date: 2024-02-16

TFE4188 - Lecture 6

Oversampling and Sigma-Delta ADCs

Goal for today

Understand **why** there are different ADCs

Introduction to **oversampling** and **delta-sigma** modulators

A few **examples**

1999, R. Walden: Analog-to-digital converter survey and analysis

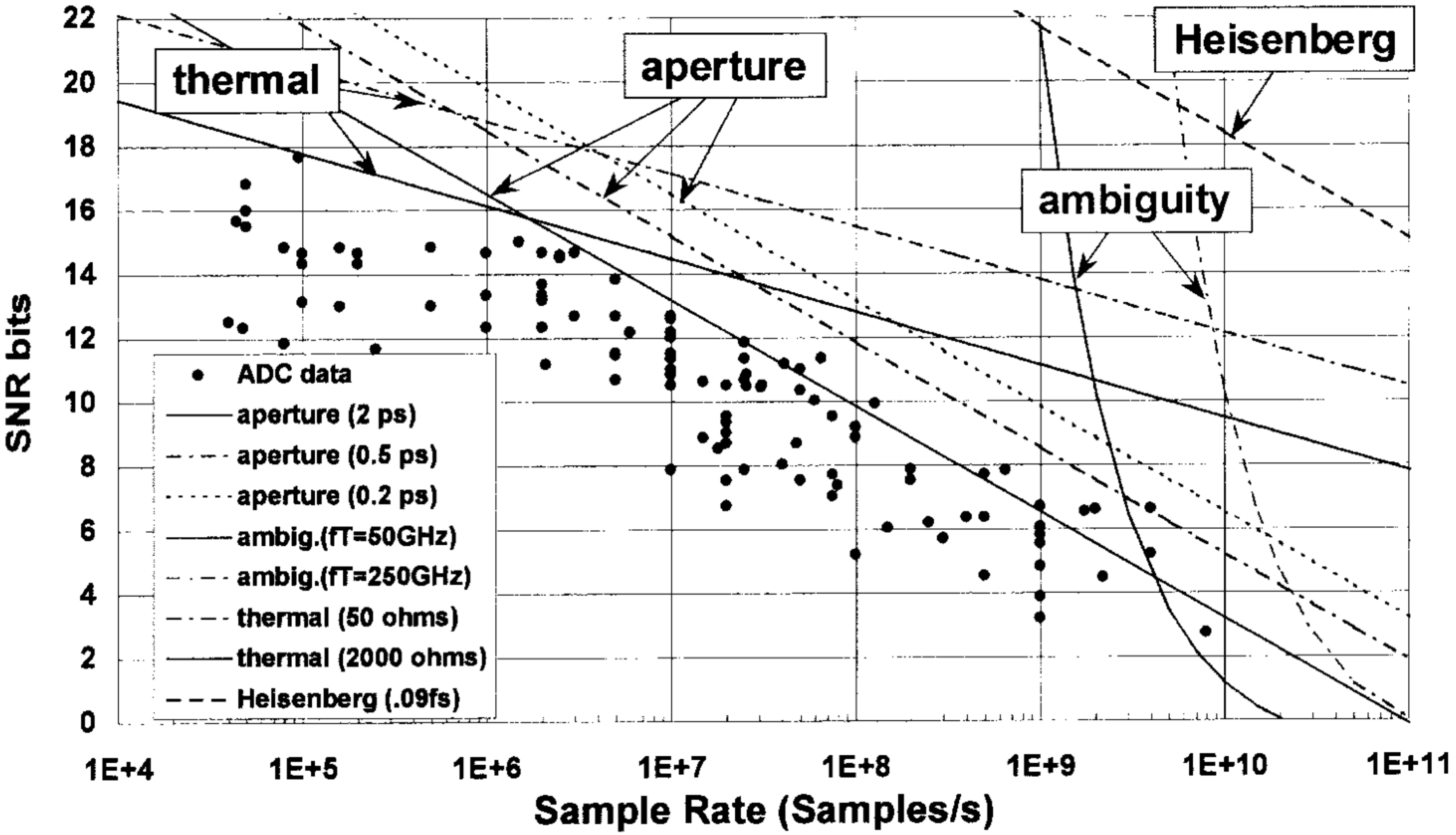
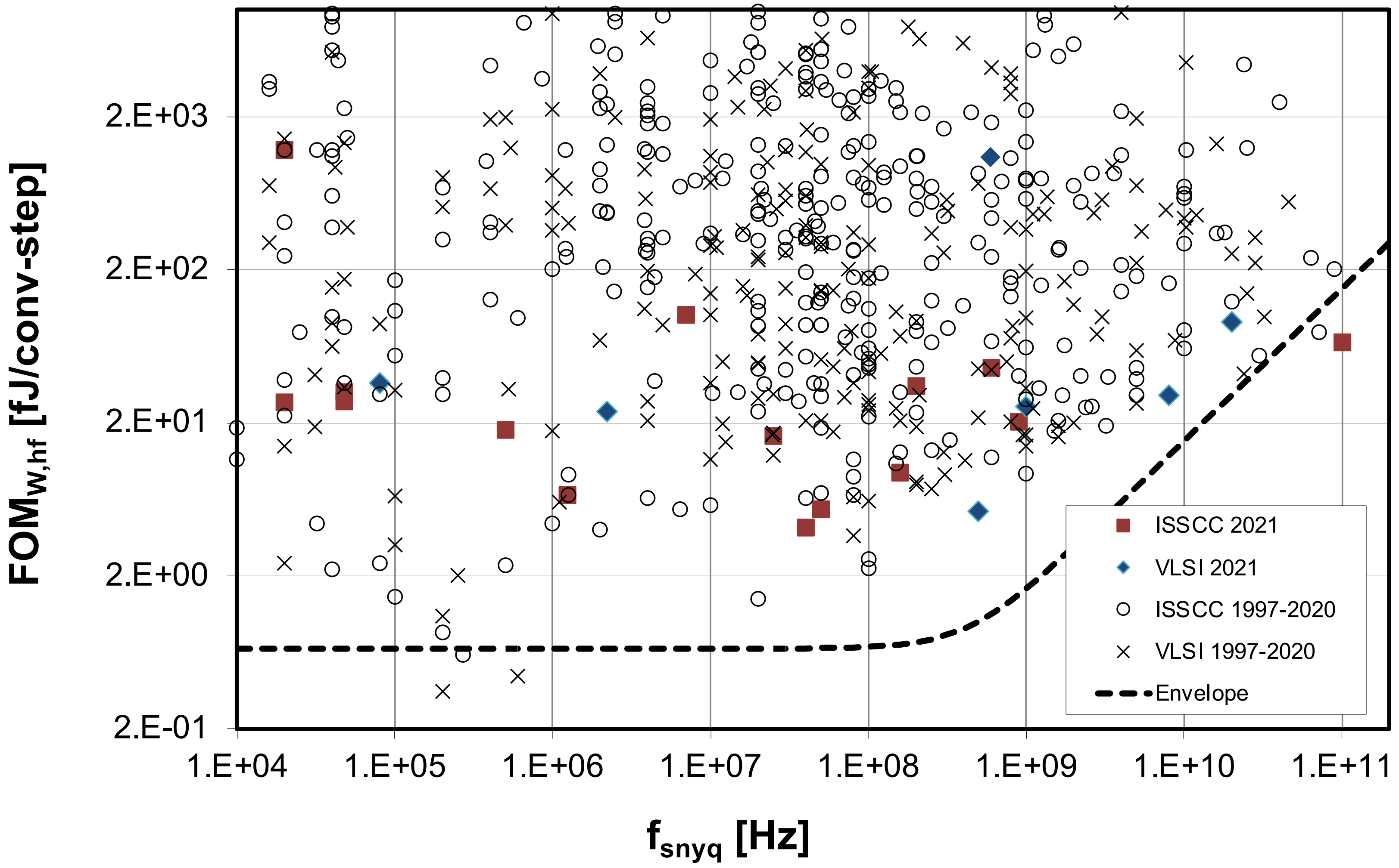


Fig. 7. Signal-to-noise ratio according to SNR-bits = (SNR(dB) − 1.76)/6.02. Three sets of curves show performance limiters due to thermal noise, aperture uncertainty, and comparator ambiguity. The Heisenberg limit is also displayed.



