date: 2024-03-08

TFE4188 - Lecture 8

Clocks and PLLs

Goal

Why do we need to generate clocks

Introduction to **PLLs**

Why

Digital

Radio

Energy harvesters

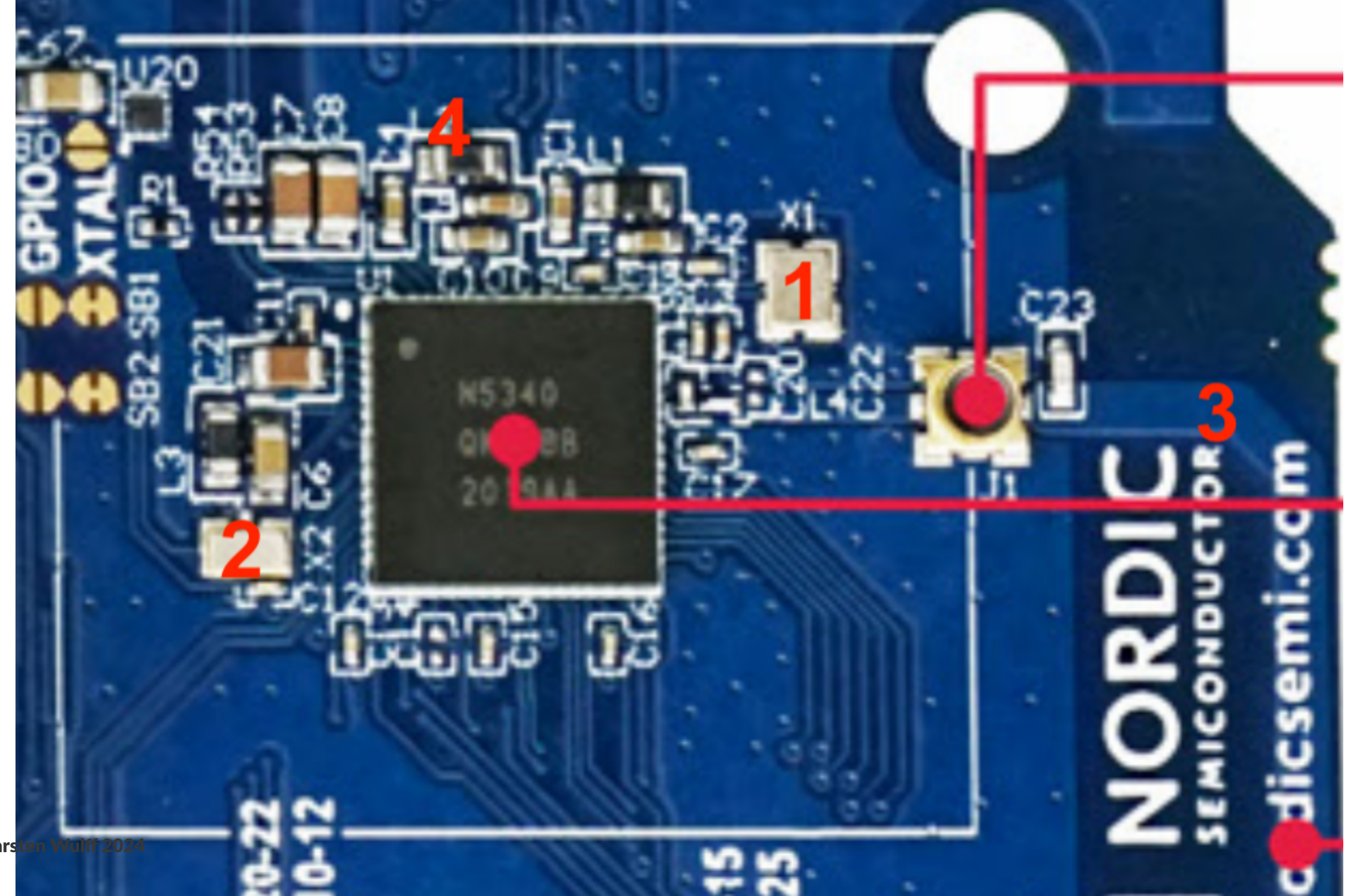
