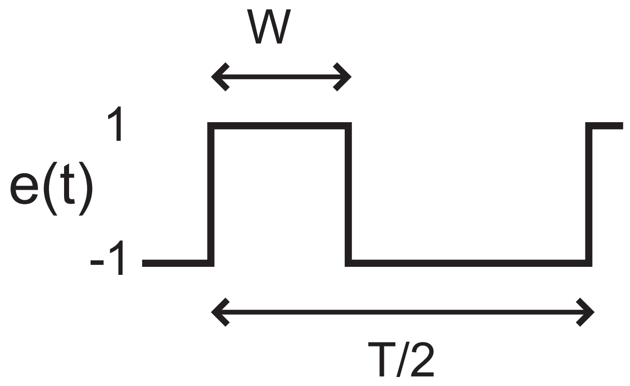
Modeling of XOR Phase Detector

- Average value of pulses is extracted by loop filter
  - Look at detector output over one cycle:



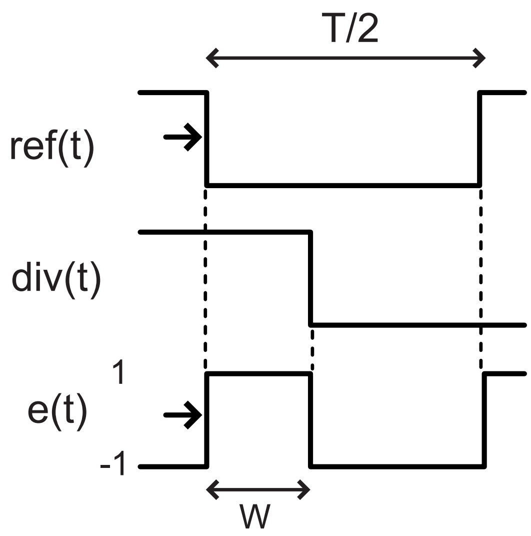
- Equation:

$$avg\{e(t)\} = -1 + 2\frac{W}{T/2}$$

Relate Pulse Width to Phase Error

- **Two cases:**

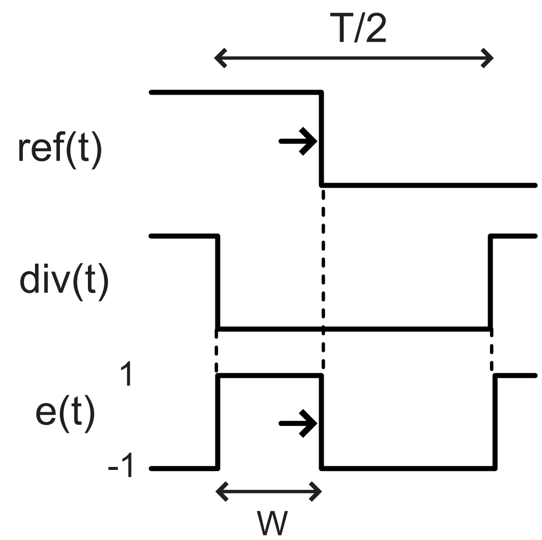
$$-\pi < \Phi_{ref} - \Phi_{div} < 0$$



$$\Downarrow$$

$$W = -\left(\frac{\Phi_{ref} - \Phi_{div}}{\pi}\right) T/2$$

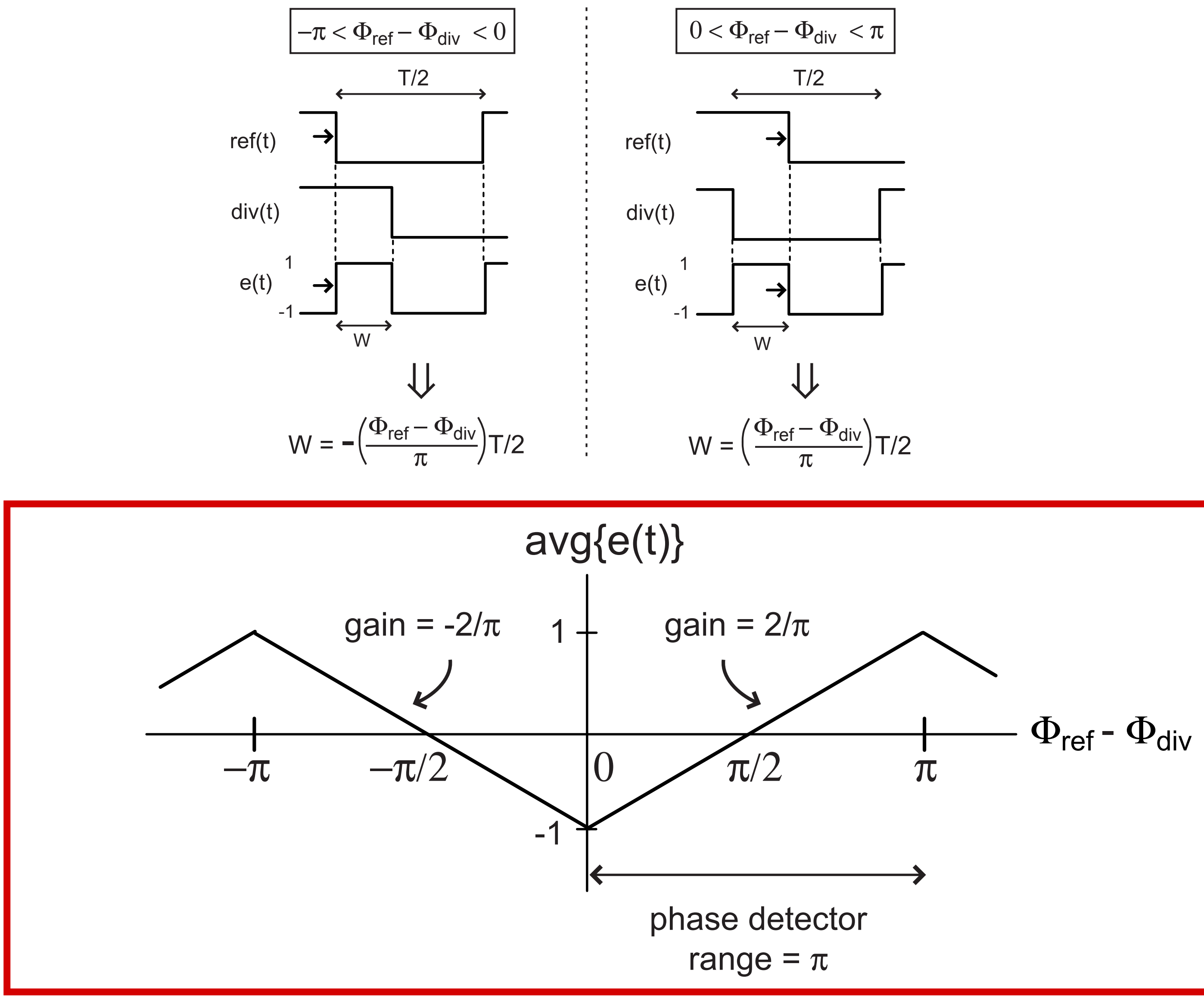
$$0 < \Phi_{ref} - \Phi_{div} < \pi$$



$$\Downarrow$$

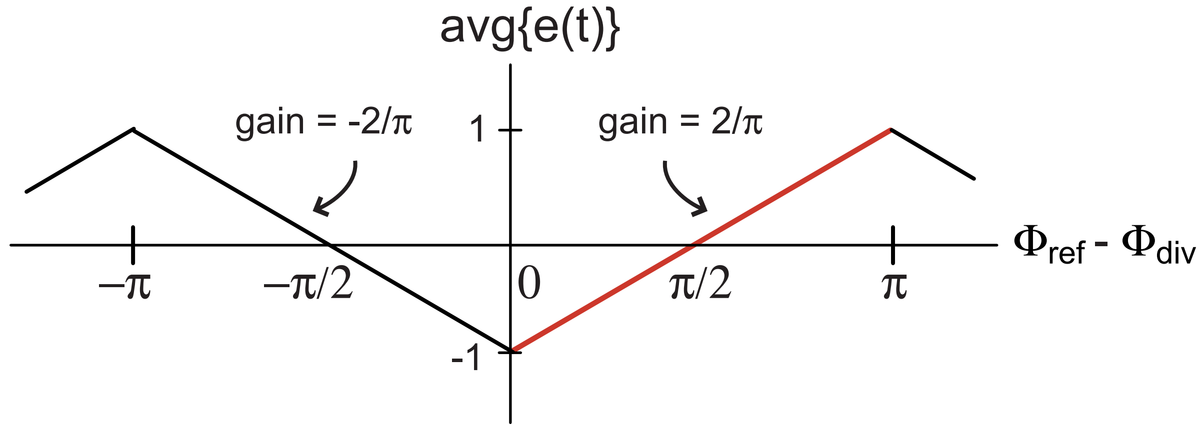
$$W = \left(\frac{\Phi_{ref} - \Phi_{div}}{\pi}\right) T/2$$

Overall XOR Phase Detector Characteristic

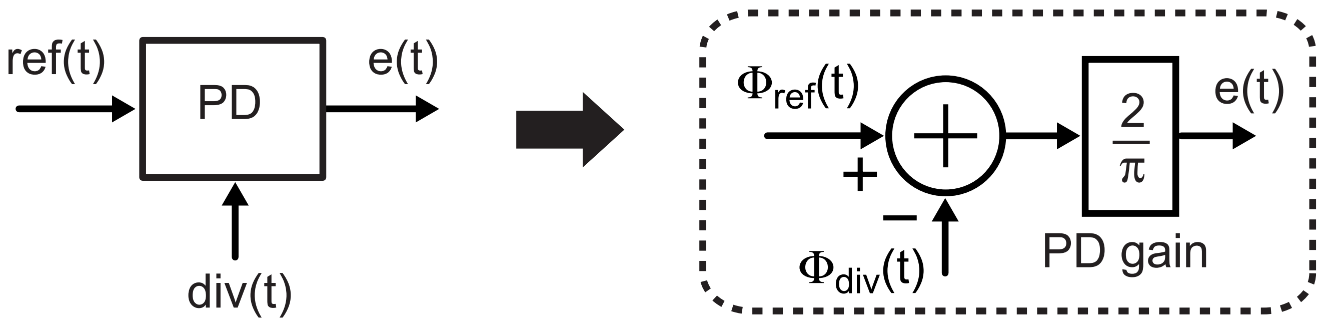


Frequency-Domain Model of XOR Phase Detector

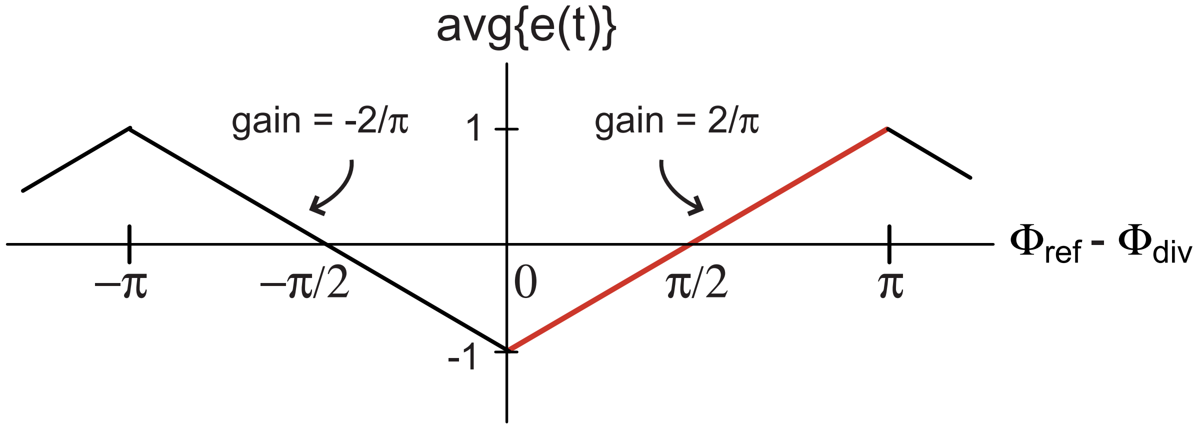
- Assume phase difference confined within 0 to π radians
  - Phase detector characteristic looks like a constant gain element



