Introduction to PLL

Ahmed Ashry

Outline

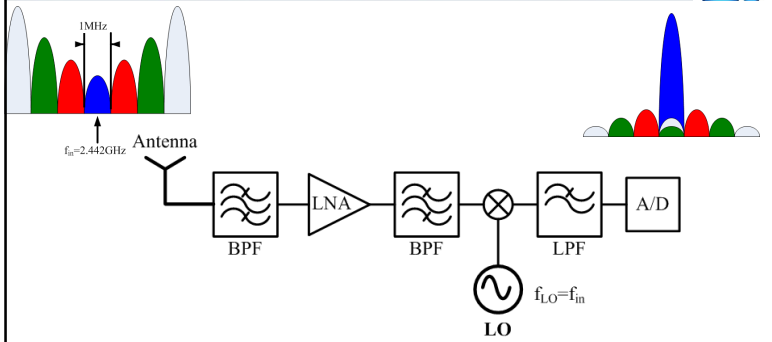
1. The need for PLL
2. PLL Specifications
3. PLL Structure
4. Type-I PLL Design
5. More PLL Types
6. Exercise (written + Matlab)

Outline (1/6)

(1) The need for PLL.

- **Why PLL is needed?**
- **Why not simple Oscillator?**
- **Why not simple VCO?**

Frequency selection



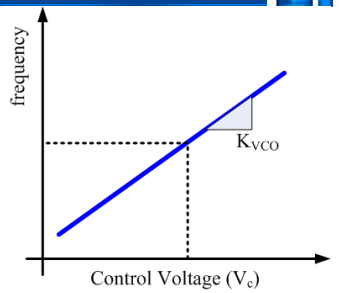
- LO frequency choose which channel to select.
- Error in LO frequency = wrong channel or more distortion

LO needed

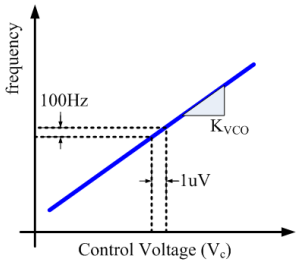
- Very accurate.
  - Example: GSM 900MHz ± 0.1ppm = ± 90Hz
  - On-chip LC oscillator = ± 10% = ± 90MHz
  - Crystal Oscillators (XO) are accurate, but:
    - Low Frequency ~ 10MHz.
    - Fixed.
- Programmable:
  - It is always needed to change the channel.
  - XOs are fixed.
  - What about VCO (Voltage Controlled Oscillator) ?

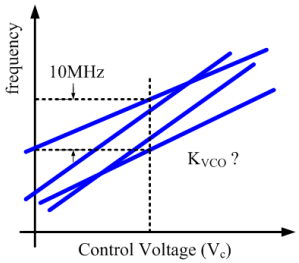
VCO is possible?

- Ideally, VCO can be used to produce any desired frequency.
- However, practically this is impossible.



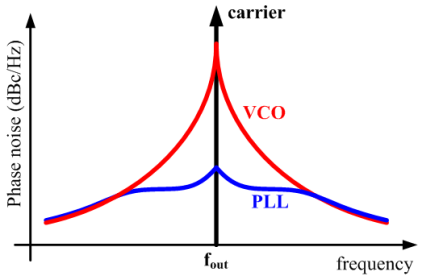
VCO is possible? No!





- The needed accuracy requires very stable control voltage
- Process and temperature variations make it more difficult

VCO is possible? No! (2)



- VCO (or generally free-running oscillator) has poor close-in phase noise compared to PLL.

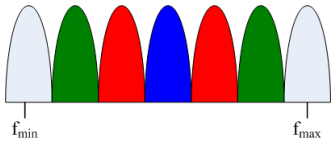
Outline (2/6)

(2) PLL Specifications.

