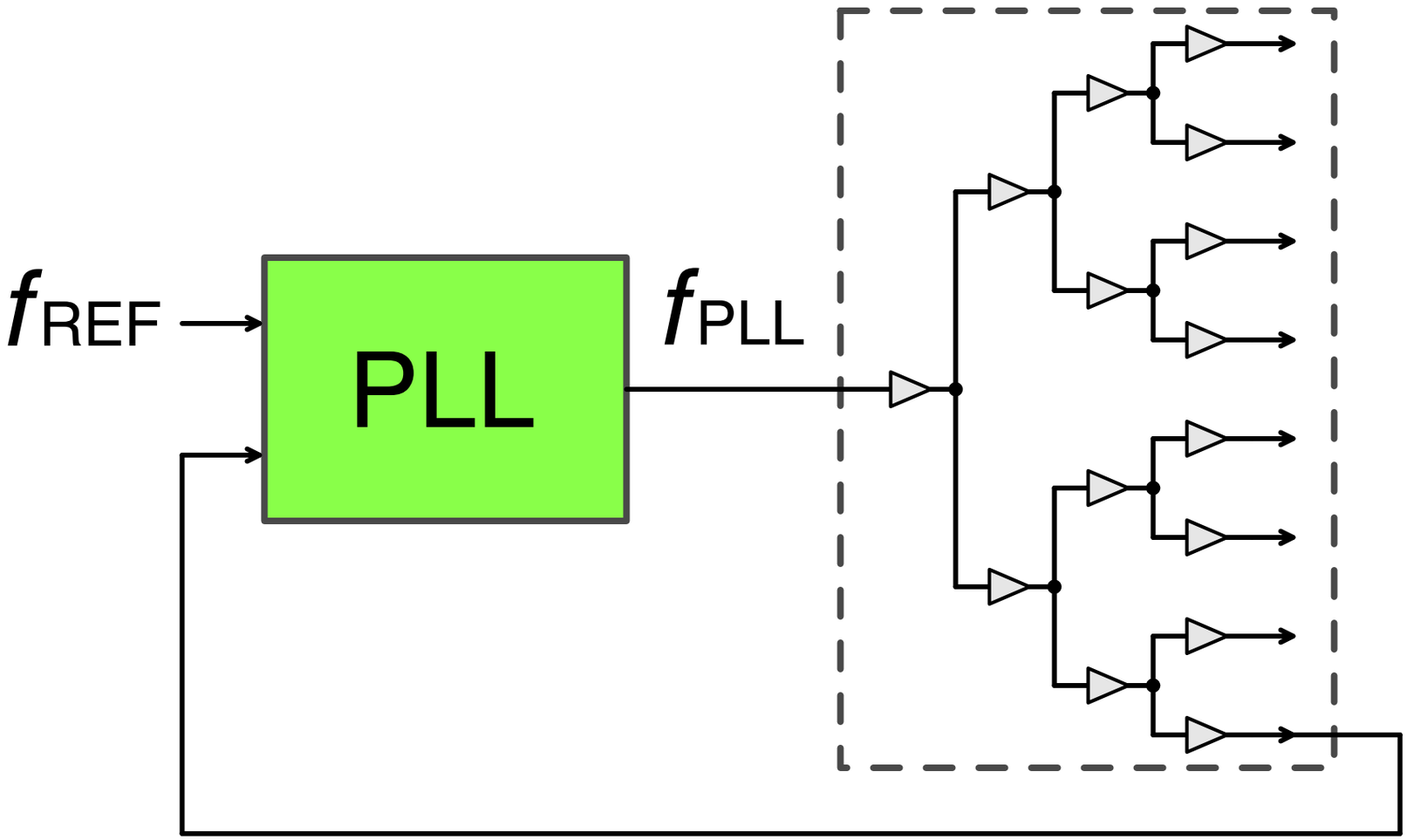
Loop Filter – Makes the control loop stable

Basic PLL

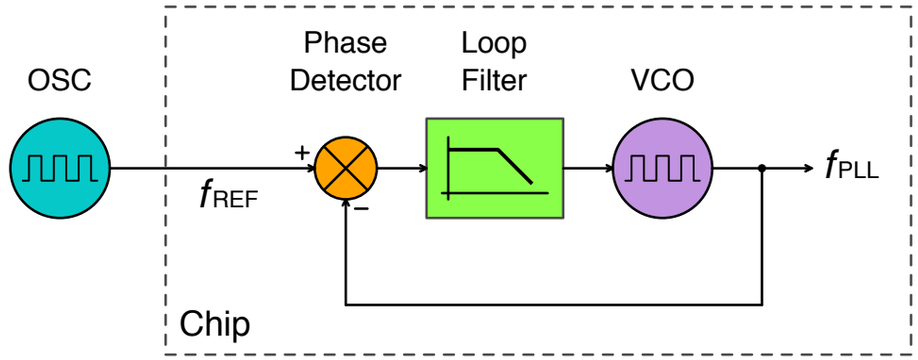


Enables:

Clock Distribution Tree



Discussion Session #1



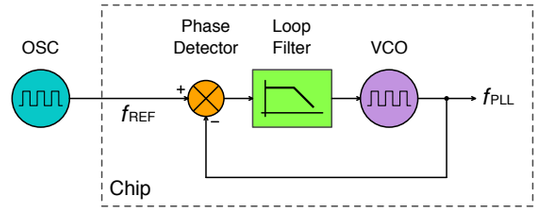
- What have we learned so far?

- What is the advantage of routing the feedback loop via the clock tree?

- What else does such a simple PLL enable?

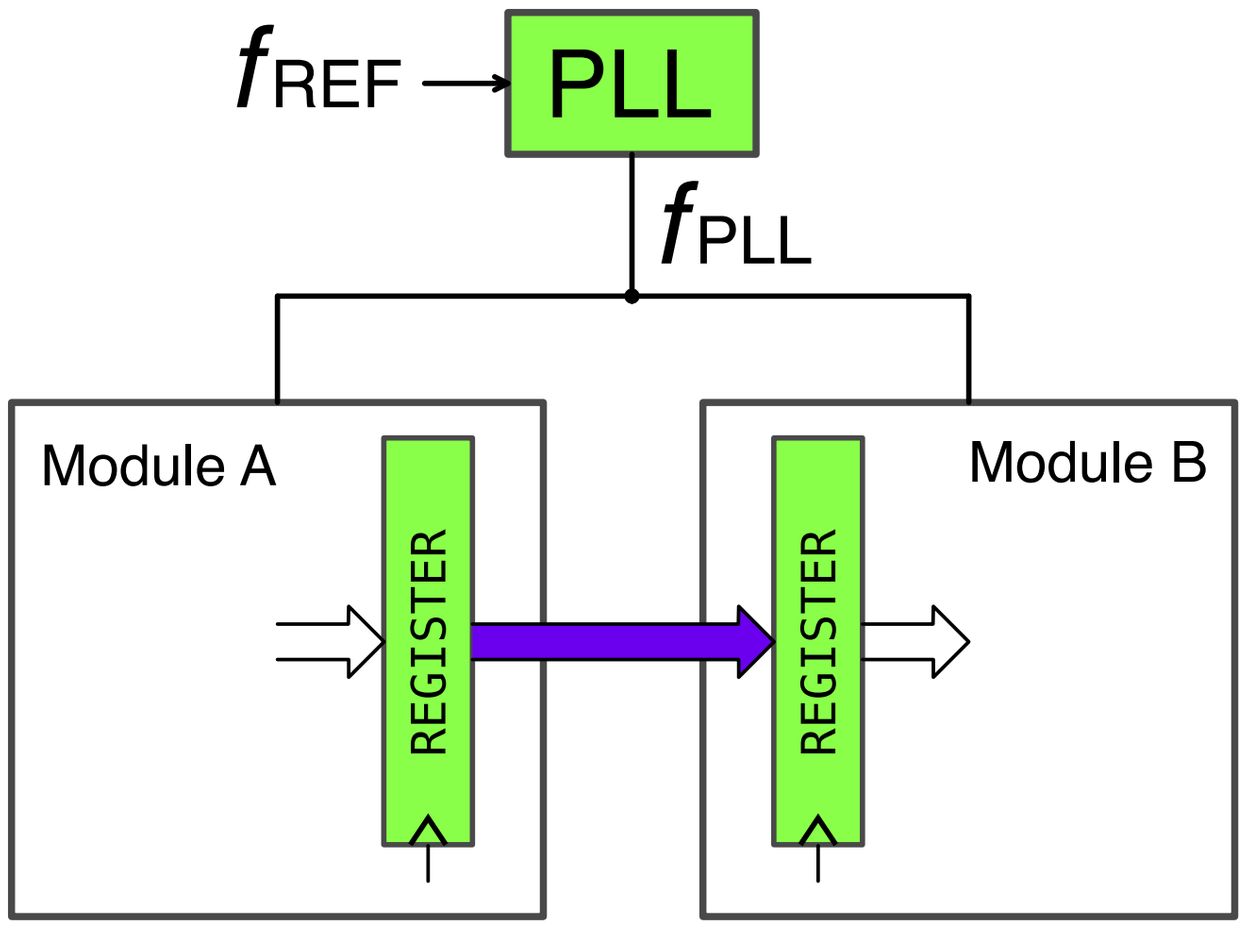
- How might we improve a PLL?