- Corresponding frequency-domain model



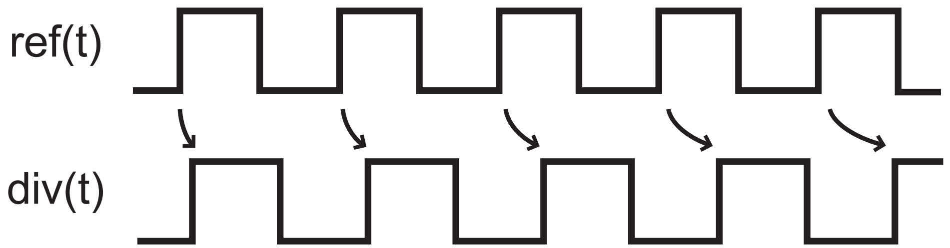
Recall Phase Detector Characteristic



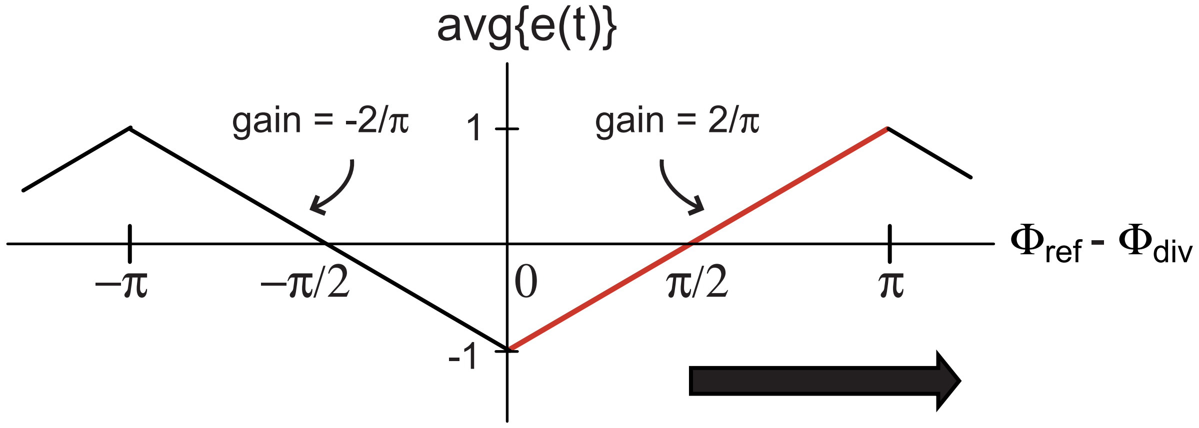
- To simplify modeling, we assumed that we always operated in a confined phase range (0 to π)
  - Led to a simple PD model
- Large perturbations knock us out of that confined phase range
  - PD behavior varies depending on the phase range it happens to be in

Cycle Slipping

- **Consider the case where there is a frequency offset between divider output and reference**
  - **We know that phase difference will accumulate**

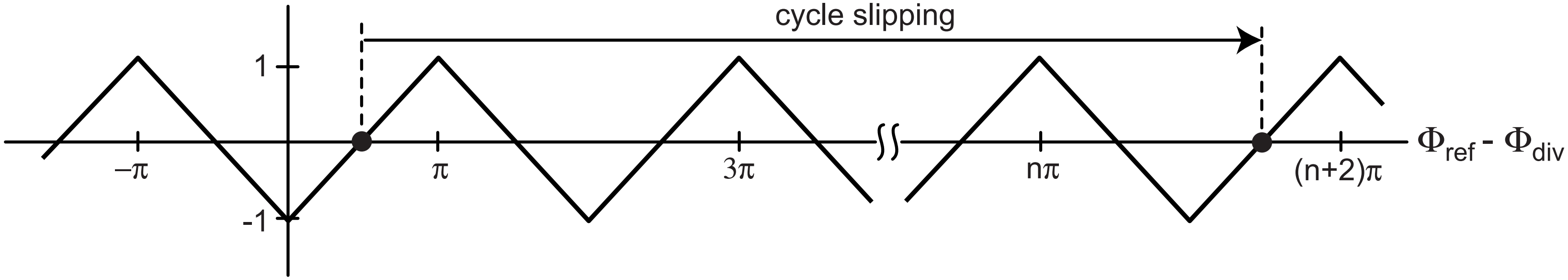


- **Resulting ramp in phase causes PD characteristic to be swept across its different regions (cycle slipping)**