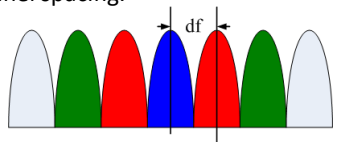
- **Tuning Range**
- **Step Size**
- **Settling Time**
- **More…….**

PLL Specifications (1)

- Tuning Range:
  - The frequency band covered by the PLL.
  - It must cover the target application.
  - GSM: 890MHz – 915MHz

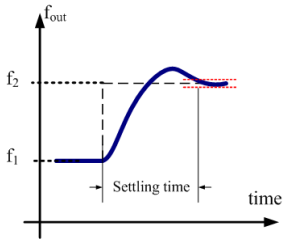


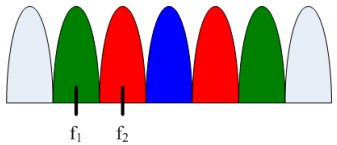
- Step size:
  - Must be at least equal to channel spacing.
  - GSM: 200kHz



PLL Specifications (2)

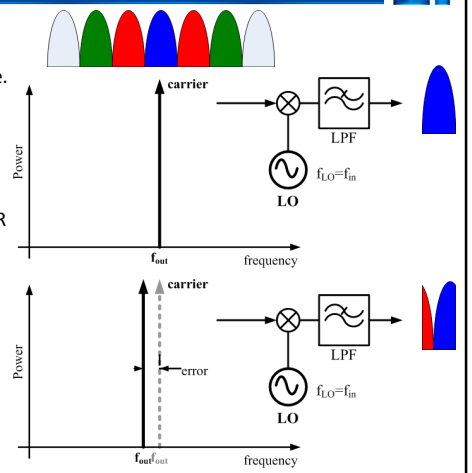
- Settling Time:
  - Fast switching between channels.
  - GSM: 280us (90Hz error)





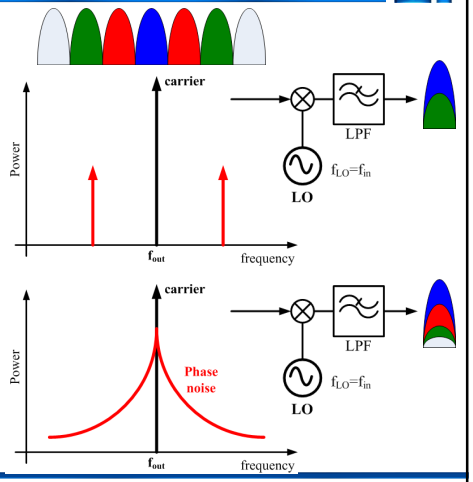
PLL Specifications (3)

- Frequency Stability:
  - Accuracy of reference source.
  - GSM: 90Hz/900MHz = 0.1ppm
  - Inaccurate frequency → wrong selection → worst BER



PLL Specifications (4)

- Spurs (undesired harmonics): cause interferes with other channels.
- Phase noise (random fluctuation of carrier frequency): cause interferes with other channels.



Outline (3/6)

(3) PLL Structure.

- **PLL Block diagram.**
- **PLL Linear Model.**
- **PLL Main Blocks.**
- **PLL Simple Design.**
- **PLL Phase Noise**

What is "Phase" ?

- It is simply the number of cycles.
- 1 cycle = 2π
- If the frequency is constant:

$$\varphi(t) = \omega \cdot t + \varphi_o$$

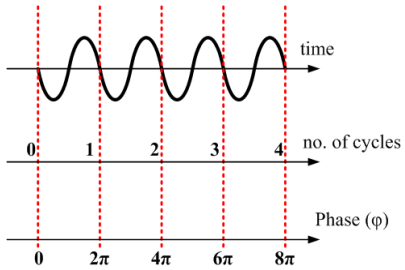
$$V(t) = A \cdot \sin(\omega \cdot t + \varphi_o)$$

$$V(t) = A \cdot \sin(\varphi(t))$$

- Or, generally:

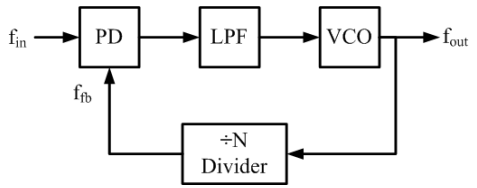
$$\varphi(t) = \int_0^t \omega(t) \cdot dt$$

$$\omega(t) = \frac{d}{dt}\phi(t)$$



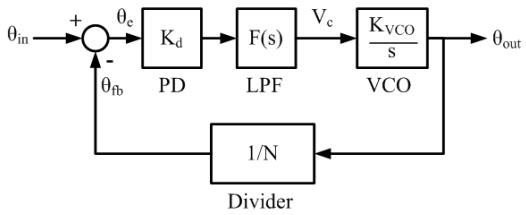
PLL (Phase-Locked Loop)

- PLL is a negative feedback system.
- It compares "Phase" not "Amplitude"
- When PLL is in lock (steady-state), feedback phase is equal to the input phase.