Basic PLL

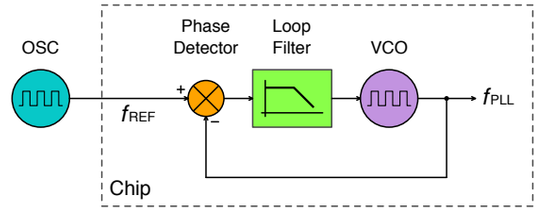


Enables:

data transfers between similar circuit modules

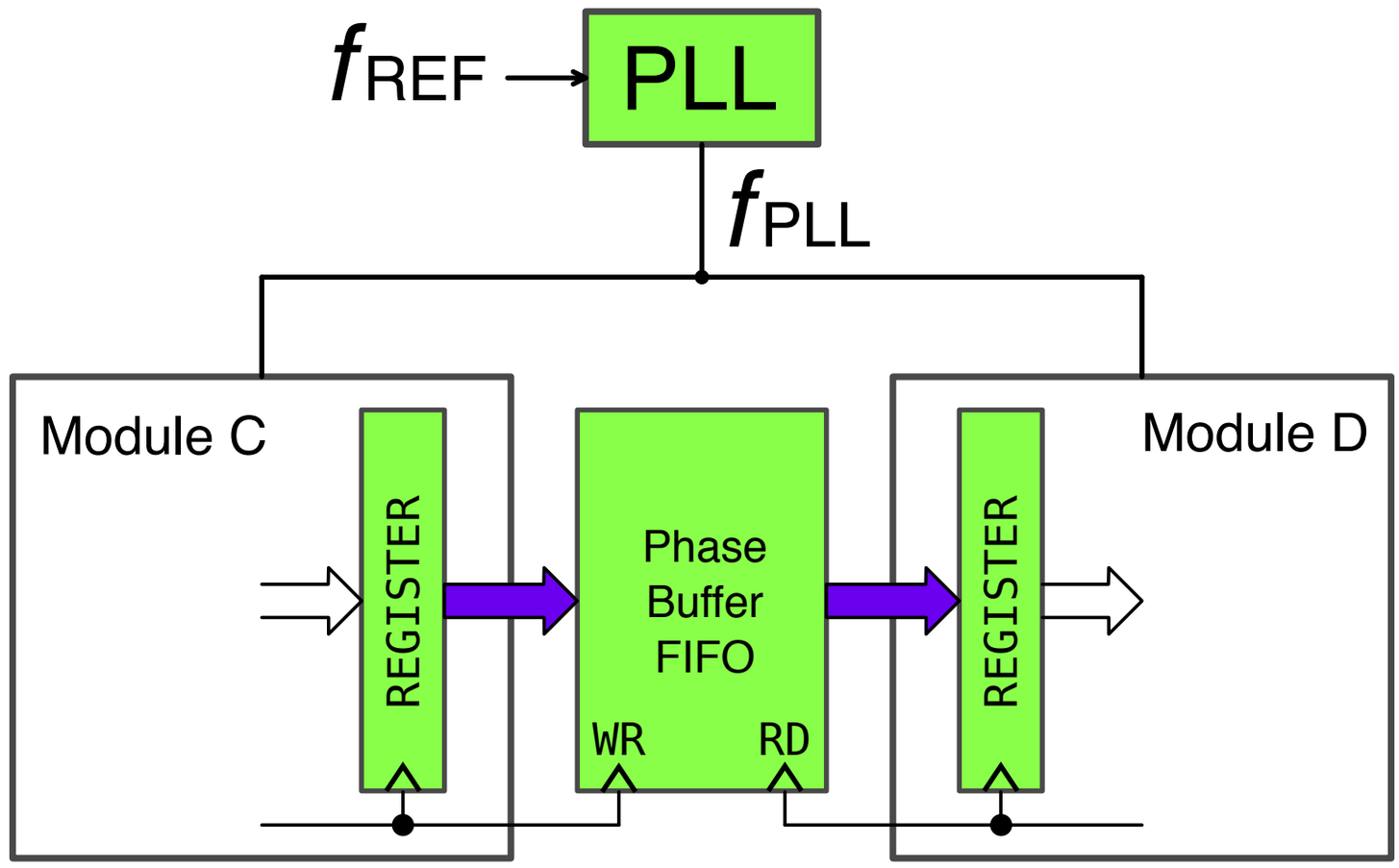


Basic PLL

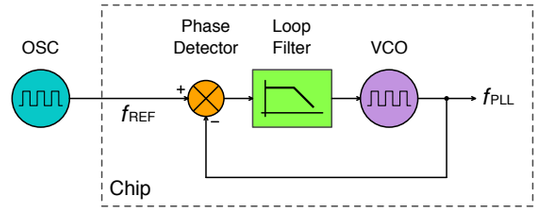


Enables:

using a Phase Buffer accommodate clock skew

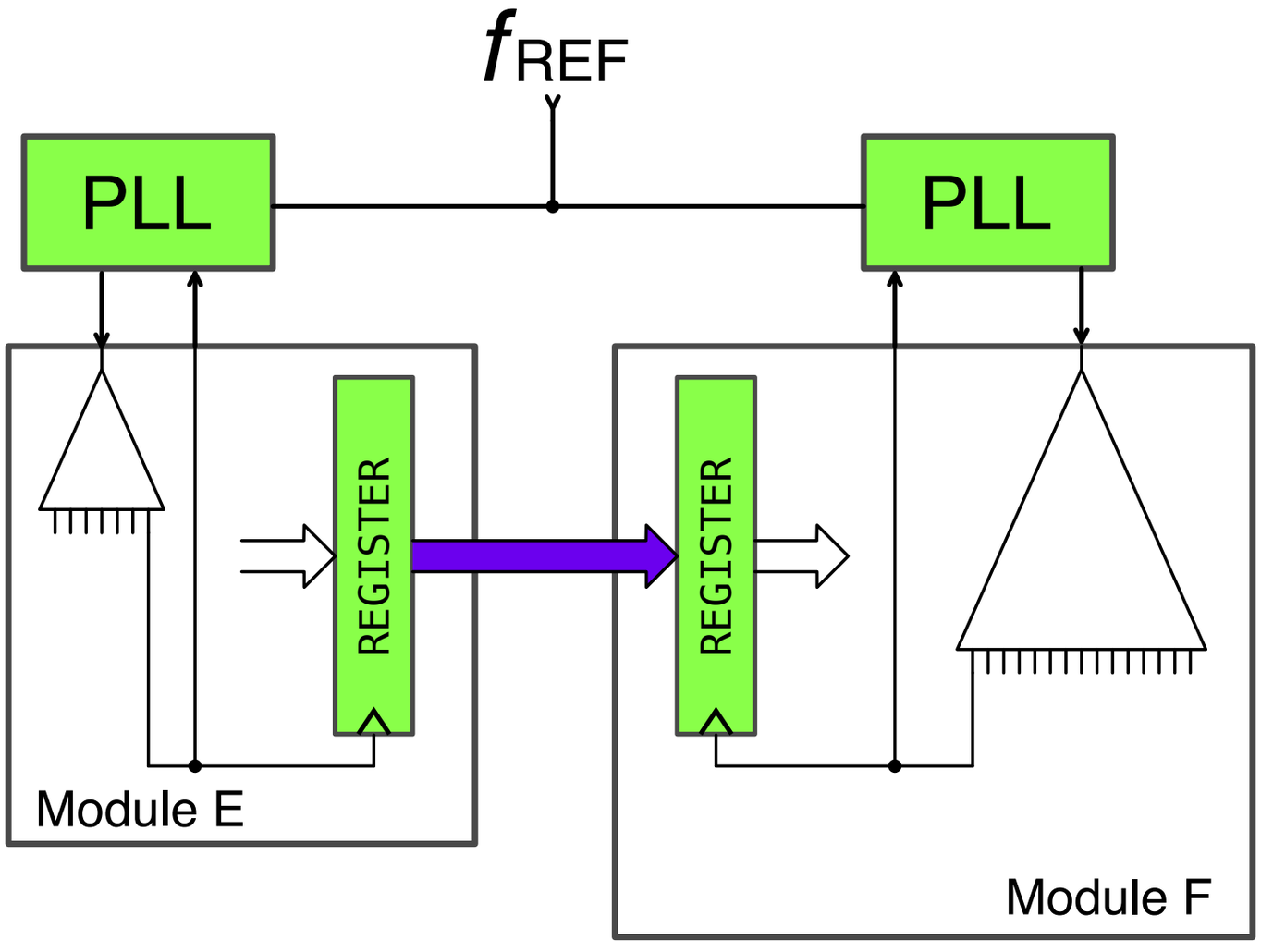


Basic PLL

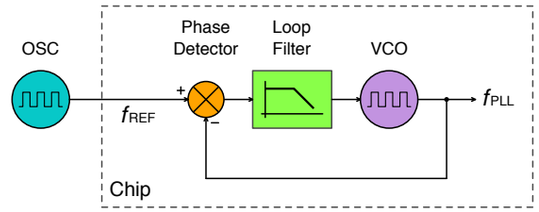


Enables:

