date: 2024-02-09

TFE4188 - Lecture 5

Switched-Capacitor Circuits

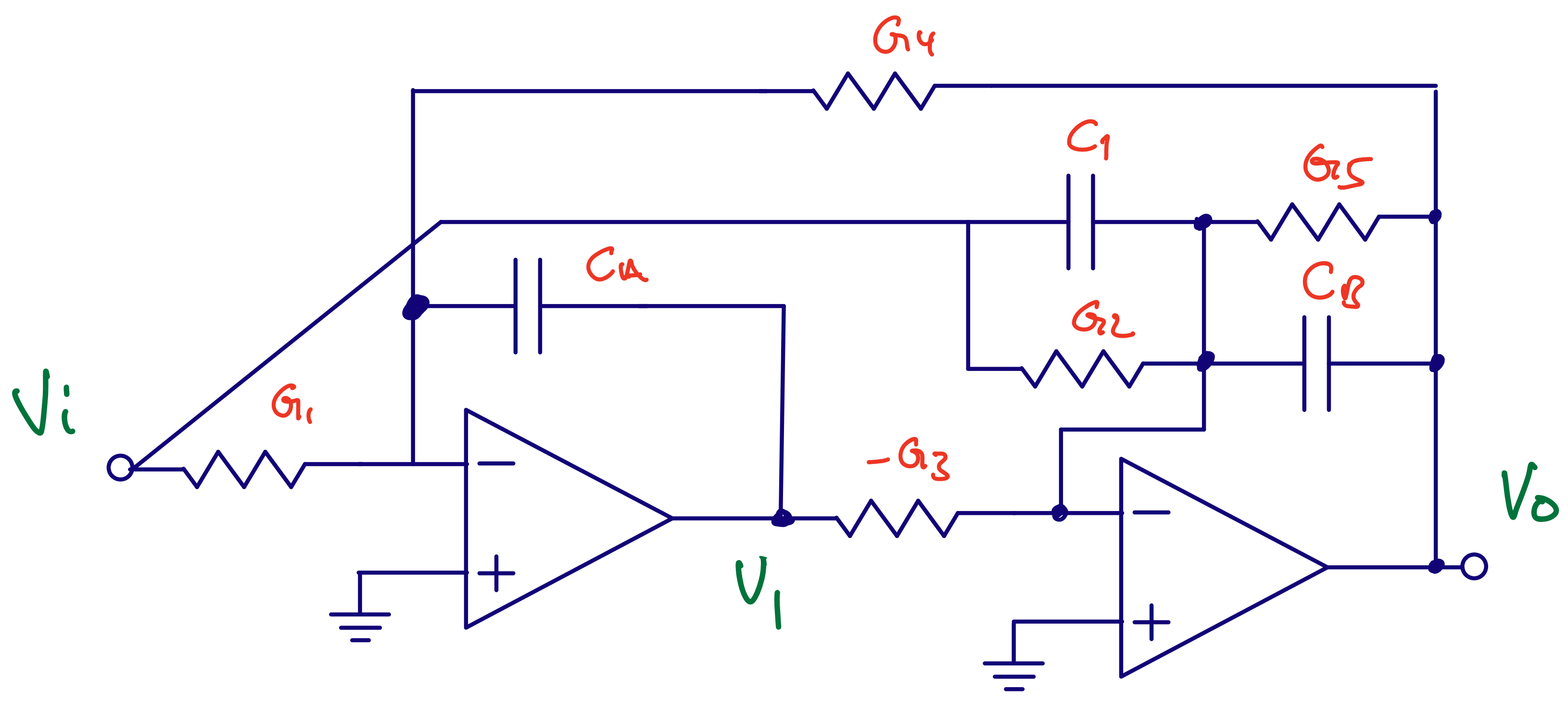
Goal

Understand **why** we would use switched capacitor circuits

Introduction to **discrete-time**, and **switched capacitor principles** and the **circuits** we need

Why

Active-RC



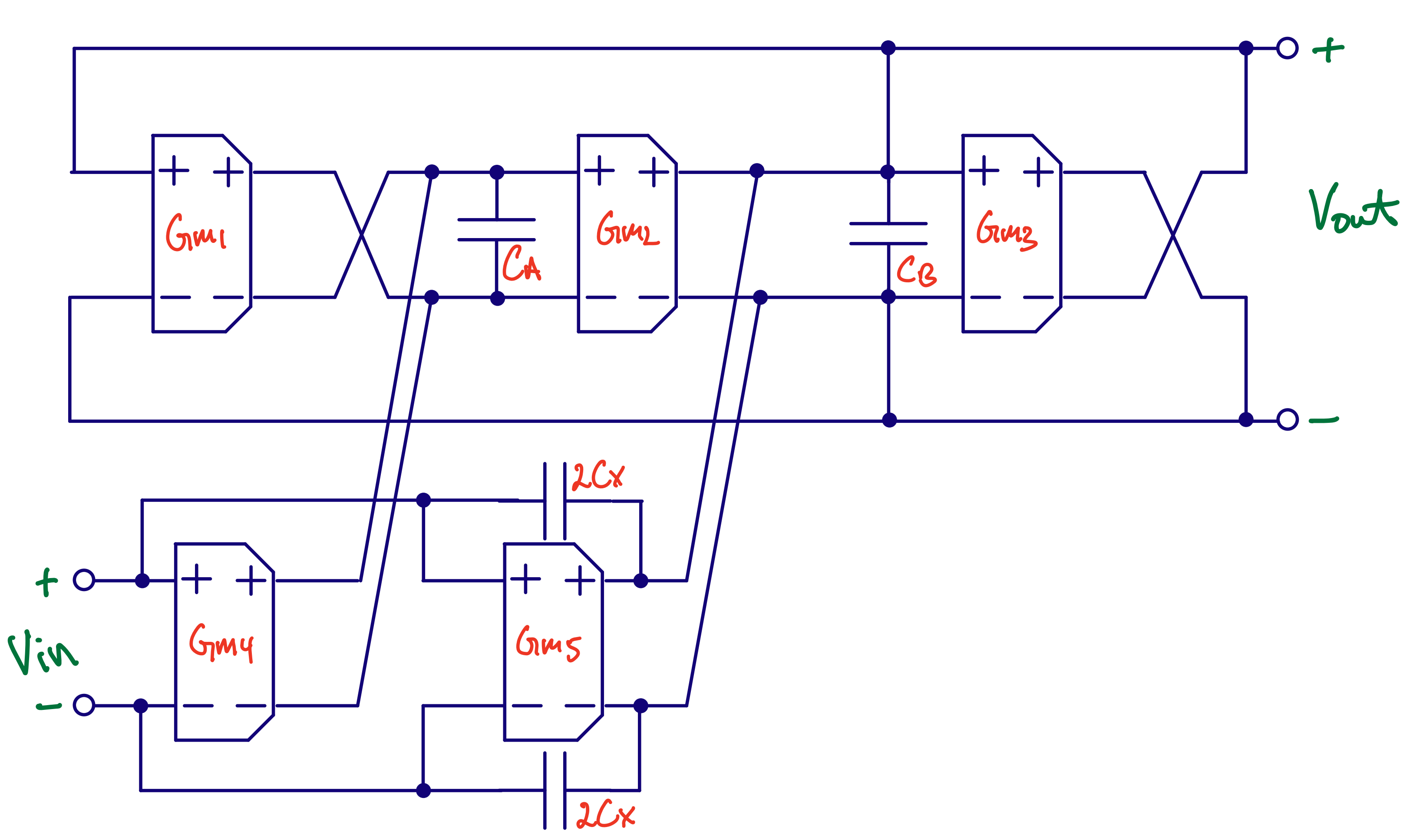
$$H(s) = \frac{\left[\frac{C_1}{C_B}s^2 + \frac{G_2}{C_B}s + \left(\frac{G_1 G_3}{C_A C_B}\right)\right]}{\left[s^2 + \frac{G_5}{C_B}s + \frac{G_3 G_4}{C_A C_B}\right]}$$

$$\omega_{p|z} \propto \frac{G}{C} = \frac{1}{RC}$$

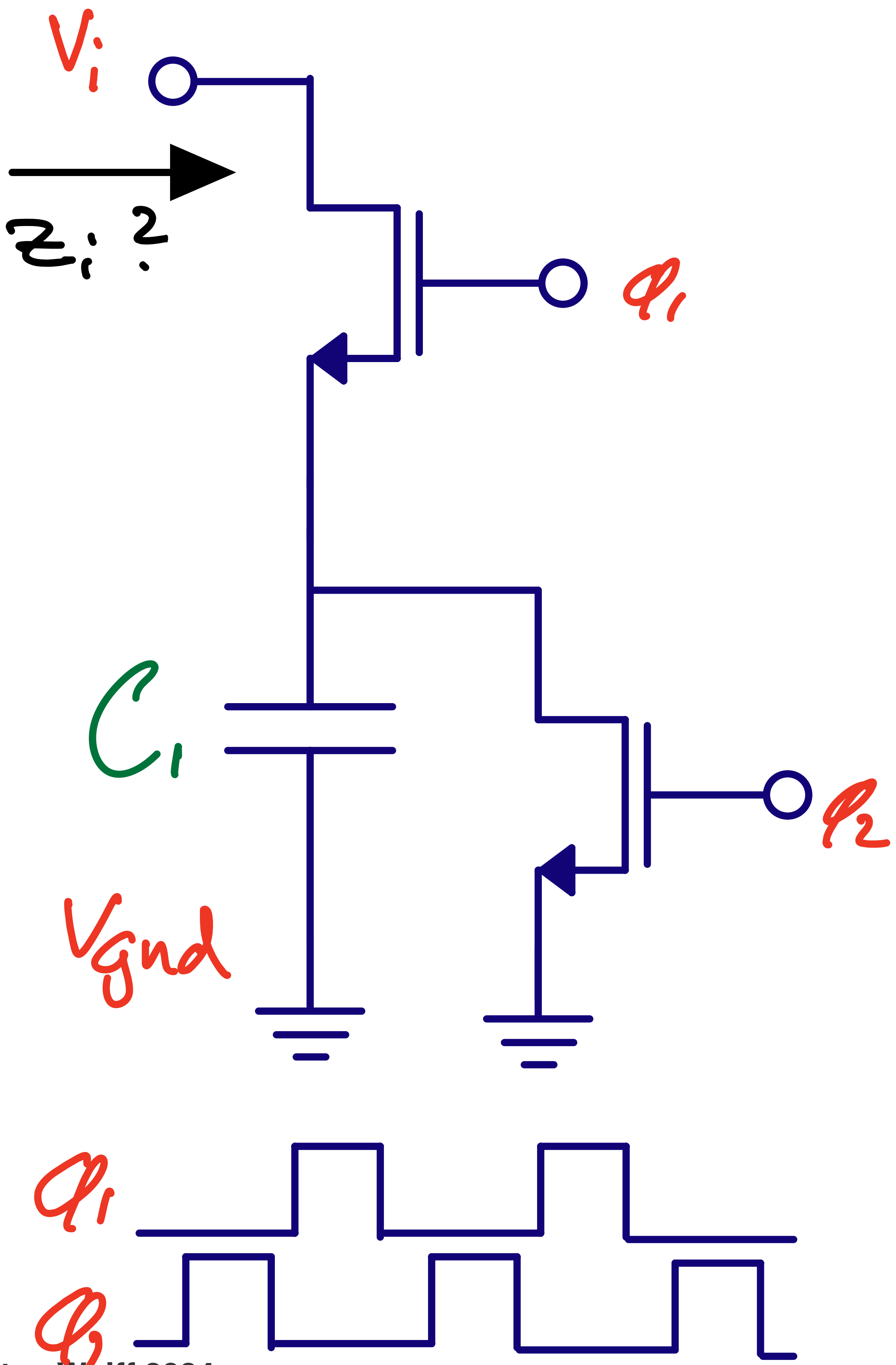
Gm-C

$$H(s) = \frac{\left[s^2 \frac{C_X}{C_X+C_B} + s\frac{G_{m5}}{C_X+C_B} + \frac{G_{m2}G_{m4}}{C_A(C_X+C_B)}\right]}{\left[s^2 + s\frac{G_{m2}}{C_X+C_B} + \frac{G_{m1}G_{m2}}{C_A(C_X+C_B)}\right]}$$