Switched regulators

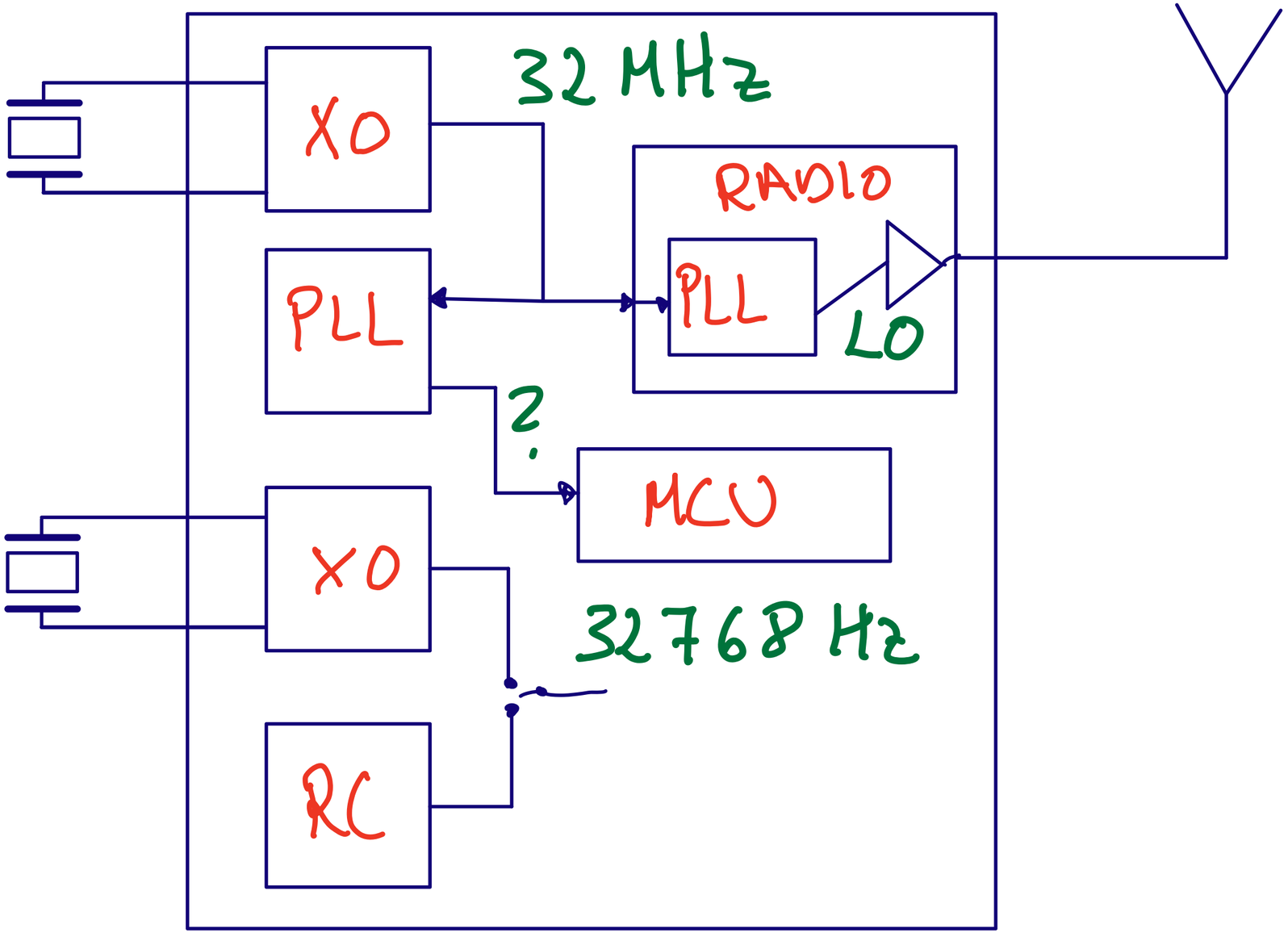
ADCs

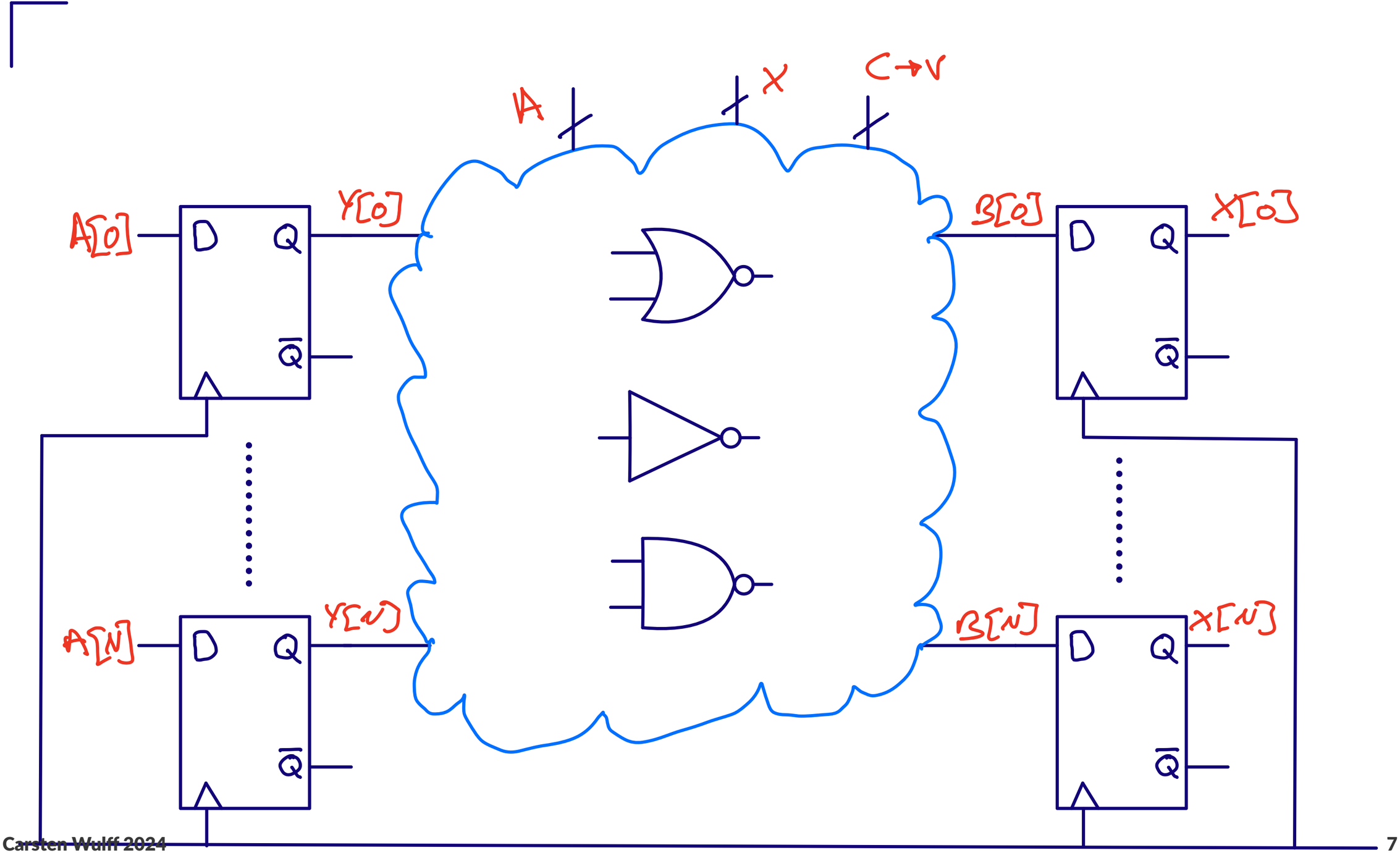
Accurate delay

SC filters

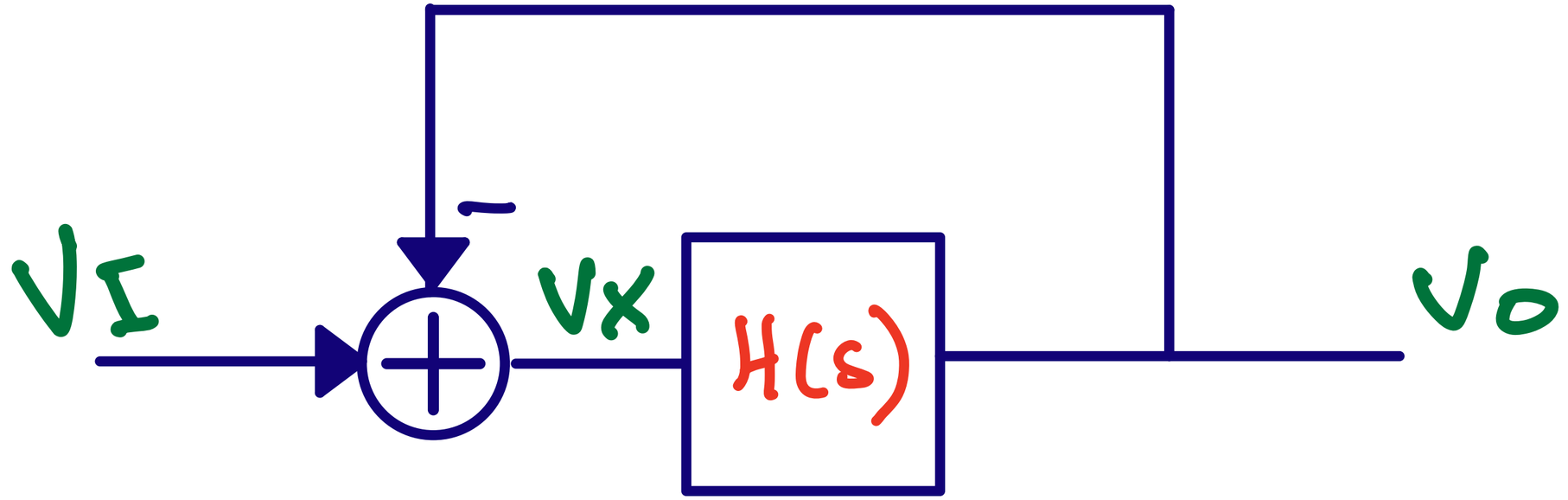
...