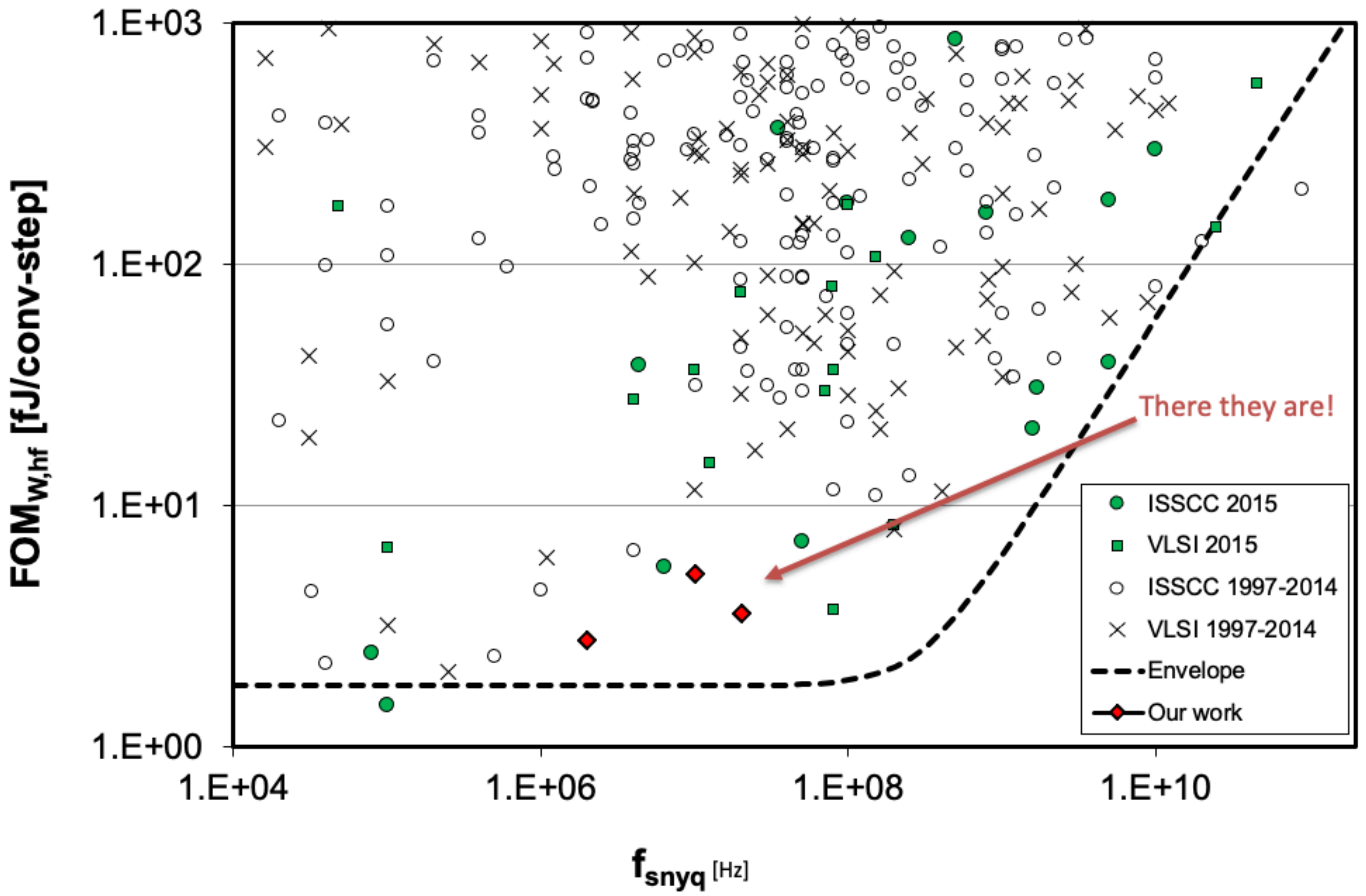
B. Murmann, ADC Performance Survey 1997-2023

$$FOM_W = \frac{P}{2^B f_s}$$

Below 10 fJ/conv.step is good.

Below 1 fJ/conv.step is extreme.

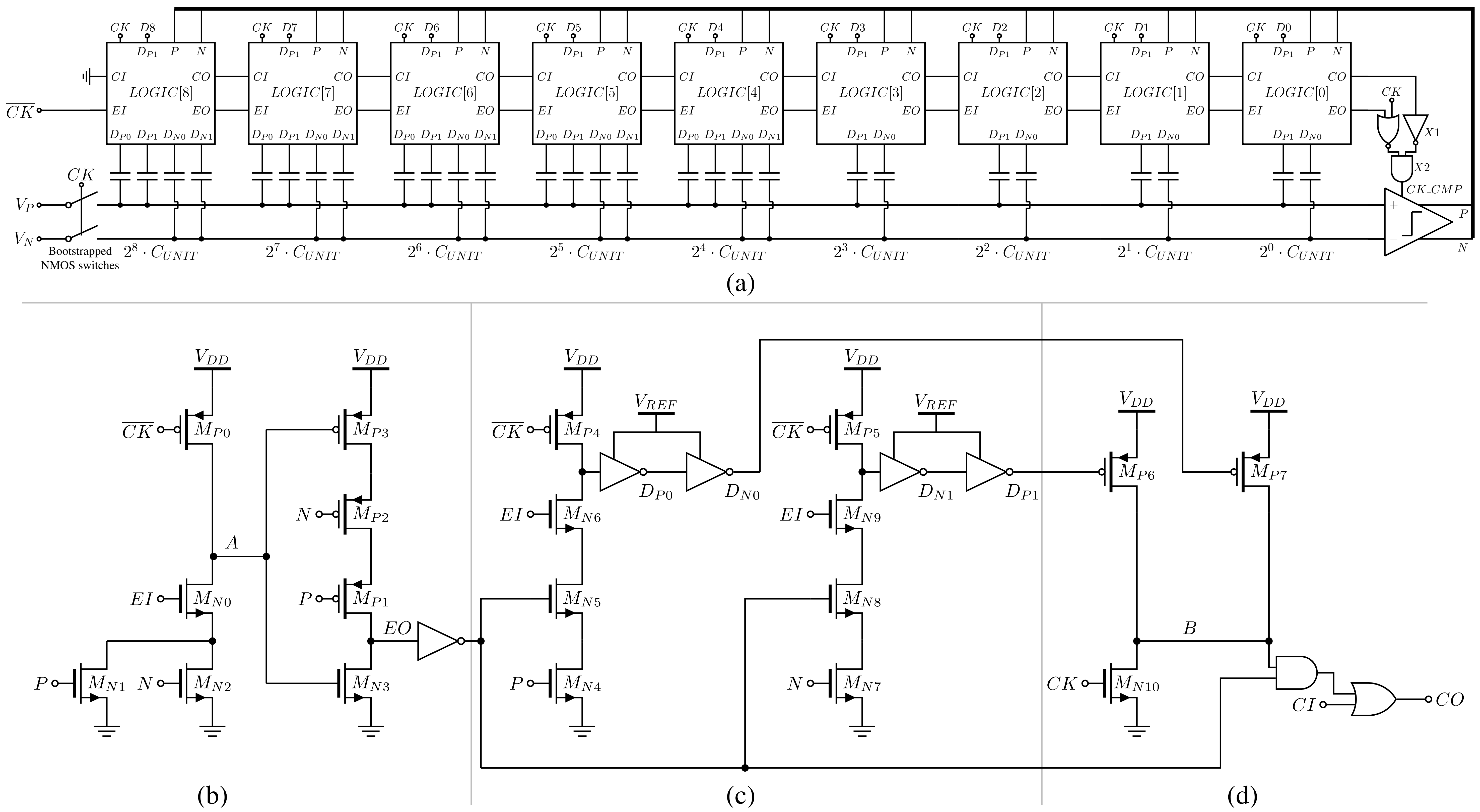
People from NTNU have made some of the worlds best ADCs