$$\omega_{p|z} \propto \frac{G_m}{C}$$



Switched capacitor



$$Q_{\phi 2\$} = C_1 V_{GND} = 0$$

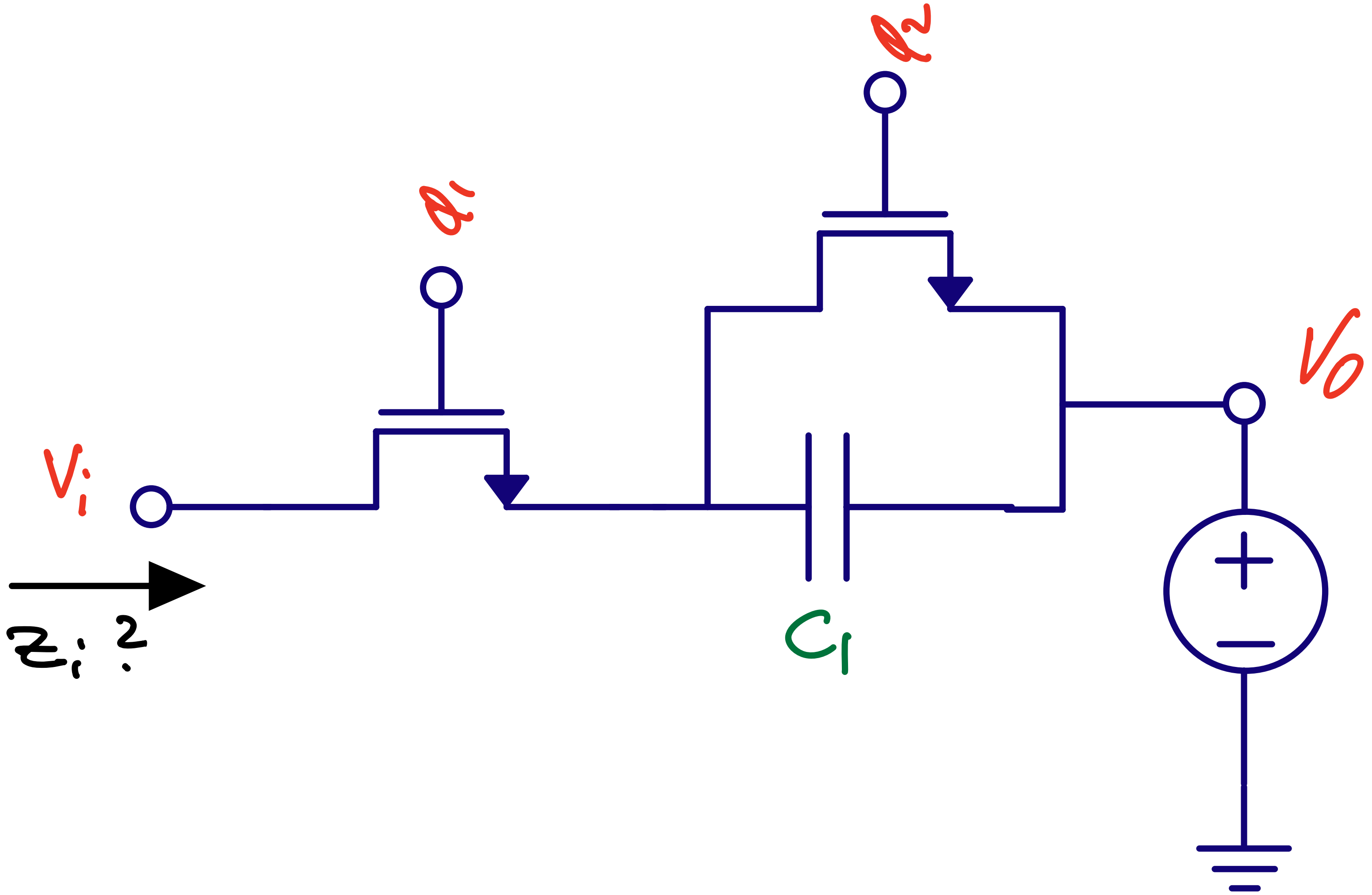
$$Q_{\phi 1\$} = C_1 V_I$$

$$Z_I = (V_I - V_{GND})/I_I$$

$$I_I = \frac{Q}{dt} = Q f_\phi$$

$$Z_I = \frac{V_I - V_{GND}}{\left(Q_{\phi 1\$} - Q_{\phi 2\$}\right) f_\phi}$$

$$Z_I = \frac{V_I}{(V_I C - 0)\, f_\phi} = \frac{1}{C_1 f_\phi}$$

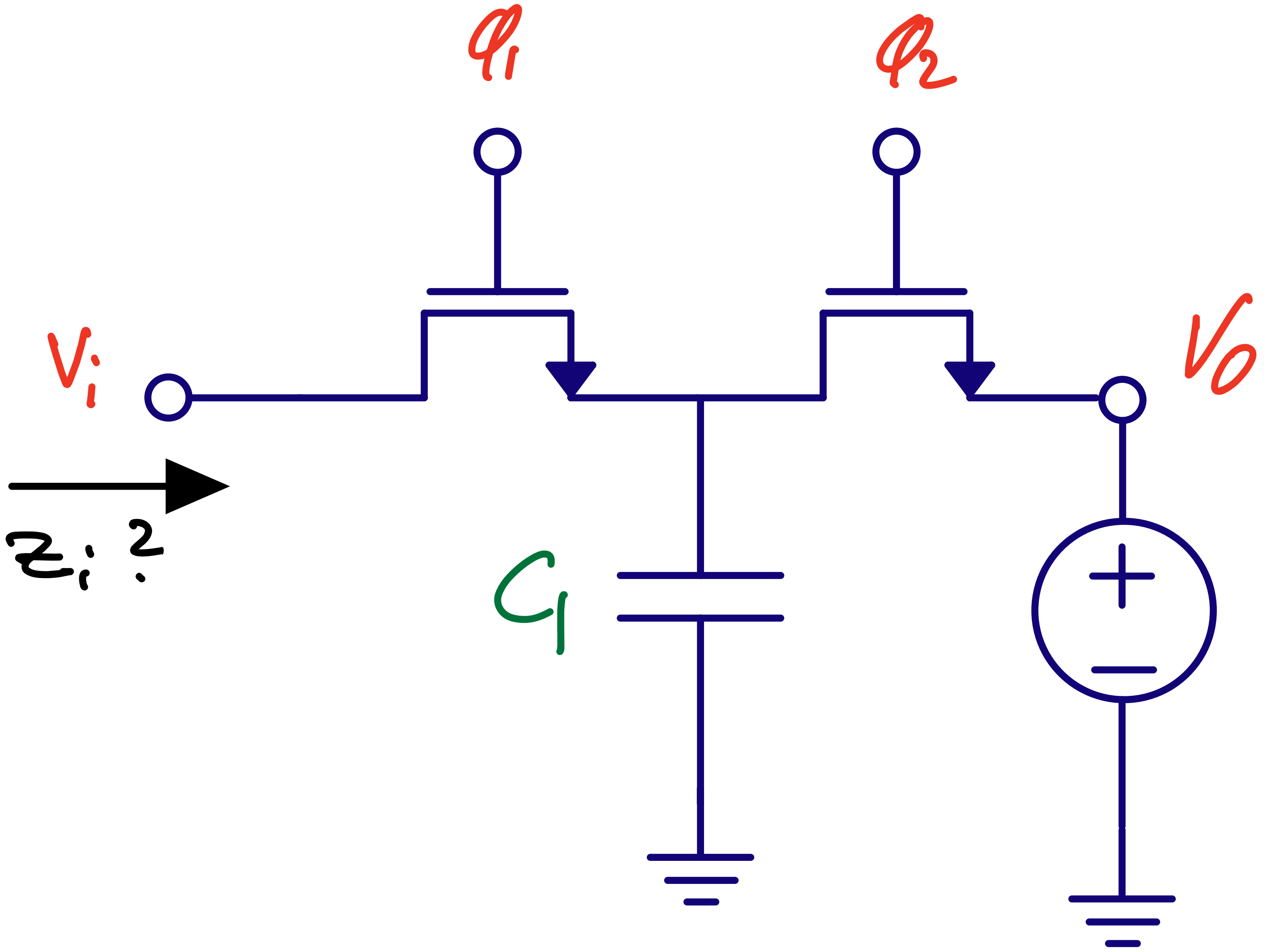


$$Z_I = \frac{V_I - V_O}{\left(Q_{\phi1\$} - Q_{\phi2\$}\right) f_\phi}$$

$$Q_{\phi1\$} = C_1(V_I - V_O)$$

$$Q_{\phi2\$} = 0$$

$$Z_I = \frac{V_I - V_O}{(C_1(V_I - V_O))\, f_\phi} = \frac{1}{C_1 f_\phi}$$



$$Z_I = \frac{V_I - V_O}{\left(Q_{\phi1\$} - Q_{\phi2\$}\right) f_\phi}$$

$$Q_{\phi1\$} = C_1 V_I)$$

$$Q_{\phi2\$} = C_1 V_O$$

$$Z_I = \frac{V_I - V_O}{(C_1 V_I - C_1 V_O)) f_\phi} = \frac{1}{C_1 f_\phi}$$

A pipelined 5-Msample/s 9-bit analog-to-digital converter

$$\omega_{p|z} \propto \frac{C_1}{C_2}$$

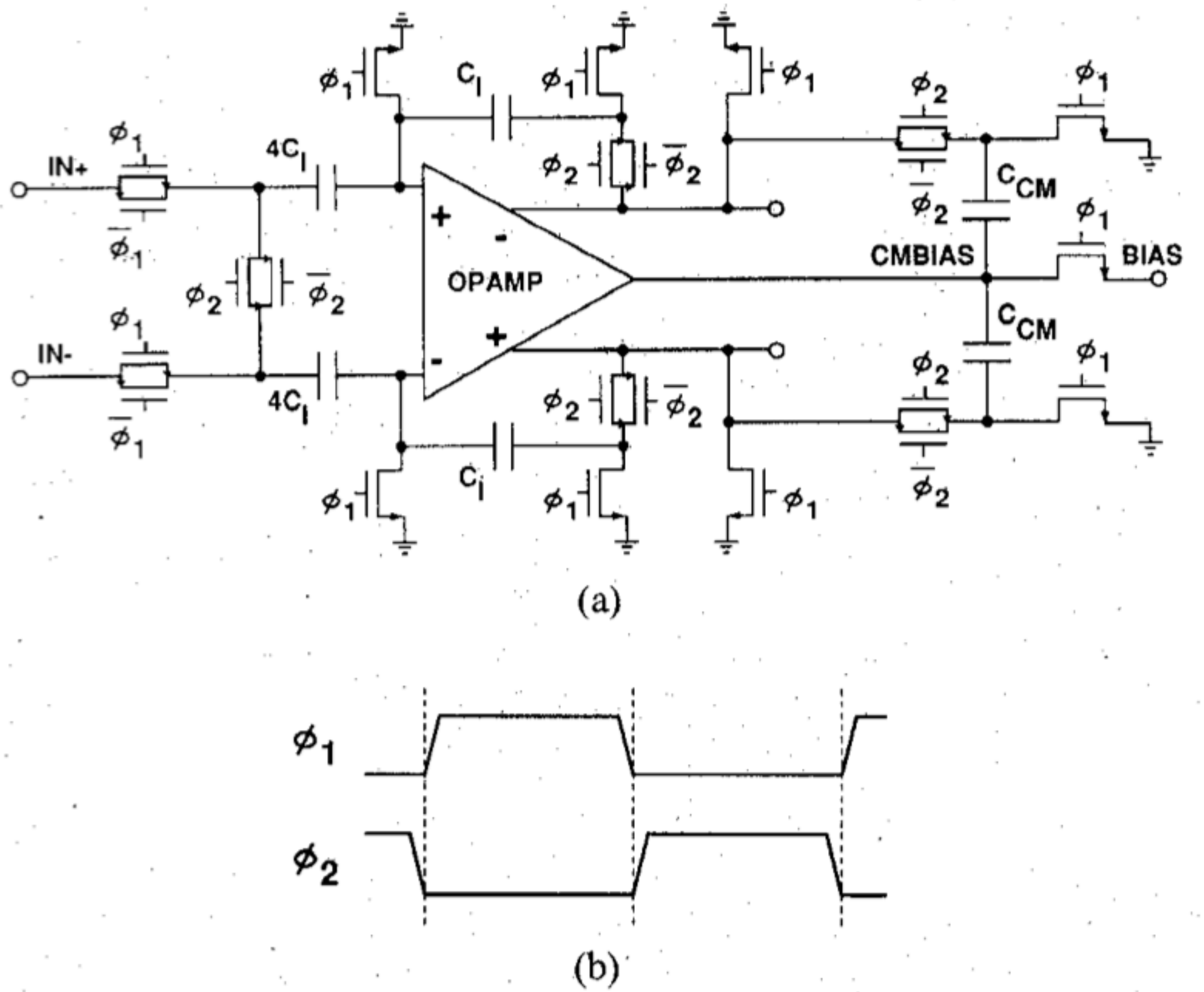


Fig. 6. (a) Schematic of S/H amplifier. (b) Timing diagram of a two-phase nonoverlapping clock.

Discrete-Time Signals

Define x_c as a continuous time, continuous value signal

Define $\ell(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases}$

Define

$$x_{sn}(t) = \frac{x_c(nT)}{\tau}\left[\ell(t - nT) - \ell(t - nT - \tau)\right]$$

Define $x_s(t) = \sum_{n=-\infty}^{\infty} x_{sn}(t)$

Think of a sampled version of an analog signal as an infinite sum of pulse trains where the area under the pulse train is equal to the analog signal.

Why do this?

If $x_s(t) = \sum_{n=-\infty}^{\infty} x_{sn}(t)$

Then $X_{sn}(s) = \frac{1}{\tau}\frac{1 - e^{-s\tau}}{s} x_c(nT)e^{-snT}$

And $X_s(s) = \frac{1}{\tau}\frac{1 - e^{-s\tau}}{s} \sum_{n=-\infty}^{\infty} x_c(nT)e^{-snT}$

Thus $\lim_{\tau \to 0} \to X_s(s) = \sum_{n=-\infty}^{\infty} x_c(nT)e^{-snT}$

Or $X_s(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(j\omega - \frac{jk2\pi}{T}\right)$

The spectrum of a sampled signal is an infinite sum of frequency shifted spectra

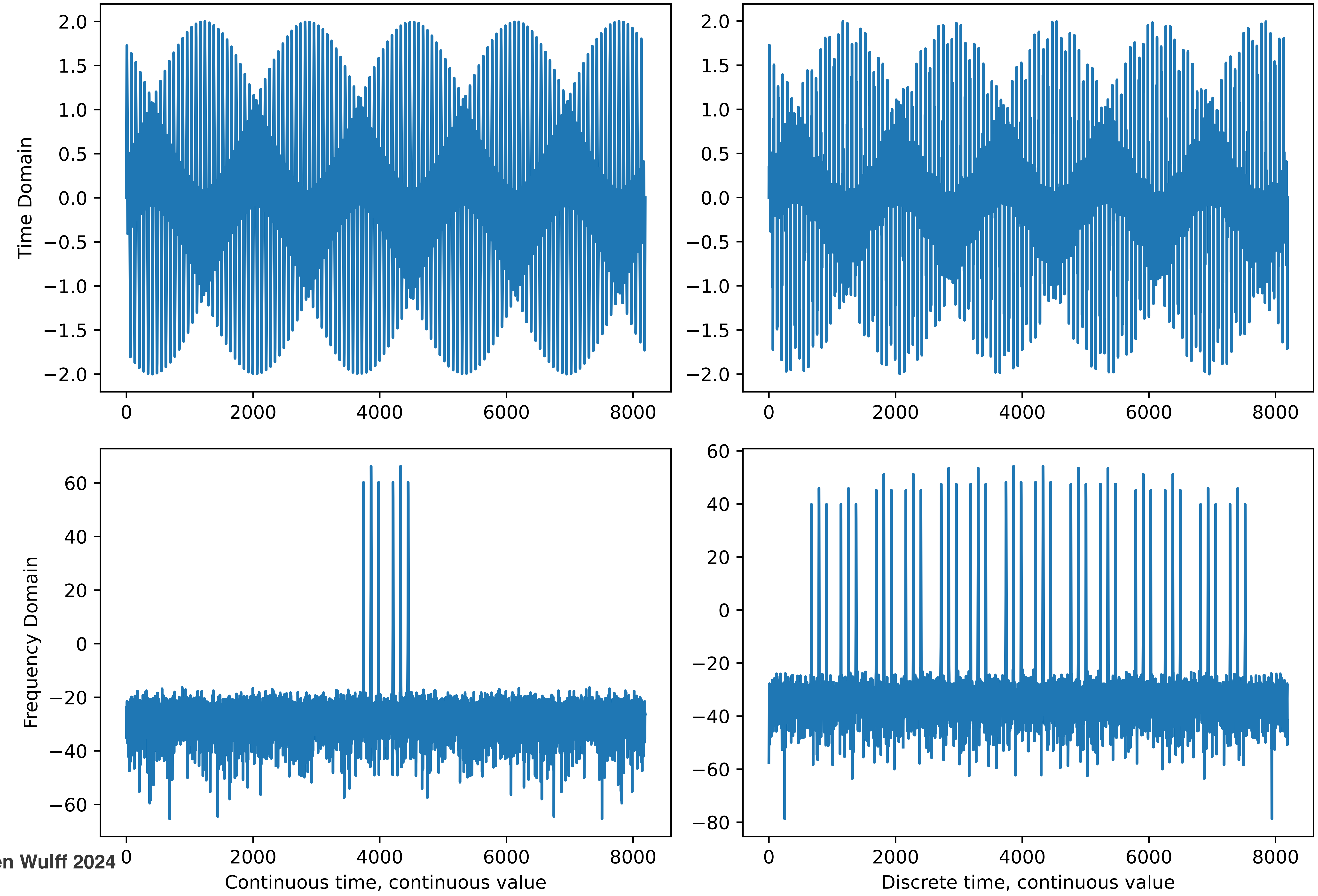
or equivalently

When you sample a signal, then there will be copies of the input spectrum at every nf_s

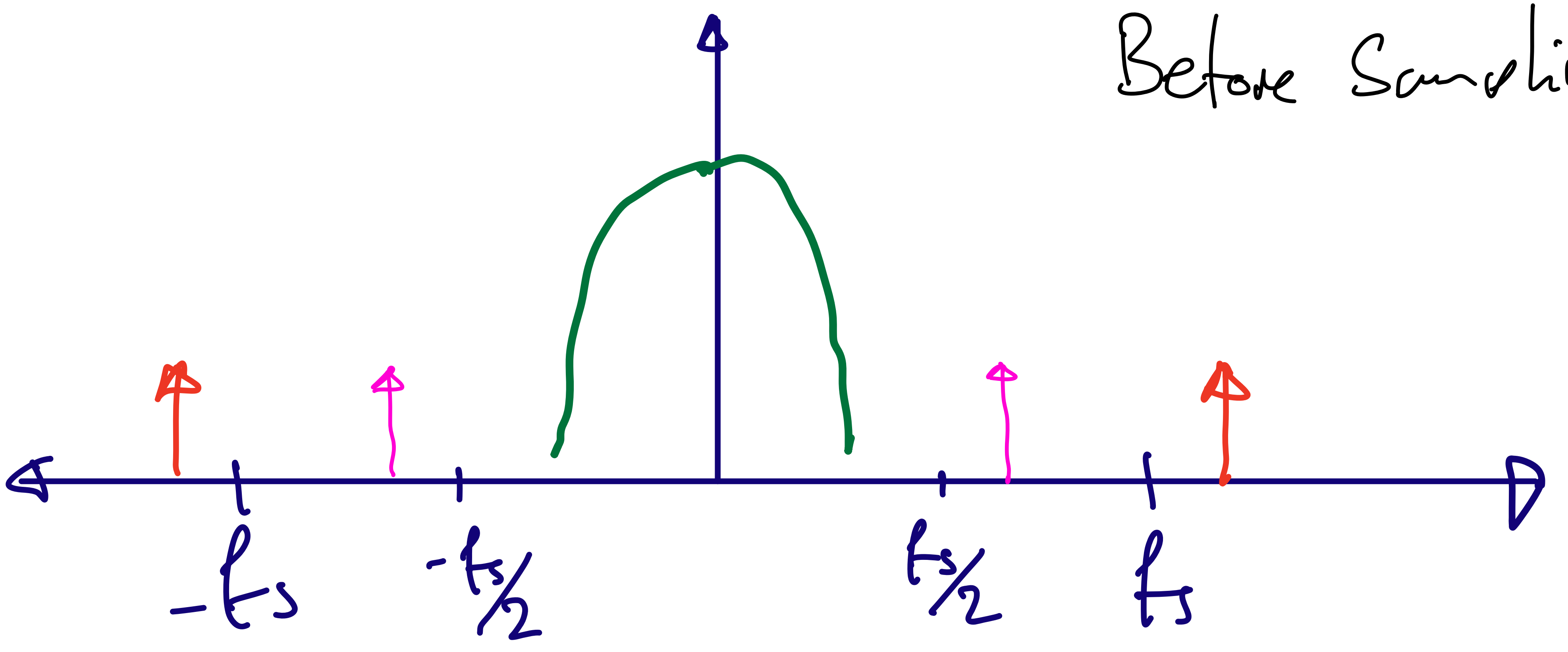
However, if you do an FFT of a sampled signal, then all those infinite spectra will fold down between $0 \to f_{s1}/2$ or $-f_{s1}/2 \to f_{s1}/2$ for a complex FFT