4
1
3
2
N5340
NORDIC
SEMICONDUCTOR
dicsemi.com


32 MHz
XO
RADIO
PLL
PLL
LO
?
MCU
XO
32768 Hz
RC

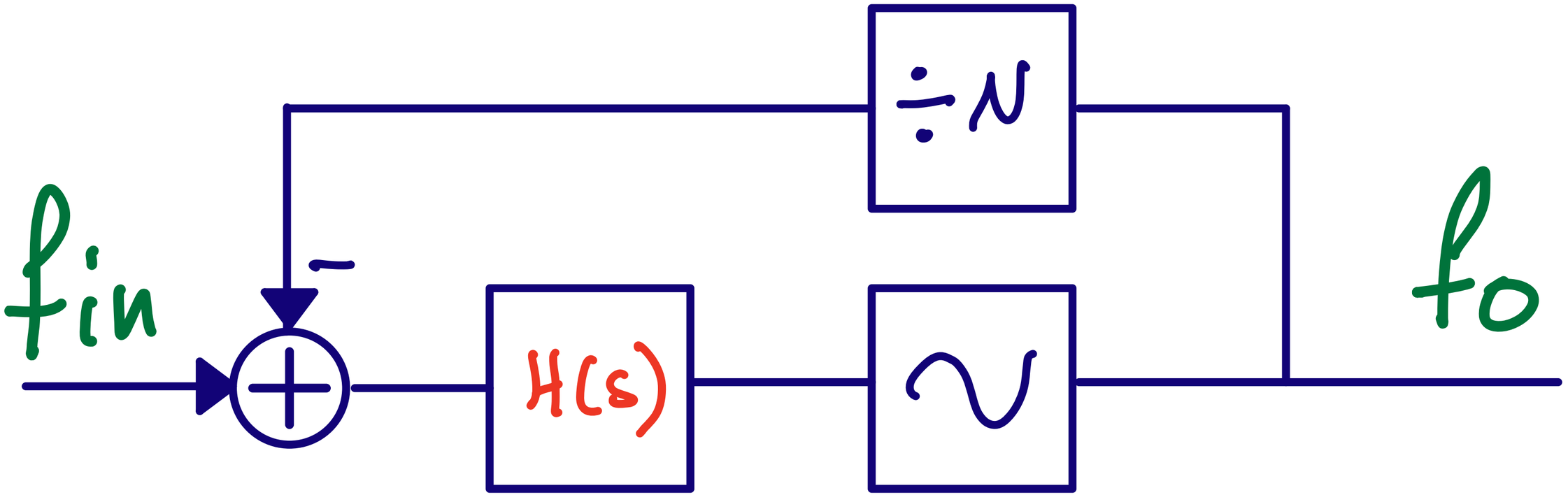

A
X
C→V
A[0]
D
Q
Q
Y[0]
B[0]
D
Q
Q
X[0]
A[N]
D
Q
Q
Y[N]
B[N]
D
Q
Q
X[N]
Clk out

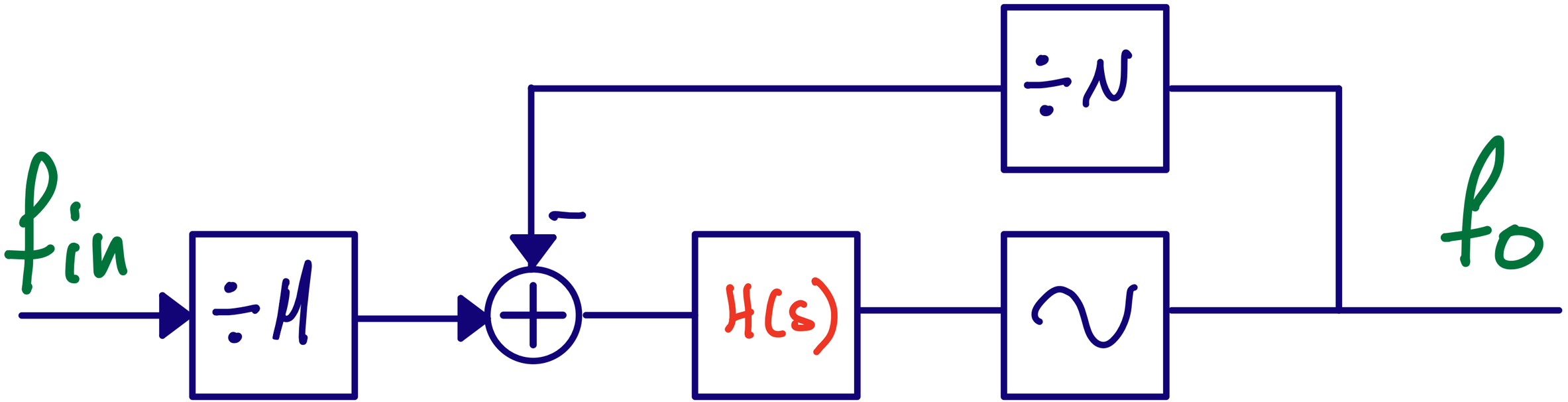

PLL

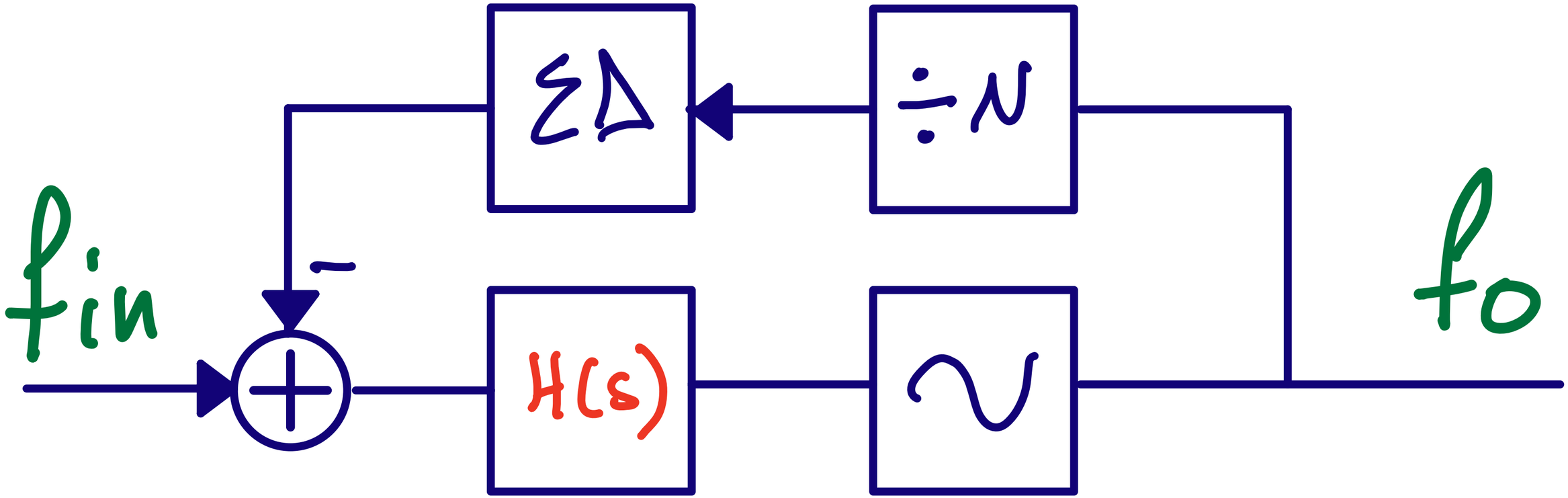


$$V_I - V_o = V_x \qquad V_o = V_x\, H(s)$$

$$V_I = V_o + \frac{V_o}{H(s)} \qquad H(s)=\infty \Rightarrow \qquad V_o = V_I$$


÷N
fin
−
H(s)
fo


fin
÷M
−
H(s)
÷N
fo


ΣΔ
÷N
fin
−
H(s)
fo

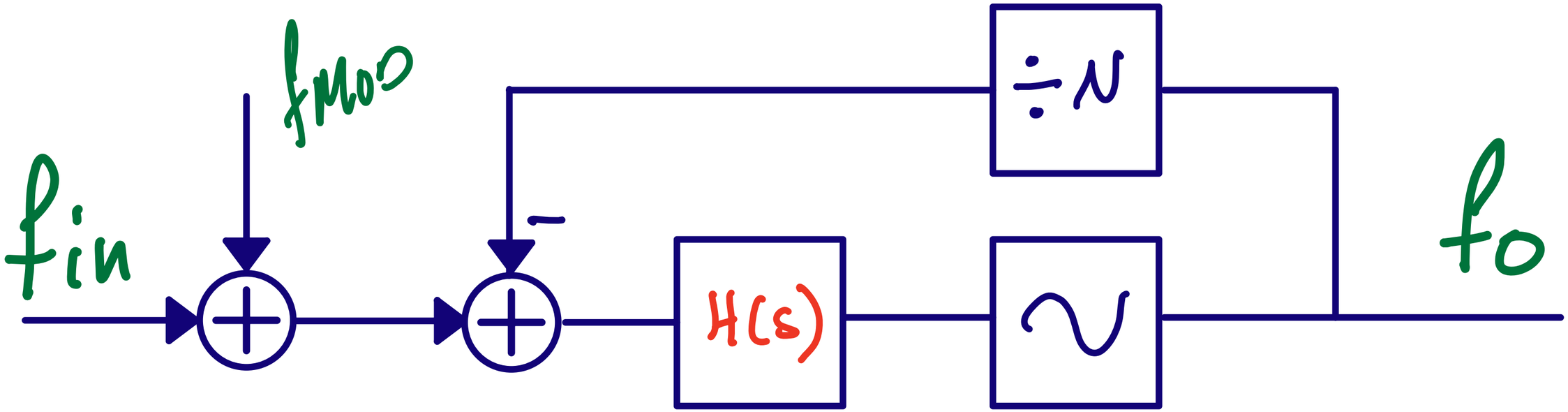
$$A_m(t) \times cos\left(2\pi f_{carrier} t + \phi_m(t)\right)$$