reliable communication between very different modules

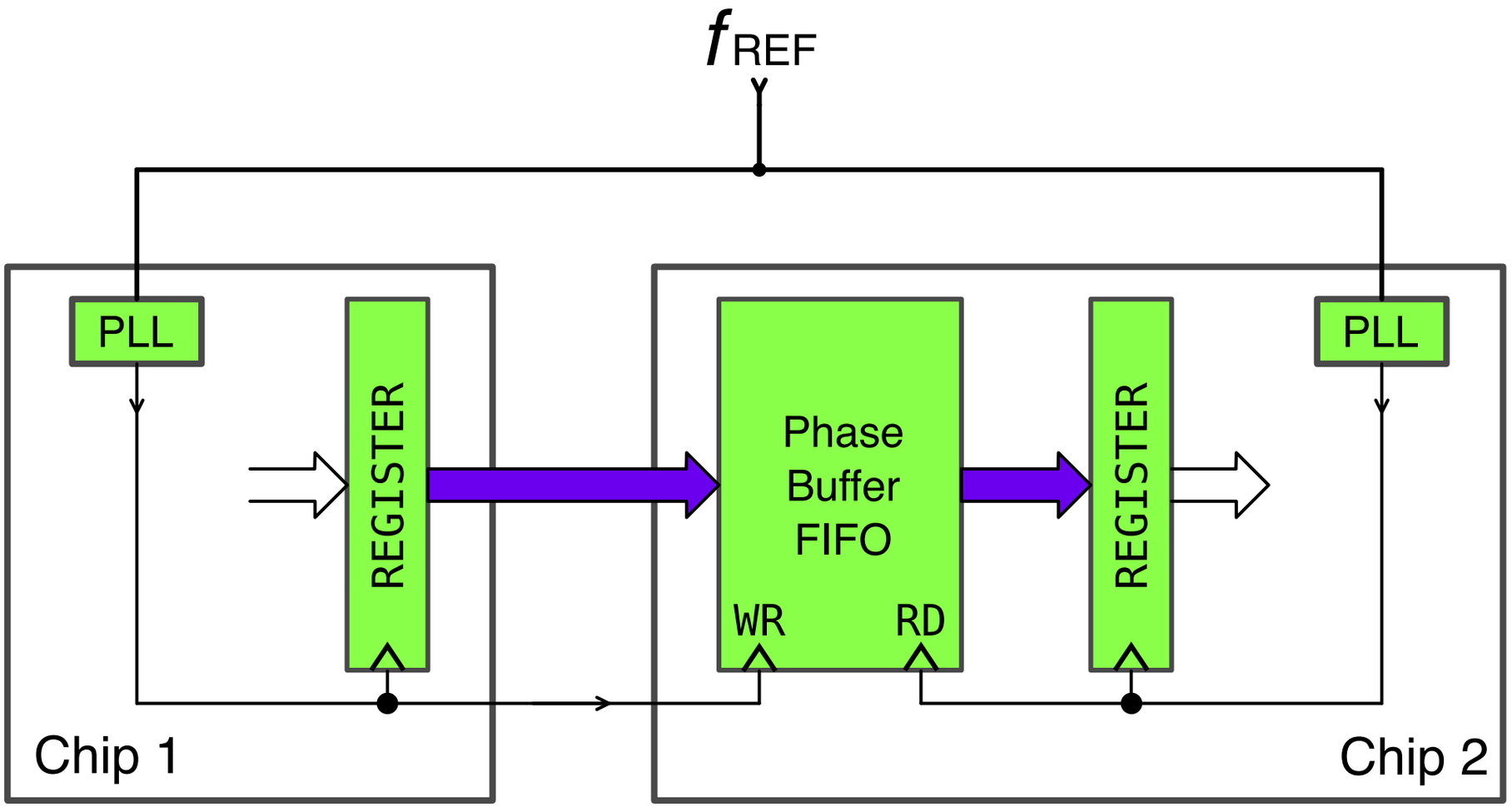


Basic PLL

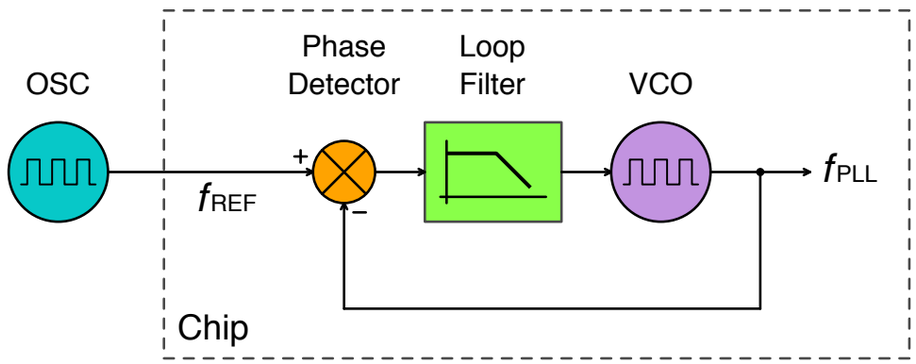


Enables:

communication between chips



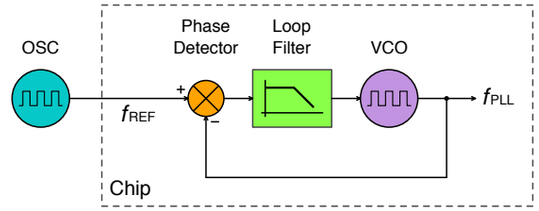
Discussion Session #1



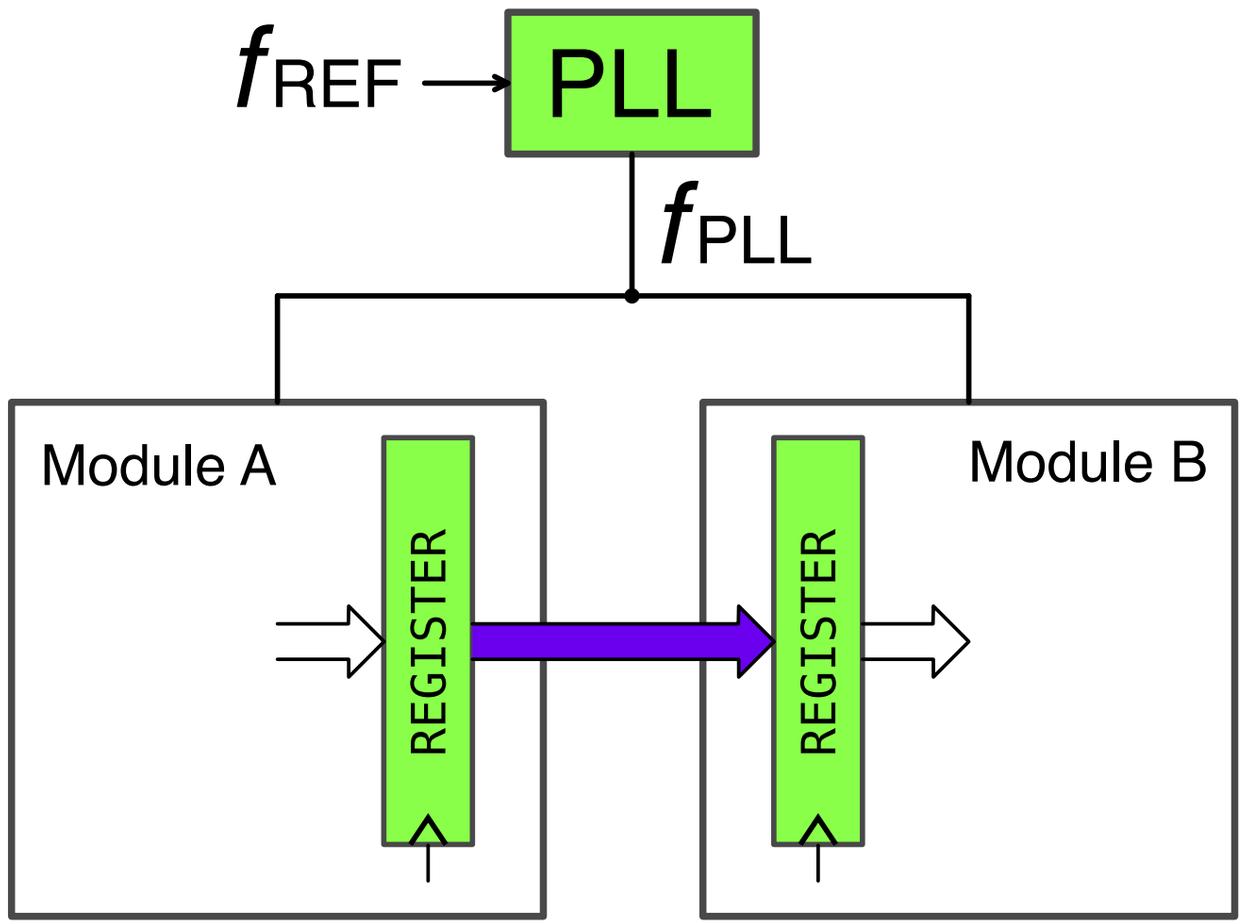
END

Return to lecture

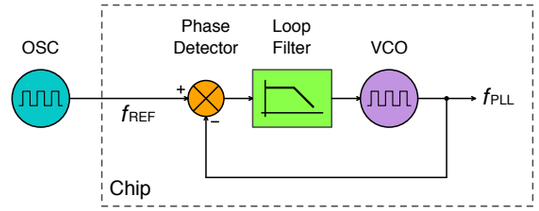
Basic PLL



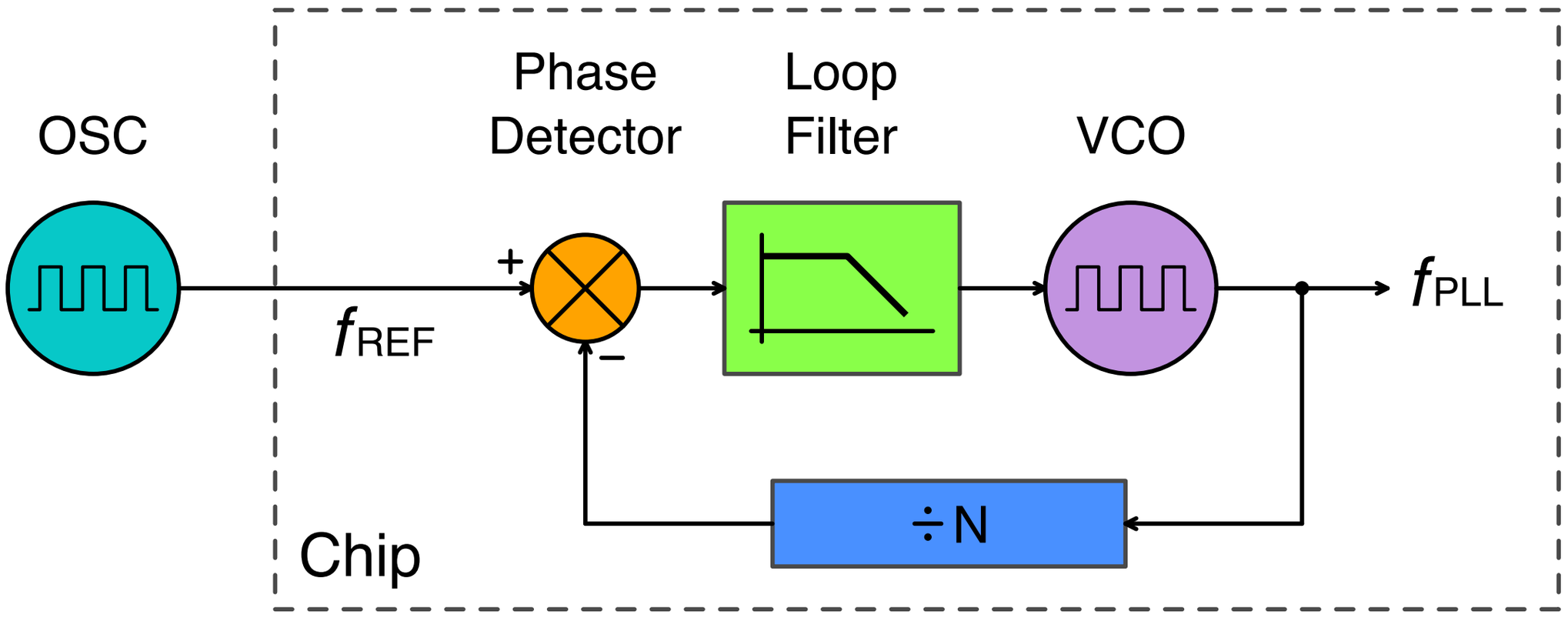
Control of clock phase enables data transfers between circuit modules, and between chips.