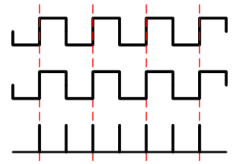
PLL Linear Model

- PLL is a non-linear system.
- However, under certain conditions (near lock), it can be “linearized”.
- State variable is the phase.

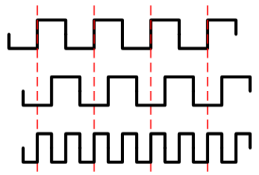


PLL Blocks (1-Phase detector) 1

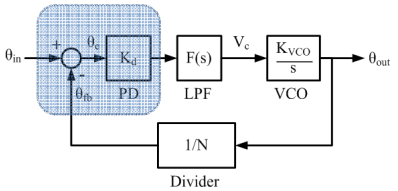
- XOR is the simplest PD



$\Delta\phi = 0 \rightarrow D = 0\% \rightarrow V_d = 0$

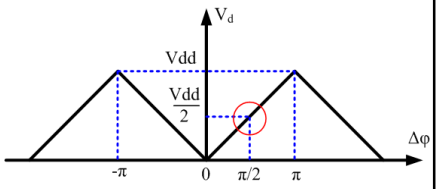


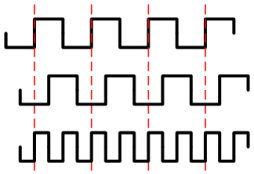
$\Delta\phi = \pm\frac{\pi}{2} \rightarrow D = 50\% \rightarrow V_d = Vdd/2$



PLL Blocks (1-Phase detector) 2

- For stable operation, the loop should be designed, such that the nominal output of the PD is Vdd/2
- $K_d = Vdd/\pi$

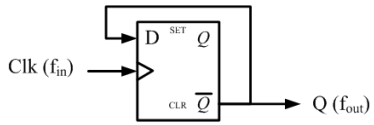


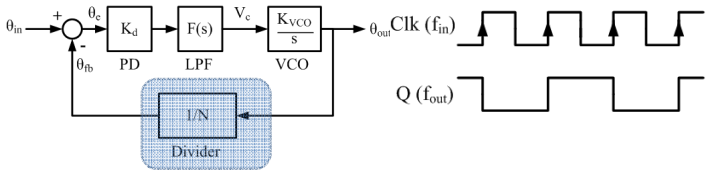


$\Delta\phi = \pm\frac{\pi}{2} \rightarrow D = 50\% \rightarrow V_d = Vdd/2$

PLL Blocks (2-Divider)

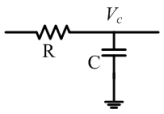
- Toggle Flip-Flop T-FF can acts a divide by 2.
- Cascading T-FFs can divide by 2^n
- Adding some logic can extend the division ratio to any integer.

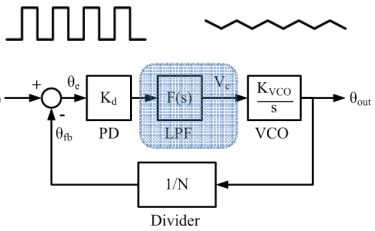


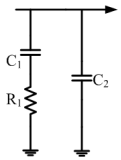


PLL Blocks (3-LPF)

- Filters the PD output.
- Weak filtering (Wide BW) leads to “Spurs”.
- Small BW leads to longer settling time.
- Filter structure determines PLL type.

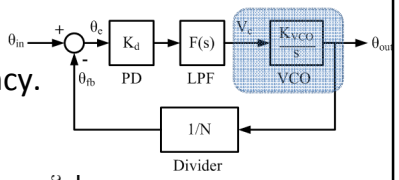






PLL Blocks (4-VCO)

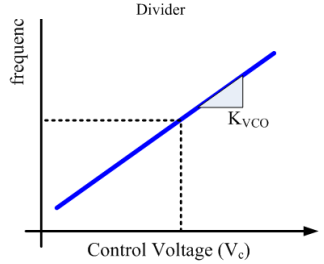
- VCO: **V**oltage **C**ontrolled **O**scillator.
- Kvco: VCO sensitivity.
- fo: Free-running frequency.