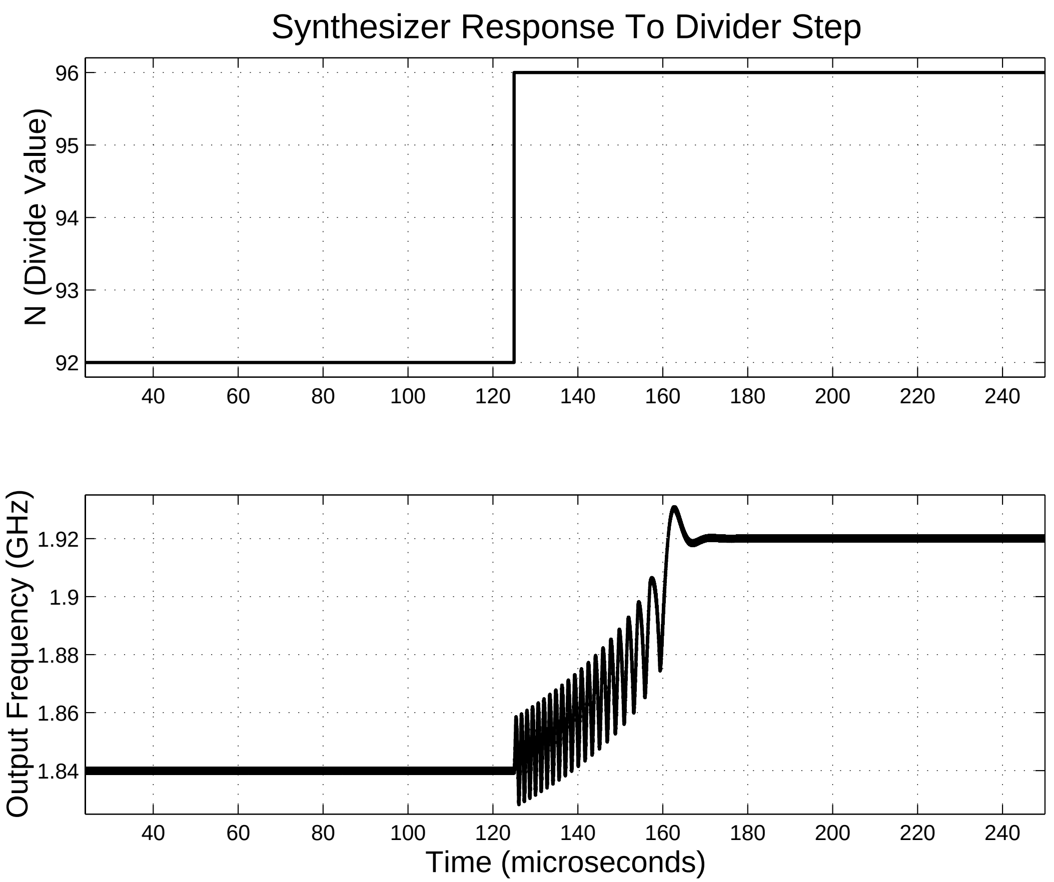
Impact of Cycle Slipping

- **Loop filter averages out phase detector output**
- **Severe cycle slipping causes phase detector to alternate between regions very quickly**
  - **Average value of XOR characteristic can be close to zero**
  - **PLL frequency oscillates according to cycle slipping**
  - **In severe cases, PLL will not re-lock**
    - PLL has finite frequency lock-in range!



Back to PLL Response Shown Previously

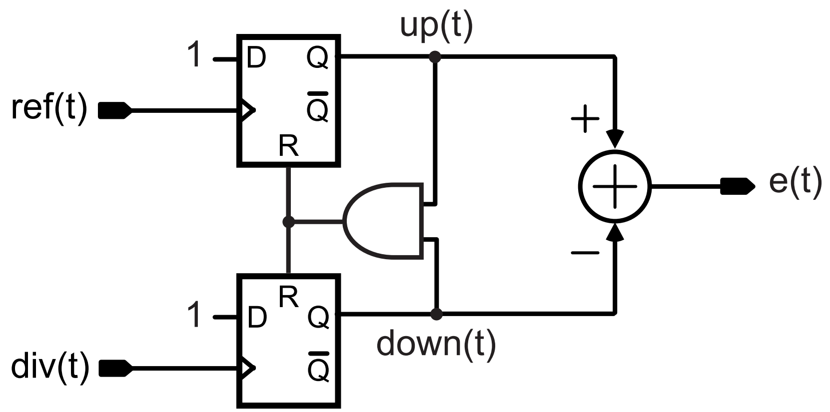
- PLL output frequency indeed oscillates
  - Eventually locks when frequency difference is small enough

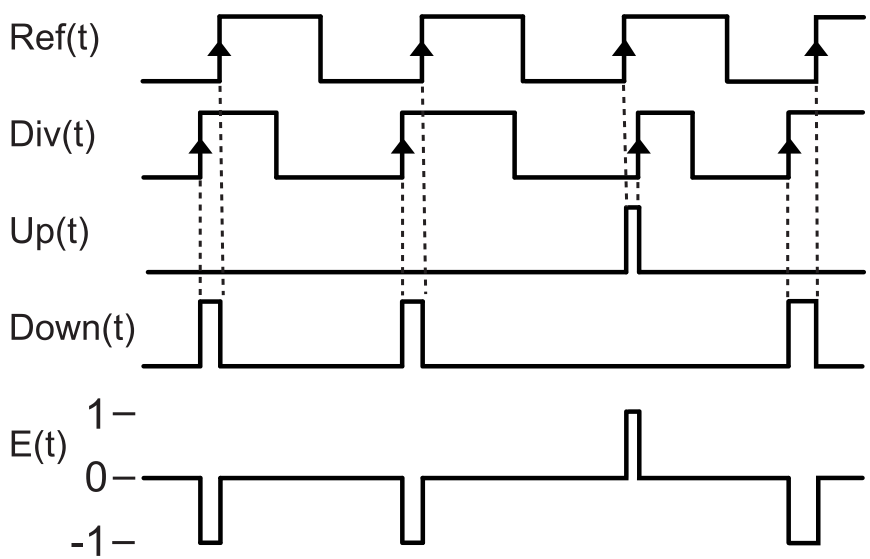


  - How do we extend the frequency lock-in range?