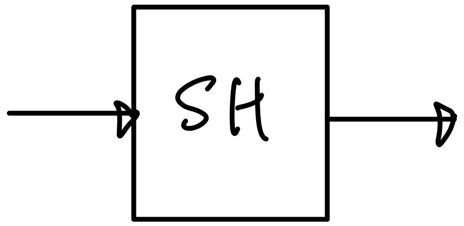
dt.py

```
#- Create a time vector
N = 2**13
t = np.linspace(0,N,N)

#- Create the "continuous time" signal with multiple sinusoidal signals and some noise
f1 = 233/N
fd = 1/N*119
x_s = np.sin(2*np.pi*f1*t) + 1/1024*np.random.randn(N) +   0.5*np.sin(2*np.pi*(f1-fd)*t) + 0.5*np.sin(2*np.pi*(f1+fd)*t)

#- Create the sampling vector, and the sampled signal
t_s_unit = [1,1,0,0,0,0,0,0]
t_s = np.tile(t_s_unit,int(N/len(t_s_unit)))
x_sn = x_s*t_s

#- Convert to frequency domain with a hanning window to avoid FFT bin
#- energy spread
Hann = True
if(Hann):
    w = np.hanning(N+1)
else:
    w = np.ones(N+1)
X_s = np.fft.fftshift(np.fft.fft(np.multiply(w[0:N],x_s)))
X_sn = np.fft.fftshift(np.fft.fft(np.multiply(w[0:N],x_sn)))
```

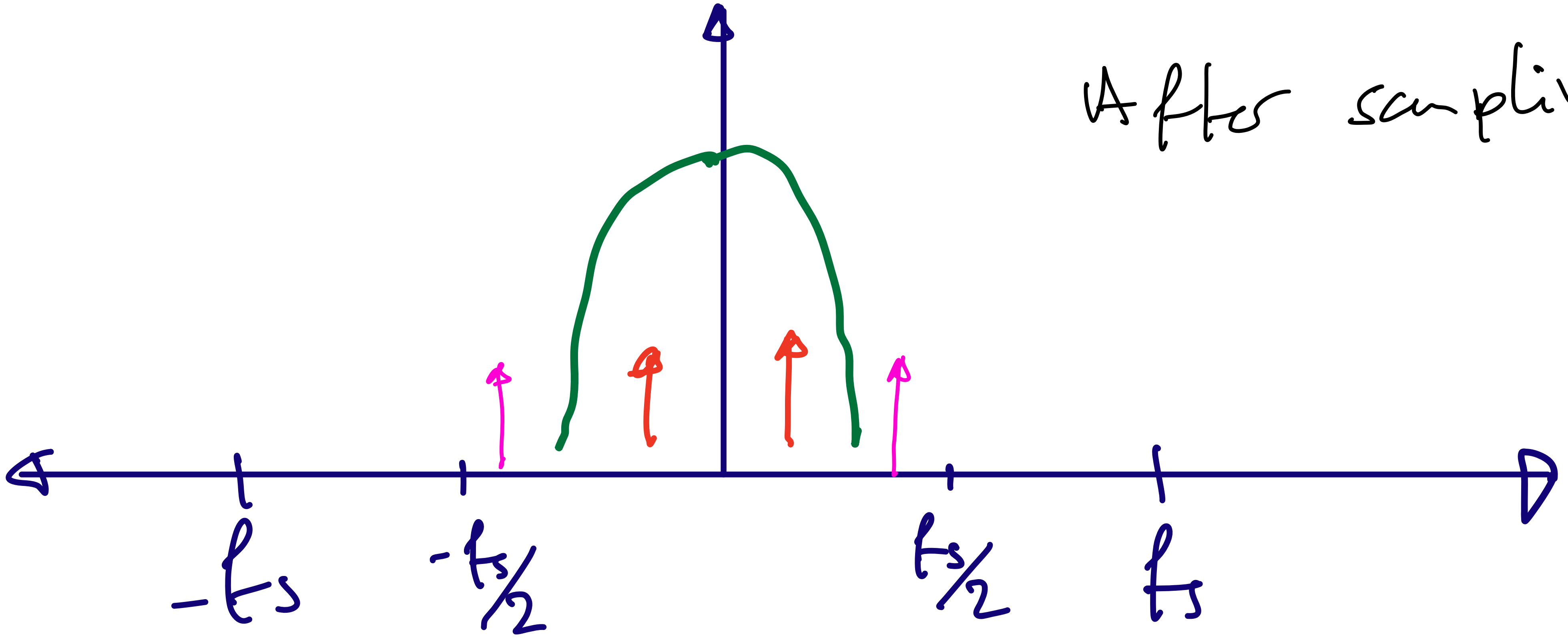

Time Domain
Frequency Domain
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
60
40
20
0
−20
−40
−60
−80
0
2000
4000
6000
8000
Continuous time, continuous value
Discrete time, continuous value