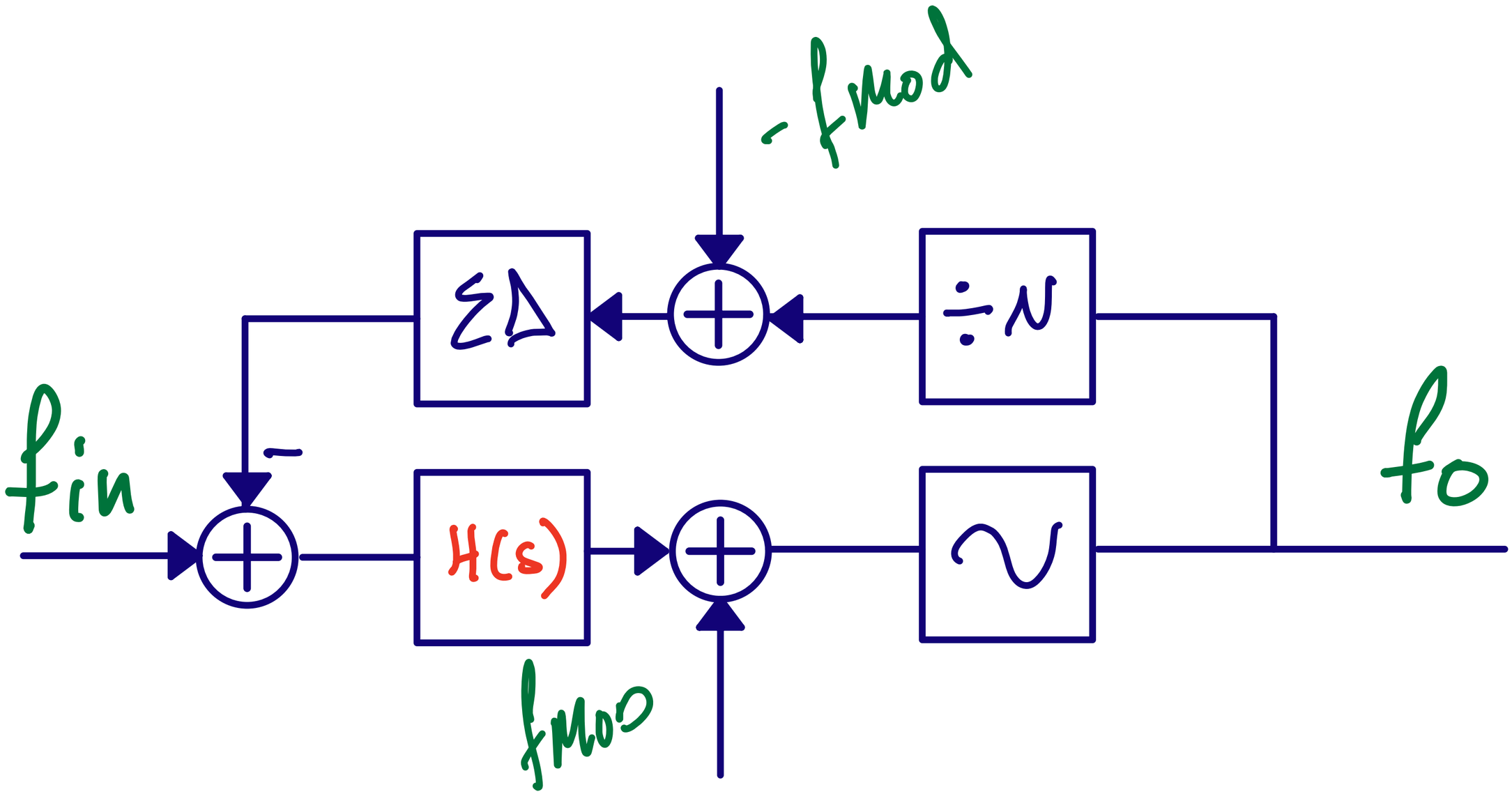
<!--pan_doc:

The Am *is the amplitude modulation, while the* $\phim$ is the phase modulation. Bluetooth Low Energy is constant envelope, so the A_m is a constant. The phase modulation is applied to the carrier, but how is it done?

One option is shown below. We could modulate our frequency reference directly. That could maybe be a sigma-delta divider on the reference, or directly modulating the oscillator.

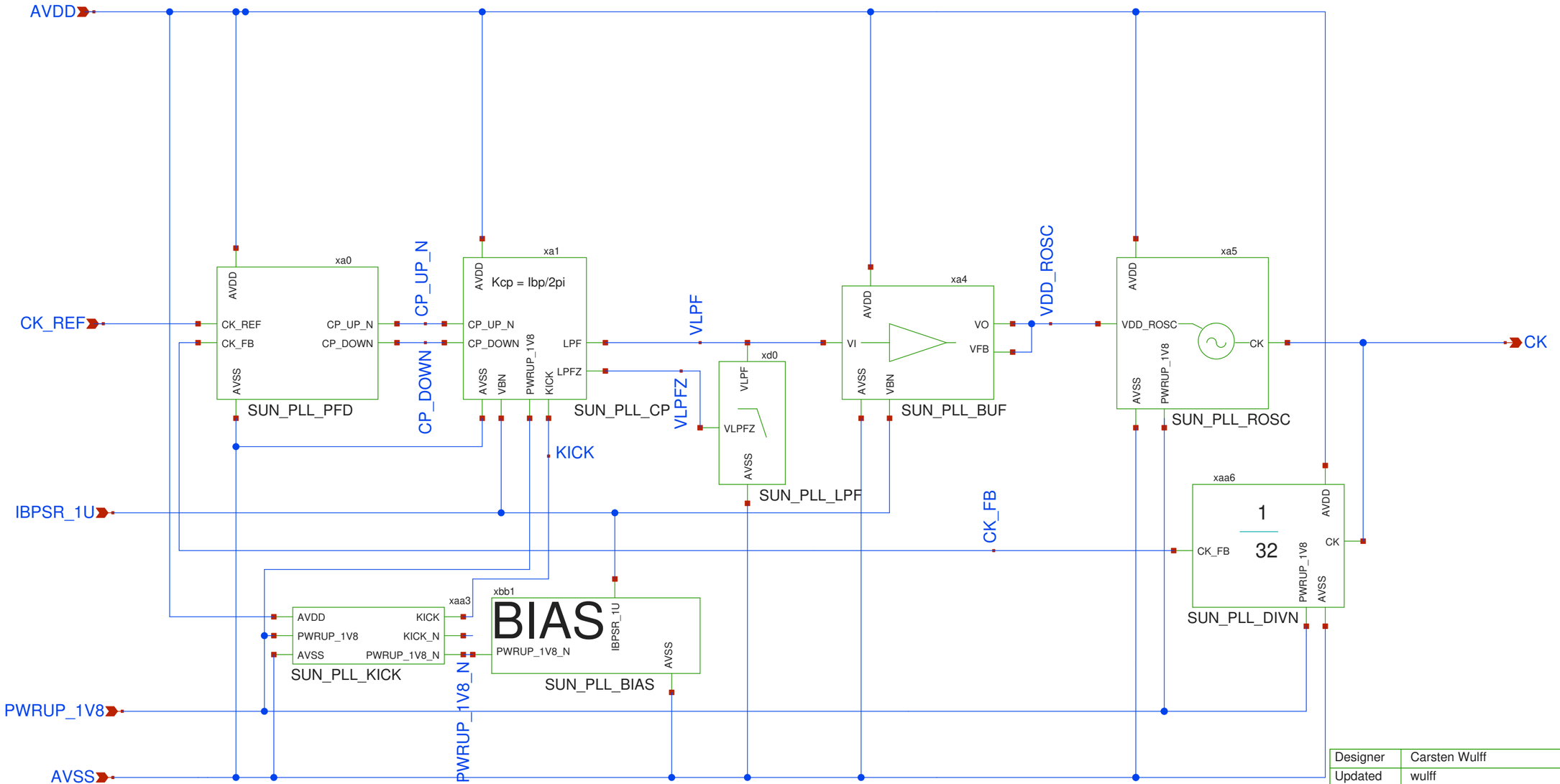
-->_


fmod
fin
−
÷N
H(s)
fo


-fmod
ΣΔ
÷N
fin
−
H(s)
fo
fmod

PLL Example

CK_REF x 32 PLL (max 512 MHz)