$$\omega(t) = K_{VCO} \cdot V_C$$

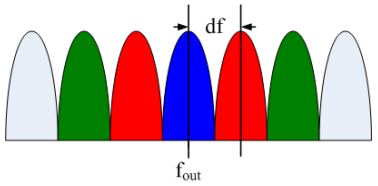
$$\varphi(t) = \int_0^t \omega(t) \cdot dt$$

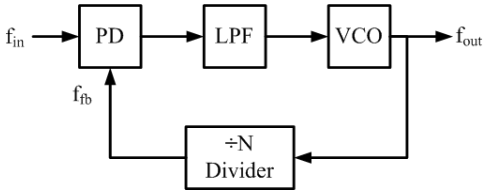
$$\varphi(s) = \frac{K_{VCO}}{s} \cdot V_C$$



PLL Design

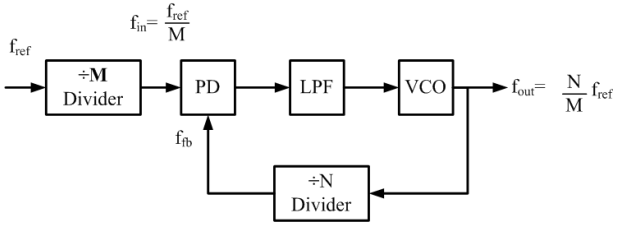
- Given:
  - Output frequency f_{out}
  - Channel spacing. df
- Required:
  - f_{ref}=?
  - N=?
- Steps:
  - $f_{ref} = df$
  - $N = f_{out} / f_{ref}$





Fractional PLL

- What if the required channel spacing is smaller than the available reference frequency?
  - Solution (1): Fractional-N PLL (out of our lecture scope)
  - Solution (2): Reference division:



Design Example

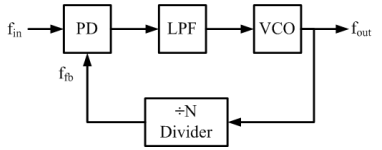
- Design a PLL for GSM (fout = 900MHz, df=200kHz)
- Available reference is 13MHz.

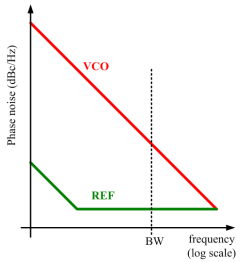
- Solution:
  - M=13MHz/200kHz = 65
  - N= 900MHz/200kHz = 4500

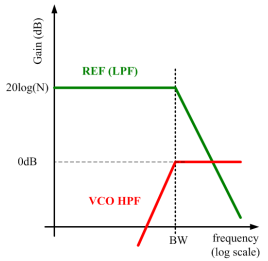
ANALOG DEVICES Analog Dialogue

PLL Phase noise (1)

- PLL phase noise is mainly due to main sources:
  1. Reference (input)
     Low phase noise (clean source).
     Noise gain (N).
     Low-pass Filter.
  2. VCO
     Higher phase noise.
     Unity noise gain.
     High-pass Filter.

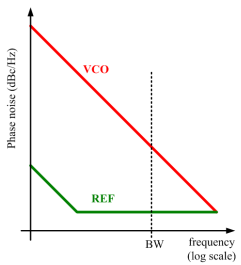


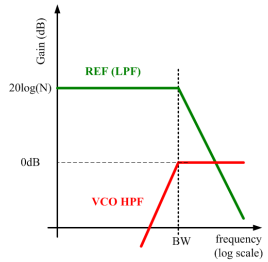


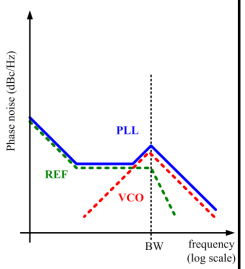


Phase noise (2)

- Each noise source is shaped with the corresponding transfer function
  - Reference: Gain=N LPF. (Close-in noise)
  - VCO: Gain=1 HPF (Far-out noise)
- It is clear that (N) and (BW) determine PLL Phase Noise.