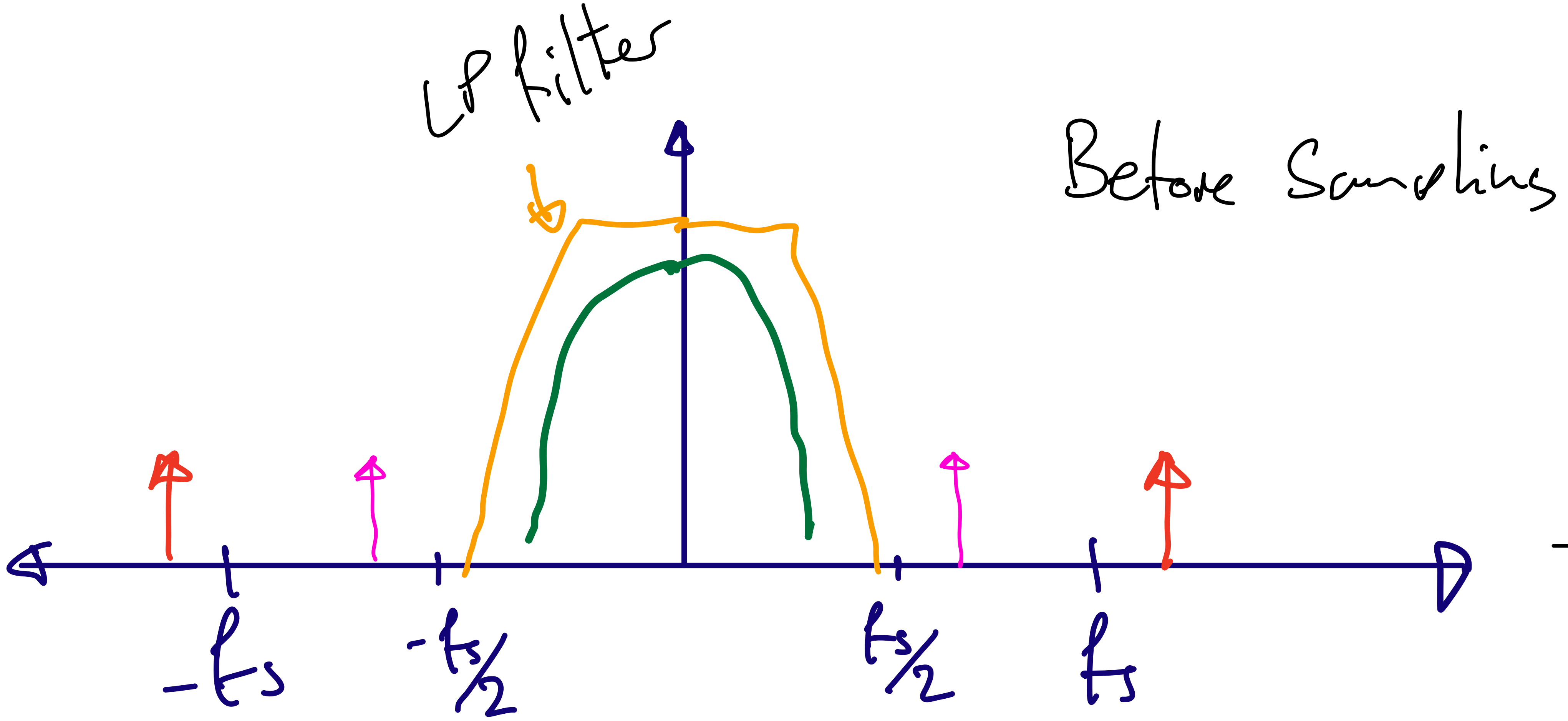
Before Sampling

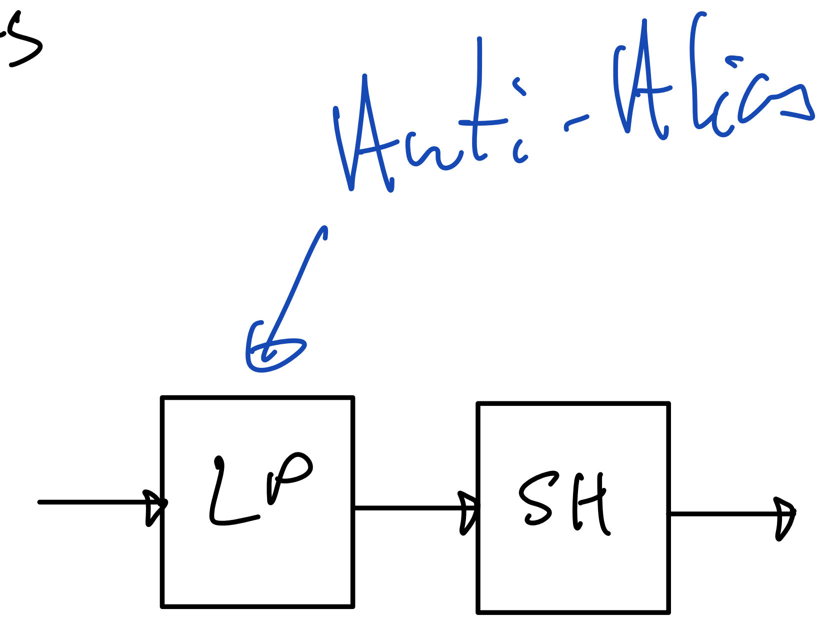


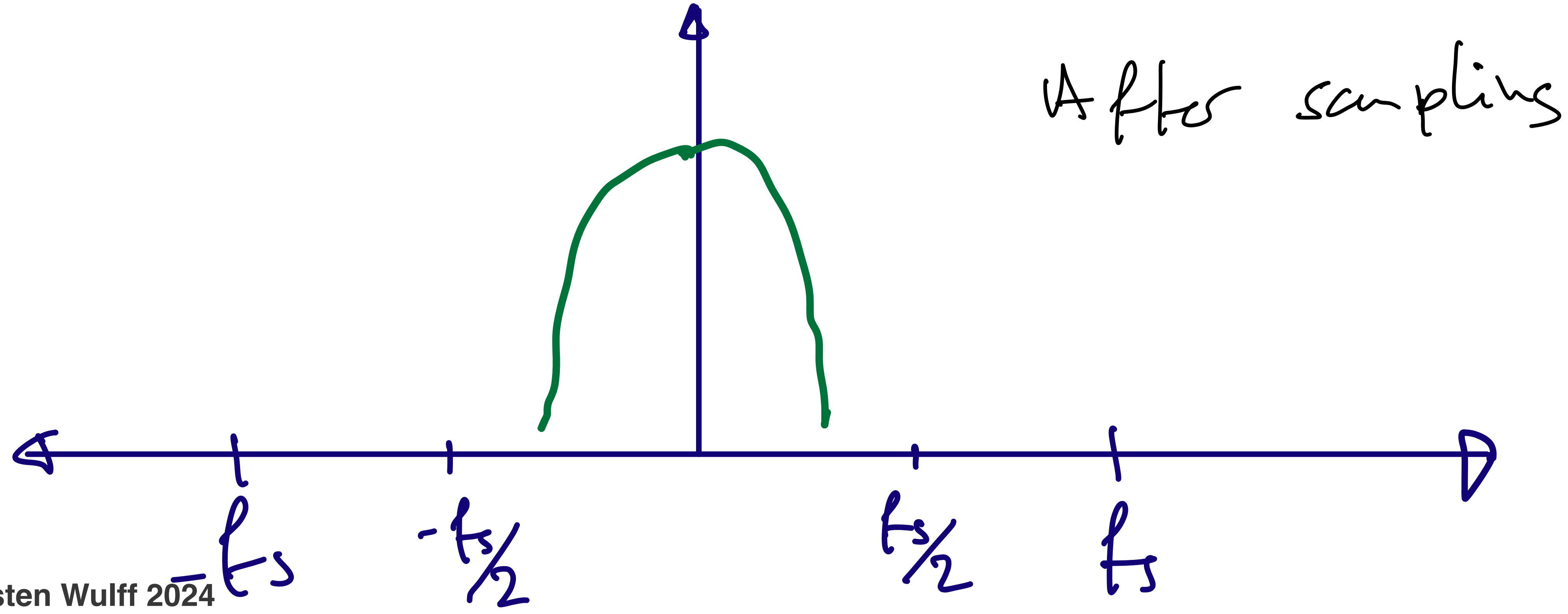


After sampling

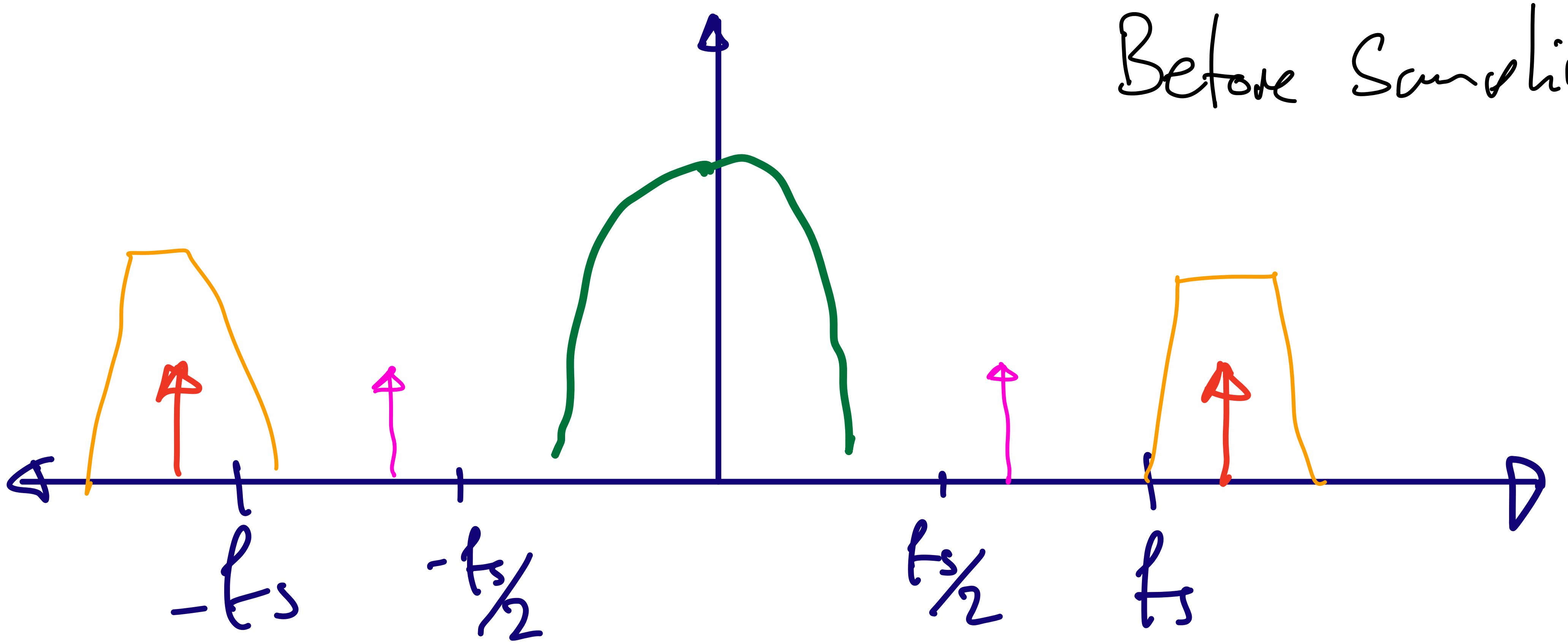


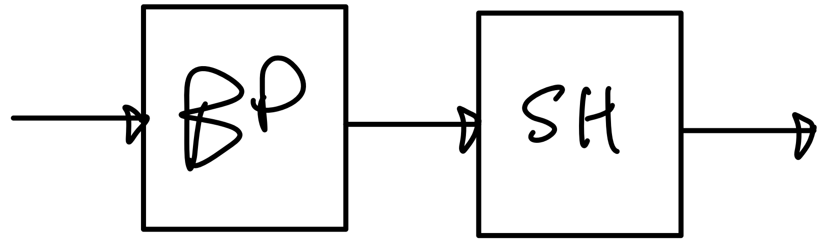

LP filter
Before Sampling
-fs
-fs/2
fs/2
fs


Anti-Alias
LP
SH

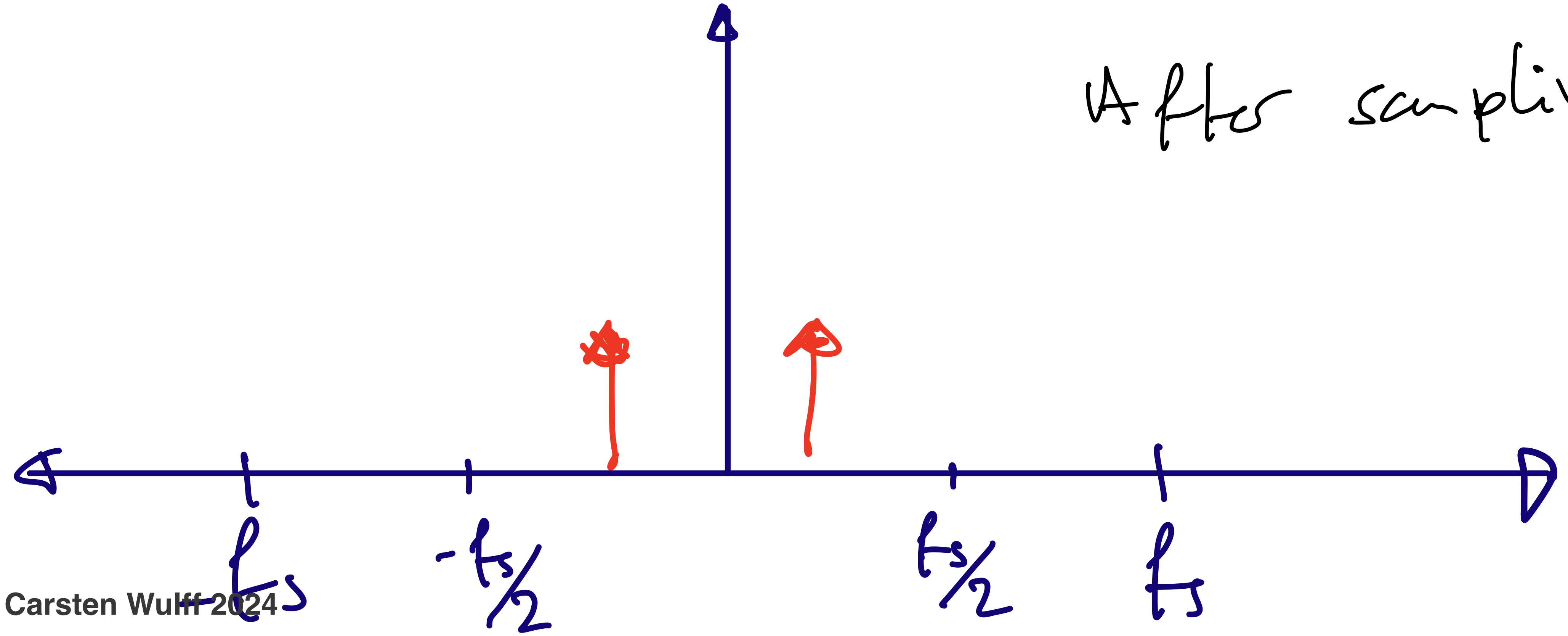

After sampling
-fs
-fs/2
fs/2
fs

Before Sampling





After sampling

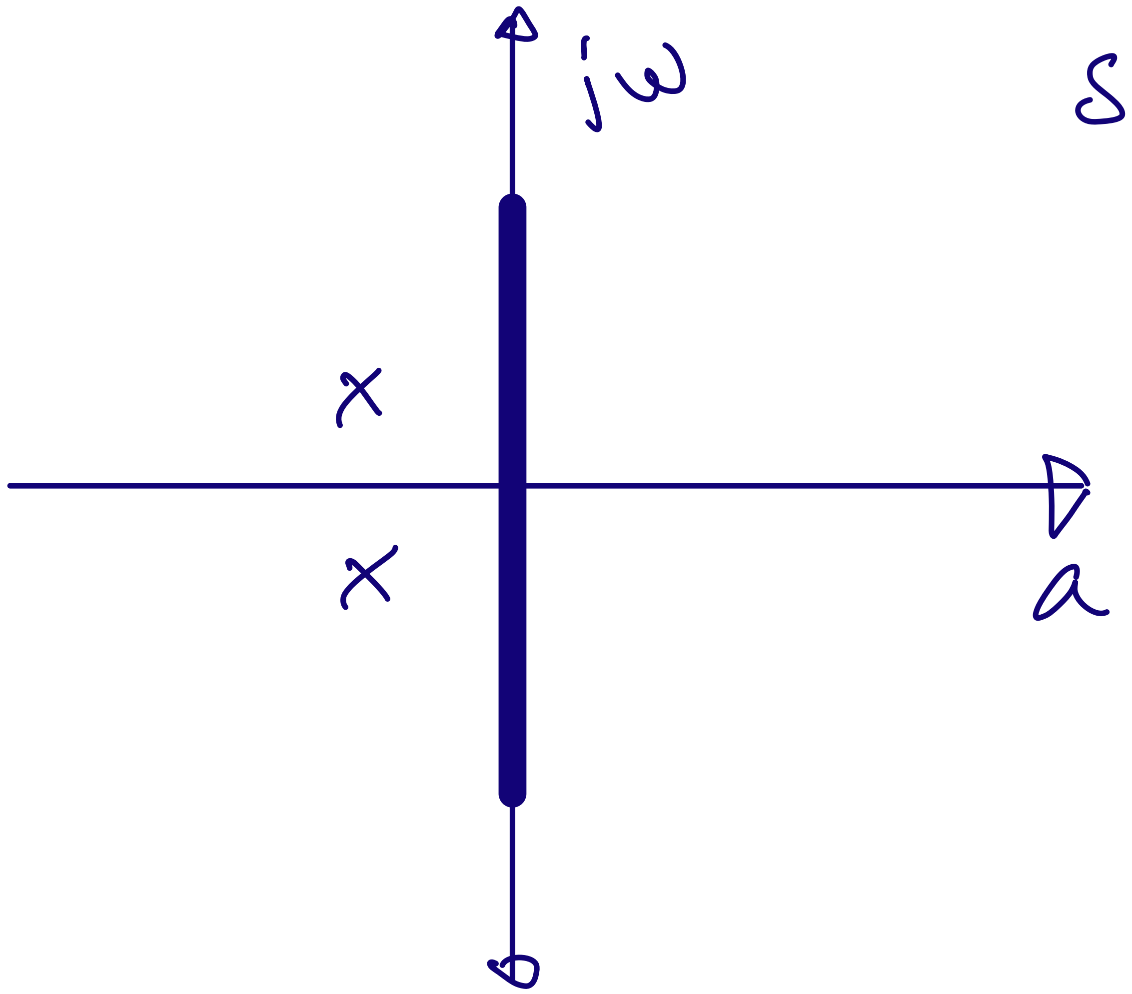


$$X_s(s) = \sum_{n=-\infty}^{\infty} x_c(nT)e^{-snT}$$

$$X_s(z) = \sum_{n=-\infty}^{\infty} x_c[n]z^{-n}$$

For discrete time signal processing we use Z-transform

If you're unfamiliar with the Z-transform, read the book or search https://en.wikipedia.org/wiki/Z-transform

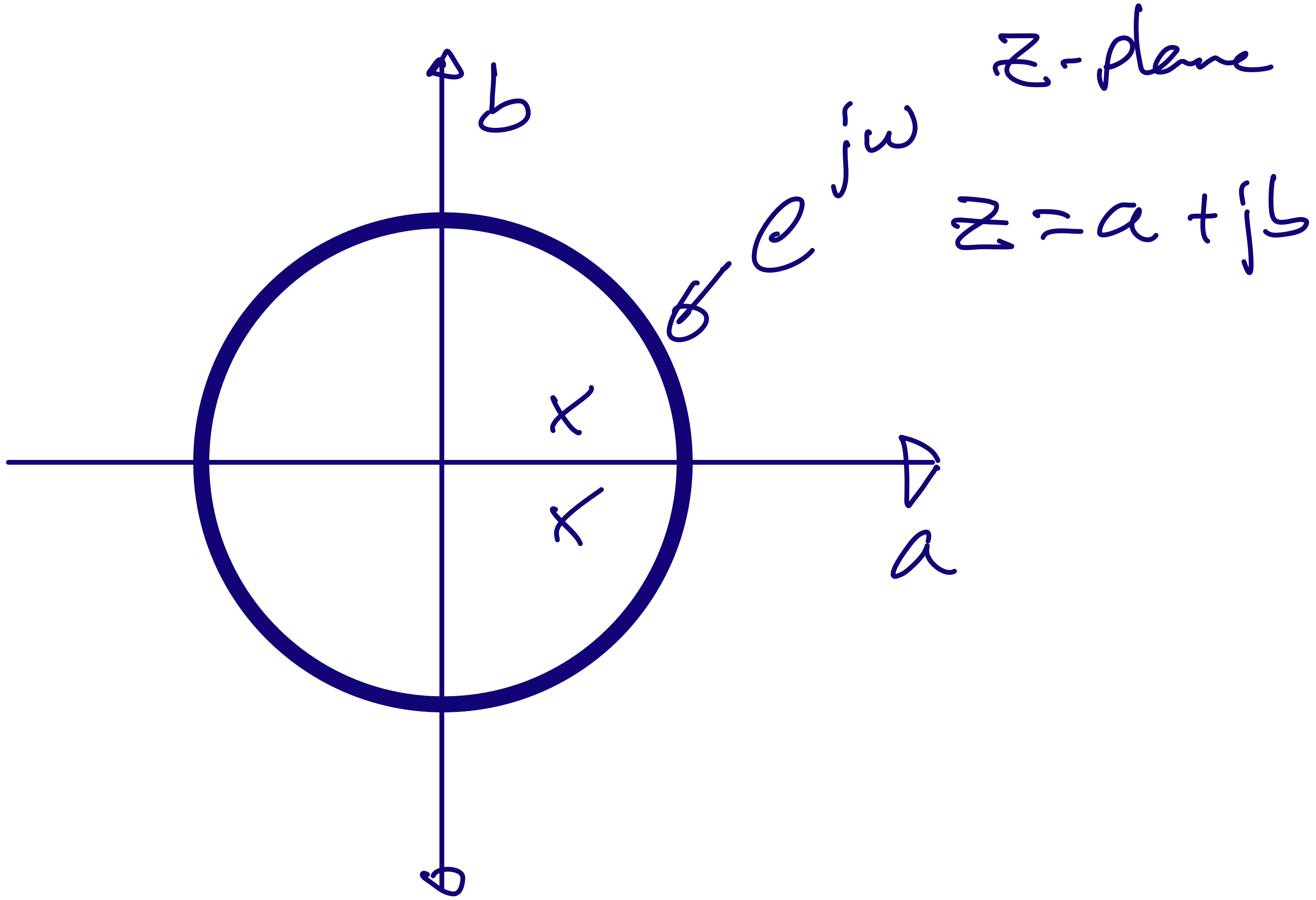


$s = a + j\omega$

Pole-Zero plots

If you're not comfortable with pole/zero plots, have a look at

[What does the Laplace Transform really tell us]



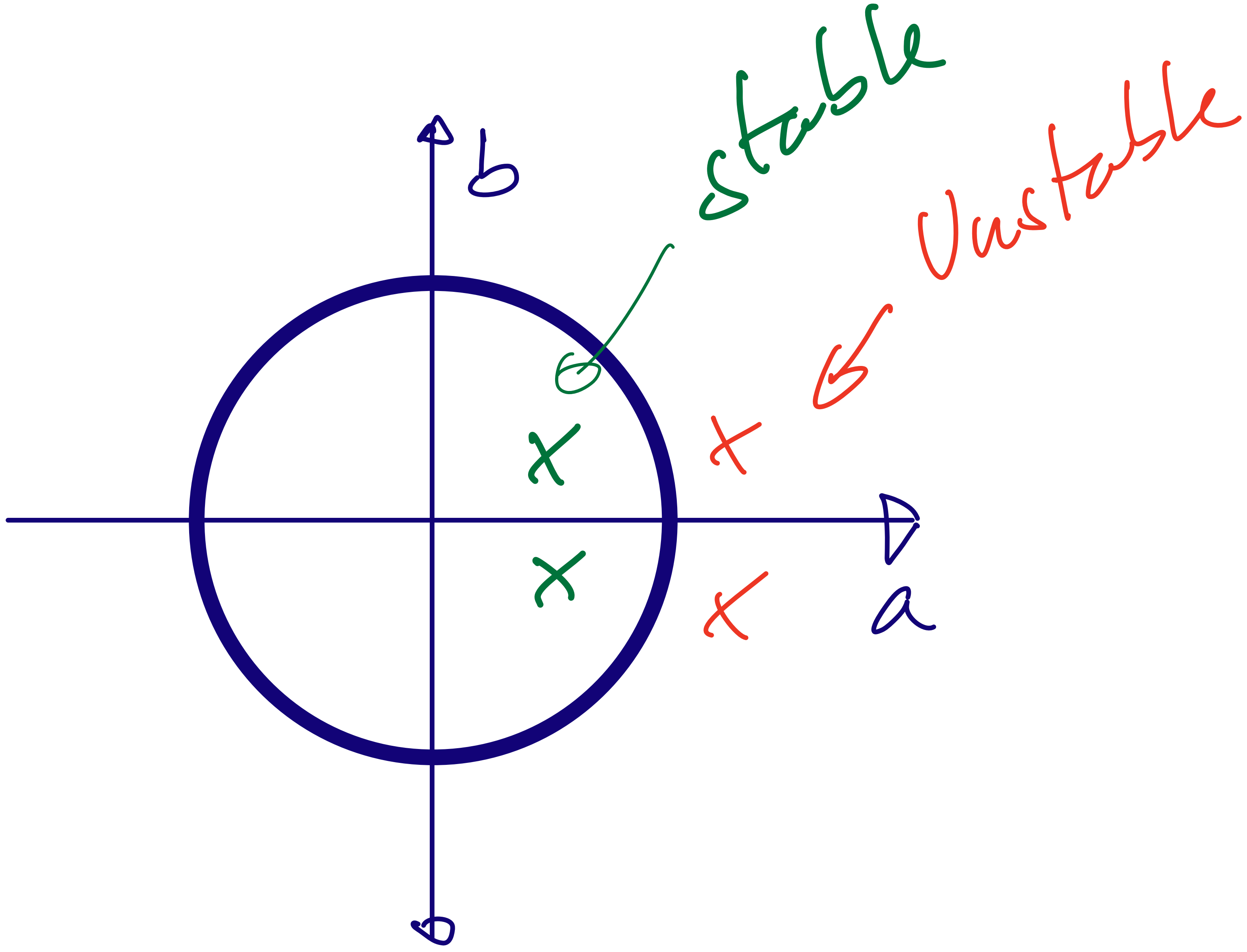
Z-domain

Spectra repeat every 2π

Bi-linear transform

$$s = \frac{z-1}{z+1}$$

Warning: First-order approximation https://en.wikipedia.org/wiki/Bilinear_transform



First order filter

$$y[n+1] = bx[n] + ay[n] \Rightarrow Yz = bX + aY$$

$$y[n] = bx[n-1] + ay[n-1] \Rightarrow Y = bXz^{-1} + aYz^{-1}$$

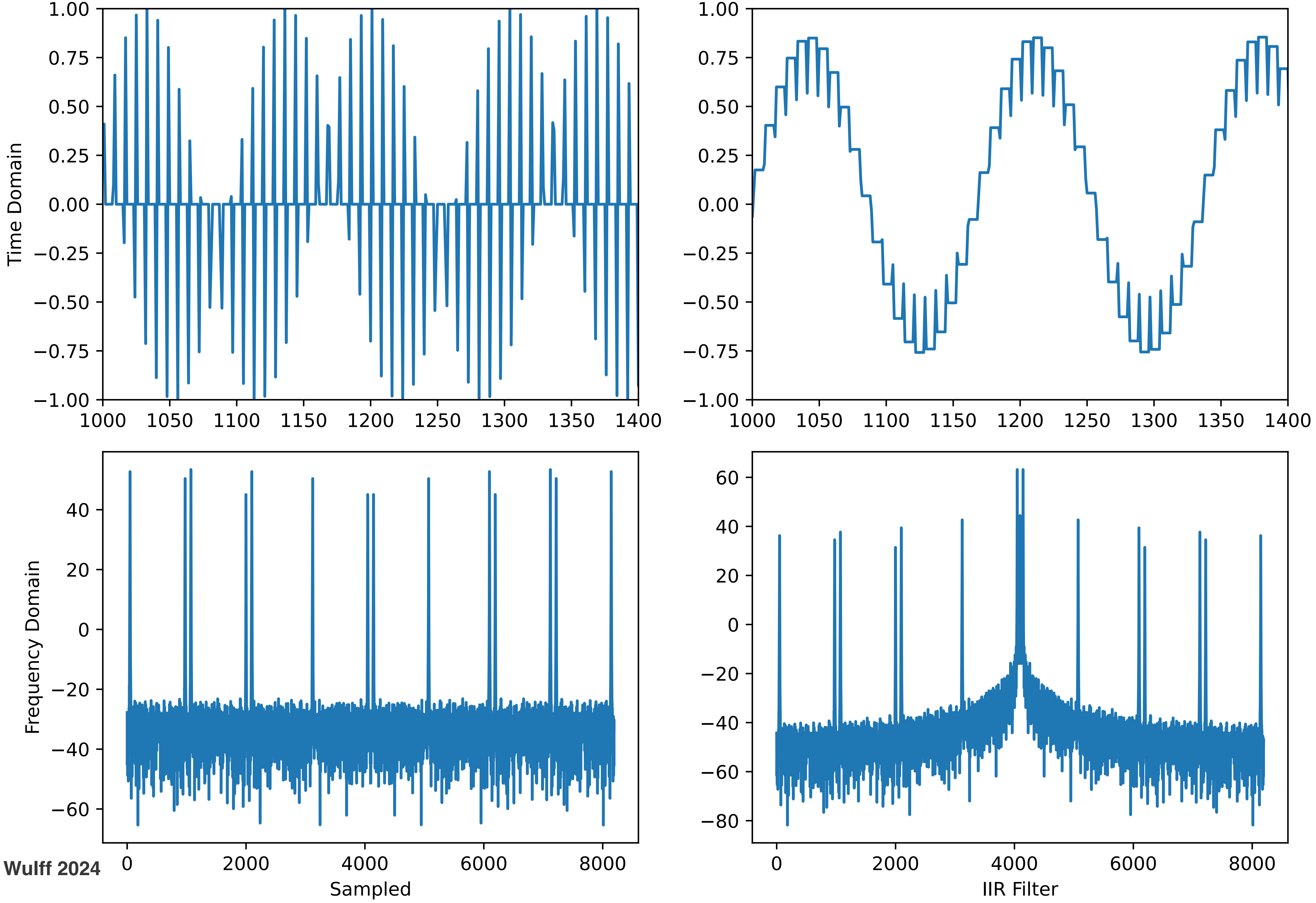
$$H(z) = \frac{b}{z-a}$$

Infinite-impulse response (IIR)

$$h[n] = \begin{cases} k & \text{if } n < 1 \\ a^{n-1}b + a^n k & \text{if } n \geq 1 \end{cases}$$