Phase Frequency Detectors (PFD)

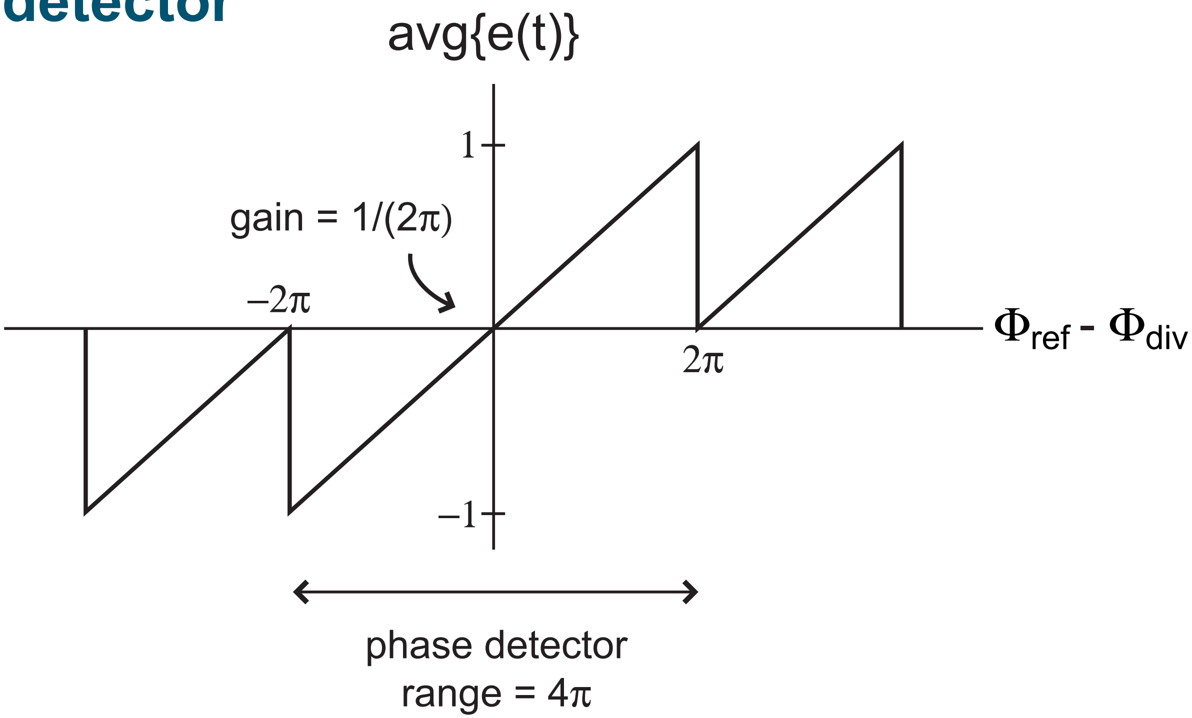
- **Example: Tristate PFD**





Tristate PFD Characteristic

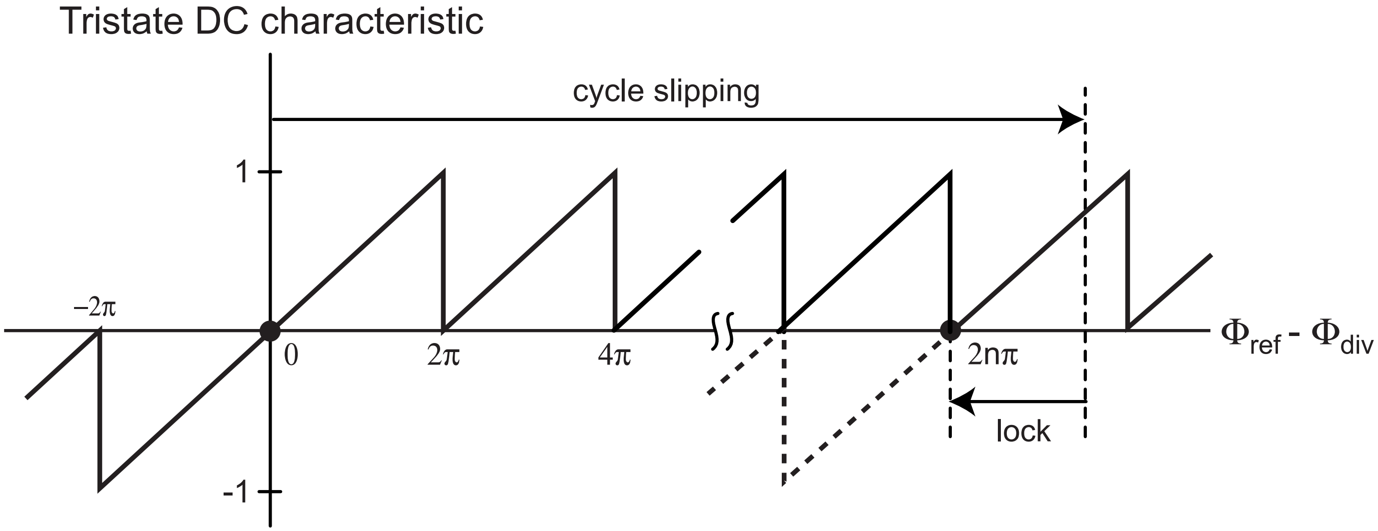
- **Calculate using similar approach as used for XOR phase detector**



- **Note that phase error characteristic is asymmetric about zero phase**
  - **Key attribute for enabling frequency detection**