Carsten Wulff 2024

Designer	Carsten Wulff
Updated	wulff
Modified	2024-02-29 14:25:41
Copyright	Carsten Wulff Software

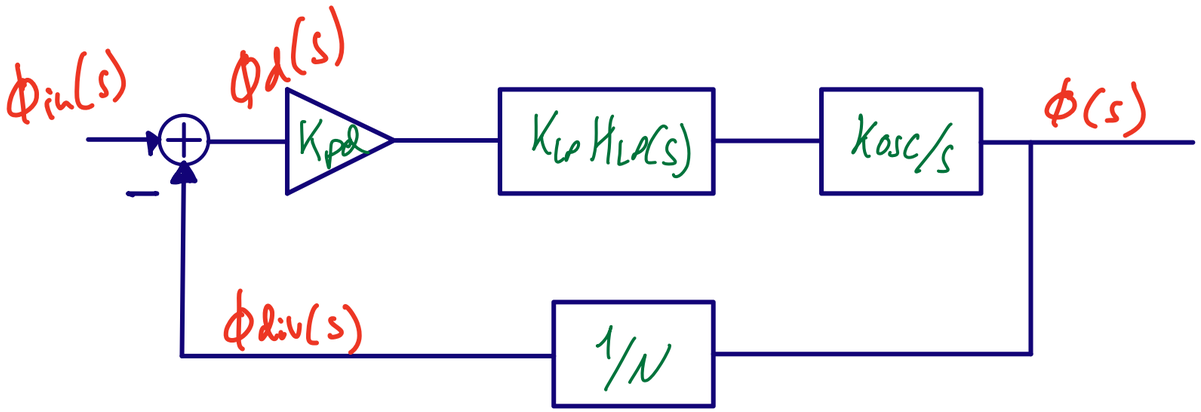
Library/Cell	SUN_PLL

PLLs need calculation!