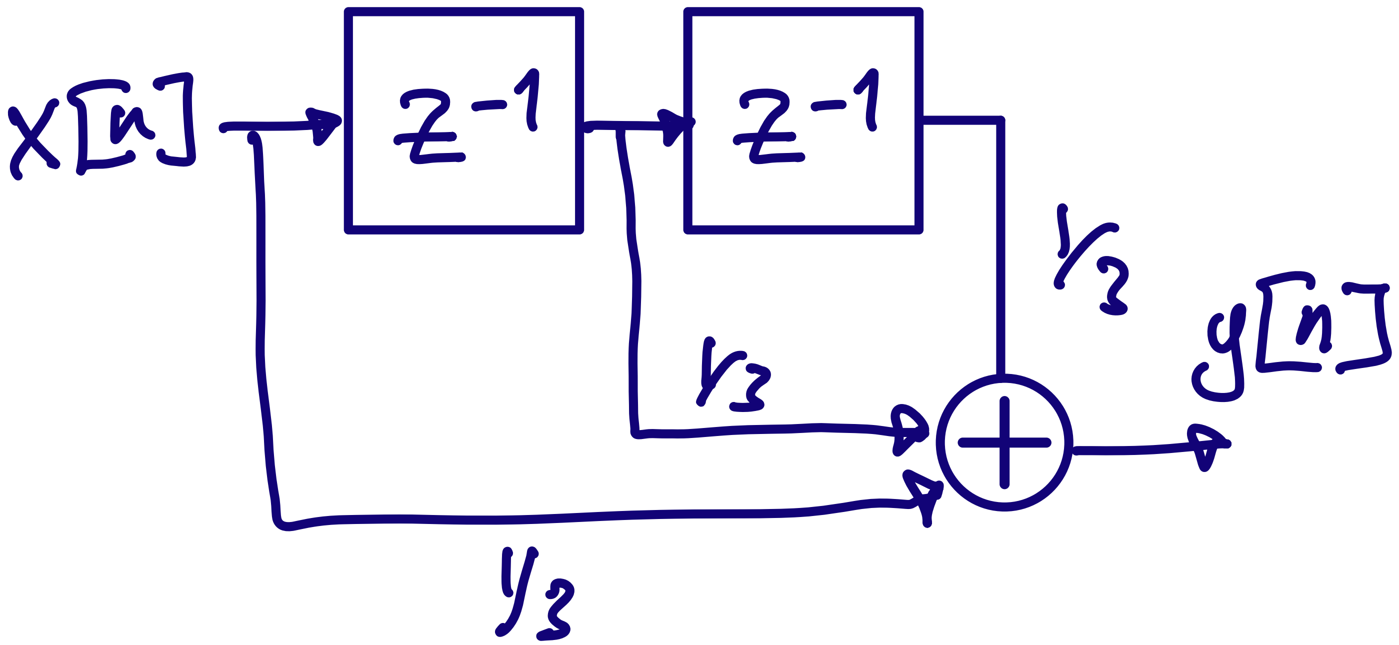
Head's up: Fig 13.12 in AIC is wrong

iir.py



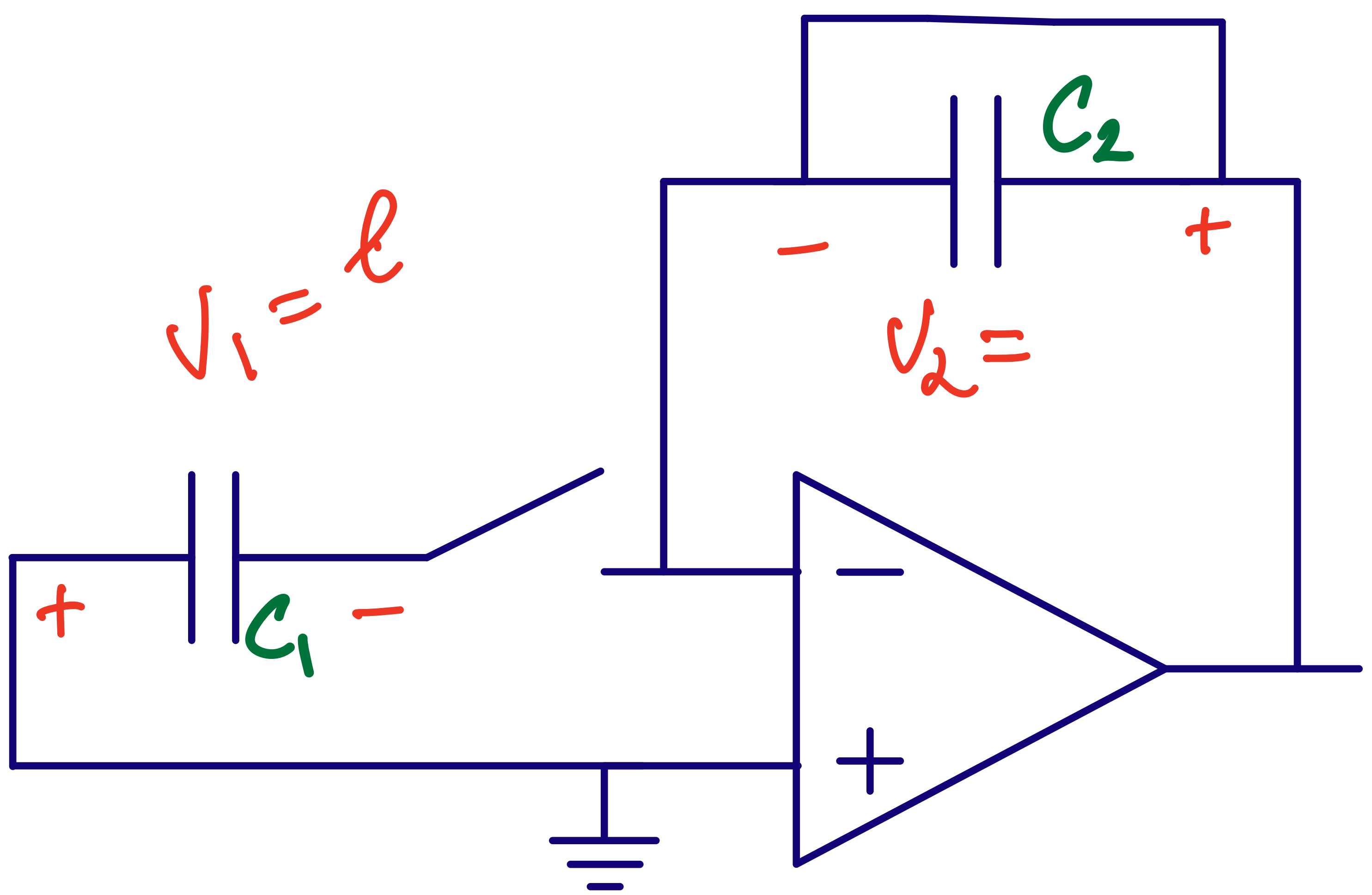
Time Domain
Frequency Domain
Sampled
IIR Filter

Finite-impulse response(FIR)



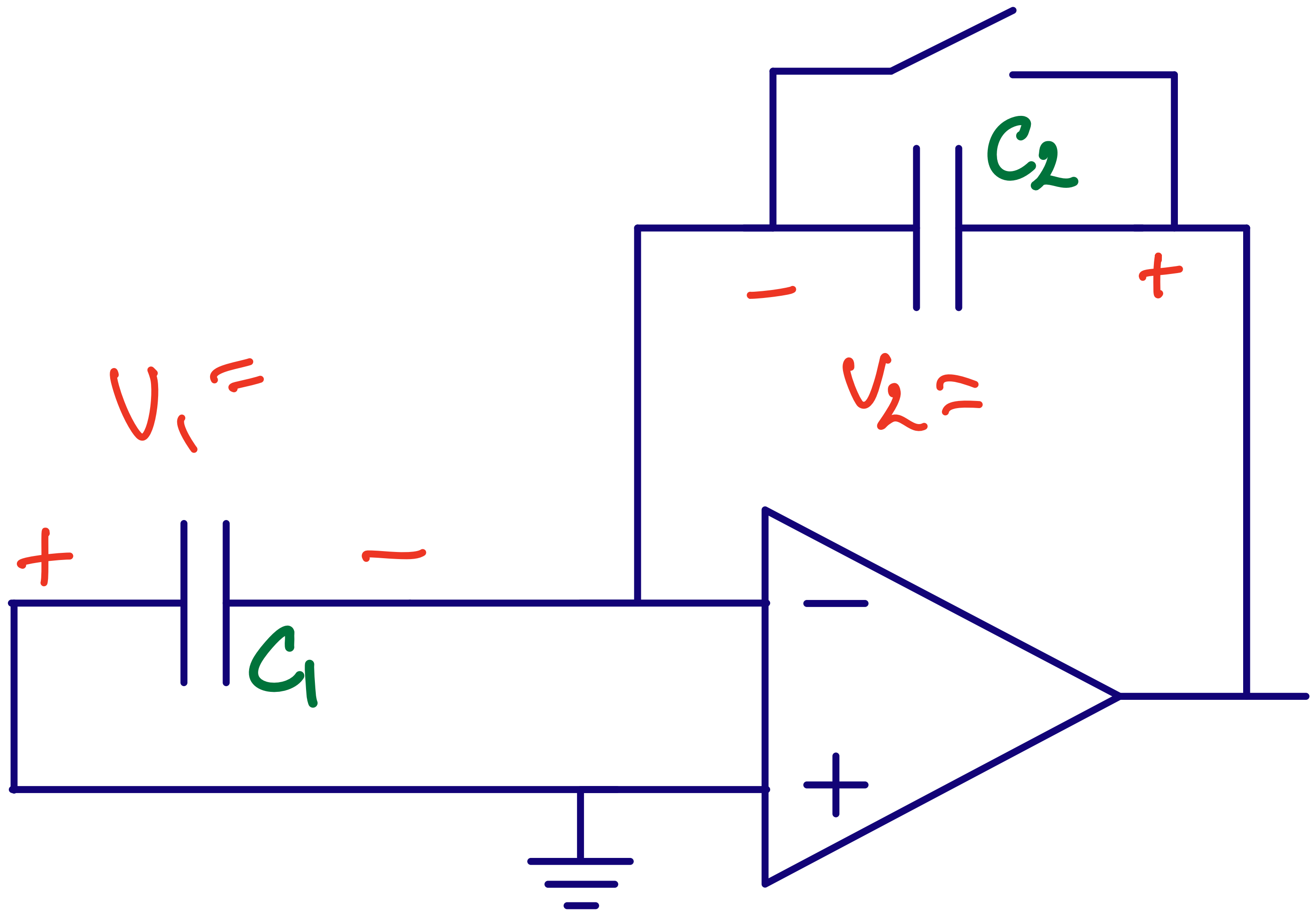
$$H(z) = \frac{1}{3}\sum_{i=0}^{2} z^{-1}$$