Outline (4/6)

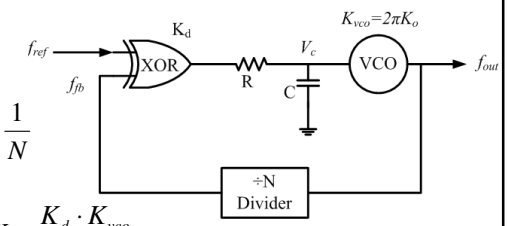
(4) Type-I PLL.

- **Type-I Design Equations.**
- **Design Example.**

Type I PLL

- Type I means there is only “one” integrator, which is the VCO itself.
- Very simple structure.
- Rarely used in communication systems. (Type-II is preferred)
- Open-loop gain:

$$A(s) = K_d \cdot F(s) \cdot \frac{K_{vco}}{s} \cdot \frac{1}{N}$$



$$A(s) = \frac{K}{s(1+sRC)} \qquad K = \frac{K_d \cdot K_{vco}}{N}$$

Type I PLL

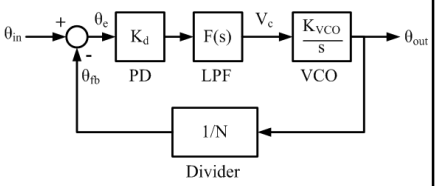
- Second order feedback system
- Open loop gain:

$$A(s) = \frac{K}{s(1+sRC)}$$

- Closed loop gain:

$$B(s) = N\frac{A(s)}{1+A(s)}$$

$$B(s) = N\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$



- Natural frequency:

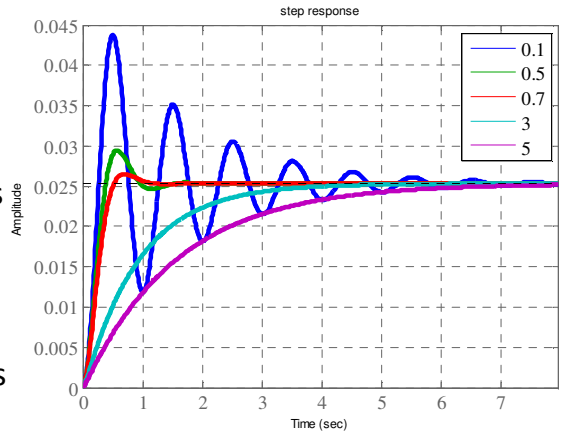
$$\omega_n = \sqrt{\frac{K}{RC}}$$

- Damping coefficient:

$$\xi = \frac{1}{2\sqrt{K \cdot RC}}$$

Zeta effect ξ

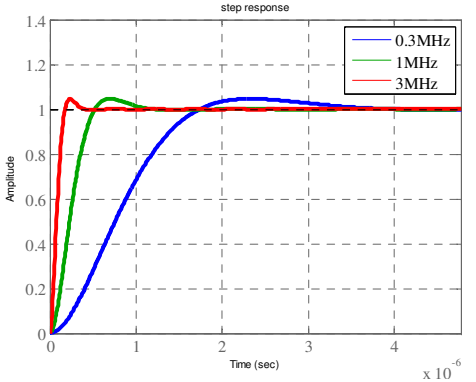
- Low values cause overshoot
- High values cause slow settling
- Optimum is **0.707**



BW effect $\omega_n = 2\pi \cdot f_n$

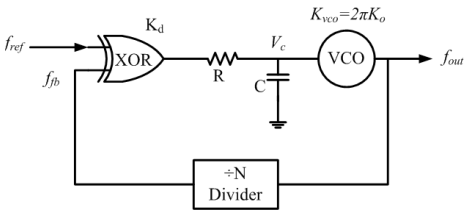
- Higher BW means fast settling.
- Settling with 5% error, Approximately:

$$t_s \approx \frac{4.6}{\xi \cdot \omega_n} \approx \frac{1}{f_n}$$



PLL Type-I Design (1)

- Given:
  - Output frequency f_{out}
  - Channel spacing. df
  - Settling time. t_s
  - XOR detector
- Required:
  - f_{ref}=?
  - N=?
  - K_o=?