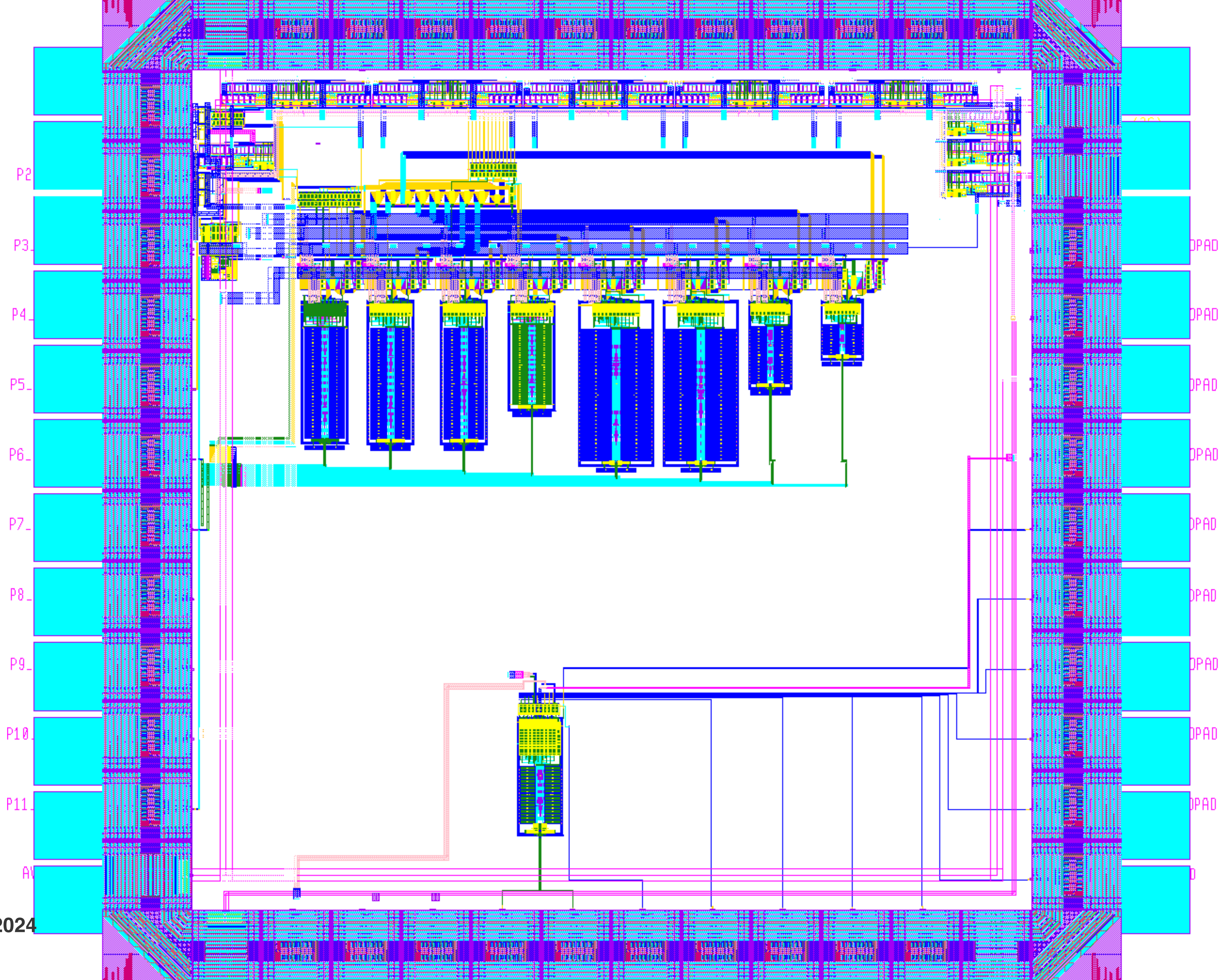


[1] A Compiled 9-bit 20-MS/s 3.5-fJ/ conv.step SAR ADC in 28-nm FDSOI for Bluetooth Low Energy Receivers

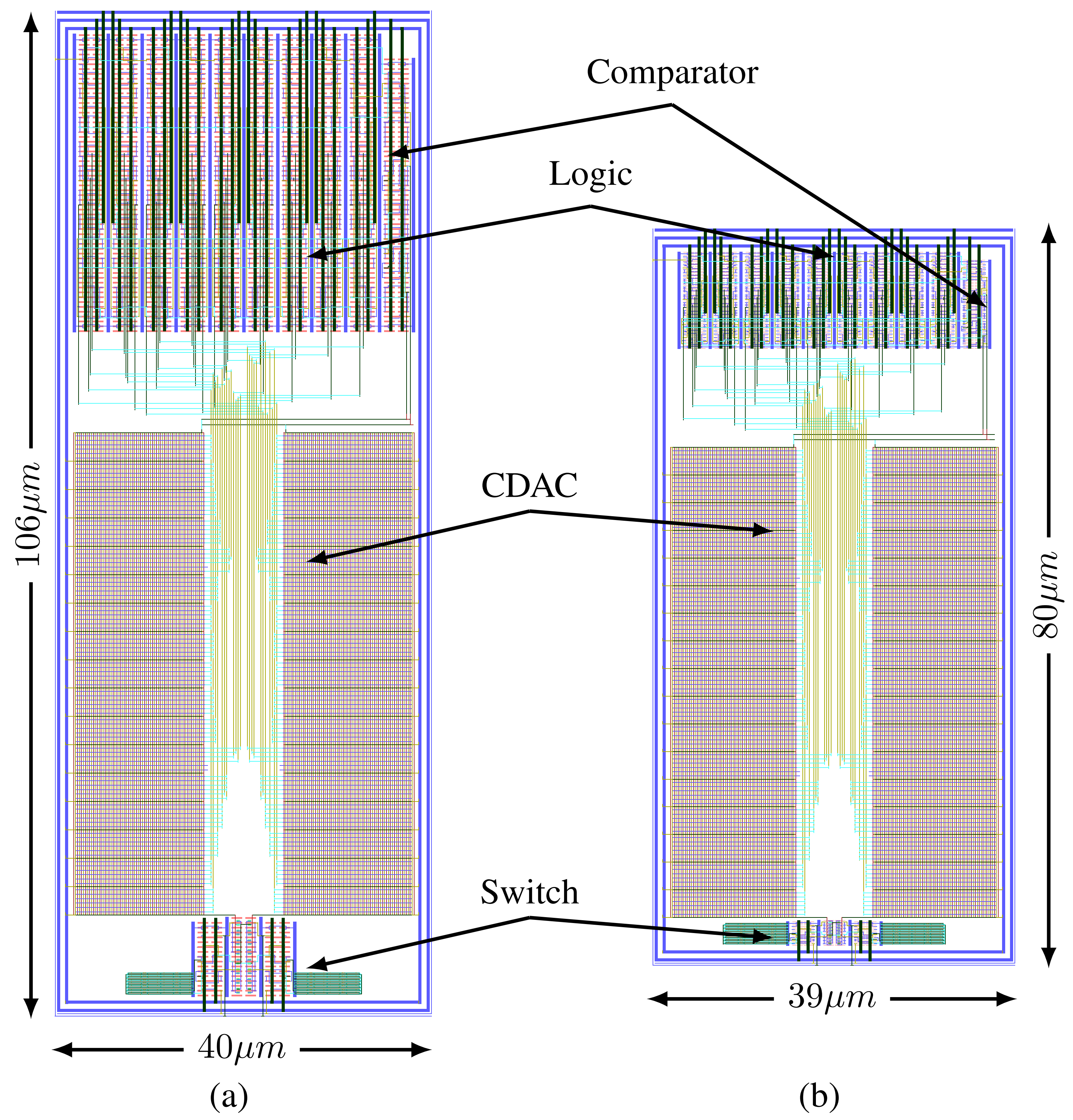
[2] A 68 dB SNDR Compiled Noise-Shaping SAR ADC With On-Chip CDAC Calibration