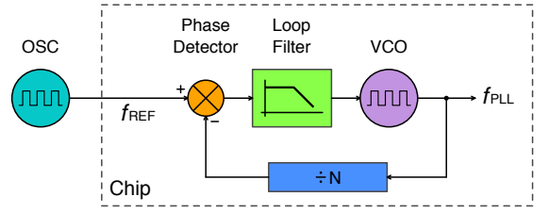
Improved PLL



$$f_{PLL} = N \times f_{REF}$$

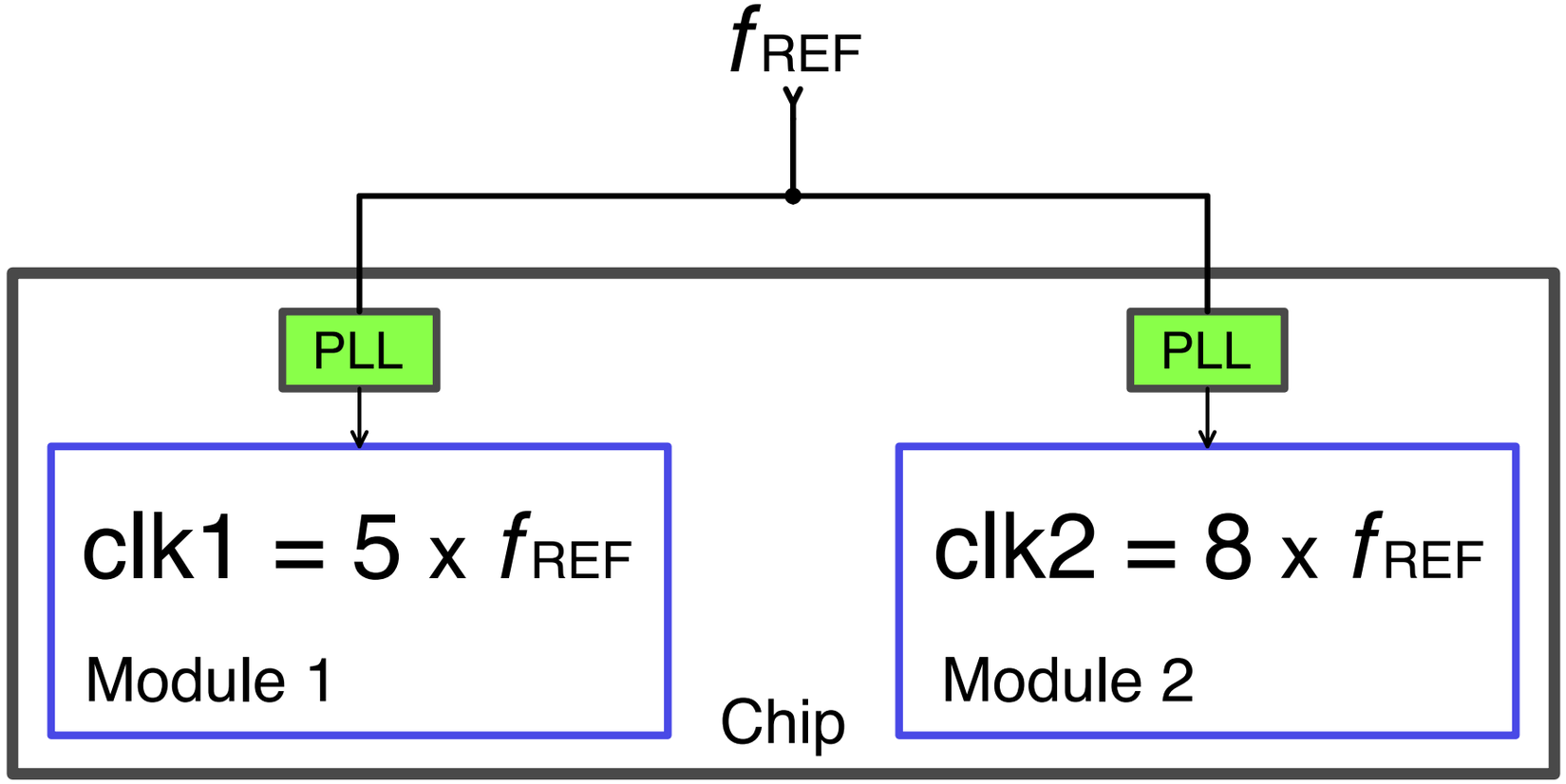


Improved PLL

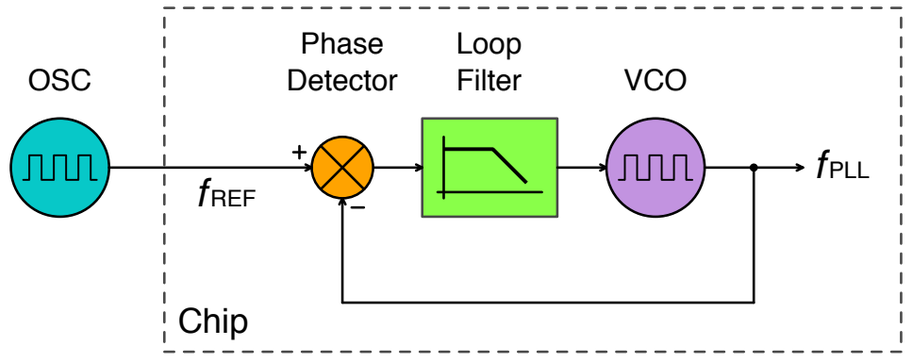


Enables:

multiple phase-locked clock frequencies



Analog PLL Components



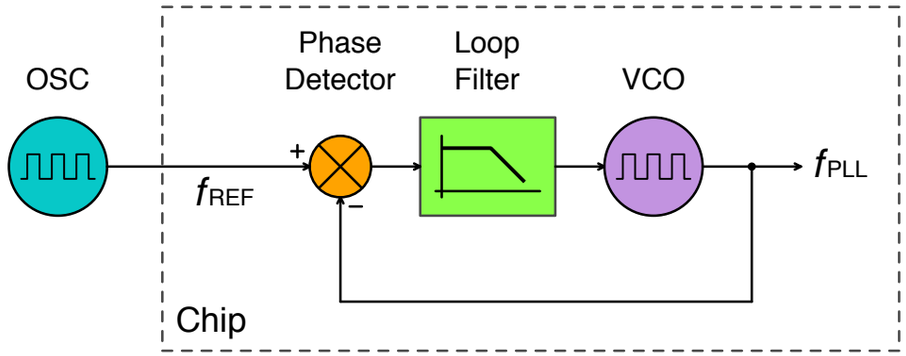
OSC – External clock reference, e.g., a quartz crystal oscillator

VCO – On-chip Voltage Controlled Oscillator, generates f_{PLL}

Phase Detector – Generates output signal that is proportional to the phase difference between f_{PLL} and f_{REF}

Loop Filter – Makes the control loop stable

Components of Basic Analog PLL



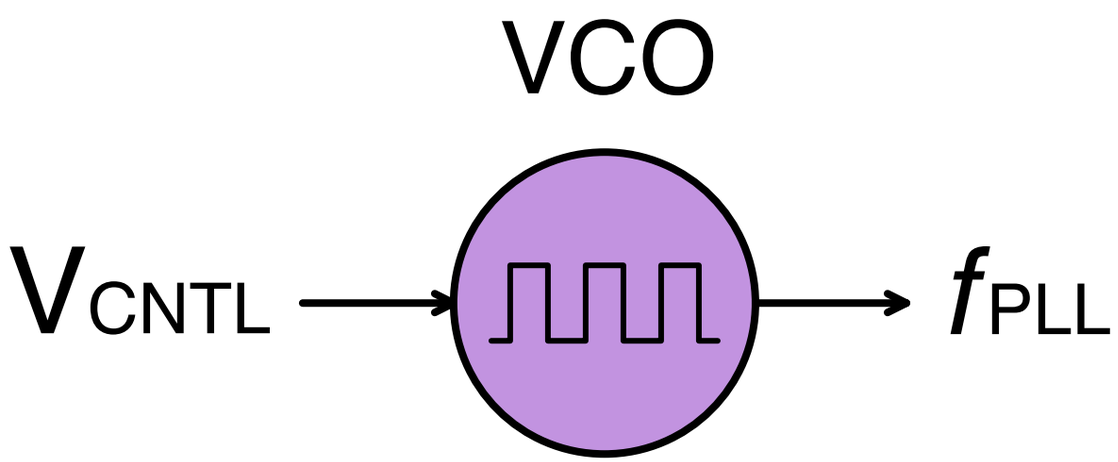
OSC – External clock reference, e.g., a quartz crustal oscillator