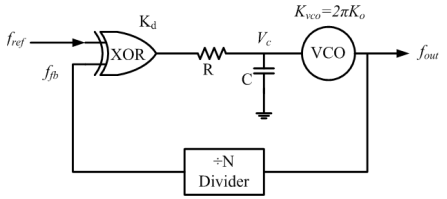
PLL Type-I Design (2)

Steps:

$f_{ref} = df$

$N = f_{out} / f_{ref}$

$f_n = 1/t_s$



$\omega_n = 2\pi \cdot f_n$ $\quad \xi = 0.707$

$RC = \frac{1}{2\xi\omega_n}$ $\quad K = \frac{\omega_n}{2\xi}$

- XOR detector -> $K_d = Vdd/\pi$

$K_{vco} = \frac{N \cdot K}{K_d}$

Exercise 1

- Design a type-I PLL with the following specifications:
  - Output center frequency 60MHz.
  - Channel spacing 1MHz.
  - Settling time 20us

Solution (1)

- $f_{ref} = df = 1MHz$
- $N = f_{out} / f_{ref} = 60\text{MHz}/1\text{MHz} = 60$
- $f_n = 1/t_s = 50kHz$

$\omega_n = 2\pi \cdot f_n = 310kr/s$ $\quad \xi = 0.707$

$RC = \frac{1}{2\xi\omega_n} = 2.3\mu s$ $\quad K = \frac{\omega_n}{2\xi} = 220kr/s$

$R = 1k\Omega \quad C = 2.3nF$

Solution (2)

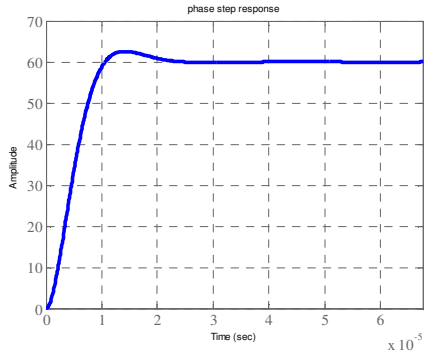
- $K_d = Vdd/\pi = 0.38V/rad$

$$K_{vco} = \frac{N \cdot K}{K_d} = 35M(r/s)/V$$

$$K_o = \frac{K_{vco}}{2\pi} = 5.6MHz/V$$

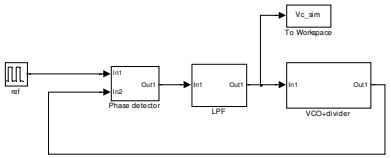
Matlab Verification (Linear Model)

- Matlab can be used to verify the linear model.

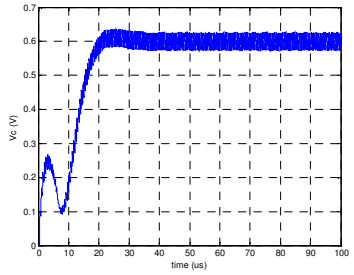


Matlab Verification (Time Model)

- Simulink Model.



- Control voltage settling.
- Ripples due to reference signal.



Outline (5/6)

(5) More PLL Types.

- **Type-II PLL.**
- **Fractional-N PLL.**
- **All-Digital PLL.**

Type-I Limitations (1)

- Open loop gain:

$$A(s)=\frac{K}{s(1+sRC)}$$

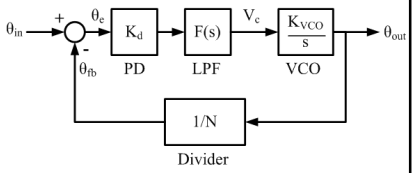
- Phase error :

$$E(s)=\frac{1}{1+A(s)}$$

$$E(s)=\frac{s\cdot\left(s+1/RC\right)}{s^2+2\xi\omega_n s+\omega_n^2}$$

- Phase step:

$$\theta_{in}(s)=\frac{\Delta\varphi}{s}$$



- Error due to phase step:

$$\theta_e(s)=\frac{\left(s+1/RC\right)\cdot\Delta\varphi}{s^2+2\xi\omega_n s+\omega_n^2}$$

$$\theta_e(s)\big|_{s=0}\neq 0 \qquad \theta_e(t)\big|_{t=\infty}\neq 0$$

- Steady-state error.