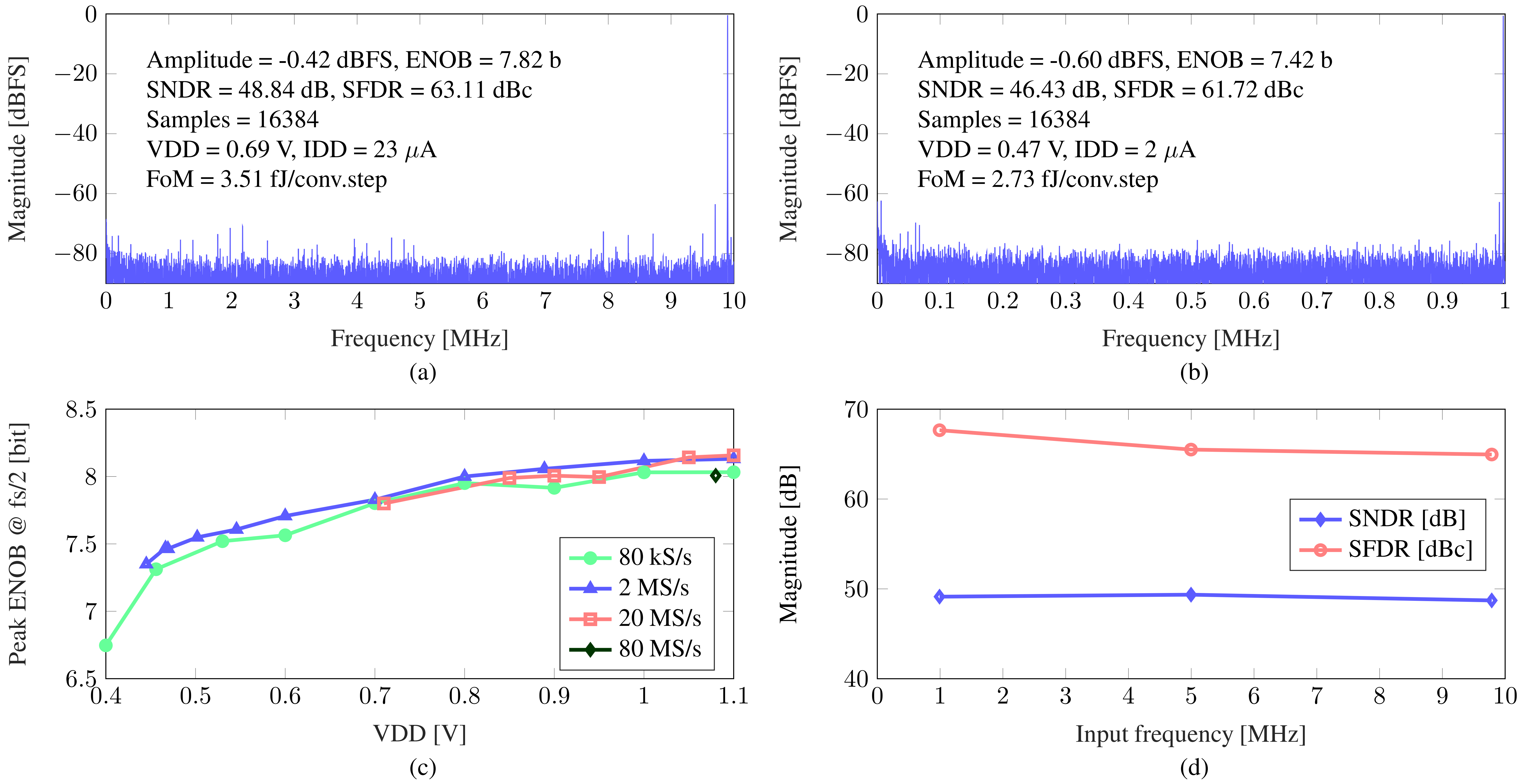
What makes a state-of-the-art ADC


CK
D8
D7
D6
D5
D4
D3
D2
D1
D0
D_{P1} P N
CI CO
EI EO
LOGIC[8]
LOGIC[7]
LOGIC[6]
LOGIC[5]
LOGIC[4]
LOGIC[3]
LOGIC[2]
LOGIC[1]
LOGIC[0]
D_{P0} D_{P1} D_{N0} D_{N1}
$\overline{CK}$
V_P
V_N
Bootstrapped NMOS switches
$2^8 \cdot C_{UNIT}$
$2^7 \cdot C_{UNIT}$
$2^6 \cdot C_{UNIT}$
$2^5 \cdot C_{UNIT}$
$2^4 \cdot C_{UNIT}$
$2^3 \cdot C_{UNIT}$
$2^2 \cdot C_{UNIT}$
$2^1 \cdot C_{UNIT}$
$2^0 \cdot C_{UNIT}$
X1
X2
CK_CMP
(a)
V_{DD}
V_{REF}
M_{P0} M_{P1} M_{P2} M_{P3} M_{P4} M_{P5} M_{P6} M_{P7}
M_{N0} M_{N1} M_{N2} M_{N3} M_{N4} M_{N5} M_{N6} M_{N7} M_{N8} M_{N9} M_{N10}
A
B
EI
EO
CI
CO
D_{P0} D_{N0} D_{N1} D_{P1}
(b)
(c)
(d)

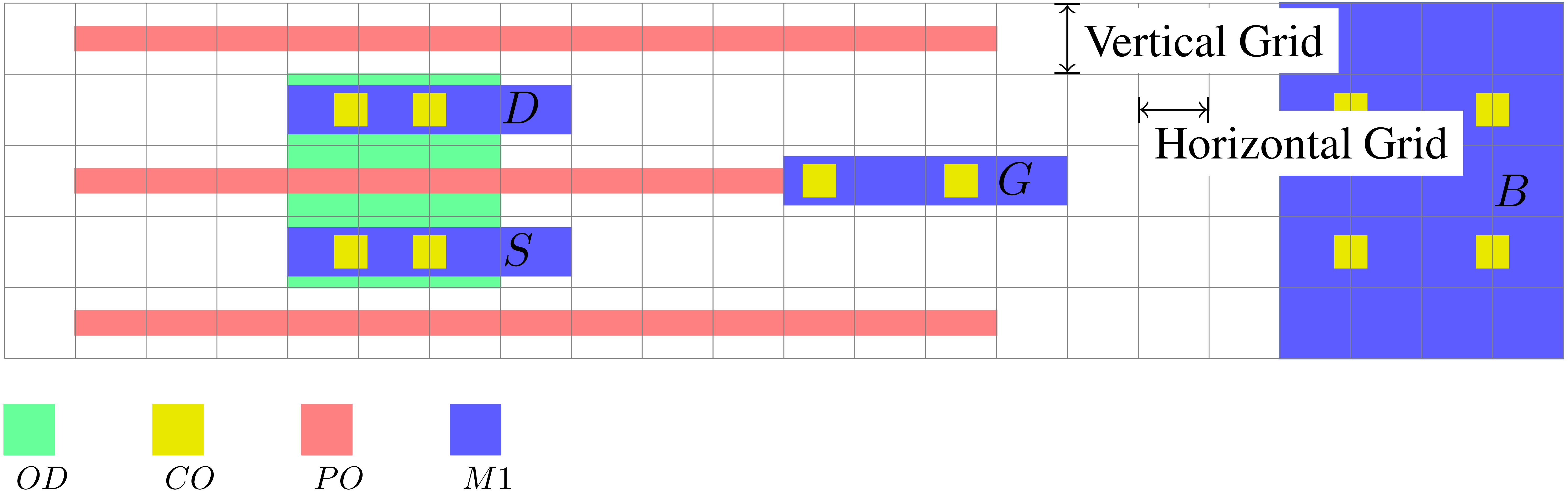


P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
PAD
PAD
PAD
PAD
PAD
PAD
PAD
PAD
PAD