VCO – On-chip Voltage Controlled Oscillator, generates f_{PLL}

Phase Detector – Generates output signal that is proportional to the phase difference between f_{PLL} and f_{REF}

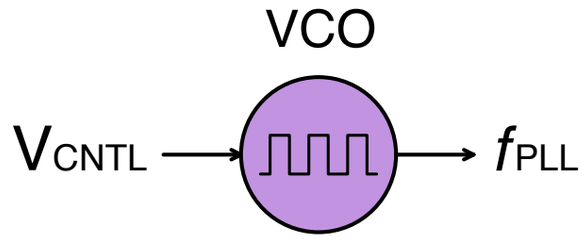
Loop Filter – Makes the control loop stable

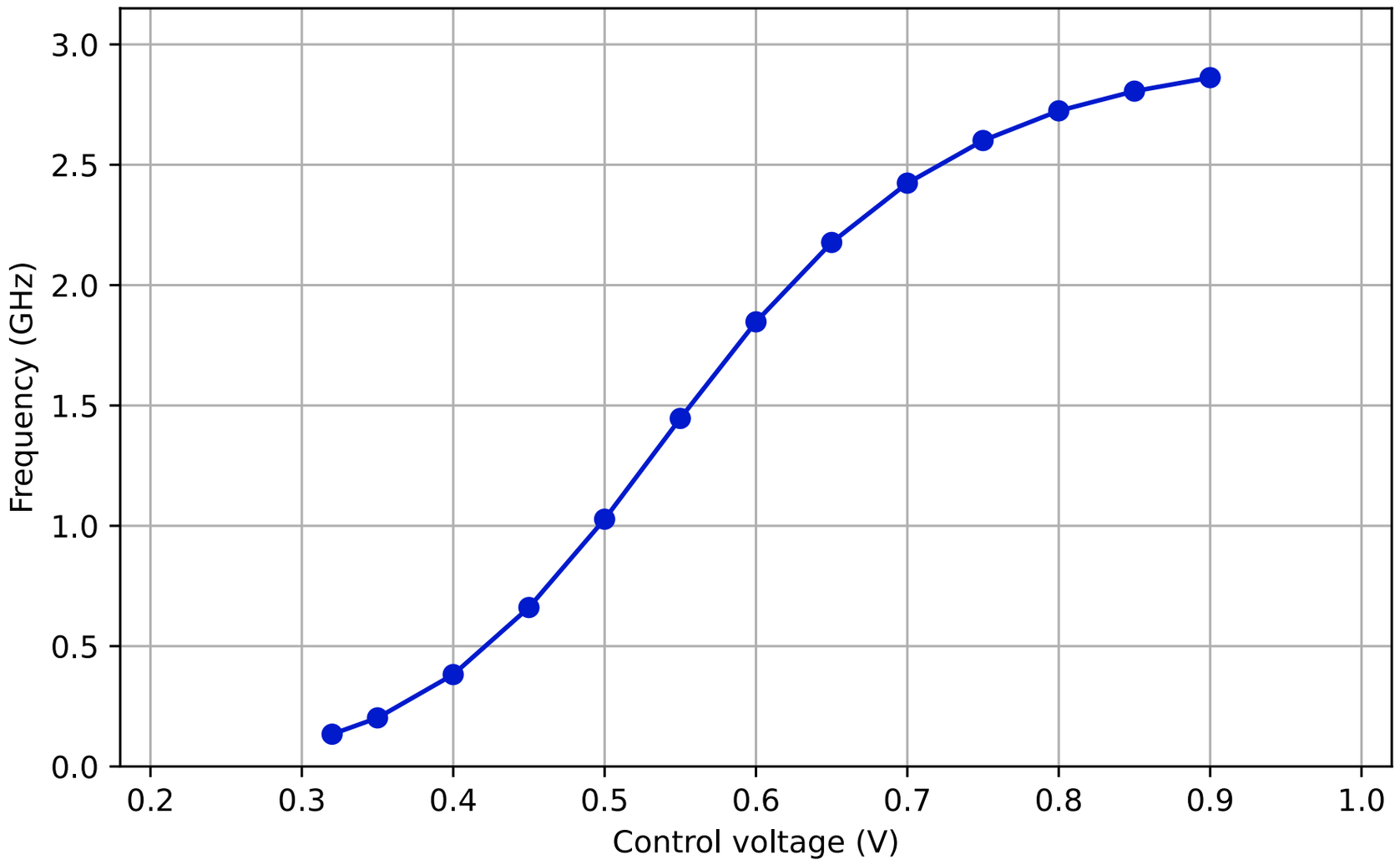
Analog VCO



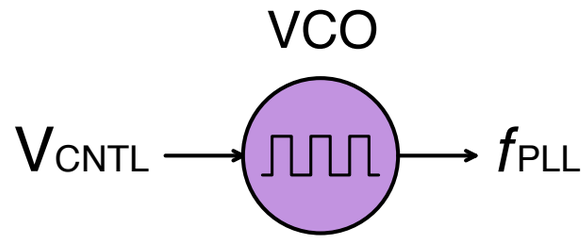
Analog control voltage V_{CNTL} adjusts frequency of oscillator to generate f_{PLL}

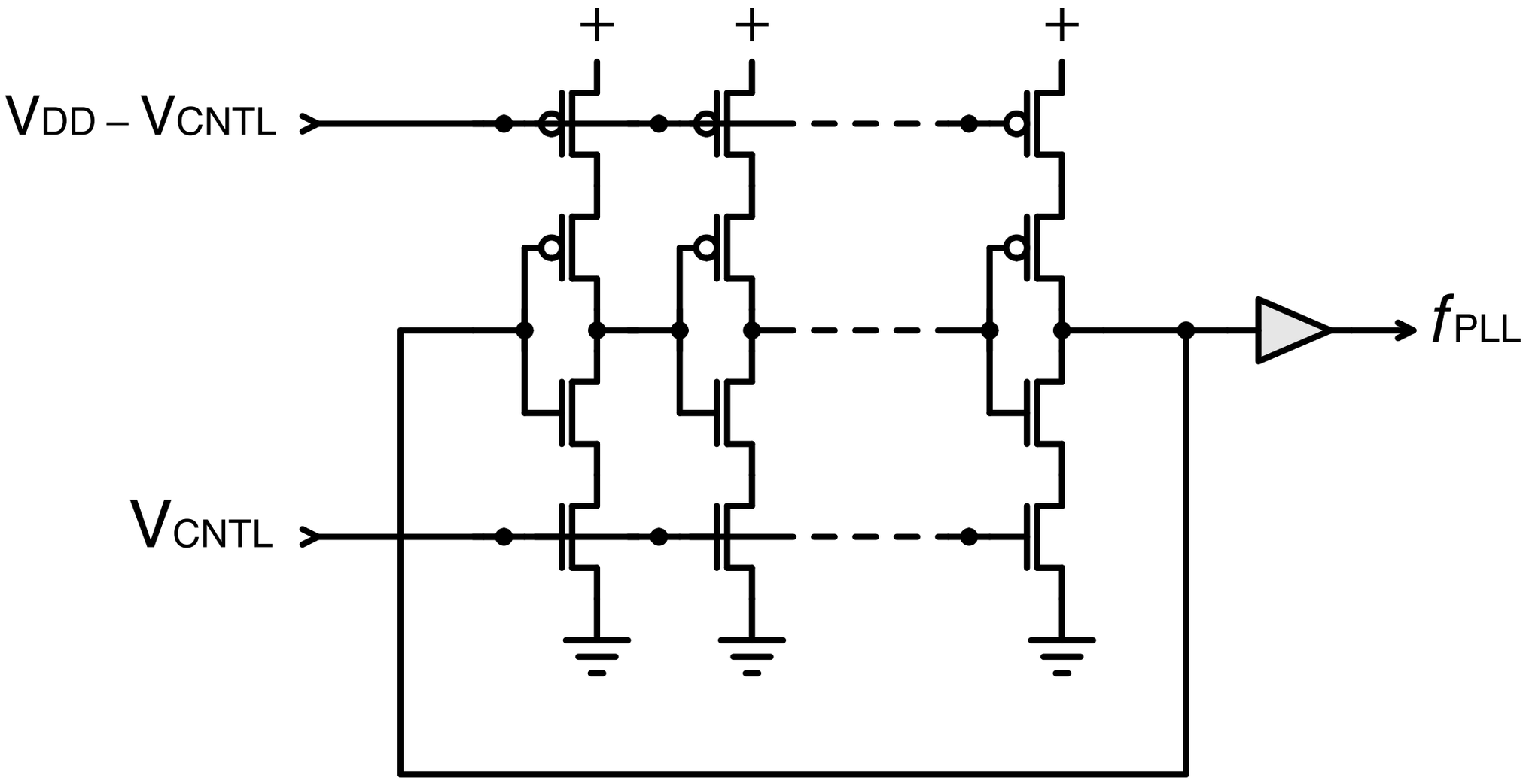
Analog VCO


VCO
VCNTL
fPLL

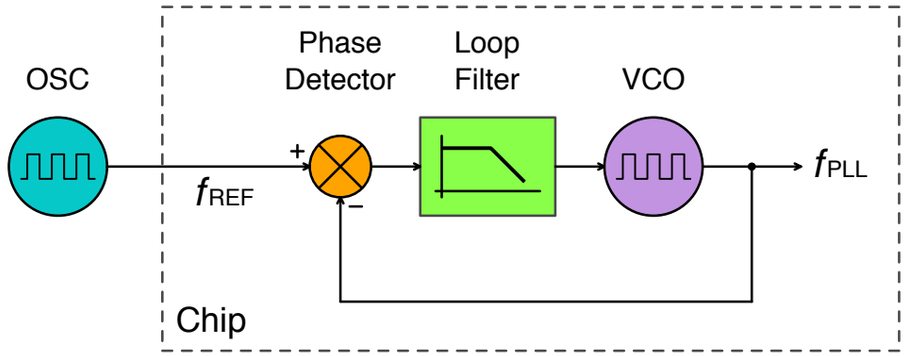

3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency (GHz)
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Control voltage (V)