#noCowboyDesign

$$\phi(t) = 2\pi \int_0^t f(t)dt$$

Loop gain

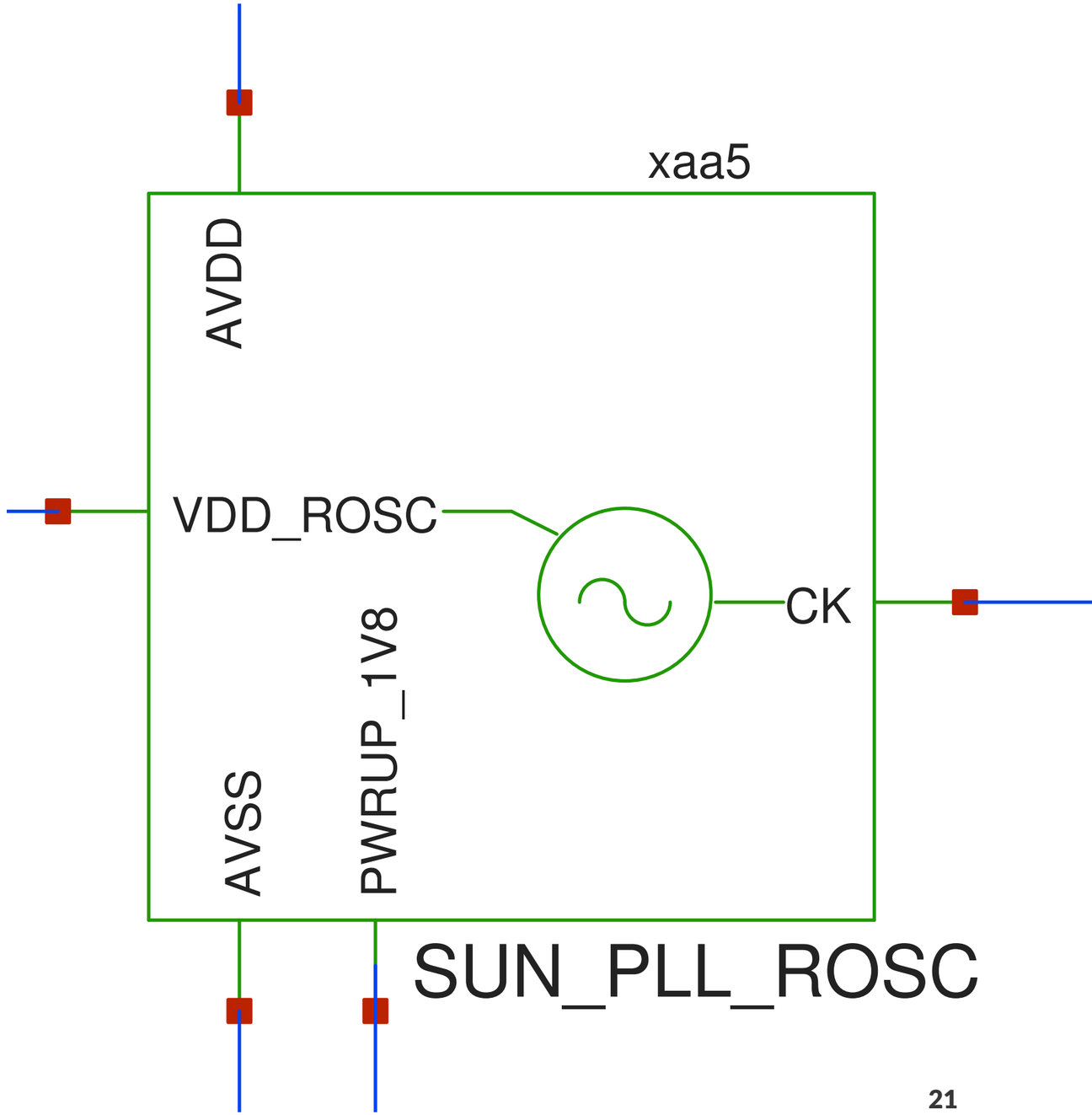


$$\frac{\phi_d}{\phi_{in}} = \frac{1}{1+L(s)}$$

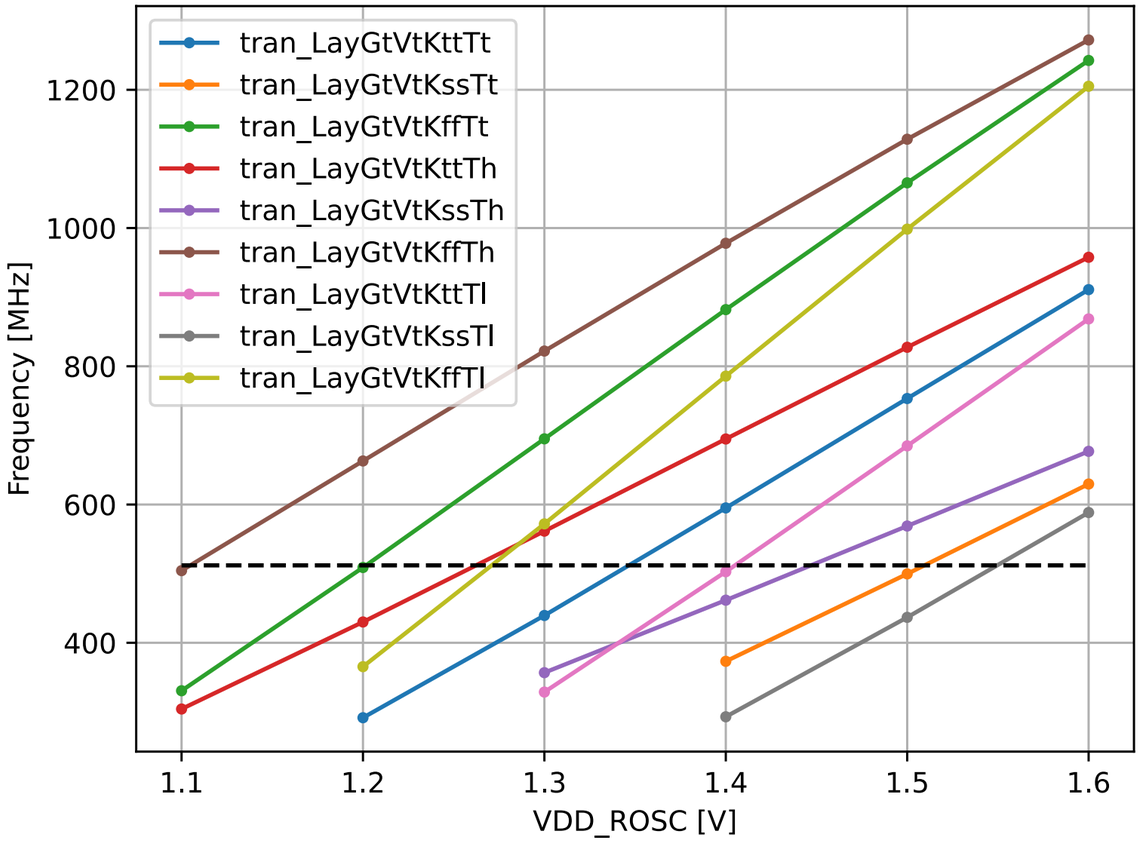
$$L(s) = \frac{K_{osc}K_{pd}K_{lp}H_{lp}(s)}{Ns}$$

