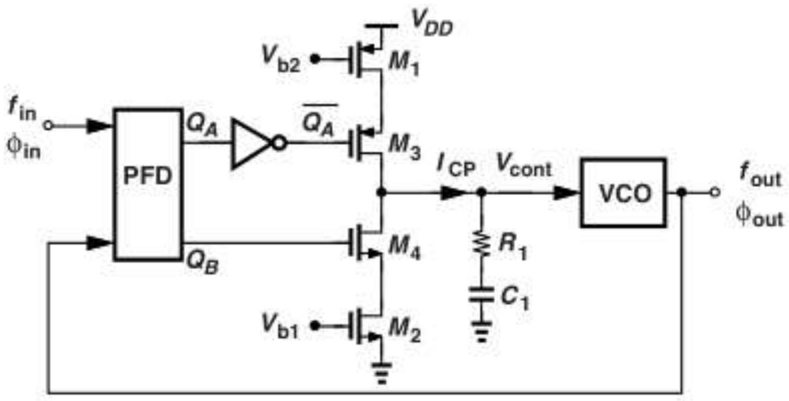
Conventional Phase – Locked Loop Design

Object

Design, model, and verify a conventional charge-pump PLL, focusing on loop dynamics, stability, lock behavior, and practical design trade-offs.

PLL Structure

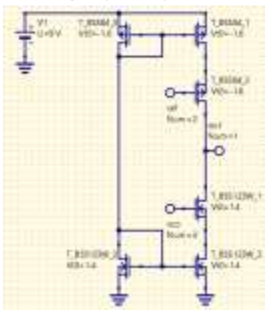


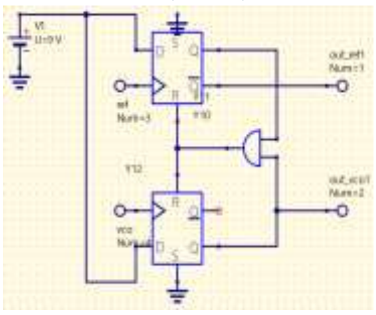
Circuit components

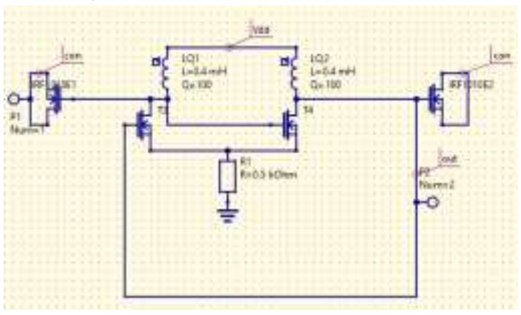
Charge pump

Phase – Frequency detector

Voltage – controlled oscillator





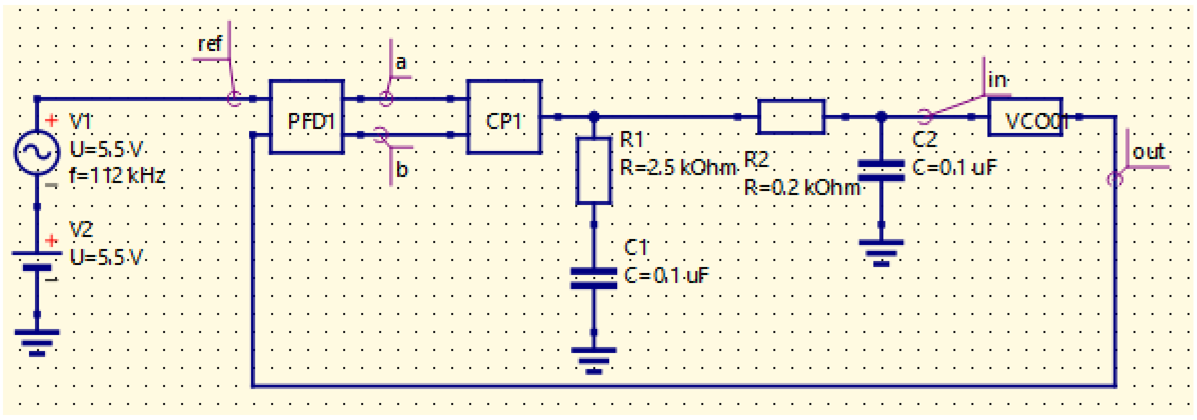


System overview

A classical PLL structure consisting of:

- Phase–Frequency Detector (PFD)
- Charge Pump (CP)
- Loop Filter
- Voltage-Controlled Oscillator (VCO)