Analog VCO





Components of Basic Analog PLL



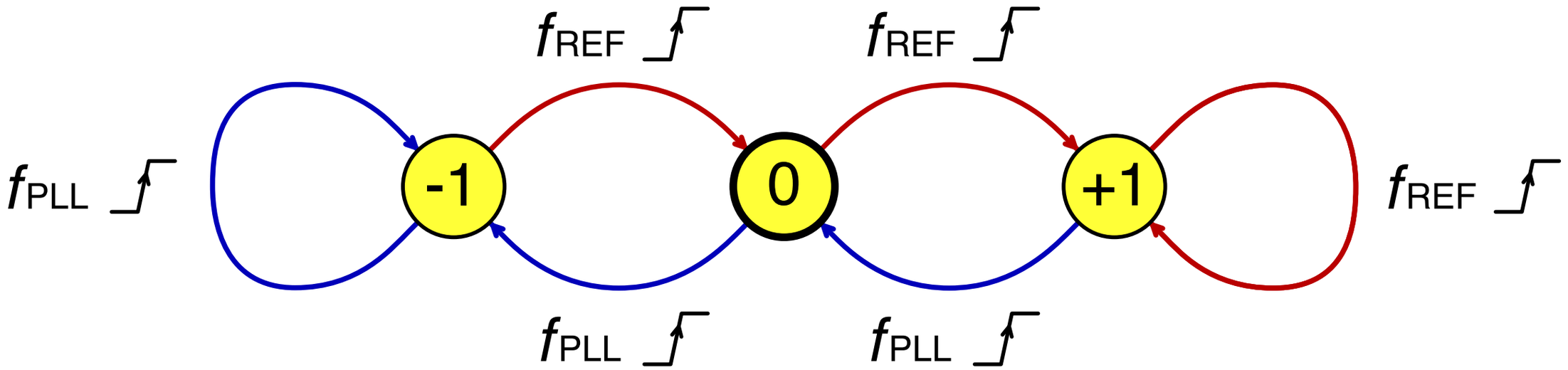
OSC – External clock reference, e.g., a quartz crustal oscillator

VCO – On-chip Voltage Controlled Oscillator, generates f_{PLL}

Phase Detector – Generates output signal that is proportional to the phase difference between f_{PLL} and f_{REF}

Loop Filter – Makes the control loop stable

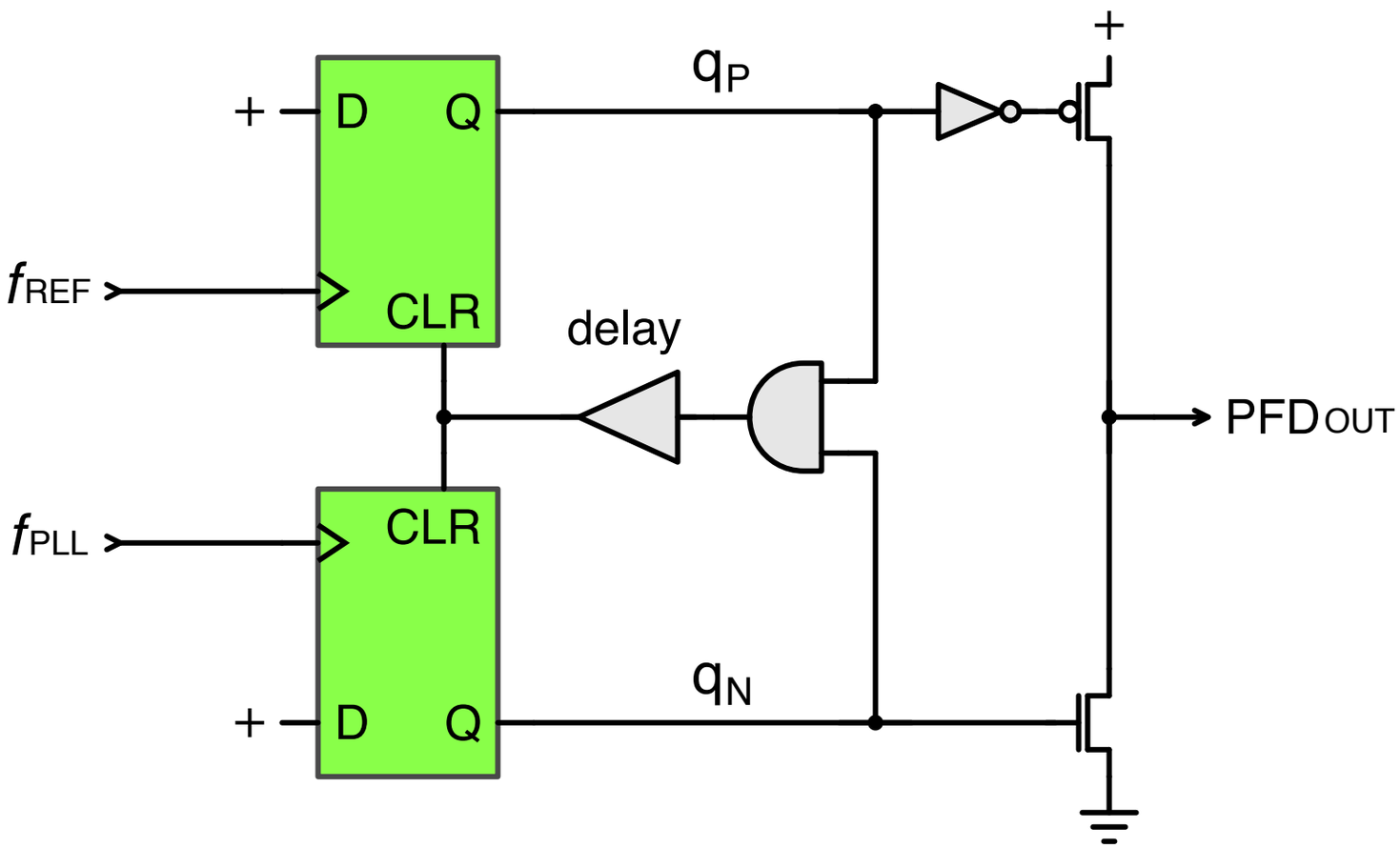
Phase Detector States



f_{REF} early: speed up VCO

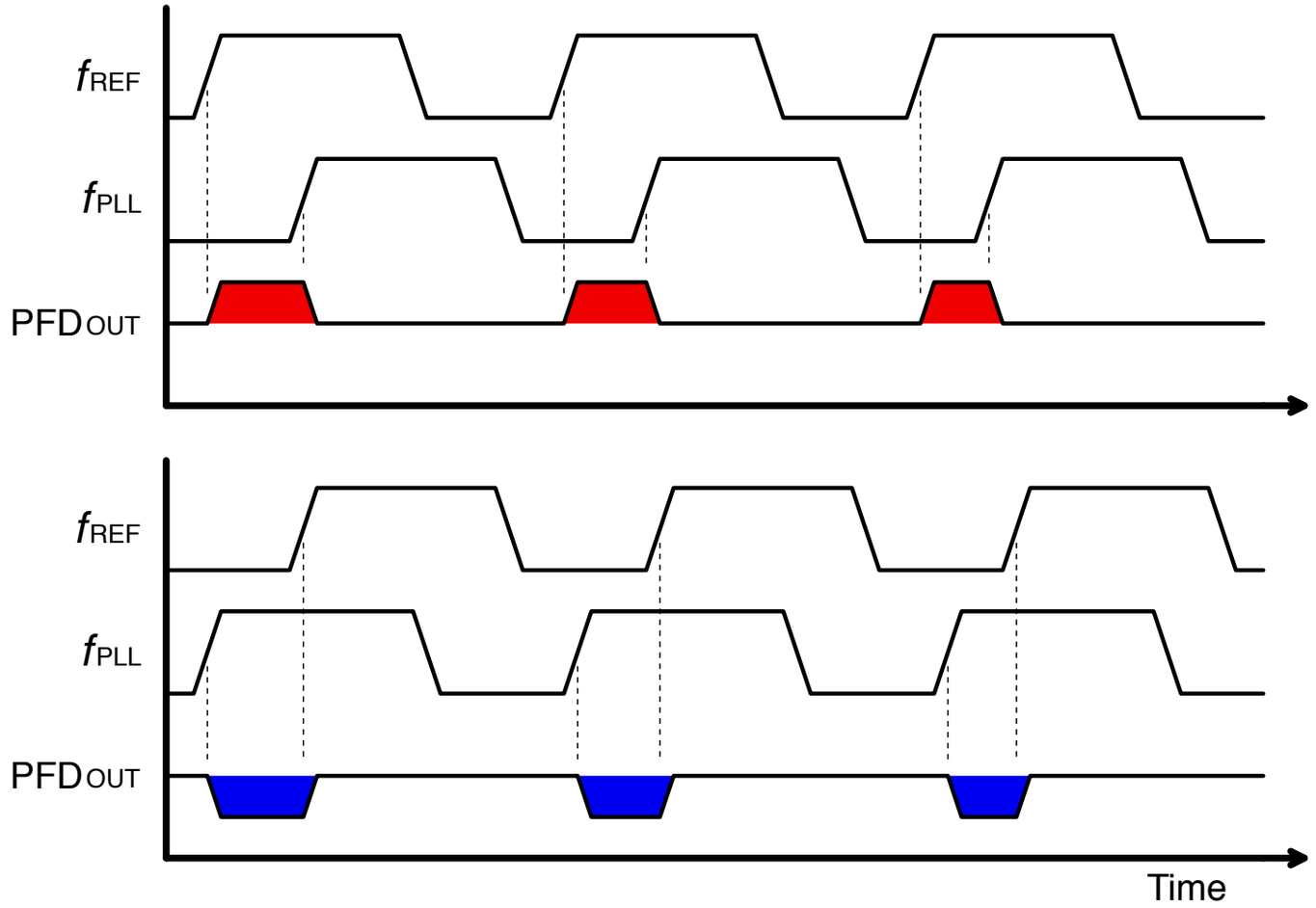
f_{PLL} early: slow down VCO

Analog PFD



Phase-Frequency Detector, PFD circuit.
The delay element helps compensate for non-ideal transistor operation.