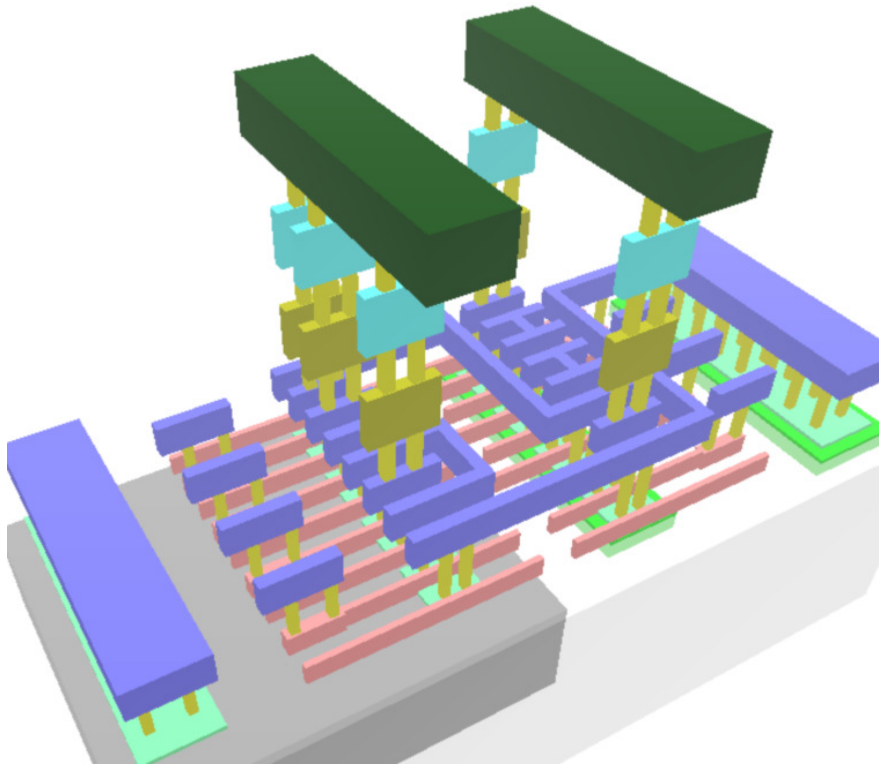

Comparator
Logic
CDAC
Switch
106μm
40μm
80μm
39μm
(a)
(b)


Amplitude = -0.42 dBFS, ENOB = 7.82 b
SNDR = 48.84 dB, SFDR = 63.11 dBc
Samples = 16384
VDD = 0.69 V, IDD = 23 μA
FoM = 3.51 fJ/conv.step
Magnitude [dBFS]
Frequency [MHz]
(a)
Amplitude = -0.60 dBFS, ENOB = 7.42 b
SNDR = 46.43 dB, SFDR = 61.72 dBc
Samples = 16384
VDD = 0.47 V, IDD = 2 μA
FoM = 2.73 fJ/conv.step
Magnitude [dBFS]
Frequency [MHz]
(b)
Peak ENOB @ fs/2 [bit]
VDD [V]
80 kS/s
2 MS/s
20 MS/s
80 MS/s
(c)
Magnitude [dB]
Input frequency [MHz]
SNDR [dB]
SFDR [dBc]
(d)

	Weaver [5]	Harpe [9]	Patil [10]	Liu [11]	This work	
Technology (nm)	90	90	28 FDSOI	28	28 FDSOI	
Fsample (MS/s)	21	2	No sampling	100	2	20
Core area (mm^2)	0.18	0.047	0.0032	0.0047	0.00312	
SNDR (dB)	34.61	57.79	40	64.43	46.43	48.84
SFDR (dBc)	40.81	72.33	30	75.42	61.72	63.11
ENOB (bits)	5.45	6.7 - 9.4	6.35	10.41	7.42	7.82
Supply (V)	0.7	0.7	0.65	0.9	0.47	0.69
Pwr (μW)	1110	1.64 -3.56	24	350	0.94	15.87
Compiled	Yes	No	No	No	Yes	
FoM (fJ/c.step)	838	2.8 - 6.6	3.7	2.6	2.7	3.5



Vertical Grid
Horizontal Grid
D
G
S
B
OD
CO
PO
M1



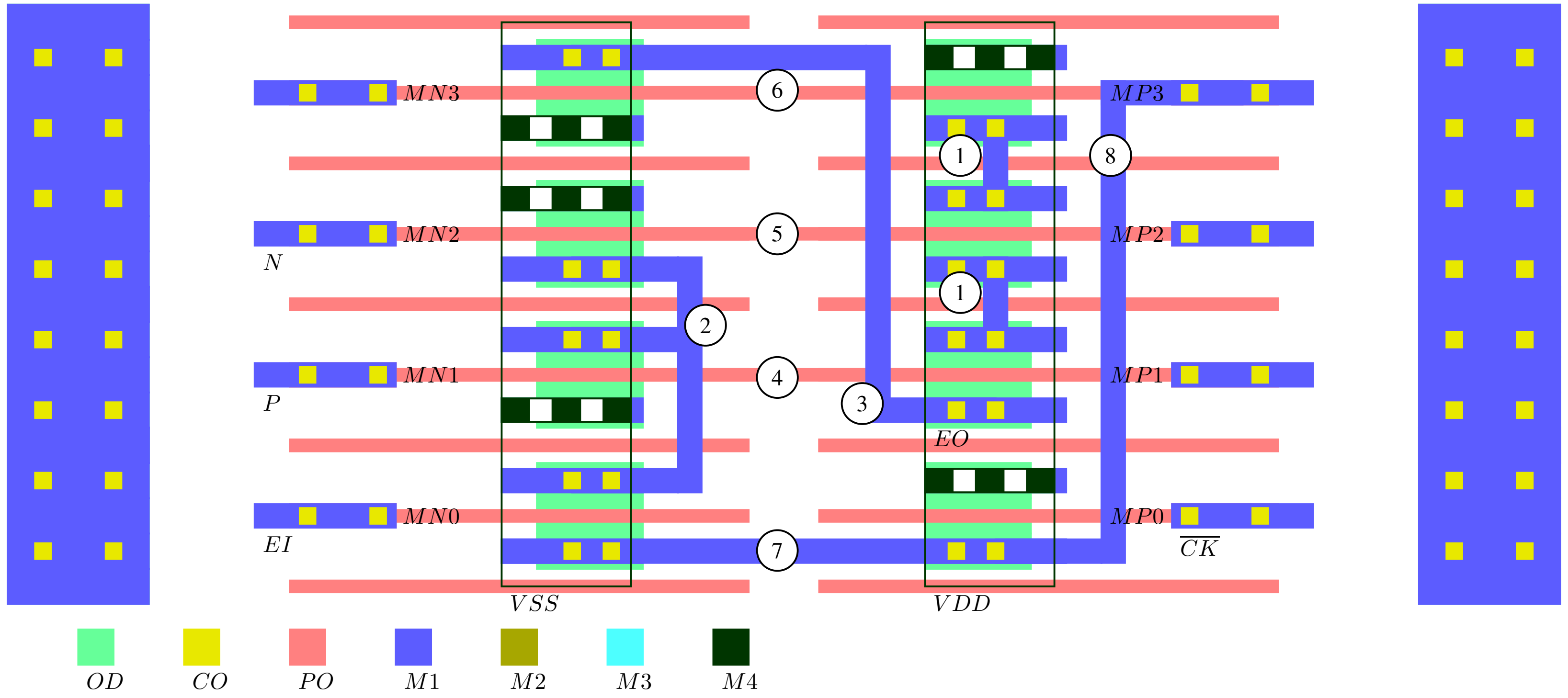
(a)

```
.SUBCKT SAREMX1_CV P  N EI EO CK_N AVDD AVSS
  MN0 N3   EI   A    AVSS NCHDL
  MN1 N3   P    AVSS AVSS NCHDL
  MN2 AVSS N    N3   AVSS NCHDL
  MN3 EO   A    AVSS AVSS NCHDL
  MP0 AVDD CK_N A    AVSS PCHDL
  MP1 N2   P    EO   AVSS PCHDL
  MP2 N1   N    N2   AVSS PCHDL
  MP3 AVDD A    N1   AVSS PCHDL
.ENDS
```