Switched-Capacitor



$$Q_{1\phi_1\$} = C_1 V_1$$

$$Q_{2\phi_1\$} = 0$$



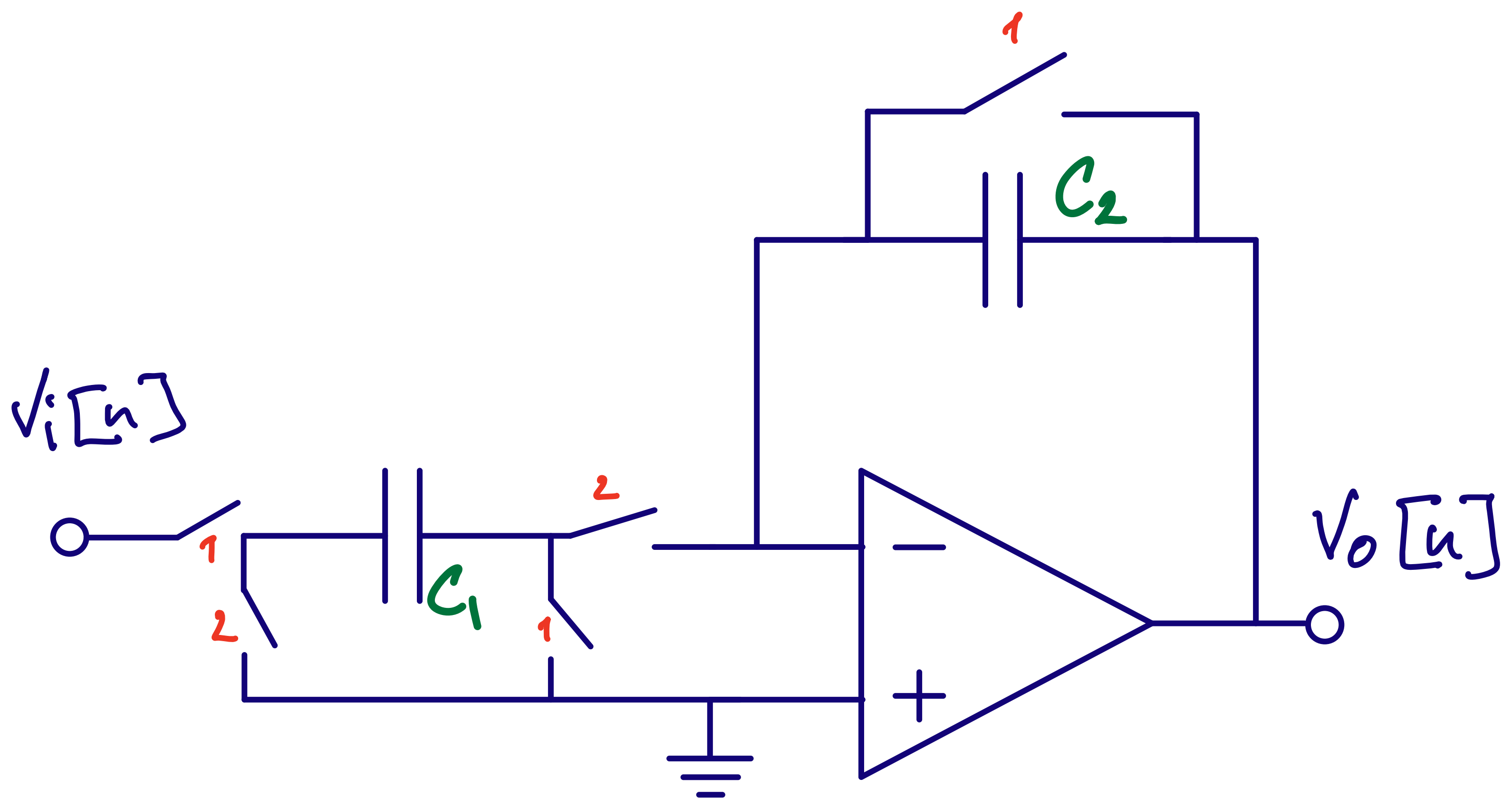
$$Q_{1\phi_2\$} = 0$$

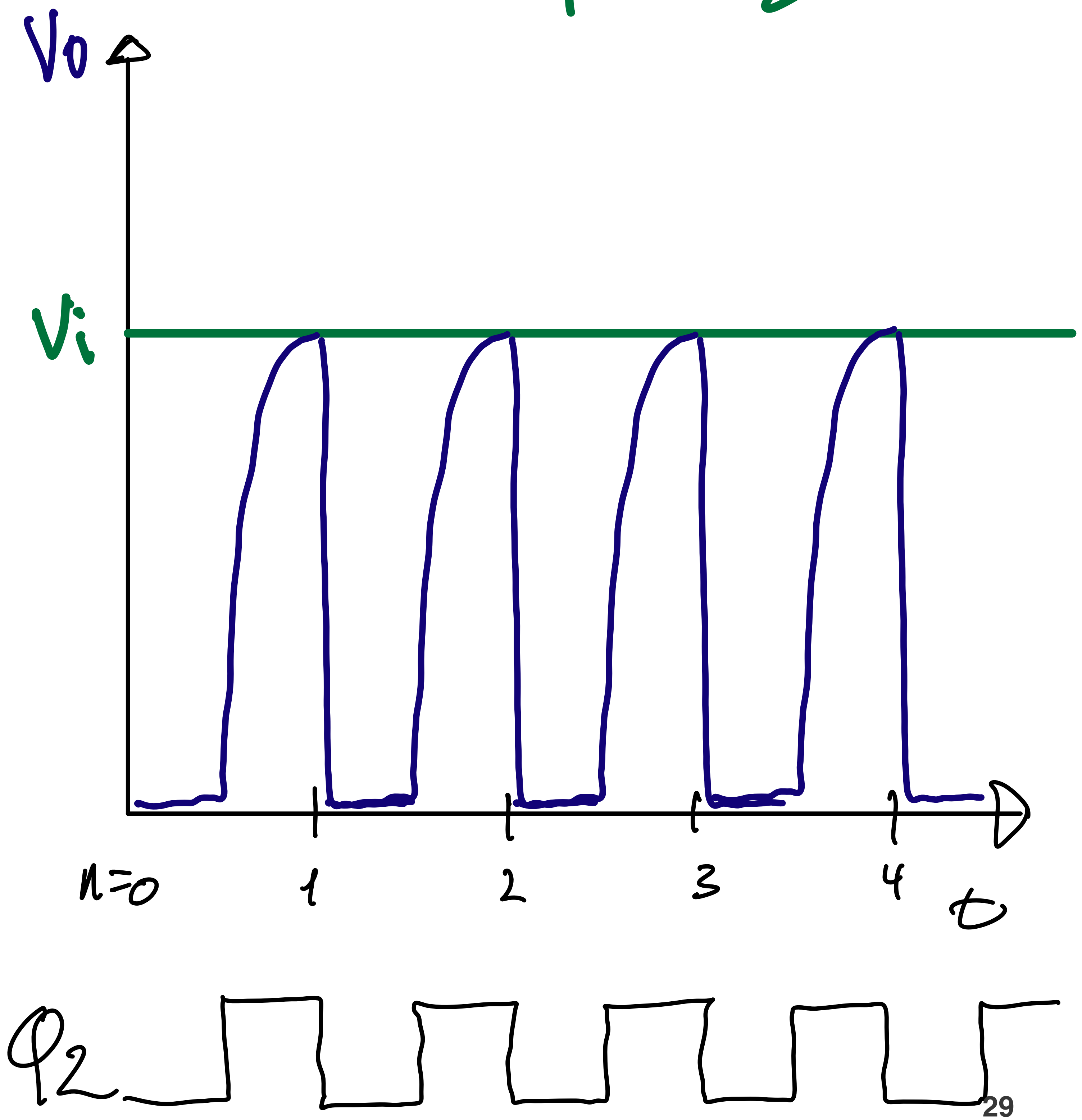
$$Q_{2\phi_2\$} = Q_{1\phi_1\$} = C_1 V_1 = C_2 V_2$$

$$\frac{V_2}{V_1} = \frac{C_1}{C_2}$$

Switched capacitor gain circuit

$C_1 = C_2$





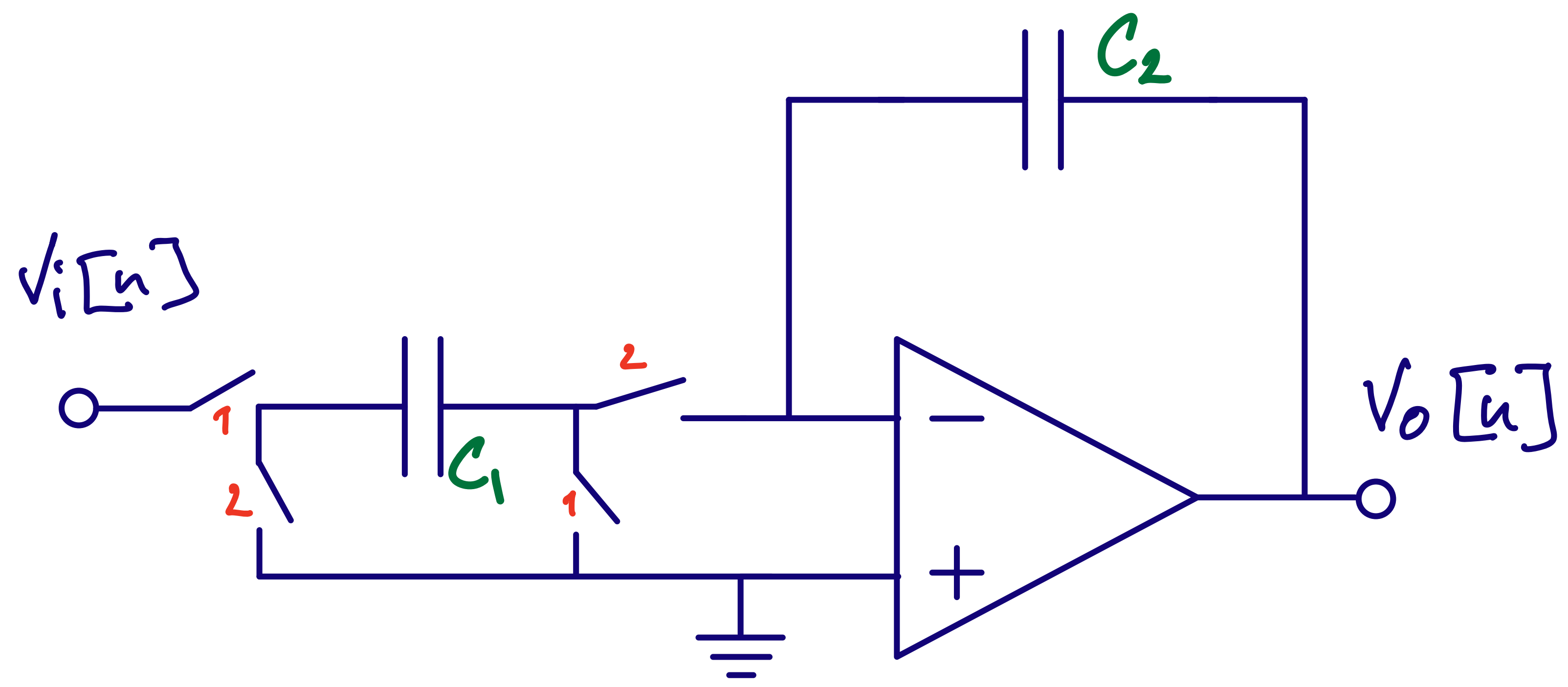
$$V_o[n+1] = \frac{C_1}{C_2} V_i[n]$$

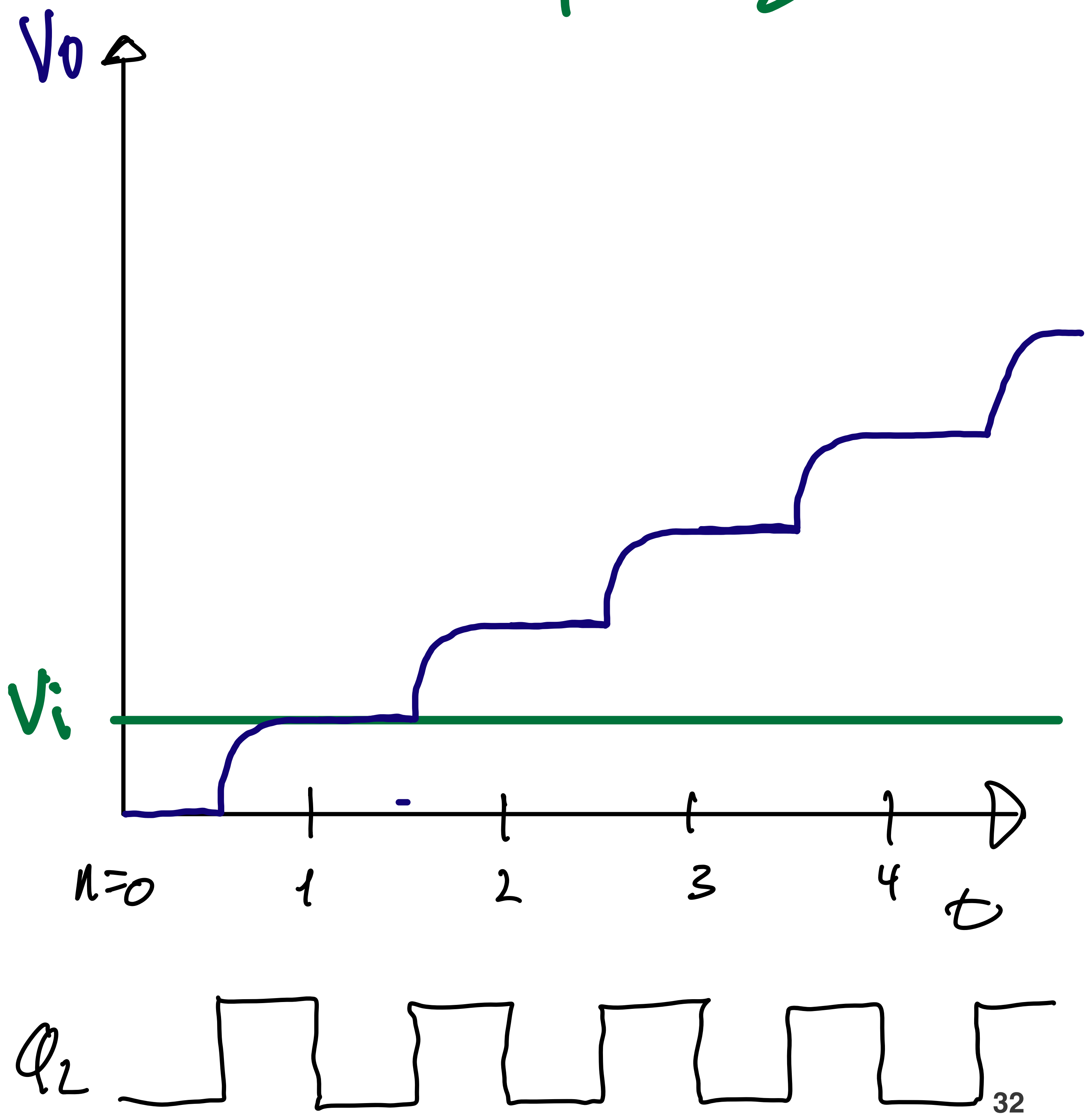
$$V_o z = \frac{C_1}{C_2} V_i$$

$$\frac{V_o}{V_i} = H(z) = \frac{C_1}{C_2} z^{-1}$$

Switched capacitor integrator

$C_1 = C_2$





$$V_o[n] = V_o[n-1] + \frac{C_1}{C_2} V_i[n-1]$$

$$V_o - z^{-1} V_o = \frac{C_1}{C_2} z^{-1} V_i$$

$$H(z) = \frac{C_1}{C_2} \frac{z^{-1}}{1 - z^{-1}} = \frac{C_1}{C_2} \frac{1}{z - 1}$$

Both phases add noise, $V_n^2 > \dfrac{2kT}{C}$

Mean

$$\overline{x(t)} = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{+T/2} x(t)dt$$

Mean Square

$$\overline{x^2(t)} = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{+T/2} x^2(t)dt$$

Variance

$$\sigma^2 = \overline{x^2(t)} - \overline{x(t)}^2$$

where σ is the standard deviation.
If mean is removed, or is zero, then

$$\sigma^2 = \overline{x^2(t)}$$

Assume two random processes, $x_1(t)$ and $x_2(t)$ with mean of zero (or removed).

$$x_{tot}(t) = x_1(t) + x_2(t)$$

$$x_{tot}^2(t) = x_1^2(t) + x_2^2(t) + 2x_1(t)x_2(t)$$

Variance (assuming mean of zero)

$$\sigma_{tot}^2 = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{+T/2} x_{tot}^2(t)dt$$

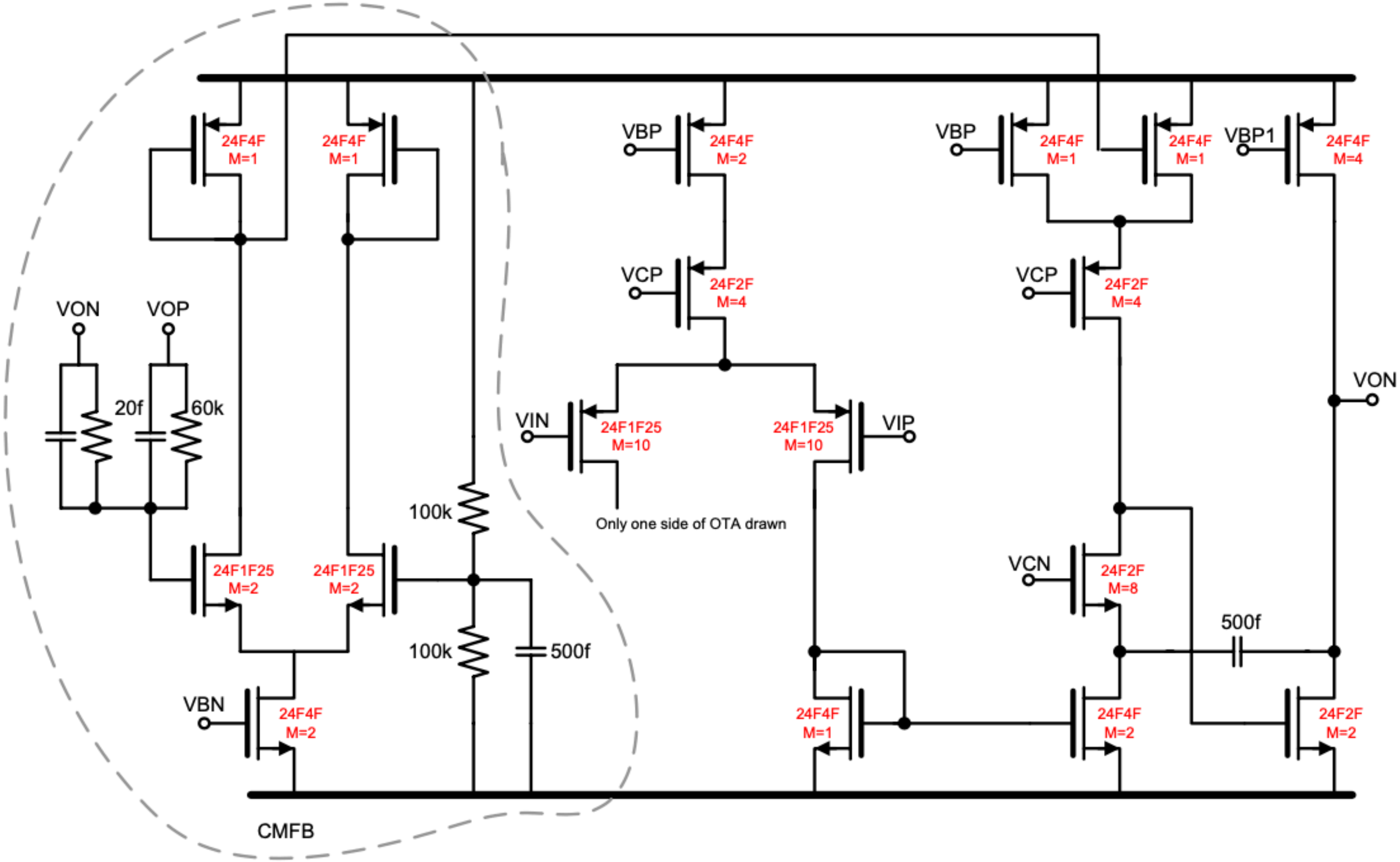
$$\sigma_{tot}^2 = \sigma_1^2 + \sigma_2^2 + \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{+T/2} 2x_1(t)x_2(t)dt$$