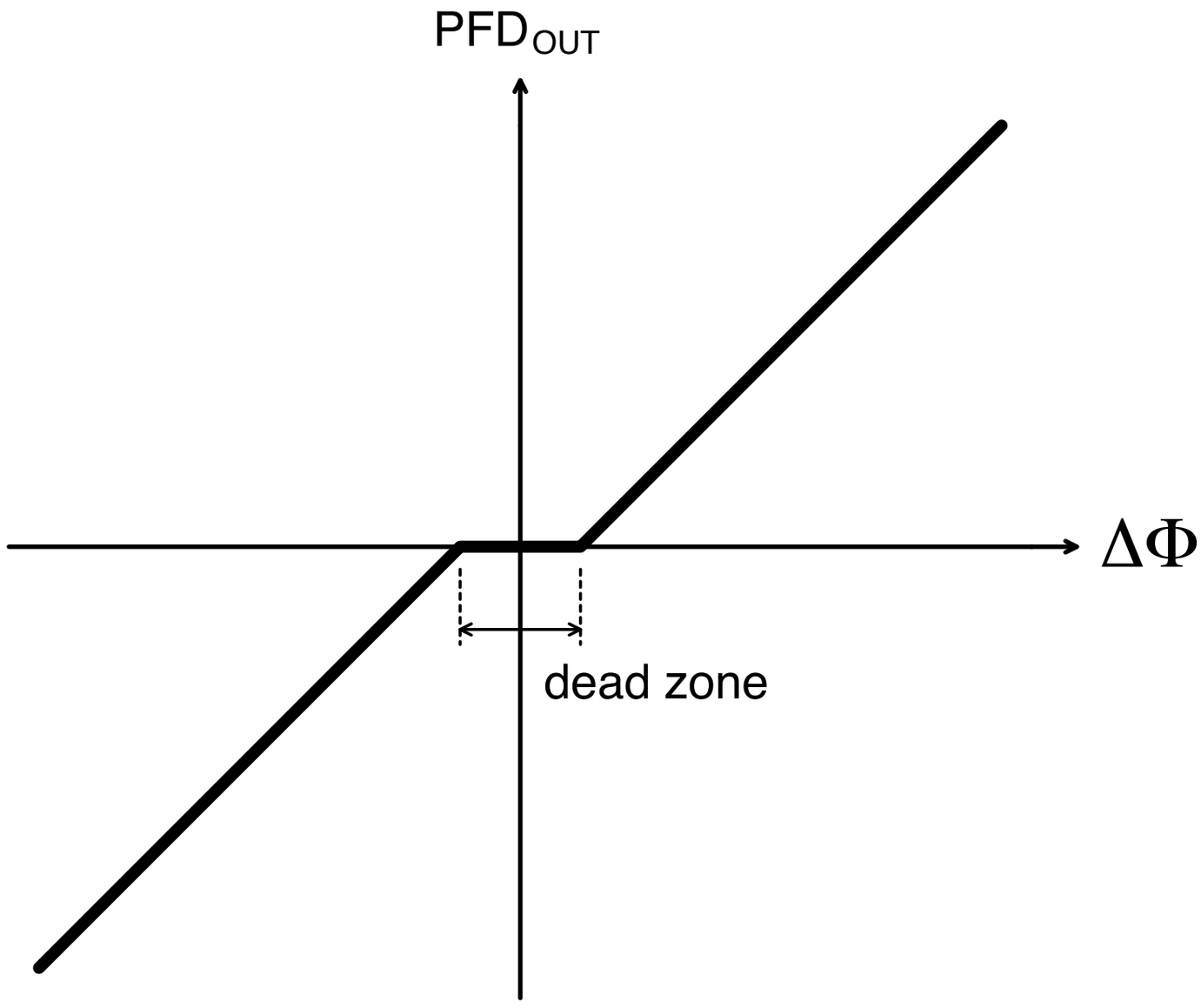
Analog PFD Waveforms



PFD_{OUT} high pulses: f_{PLL} leads f_{REF}

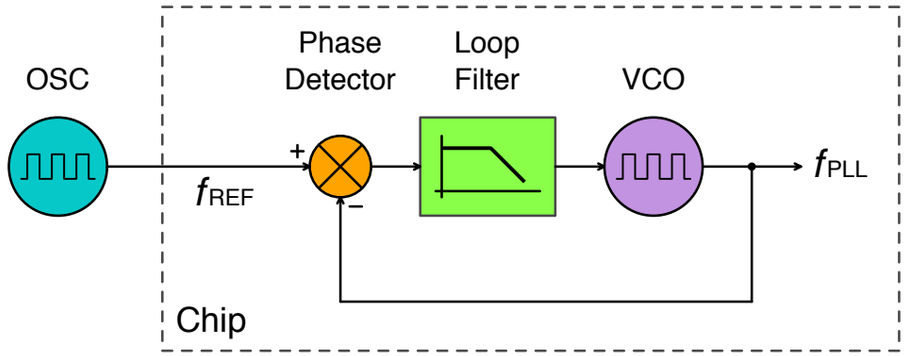
PFD_{OUT} low pulses: f_{PLL} lags f_{REF}

Analog PFD Dead Zone



The delay element removes the dead zone.

Components of Basic Analog PLL



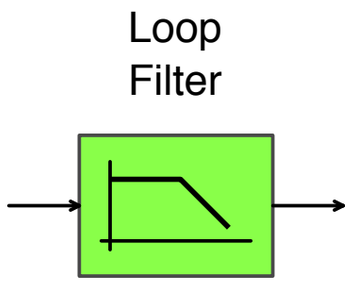
OSC – External clock reference, e.g., a quartz crustal oscillator

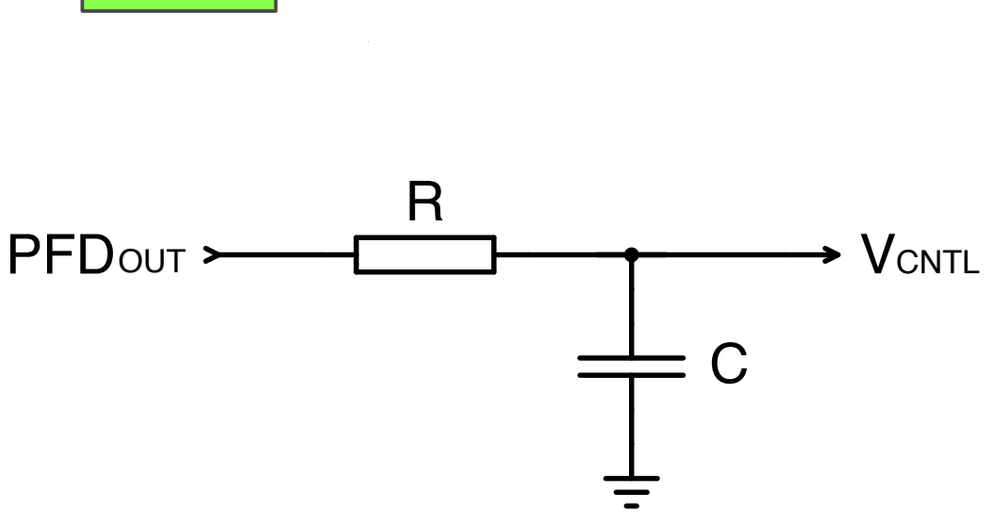
VCO – On-chip Voltage Controlled Oscillator, generates f_{PLL}

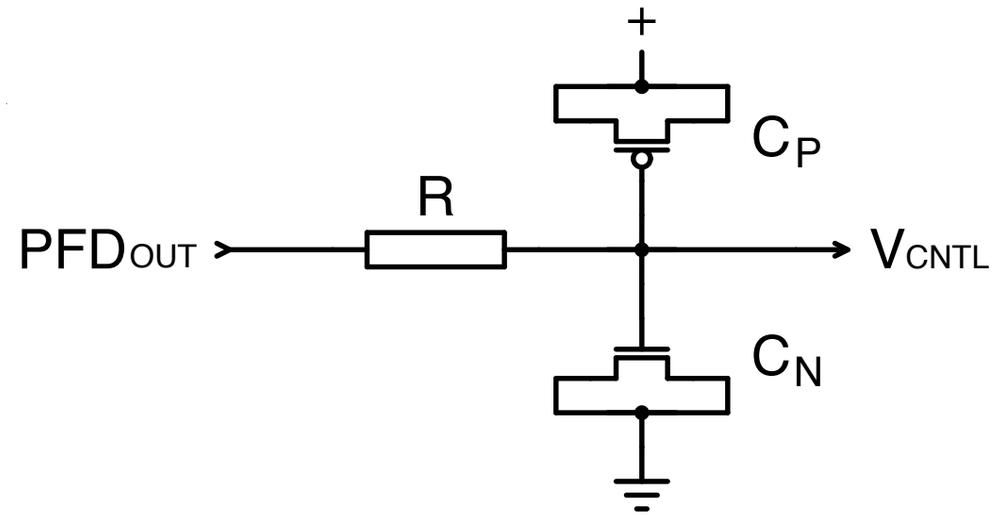
Phase Detector – Generates output signal that is proportional to the phase difference between f_{PLL} and f_{REF}