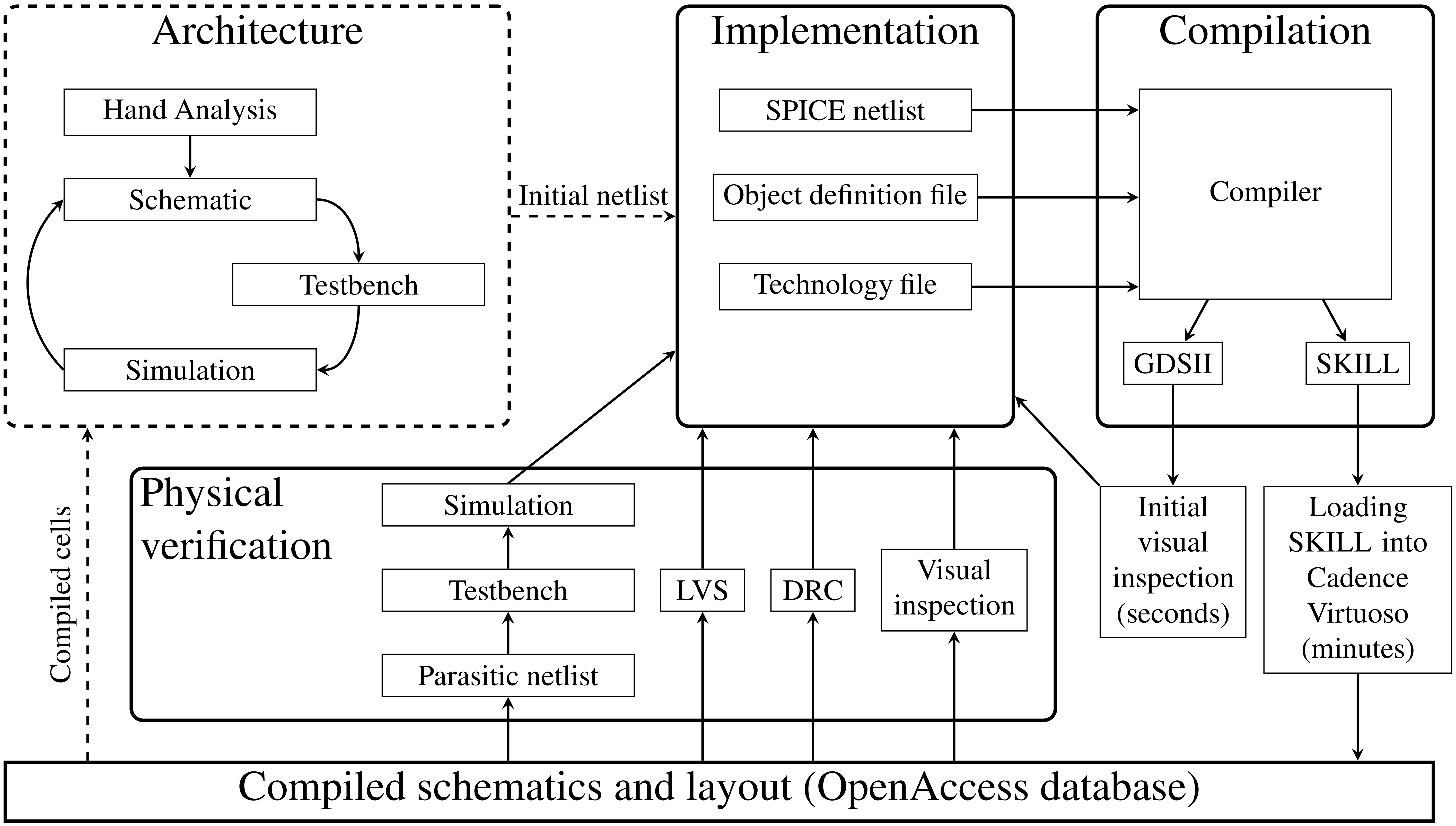
(b)

```
{ "name": "SAREMX1_CV",
  "class" : "Layout::LayoutDigitalCell",
  "addConnectivityRoutes": [
      ["M1","N1|N2","||",""],       1
      ["M1","N3","-|",""],          2
      ["M1","EO","--|-","onTopR"]   3
        ],
    "addDirectedRoutes" : [
      ["PO","P","MN1:G-MP1:G"],     4
      ["PO","N","MN2:G-MP2:G"],     5
      ["PO","A","MN3:G-MP3:G"],     6
      ["M1","A","MN0:S-MP0:S"],     7
      ["M1","A","MP0:S-|--MP3:G"]   8
      ]
    }
}
```

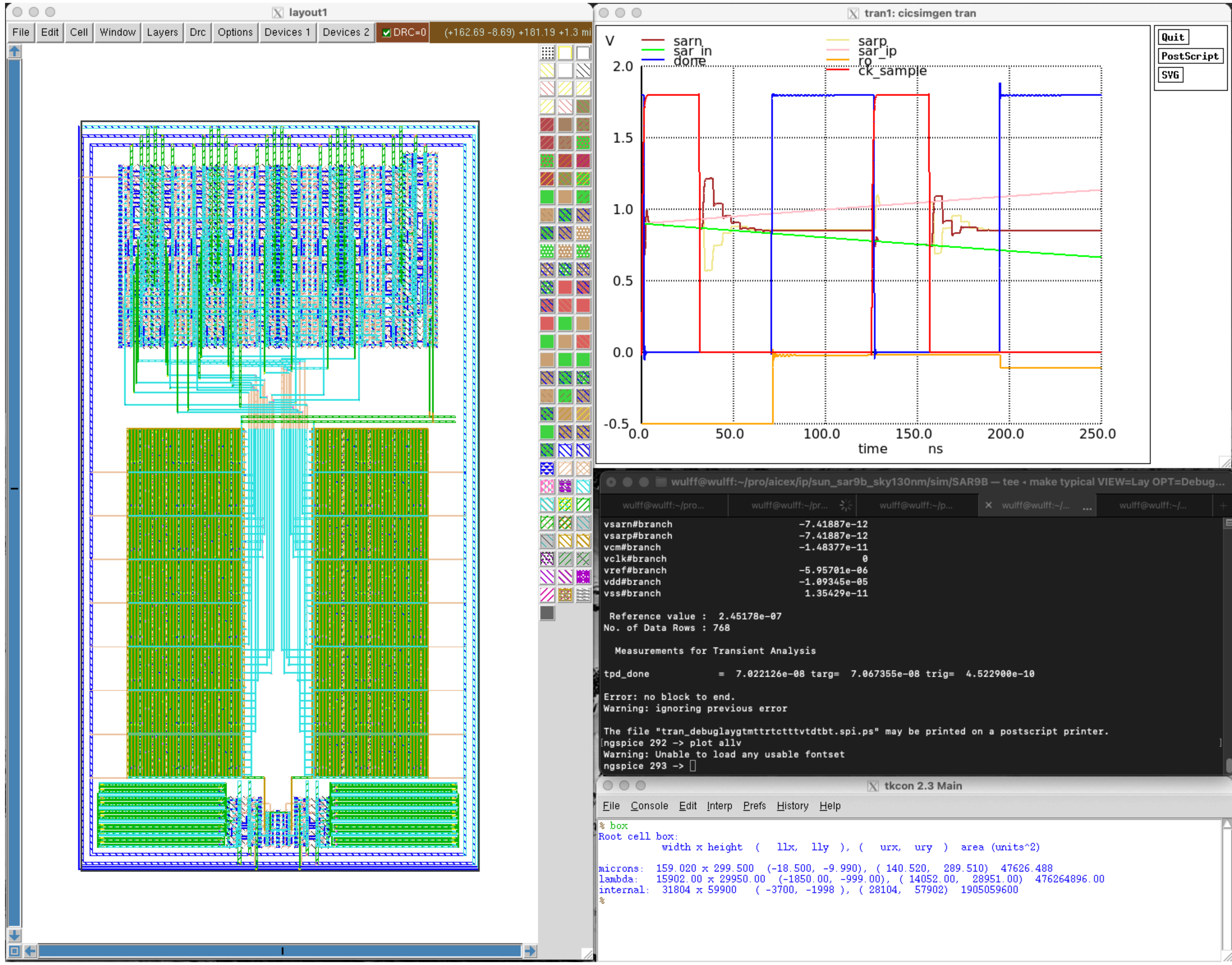
(c)