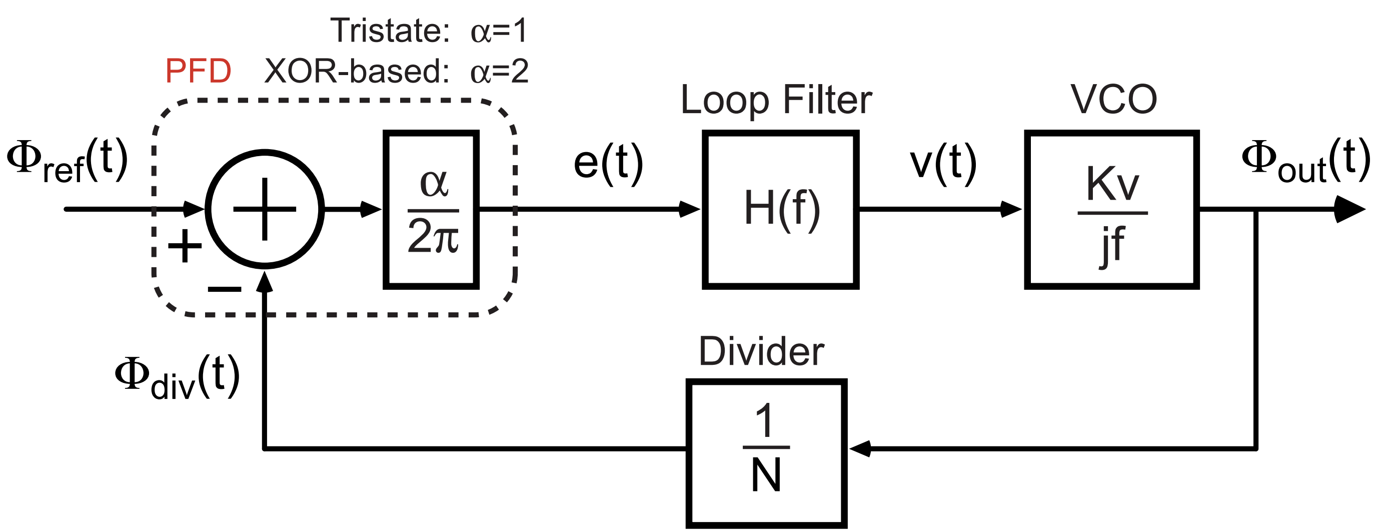
PFD Enables PLL to Always Regain Frequency Lock

- **Asymmetric phase error characteristic allows positive frequency differences to be distinguished from negative frequency differences**
  - **Average value is now positive or negative according to sign of frequency offset**
  - **PLL will always relock**



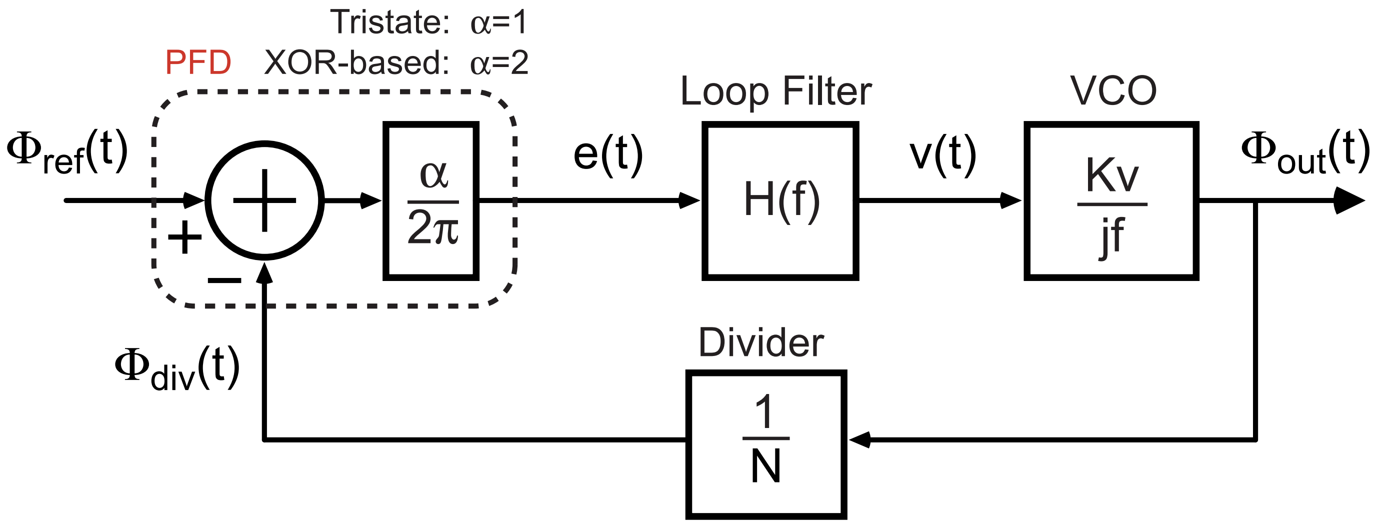
Linearized PLL Model With PFD Structures

- **Assume that when PLL in lock, phase variations are within the linear range of PFD**
  - **Simulate impact of cycle slipping if desired (do not include its effect in model)**
- **Same frequency-domain PLL model as before, but PFD gain depends on topology used**



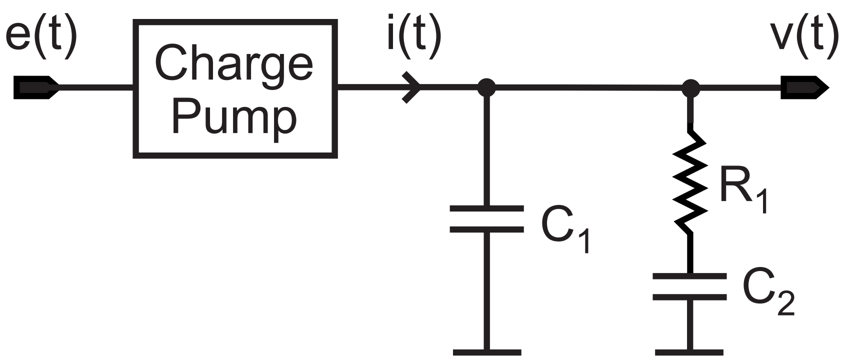
Type I versus Type II PLL Implementations