Assuming uncorrelated processes (covariance is zero), then

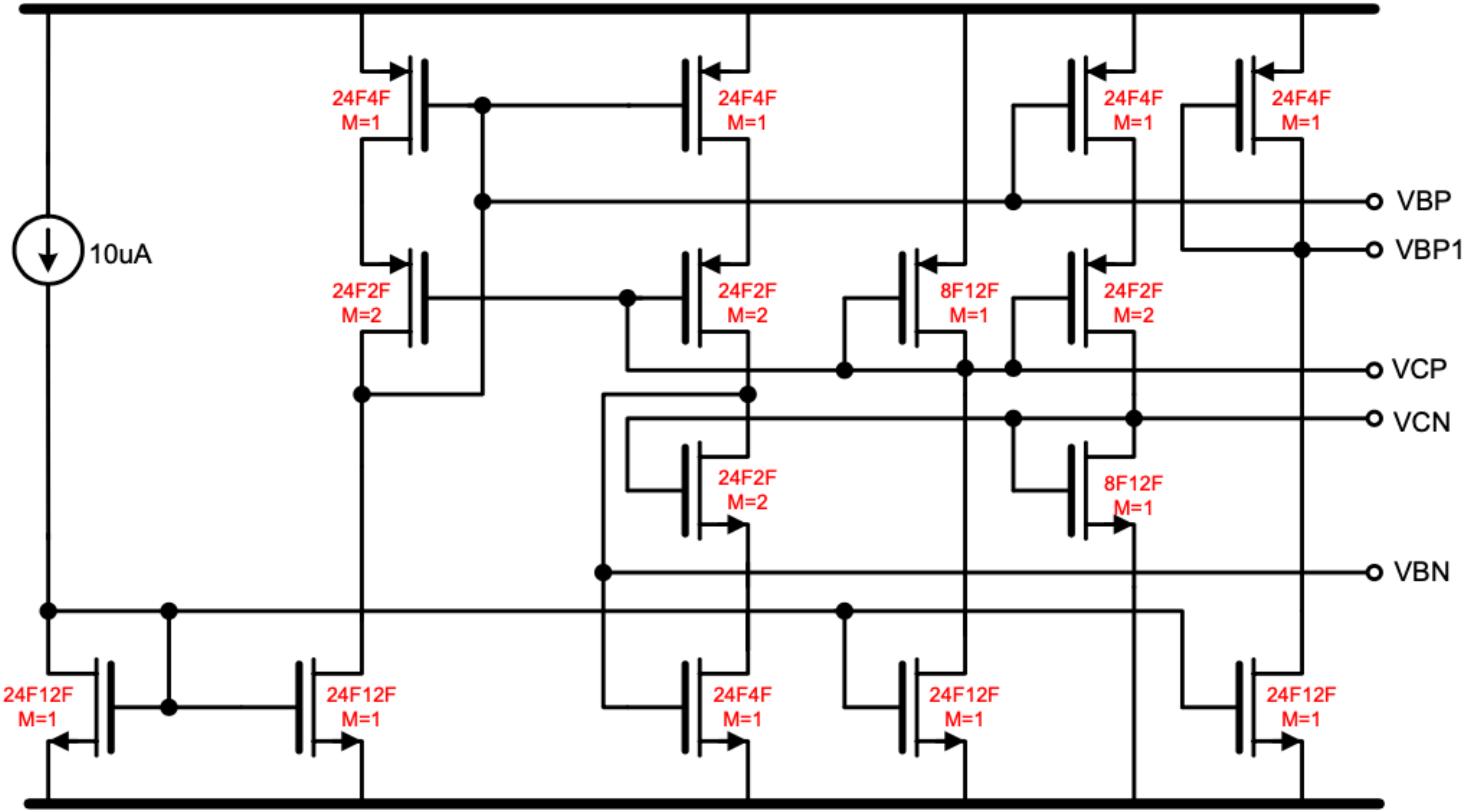
$$\sigma_{tot}^2 = \sigma_1^2 + \sigma_2^2$$

Sub-circuits for SC-circuits

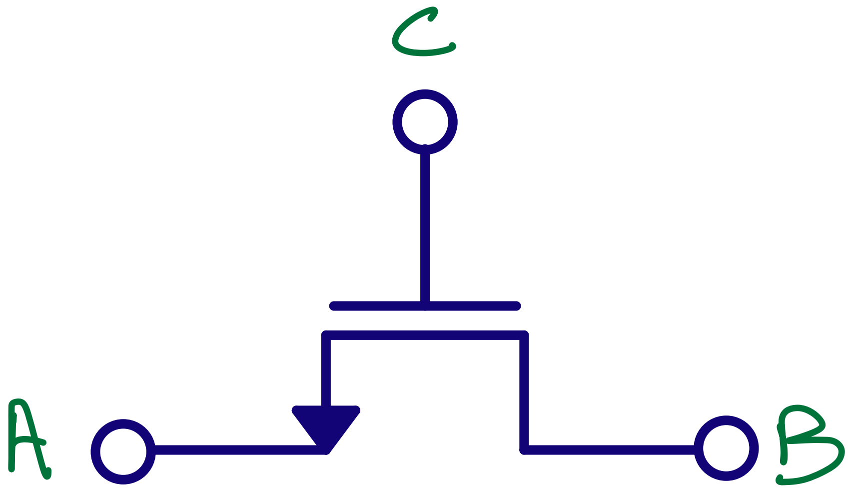
OTA

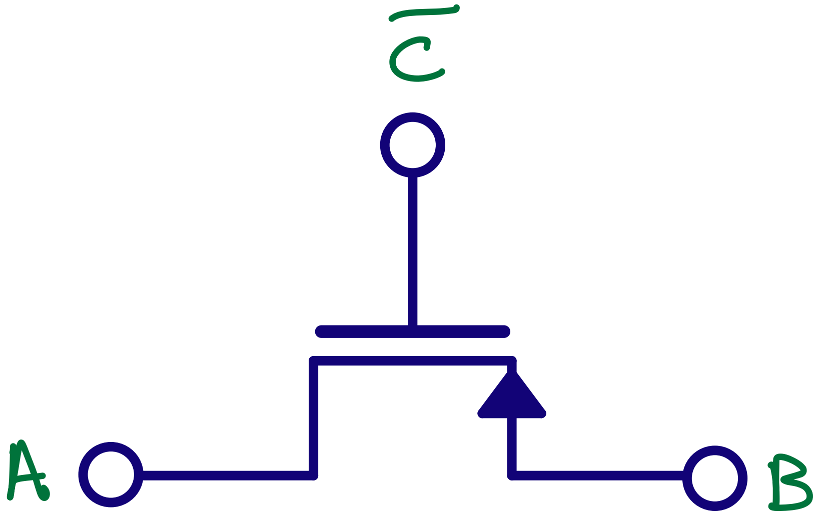


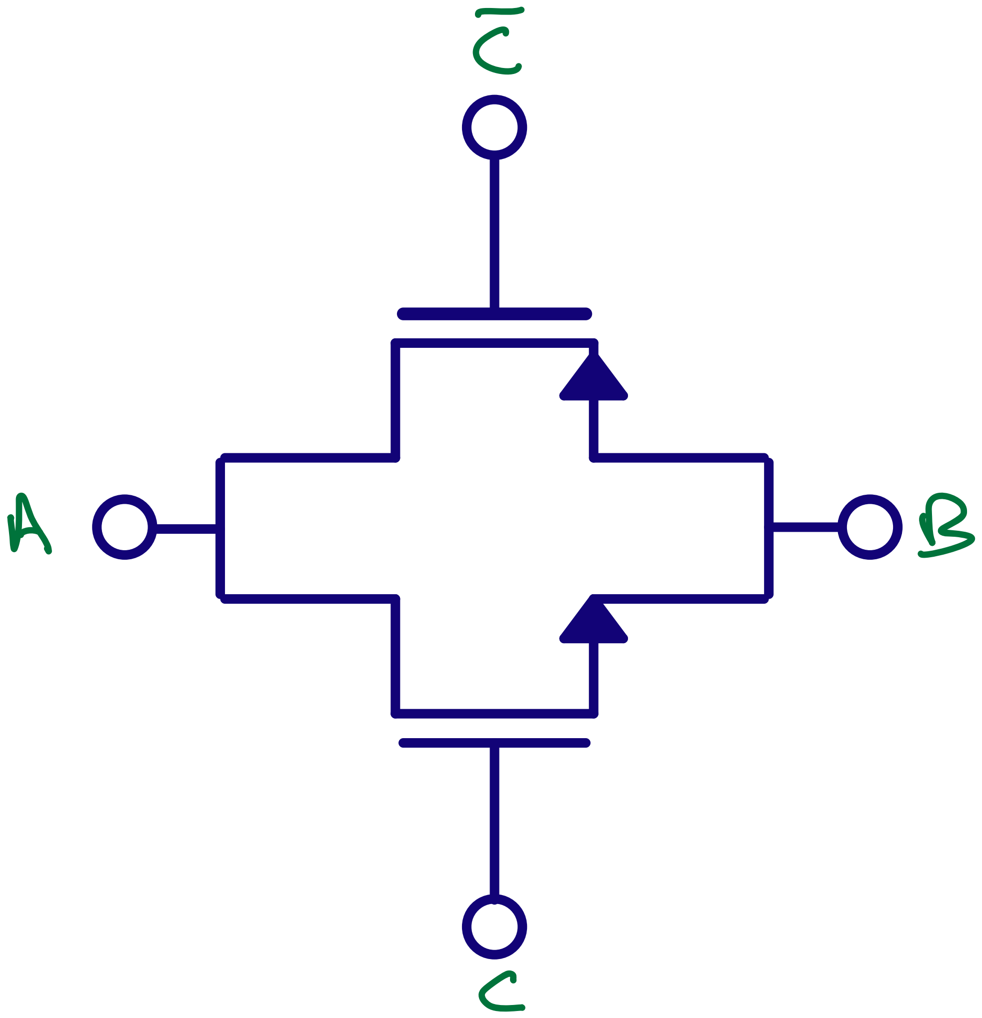
F = minimum transistor gate length. For example 24F4F => W = 24 x min gate, L= 4 x min gate

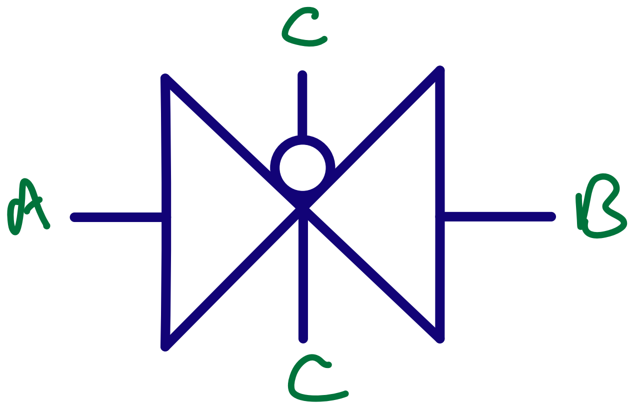

24F4F
M=1
24F4F
M=1
24F4F
M=1
24F4F
M=1
VBP
VBP1
10uA
24F2F
M=2
24F2F
M=2
8F12F
M=1
24F2F
M=2
VCP
VCN
24F2F
M=2
8F12F
M=1
VBN
24F12F
M=1
24F12F
M=1
24F4F
M=1
24F12F
M=1
24F12F
M=1