Overall system – with charge capacitor and loop filer addition



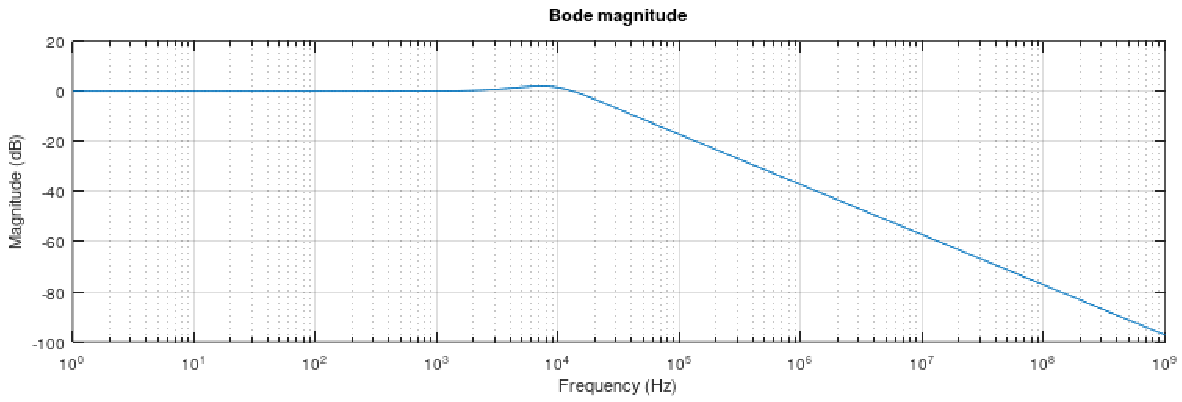
Specification

Specification	Desired value	Simulated value
Operating frequency	100 kHz	100 kHz
Damping factor	0.7	0.7329
Loop bandwidth	20 kHz	19.421 kHz
Lock time	< 1ms	0.412 ms
Phase margin	60 degree	66.83 degree
Lock range	90 kHz – 111 kHz (10%)	87 kHz – 113 kHz
Kvco	--	3 kHz/V
Charge pump current	--	72 mA

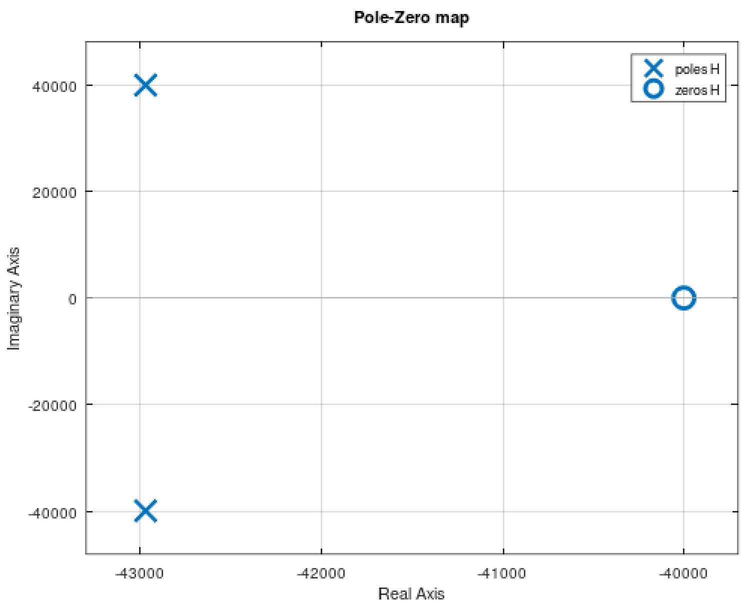
Design Methodology

1. Define PLL requirements.
2. Compute theoretical loop bandwidth, phase margin.
3. Select Charge Pump current to create enough loop gain.
4. Tune VCO gain to match lock range and linearity constraints.
5. Simulate using QUCS S + NgSpice backend for transient, AC, and parametric analysis.

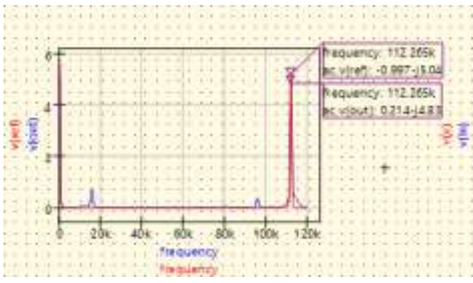
Bode plot – determine loop band width



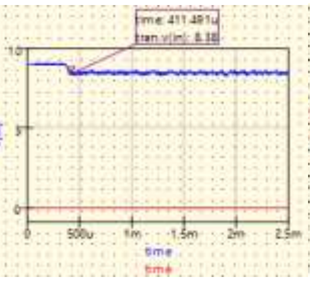
Poles – zeros map: determine damping factor



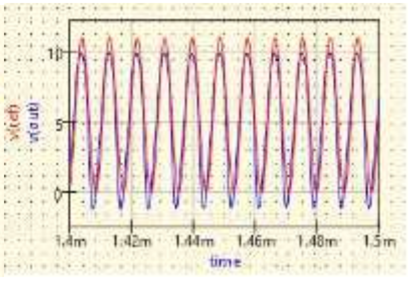
At 112 kHz: Spectrum



VCO(input signal)



Transient response



➔ Validate design by spectrum and transient response, determine the lock time by VCO input

Design trade-offs:

Loop Bandwidth vs Jitter suppression: about 0.1 – 0.2 of reference is recommended.

Phase margin vs Lock speed: 60 degrees for stable, but slower time response.

Charge pump current vs Spur level: high current for getting faster lock, but causes spurs .

Skills Demonstrated

- PLL theory: loop dynamics, stability, damping, bandwidth.
- Analog/mixed-signal modeling.
- Practical trade-off analysis.
- Simulation scripting & data verification.

Tools used

- QUCS S (NgSpice backend) — circuit modeling and simulation.
- Octave — data processing and measurement extraction.