Controlled oscillator

$$K_{osc} = 2\pi \frac{df}{dV_{cntl}}$$

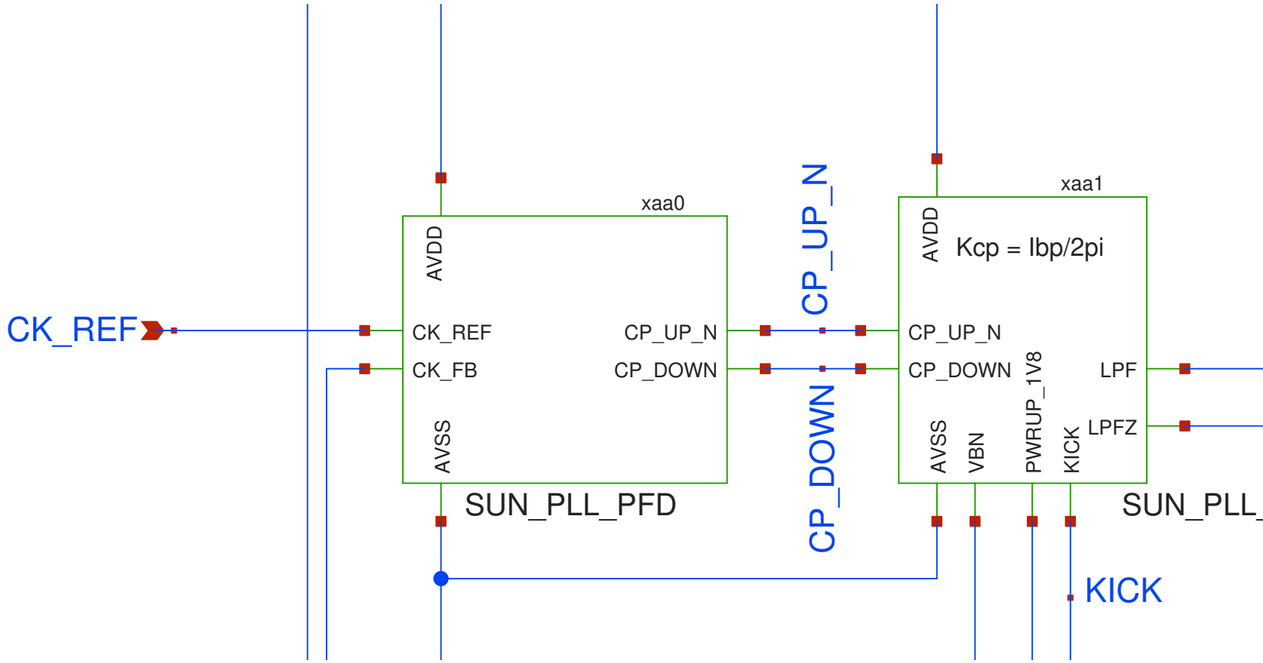


SUN_PLL_SKY130NM/sim/ROSC/



Phase detector and charge pump

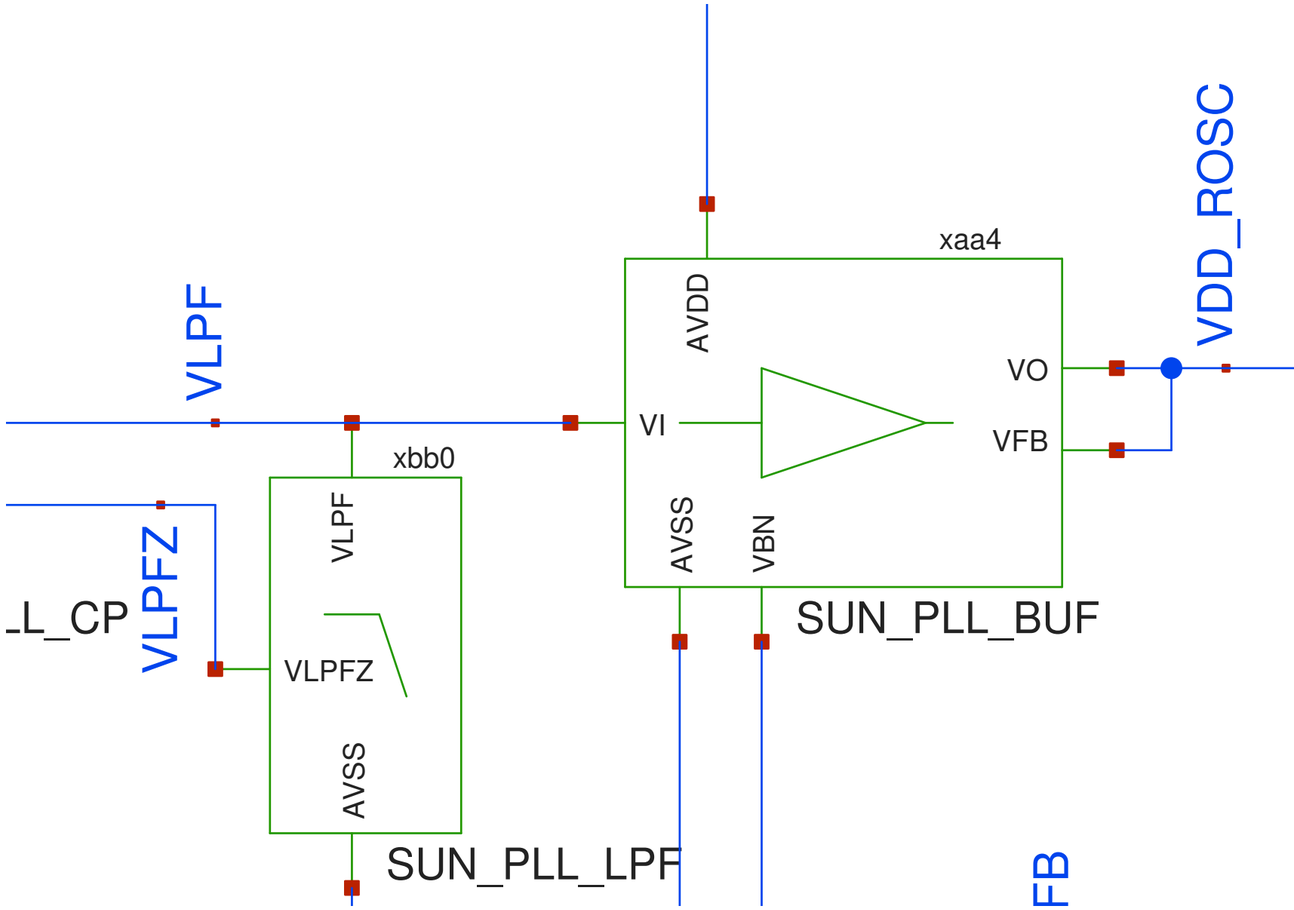
$$K_{pd} = \frac{I_{cp}}{2\pi}$$



Loop filter

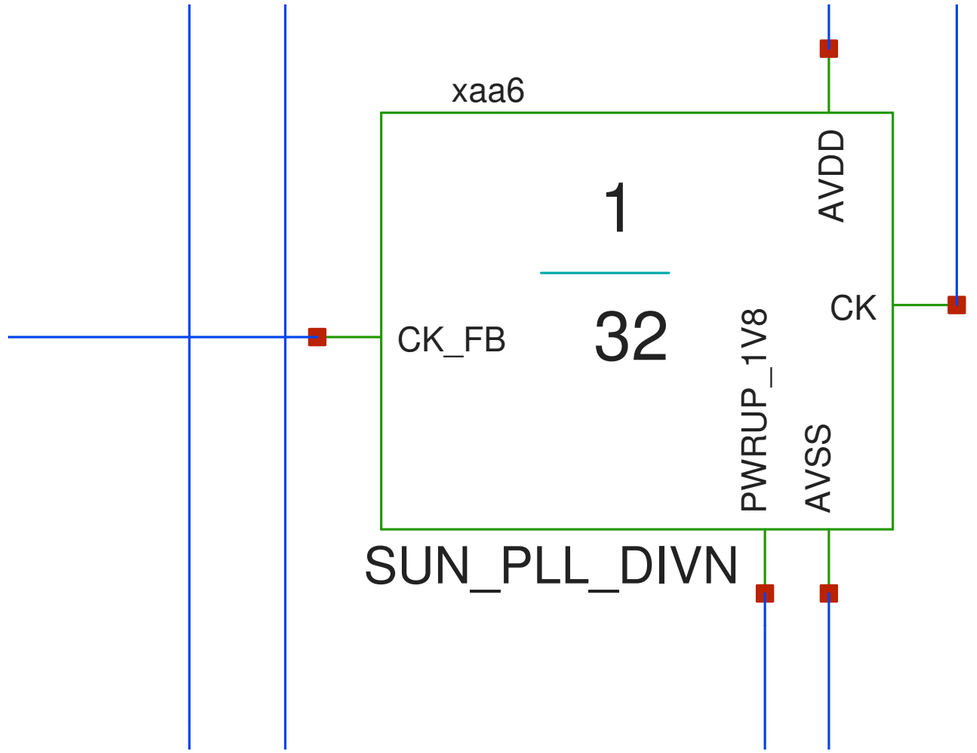
$$K_{lp}H_{lp}(s) = K_{lp}\left(\frac{1}{s} + \frac{1}{\omega_z}\right)$$

$$K_{lp}H_{lp}(s) = \frac{1}{s(C_1 + C_2)} \frac{1 + sRC_1}{1 + sR\frac{C_1C_2}{C_1+C_2}}$$