Type-I Limitations (2)

- Solution:
  - Add extra pole at origin, i.e. additional integrator.
  - This adds additional "s" to the equation:

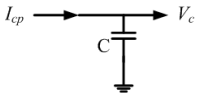
$$E(s)=\frac{s^2\cdot\left(s+1/RC\right)}{s^2+2\xi\omega_n s+\omega_n^2} \qquad \theta_e(s)=\frac{s\left(s+1/RC\right)\cdot\Delta\varphi}{s^2+2\xi\omega_n s+\omega_n^2}$$

$$\theta_e(s)\big|_{s=0}=0 \qquad \theta_e(t)\big|_{t=\infty}=0$$

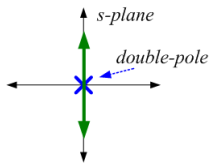
  - Now, we have "2" poles at origin (Filter + VCO) → **Type-II**

Type-II Filter Implementation (1)

- Integrator: Capacitor.
- But, with 2 poles at origin, the root-locus lies on the imaginary axis.
- The PLL is unstable (marginally).
- Filter modification is needed.

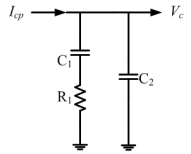


$$Z(s)=\frac{1}{s\cdot C}$$

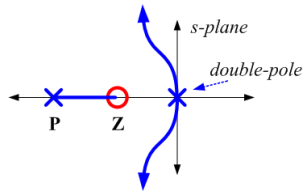


Type-II Filter Implementation (2)

- Adding extra RC section.
- Adds additional pole and zero.
- Z<P → system is conditionally stable.
- By proper design, the desired BW and damping coefficient (zeta) can be obtained.

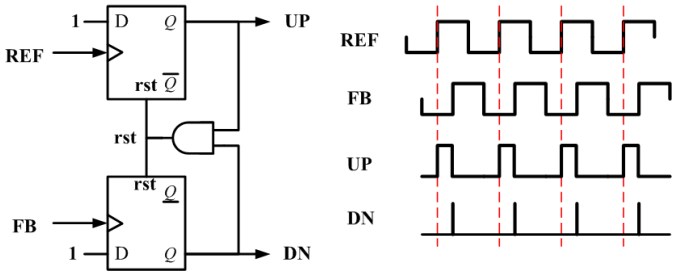


$$Z(s)=\frac{k(s+Z)}{s\cdot(s+P)}$$



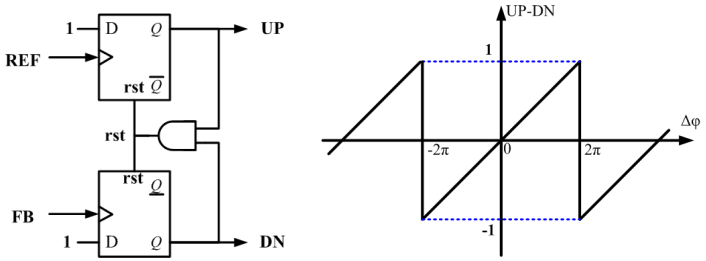
Type-II PFD (1)

- Phase detection is usually done using PFD (Phase-Frequency Detector) “instead of XOR”



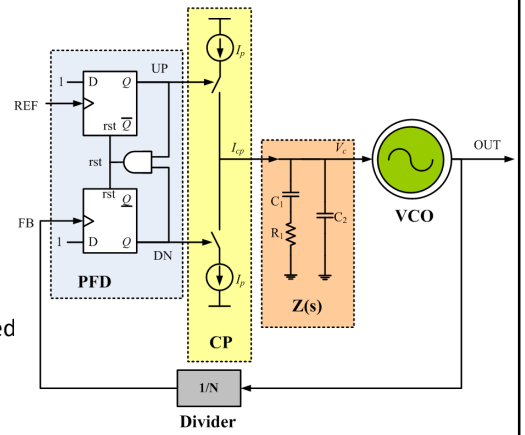
Type-II PFD (2)

- Better phase and frequency detection range.



Complete Type-II PLL

- PFD:
  - Phase-Frequency Detector.
- CP:
  - Charge Pump.
- Z(s):
  - Filter
- VCO:
  - Voltage-Controlled Oscillator



Outline (6/6)

(6) Exercise.

- **Written part.**
- **Matlab part.**

References

- **"Introduction to Charge Pump PLL Frequency Synthesizers", Ayman Ahmed, Si-Ware Systems.**

- **National Application Note:**

http://www.national.com/an/AN/AN-1001.pdf#page=1

- **Fujitsu Application Note:**

http://www.siliconrfsystems.com/Papers/U11614%20PLL%20Basics-%20Fujitsu.pdf

Thank you

Questions?