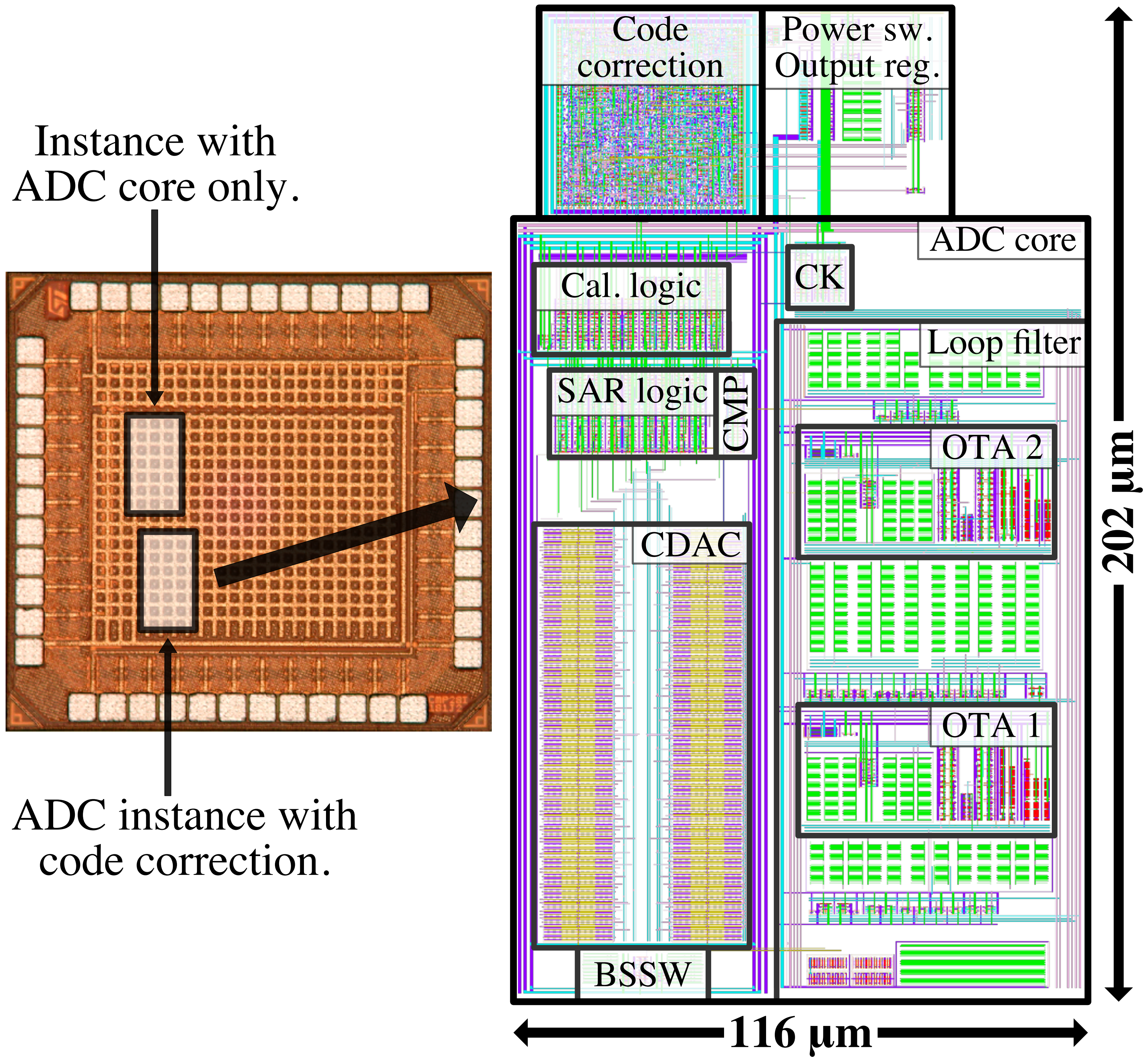


(d)


Architecture
Hand Analysis
Schematic
Testbench
Simulation
Initial netlist
Implementation
SPICE netlist
Object definition file
Technology file
Compilation
Compiler
GDSII
SKILL
Compiled cells
Physical verification
Simulation
Testbench
Parasitic netlist
LVS
DRC
Visual inspection
Initial visual inspection (seconds)
Loading SKILL into Cadence Virtuoso (minutes)
Compiled schematics and layout (OpenAccess database)

SUN_SAR9B_SKY130NM

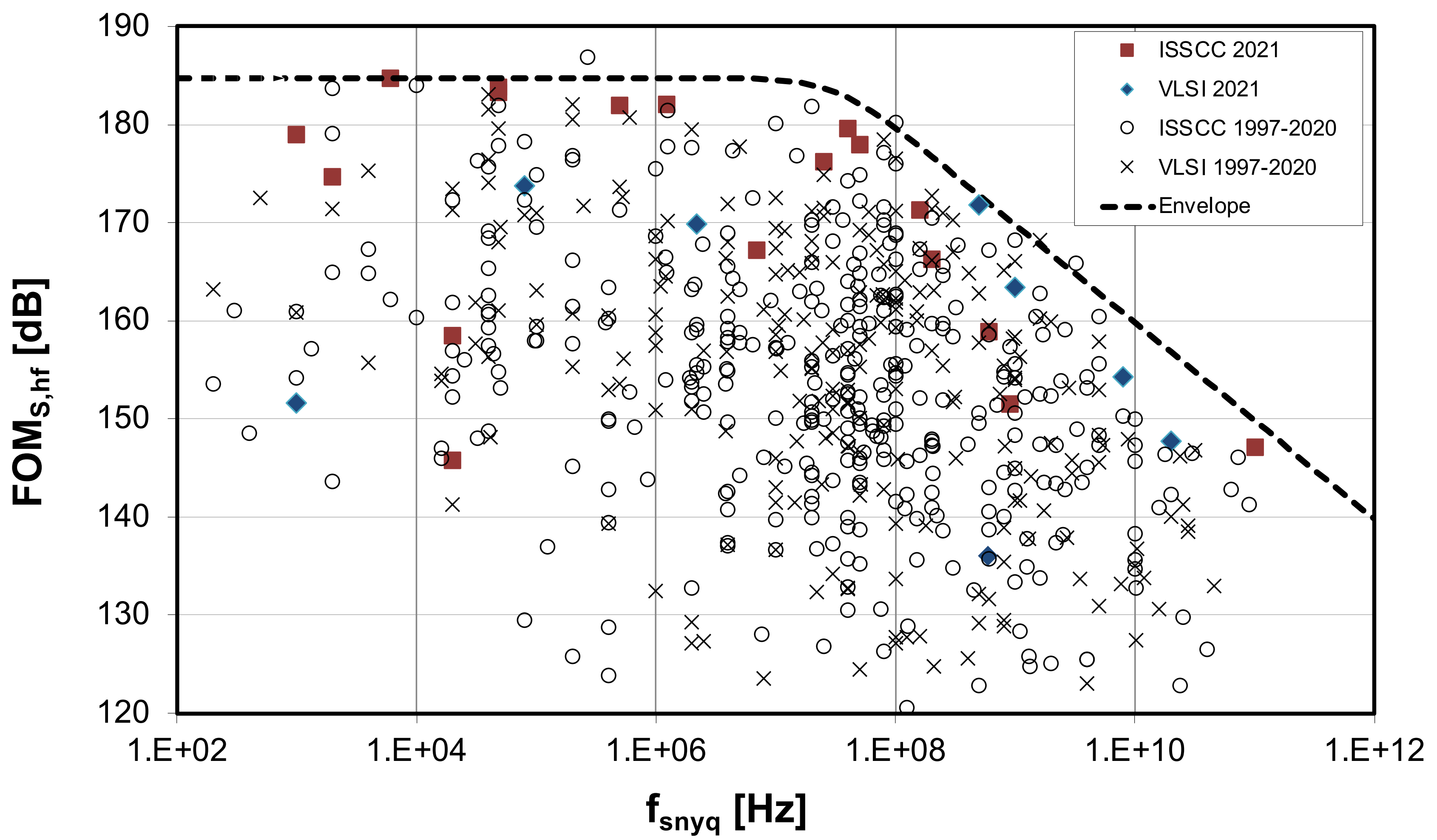



Instance with
ADC core only.
ADC instance with
code correction.
Code
correction
Power sw.
Output reg.
ADC core
CK
Cal. logic
Loop filter
SAR logic
CMP
OTA 2
CDAC
OTA 1
BSSW
202 µm
116 µm