Loop Filter – Makes the control loop stable

Analog Low-Pass Loop Filter

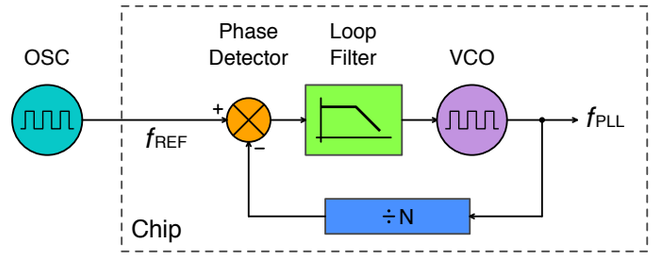


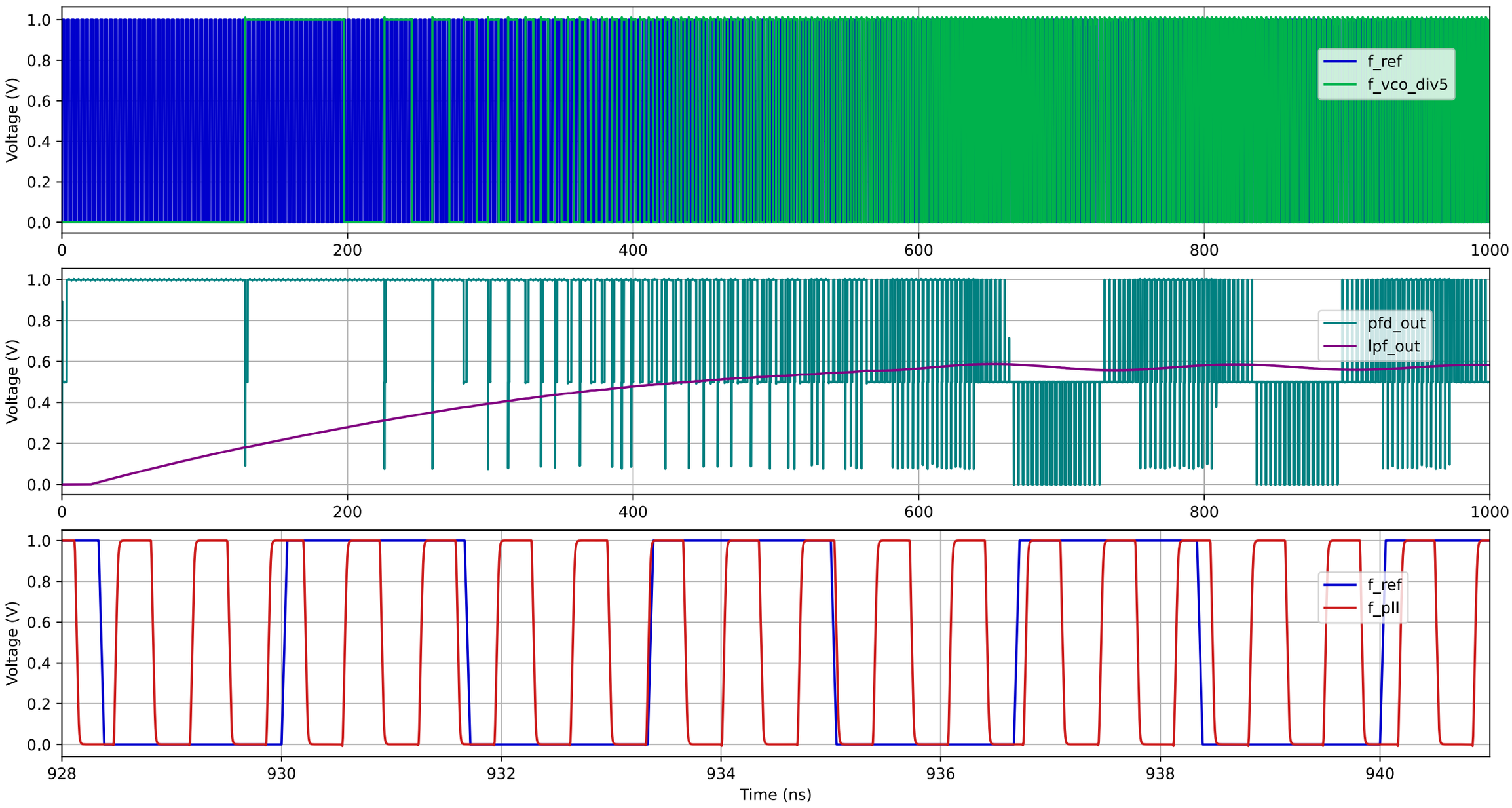




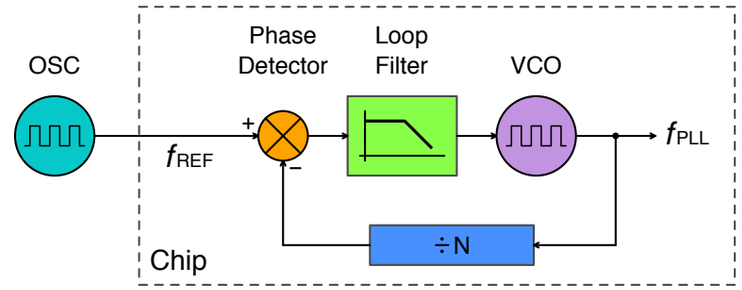
CMOS capacitors are non-ideal, but 10x smaller than the capacitance between metal layers.

Analog PLL


OSC
Phase Detector
Loop Filter
VCO
f_{REF}
f_{PLL}
+
−
÷N
Chip

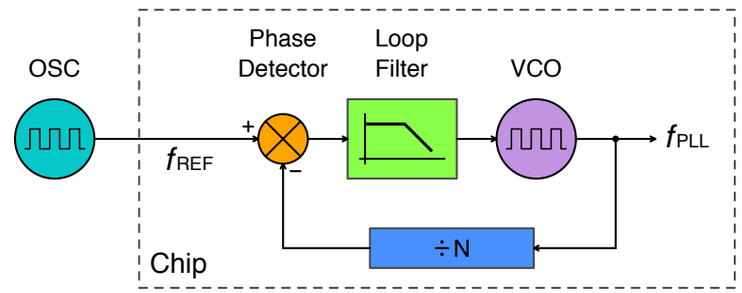

Voltage (V)
f_ref
f_vco_div5
pfd_out
lpf_out
f_ref
f_pll
Time (ns)

Discussion Session #2



- What have we learned so far?
- How does the Phase-Frequency Detector behave when $f_{REF} = 2 \times (f_{PLL} / N)$ and f_{REF} lags (f_{PLL} / N) ?
- Generate two phase-related clocks on the same chip, at 800MHz and 3.2GHz from a 160MHz f_{REF} ?
- The fabricated chip doesn't meet spec: the 3.2GHz module has a max frequency of 2.6GHz. What can we do?

Discussion Session #2



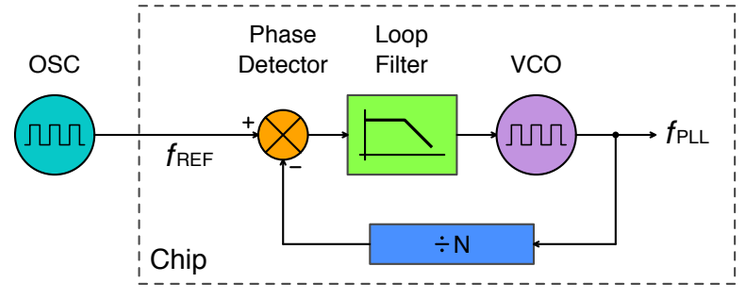
- The Phase-Frequency Detector behavior

when $f_{REF} = 2 \times (f_{PLL} / N)$:

Initially: generates pulses -ve pulses to slow the VCO down
Then: generates short pulses that speed up the VCO until

$f_{REF} = f_{PLL} / N$

Discussion Session #2



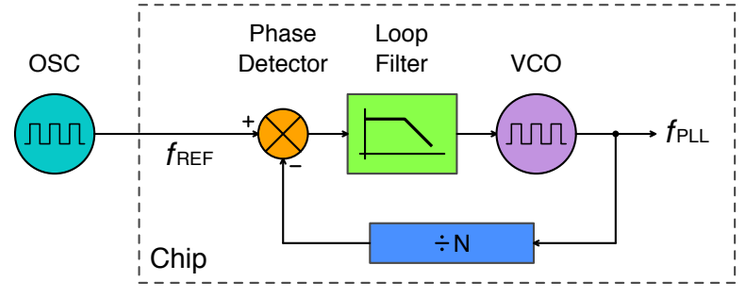
- Two phase-related clocks of 800MHz and 3.2GHz from a 160MHz f_{REF} ?

  Multiple ways, here's one way using two PLLs:

  PLL1:f_{REF} multiplier ratio is 5:1.
  PLL1:PLL2 clock ratio is 1:4.

  PLL1: N = 10, add a divide-by-2 at output ==> 800MHz
  PLL2: N = 20 ==> 3.2GHz

Discussion Session #2



- The fabricated chip doesn't meet spec: the 3.2GHz module has a max frequency of 2.6GHz. What can we do?

  Multiple ways, here's one:

  PLL2: N = 16 ==> 2.56GHz

  but might be better to preserve the 1:4 clock ratio:

  PLL1: N = 8, add a divide-by-2 at output ==> 640MHz
  PLL2: N = 16 ==> 2.56GHz

Next Lecture Session

Part 1:

- Overview
- Intro to PLLs
- Clocking with PLLs
- Analog PLLs

Part 2:

- Recap
- Digital PLLs
- Future challenges
- Summary