Divider

$$K_{div} = \frac{1}{N}$$



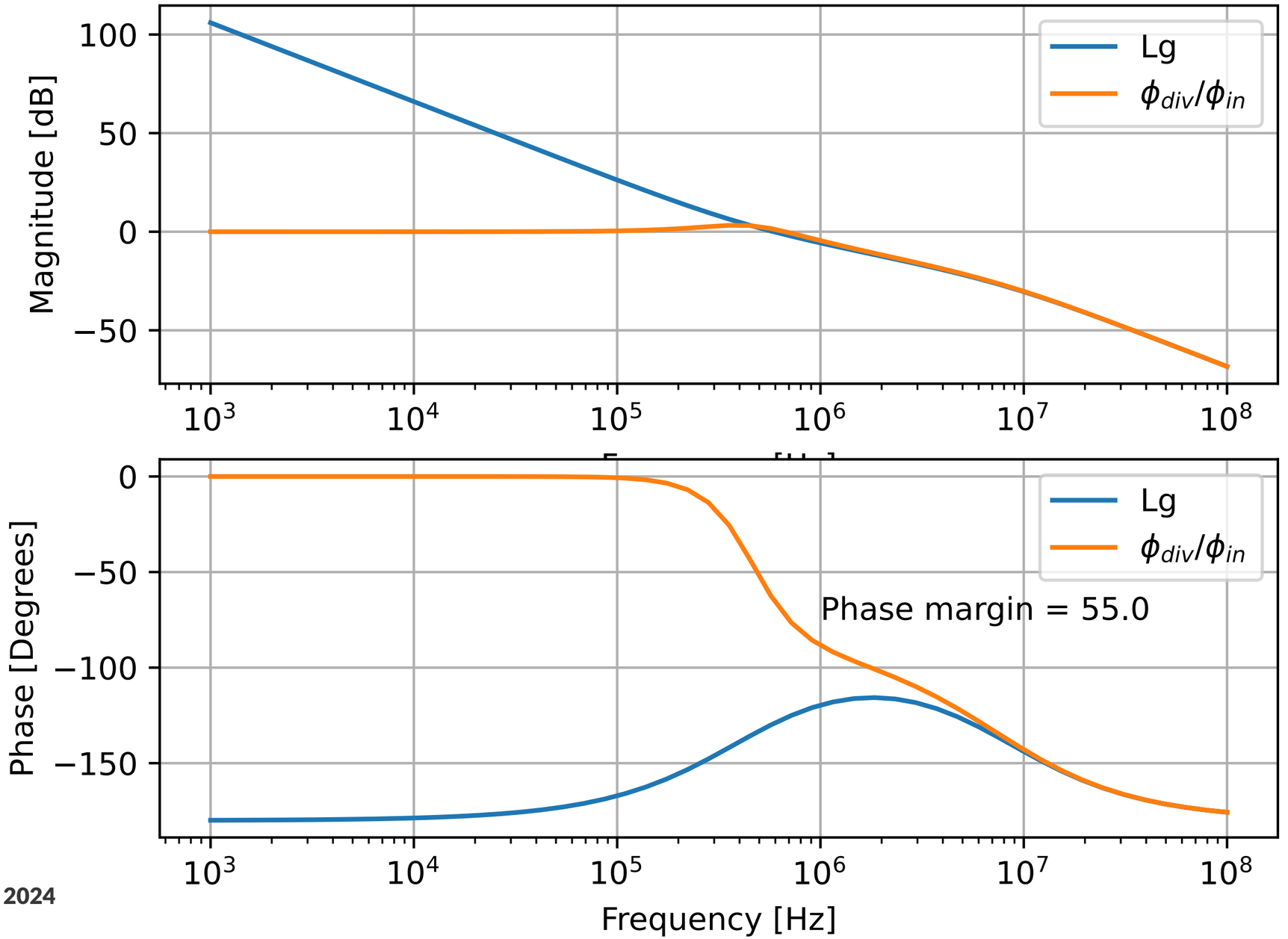
Loop transfer function

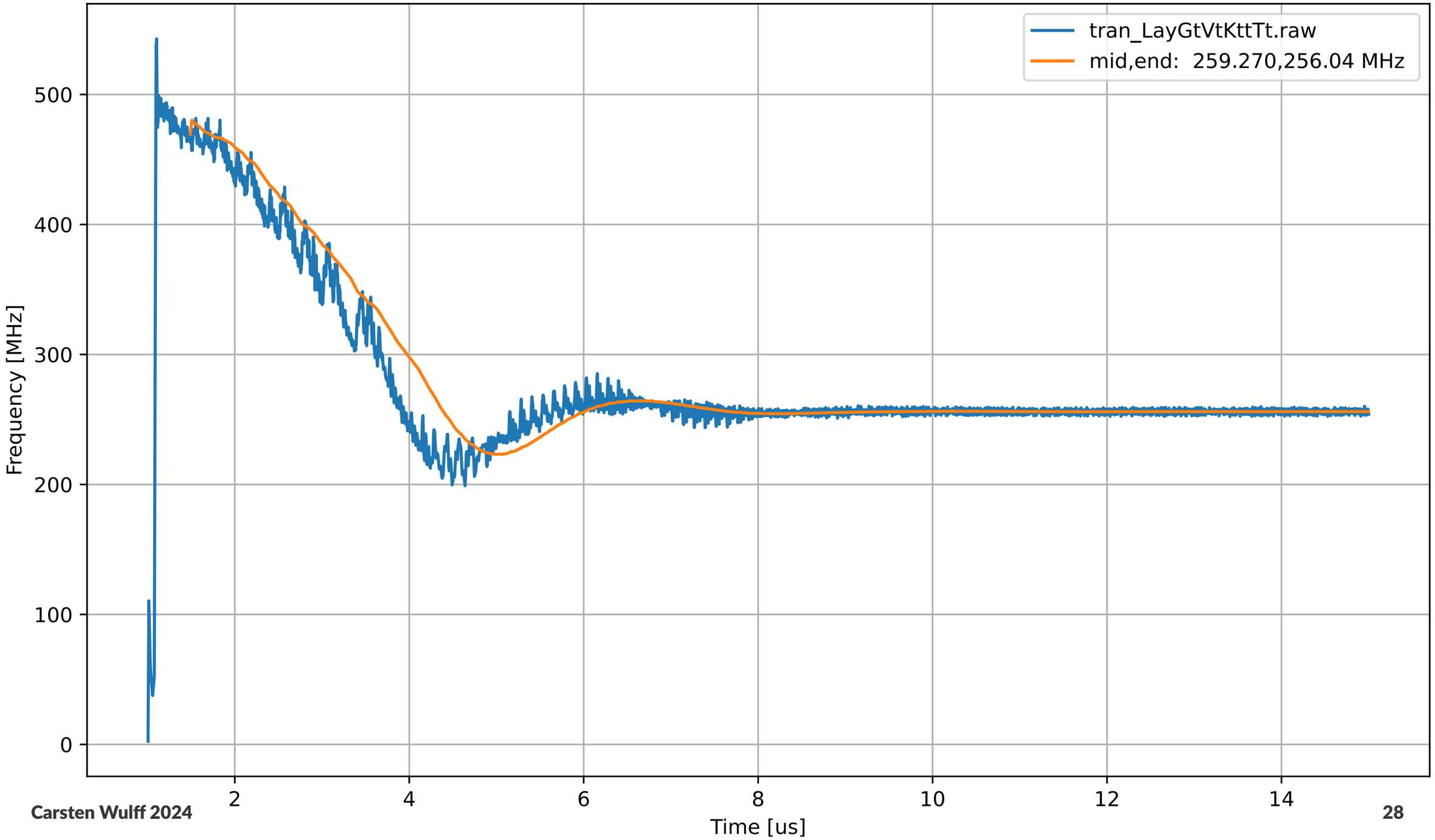
$$L(s) = \frac{K_{osc} K_{pd} K_{lp} H_{lp}(s)}{Ns}$$

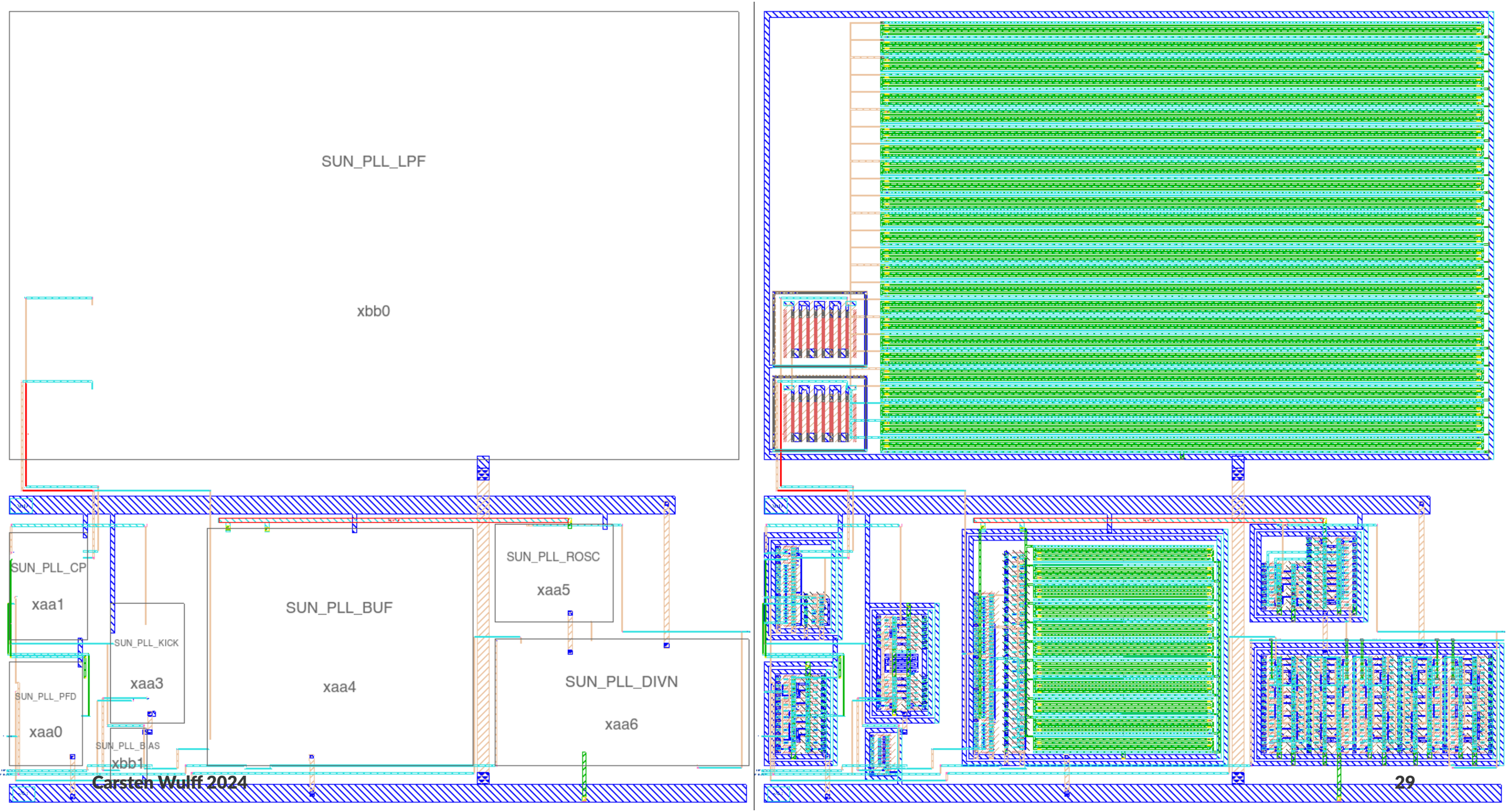
Python model

sun_pll_sky130nm/jupyter/pll

sun_pll_sky130nm/jupyter/pfd


Magnitude [dB]
100
50
0
−50
10^3
10^4
10^5
10^6
10^7
10^8
Lg
ϕ_{div}/ϕ_{in}
Phase [Degrees]
0
−50
−100
−150
Phase margin = 55.0
Frequency [Hz]


tran_LayGtVtKttTt.raw
mid,end: 259.270,256.04 MHz
Frequency [MHz]
0
100
200
300
400
500
2
4
6
8
10
12
14
Time [us]


SUN_PLL_LPF
xbb0
SUN_PLL_CP
xaa1
SUN_PLL_KICK
xaa3
SUN_PLL_PFD
xaa0
SUN_PLL_BIAS
xbb1
SUN_PLL_BUF
xaa4
SUN_PLL_ROSC
xaa5
SUN_PLL_DIVN
xaa6

SUN_PLL_SKY130NM

Thanks!