- 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



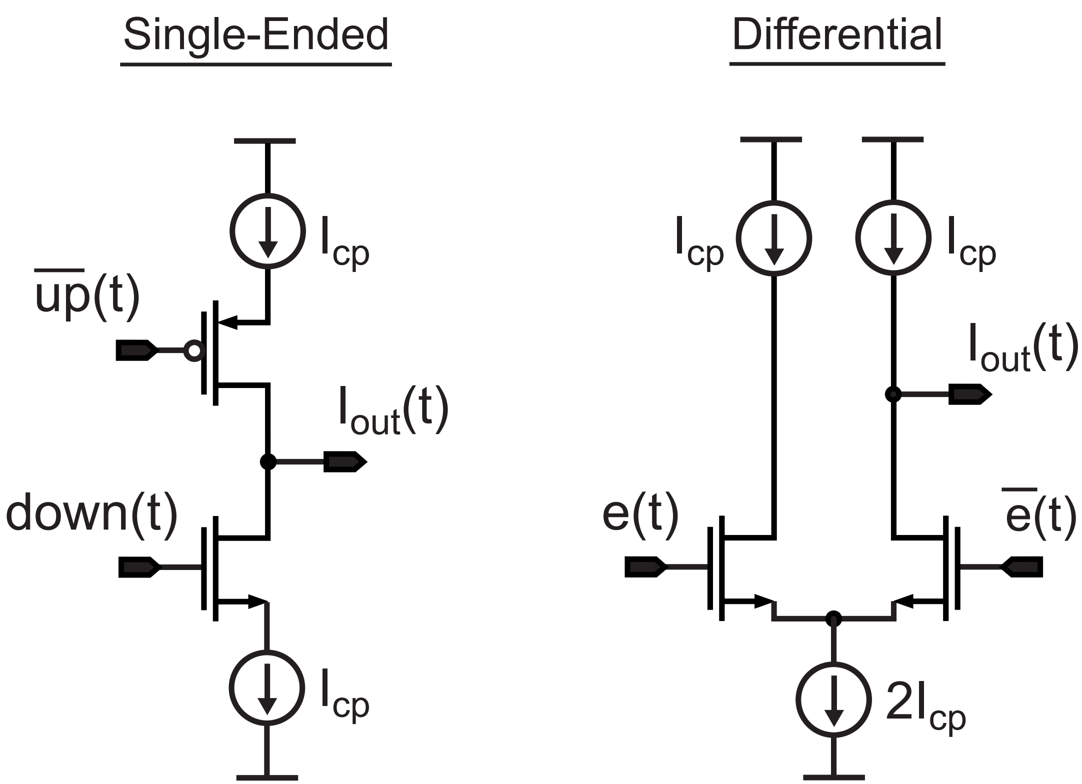
A Common Loop Filter for Type II PLL Implementation

- **Use a charge pump to create the integrator**
  - **Current onto a capacitor forms integrator**
  - **Add extra pole/zero using resistor and capacitor**
- **Gain of loop filter can be adjusted according to the value of the charge pump current**
- **Example: lead/lag network**



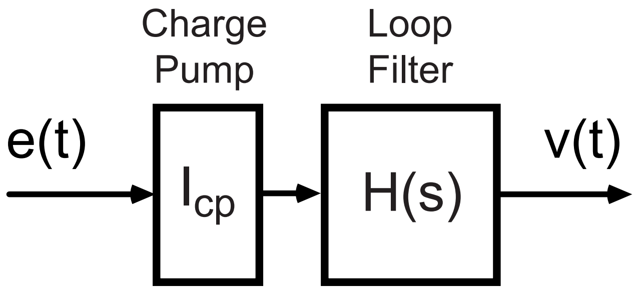
Charge Pump Implementations

- **Switch currents in and out:**



Modeling of Loop Filter/Charge Pump

- Charge pump is gain element
- Loop filter forms transfer function



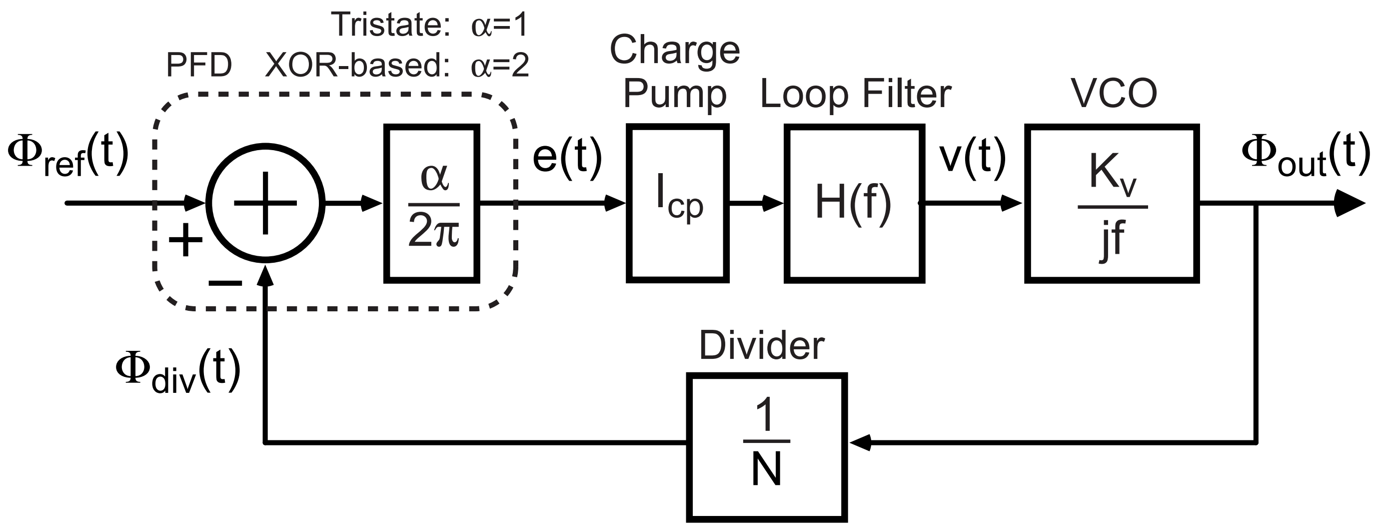
- Example: lead/lag network from previous slide

$$H(f) = \left(\frac{1}{sC_{sum}}\right)\frac{1+jf/f_z}{1+jf/f_p}$$

$$C_{sum} = C_1 + C_2, \quad f_z = \frac{1}{2\pi R_1 C_2}, \quad f_p = \frac{C_1 + C_2}{2\pi R_1 C_1 C_2}$$