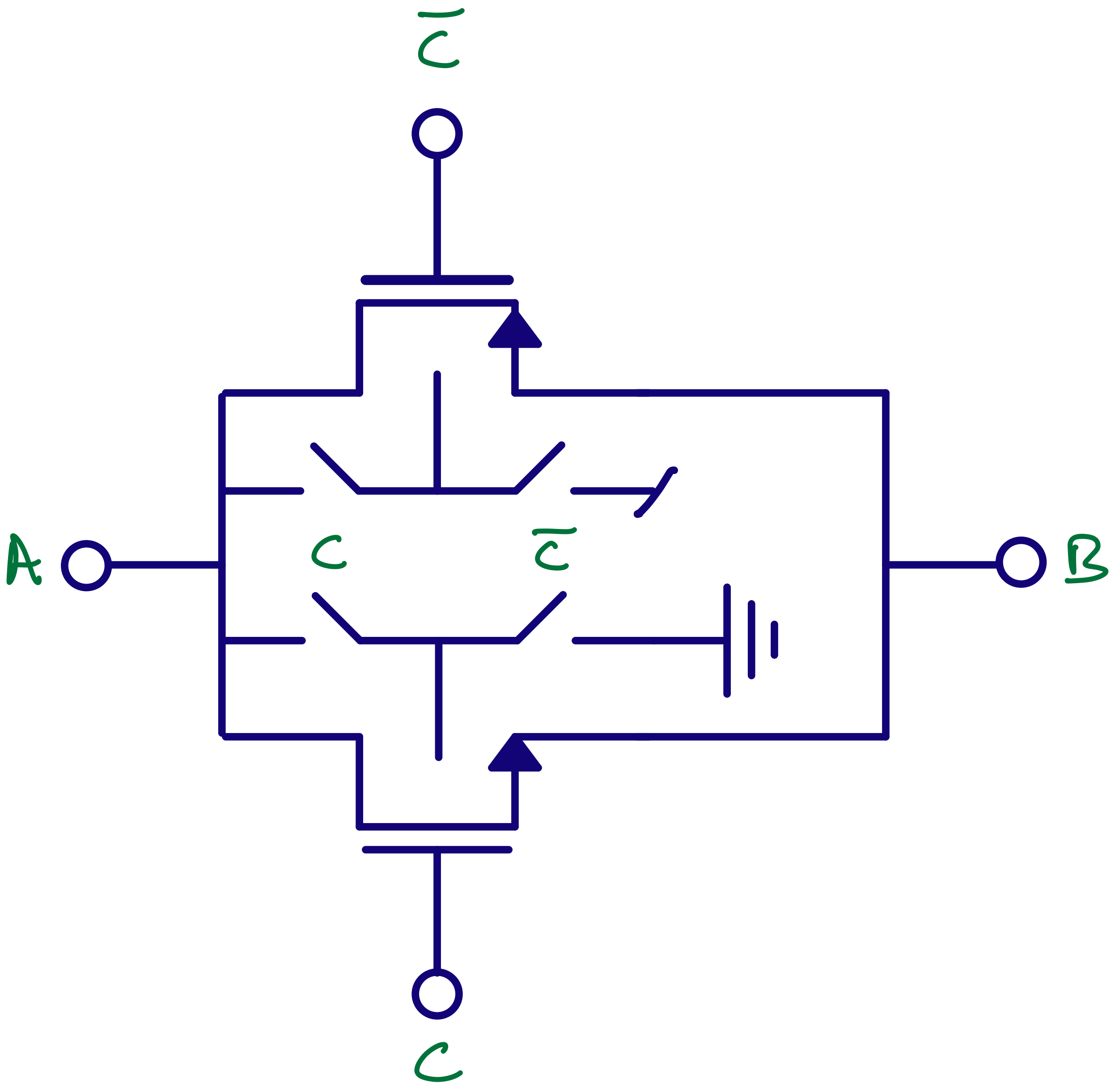
Switches

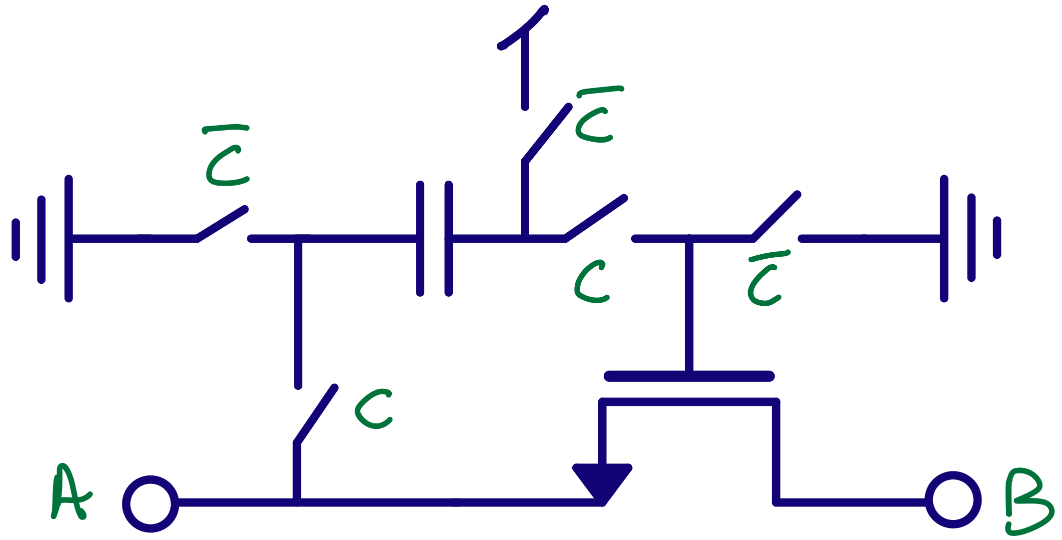

C
A
B


$\overline{C}$
A
B

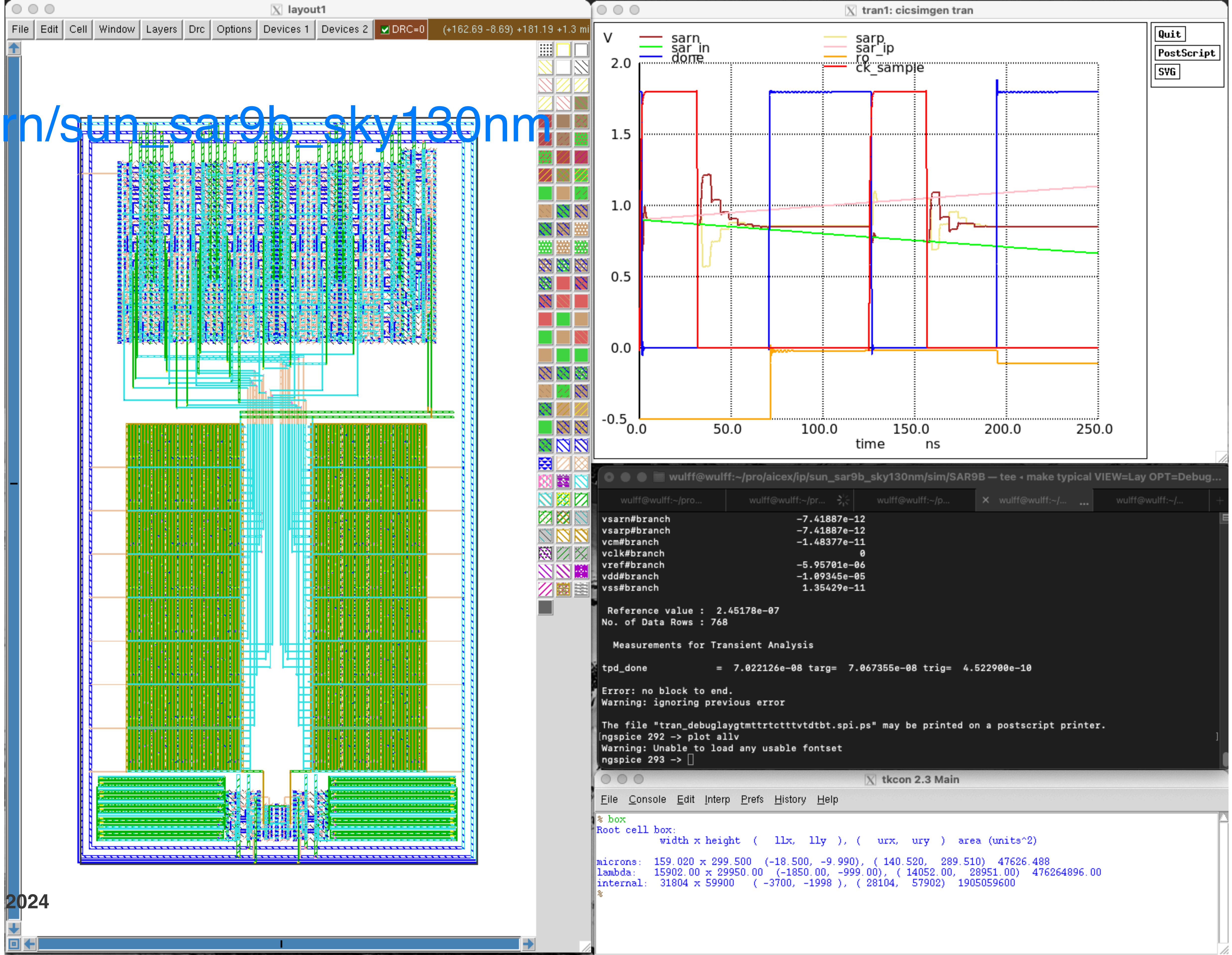

$\overline{C}$
A
B
C

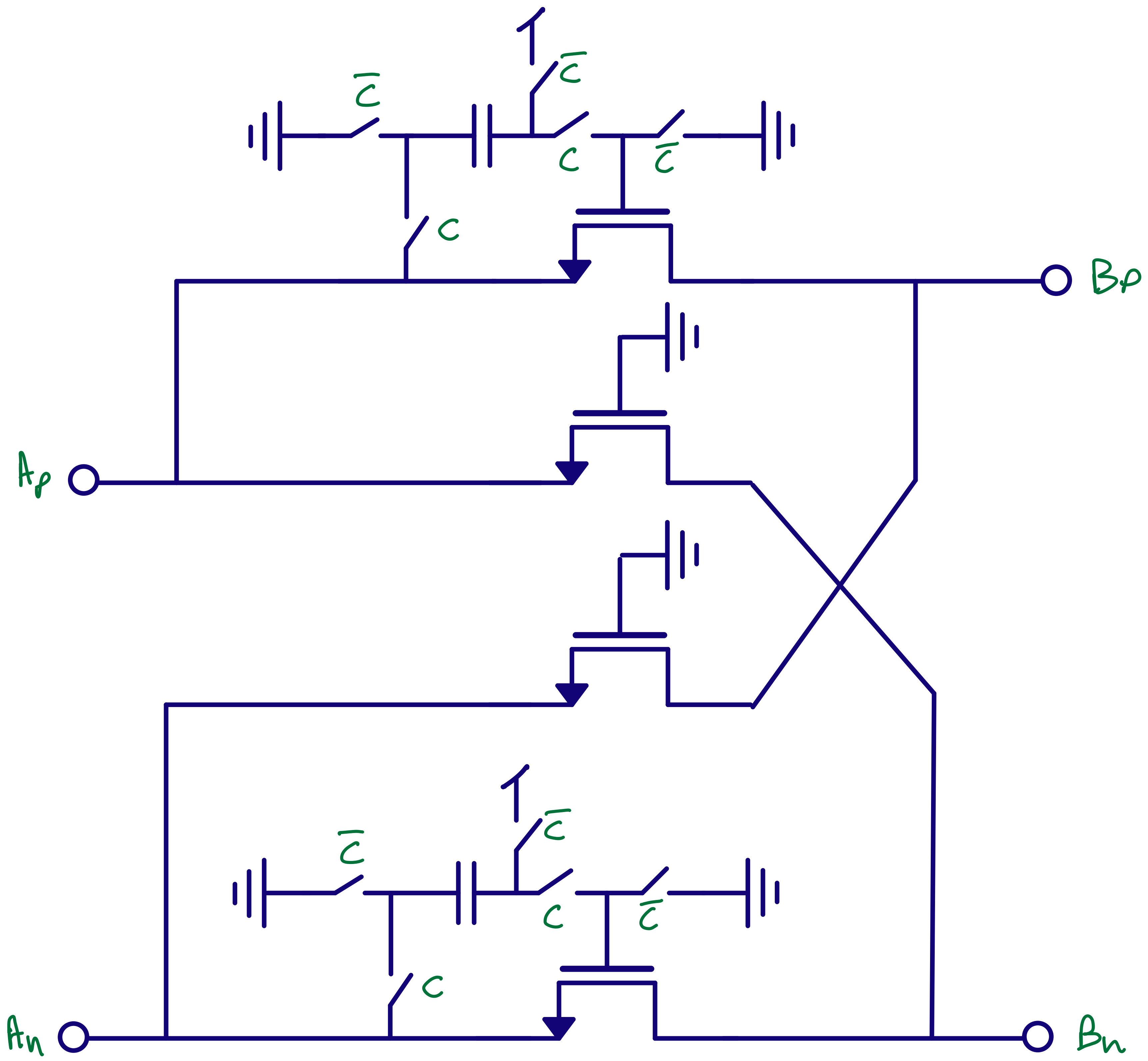

C
A
B
C


C̄
A
C
C̄
B
C


C̄
C̄
C
C̄
C
A
B

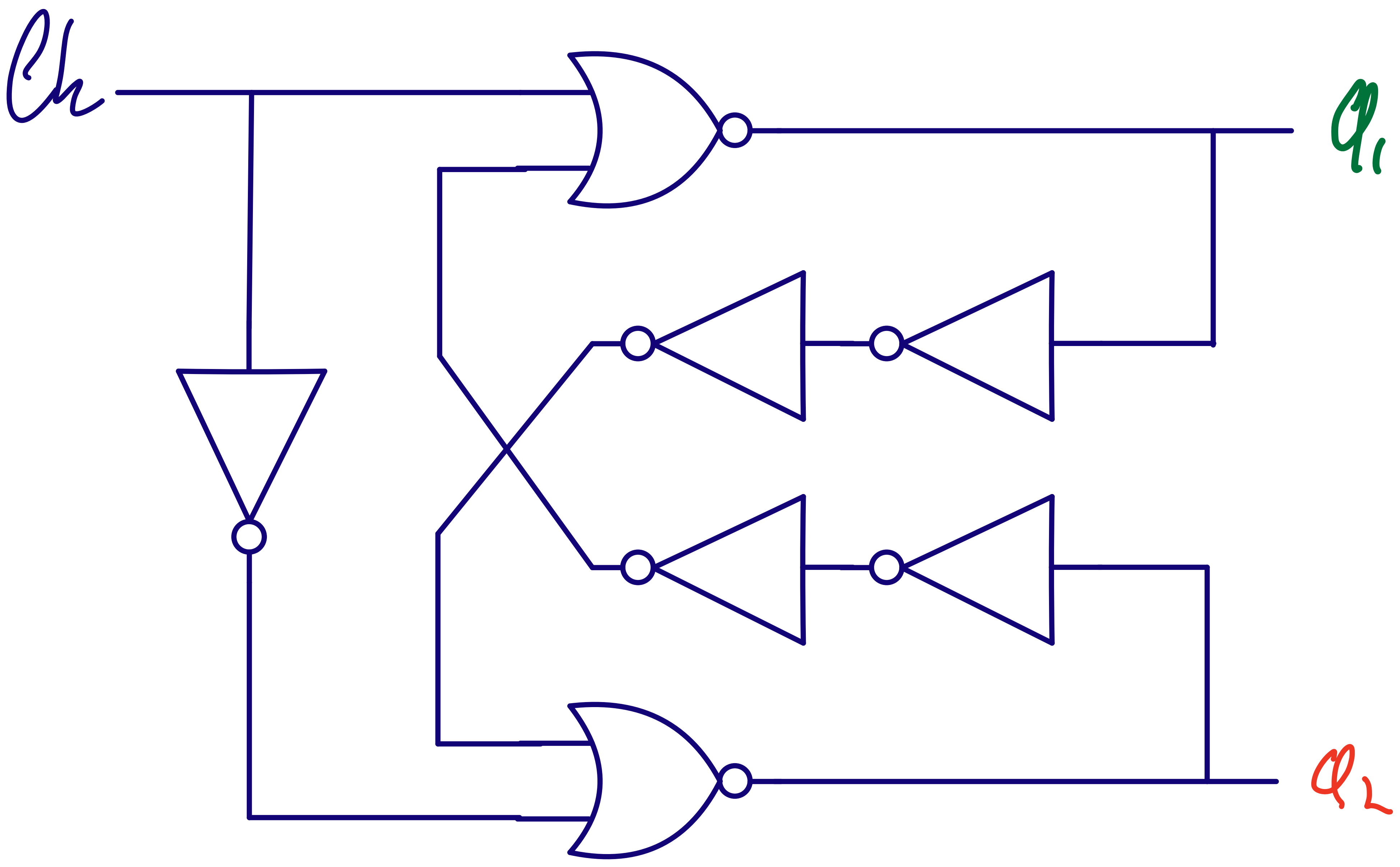
wulffern/sun_sar9b_sky130nm

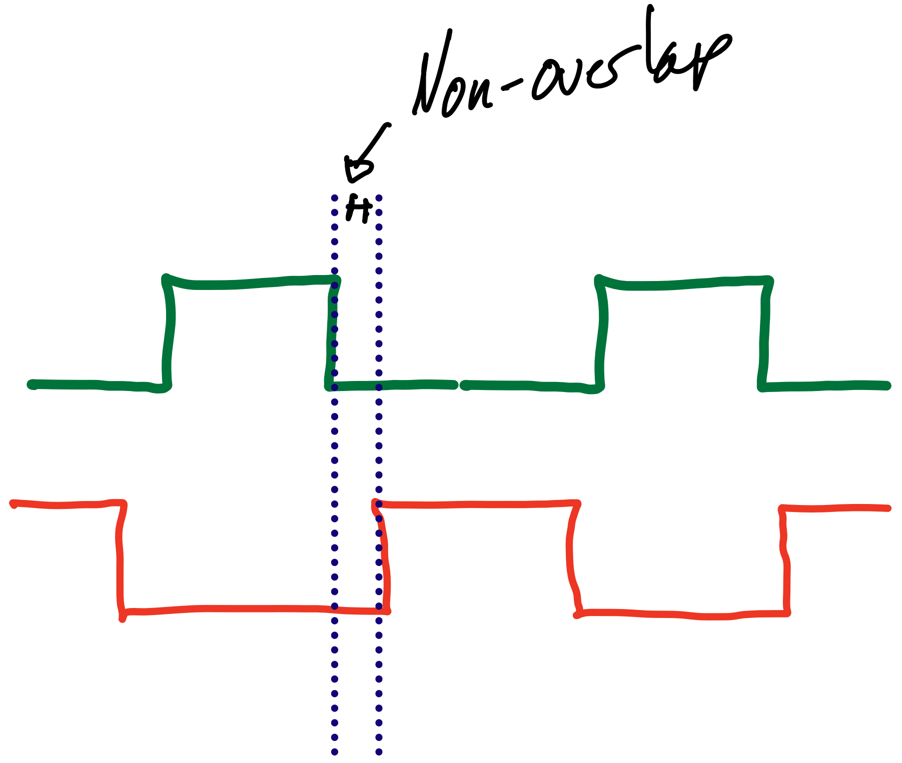




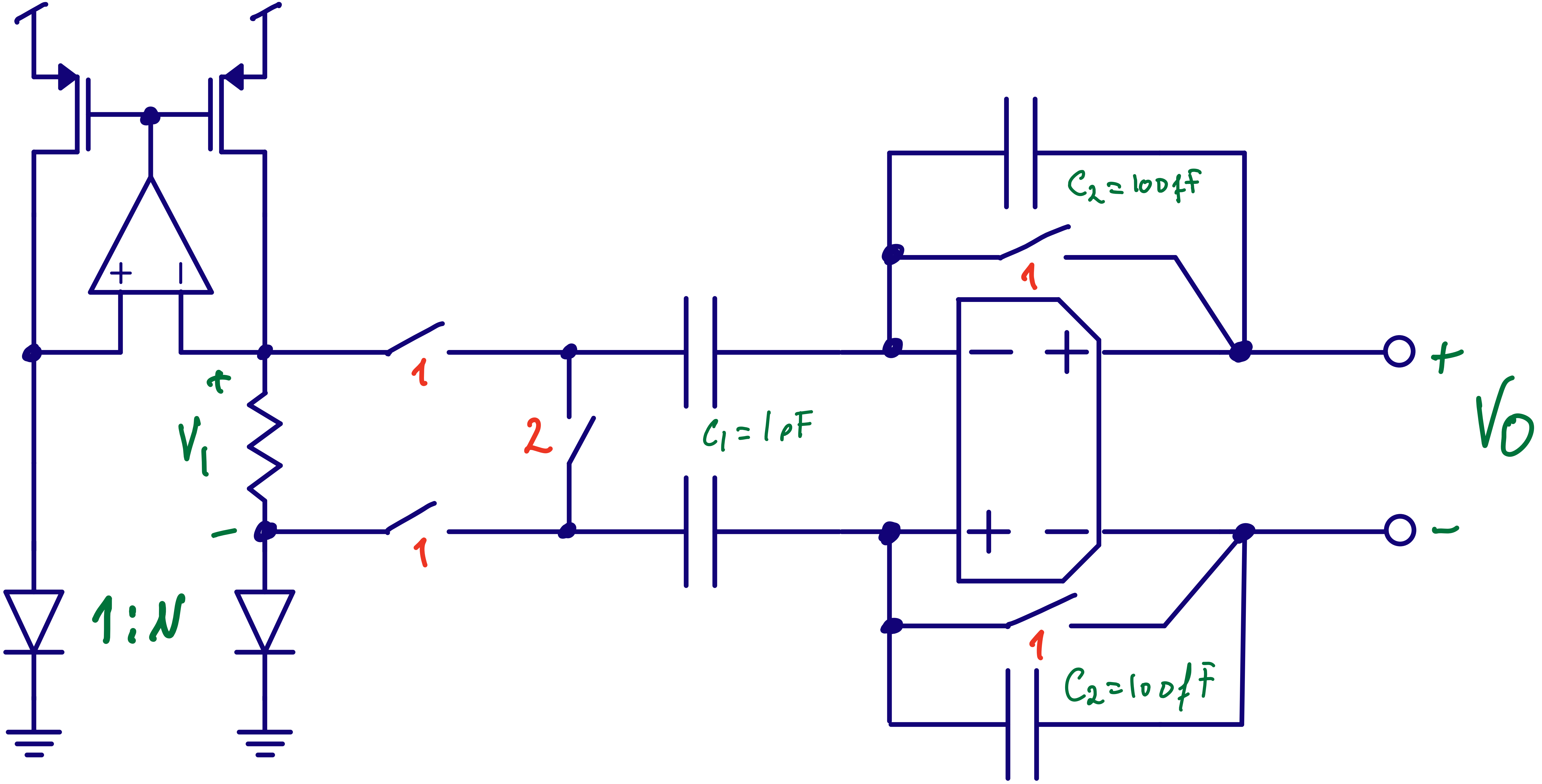
$\bar{c}$
$\bar{c}$
c
$\bar{c}$
c
Bp
Ap
$\bar{c}$
$\bar{c}$
c
$\bar{c}$
c
An
Bn

Non-overlapping clocks


Clk
ϕ_1
ϕ_2


Non-overlap

Example



+
−
+
−
+
V_1
1
1
2
$C_1 = 1pF$
$C_2 = 100fF$
1
1
$C_2 = 100fF$
+
−
V_O
1:N

Thanks!