PLL Design with Lead/Lag Filter

- **Overall PLL block diagram**



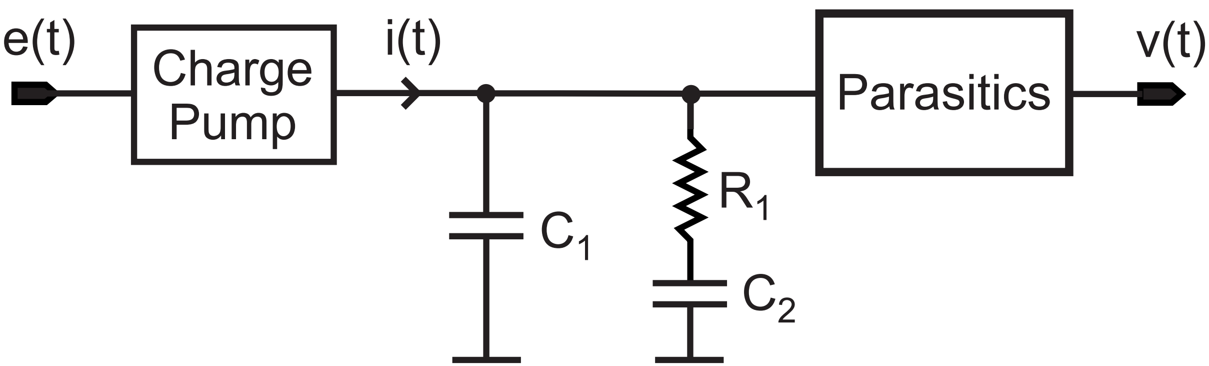
- **Loop filter**

$$H(f) = \left(\frac{1}{sC_{sum}}\right)\frac{1 + jf/f_z}{1 + jf/f_p}$$

- **Set open loop gain to achieve adequate phase margin**
  - Set f_z lower than and f_p higher than desired PLL bandwidth

Impact of Parasitics When Lead/Lag Filter Used

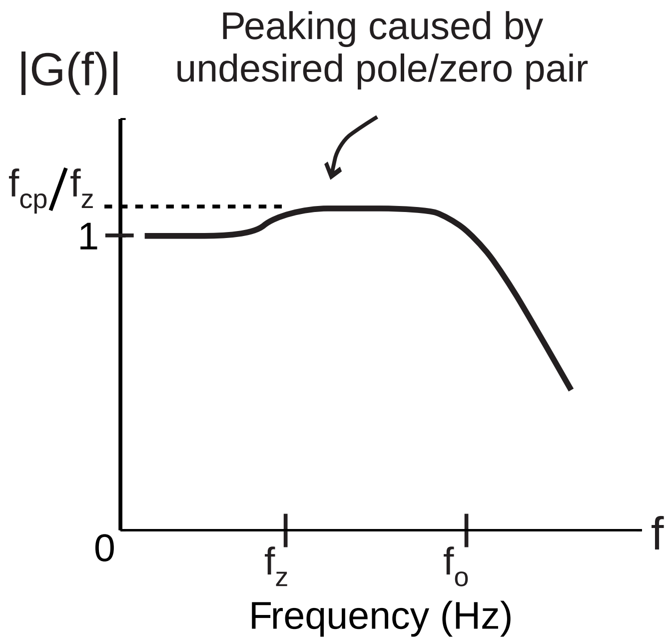
- **We can again model impact of parasitics by including them in loop filter transfer function**

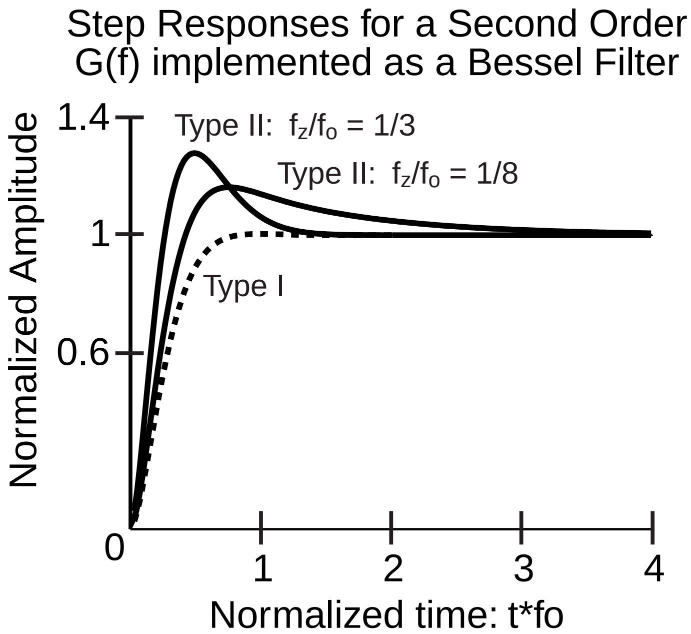


- **Example: include two parasitic poles with the lead/lag transfer function**

$$H(f) = \left(\frac{1}{sC_{sum}}\right)\frac{1 + jf/f_z}{1 + jf/f_p}\left(\frac{1}{1 + jf/f_{p2}}\right)\left(\frac{1}{1 + jf/f_{p3}}\right)$$

Negative Issues For Type II PLL Implementations





- **Parasitic pole/zero pair causes**
  - **Peaking in the closed loop frequency response**
    - A big issue for CDR systems, but not too bad for wireless
  - **Extended settling time due to parasitic "tail" response**
    - Bad for wireless systems demanding fast settling time