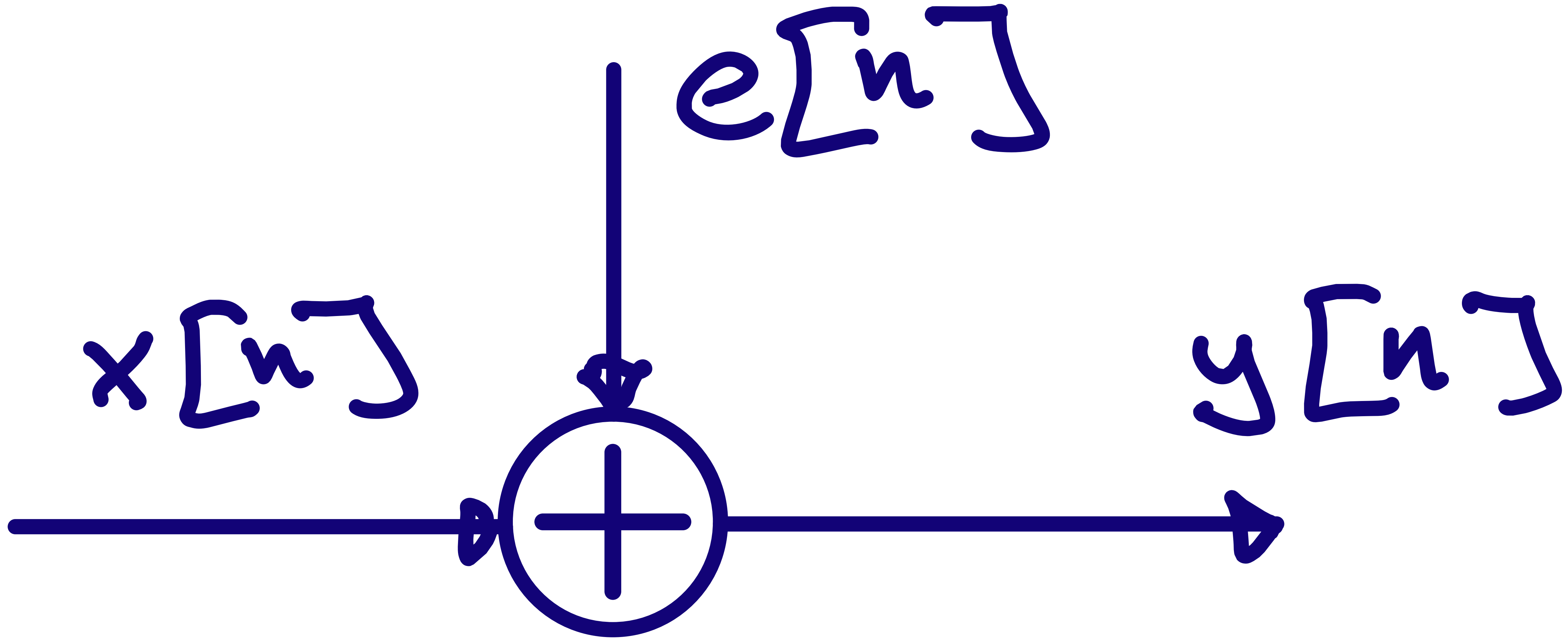
B. Murmann, ADC Performance Survey 1997-2022 (ISSCC & VLSI Symposium)

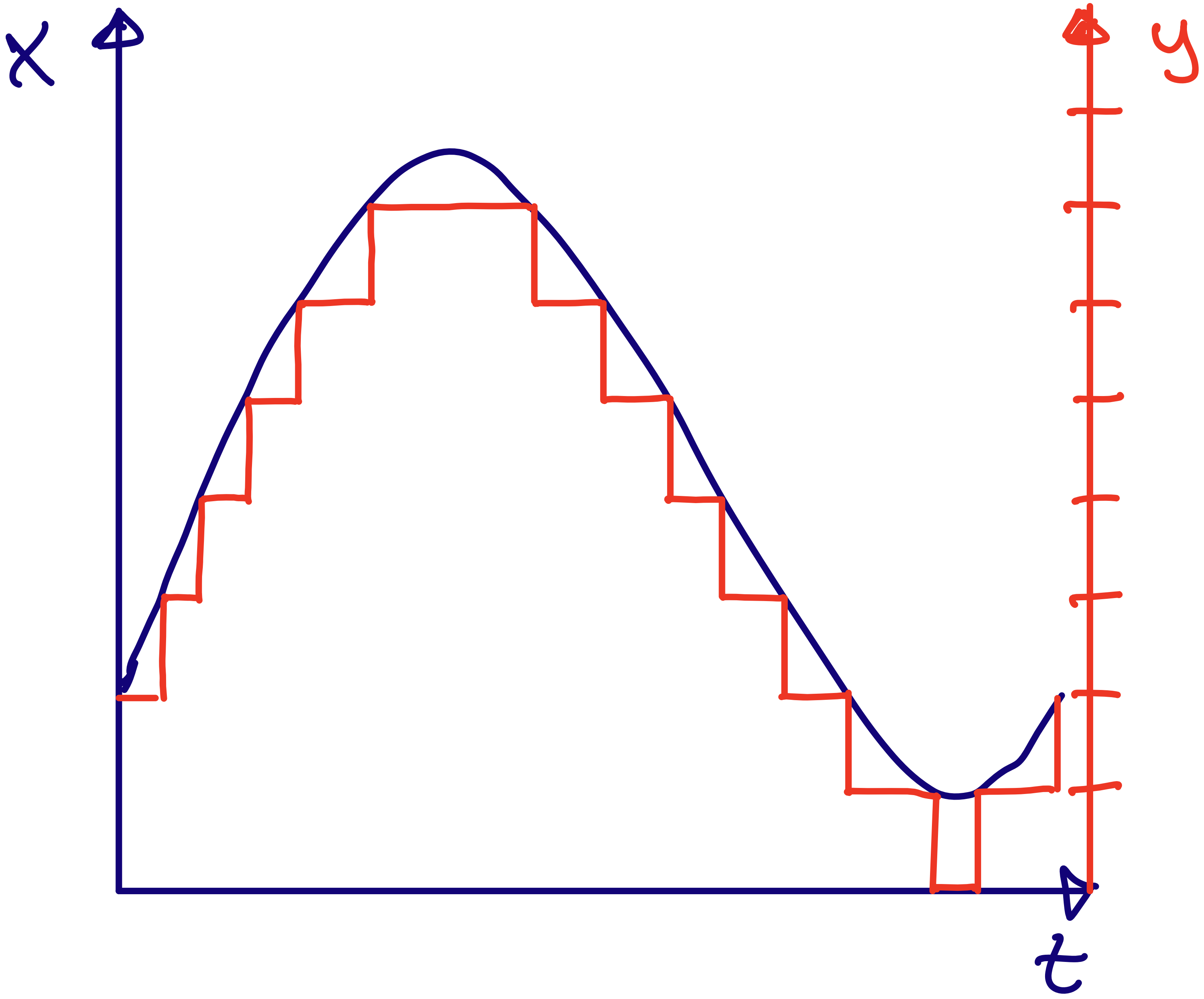
$$FOM_S = SNDR + 10\log\left(\frac{f_s/2}{P}\right)$$

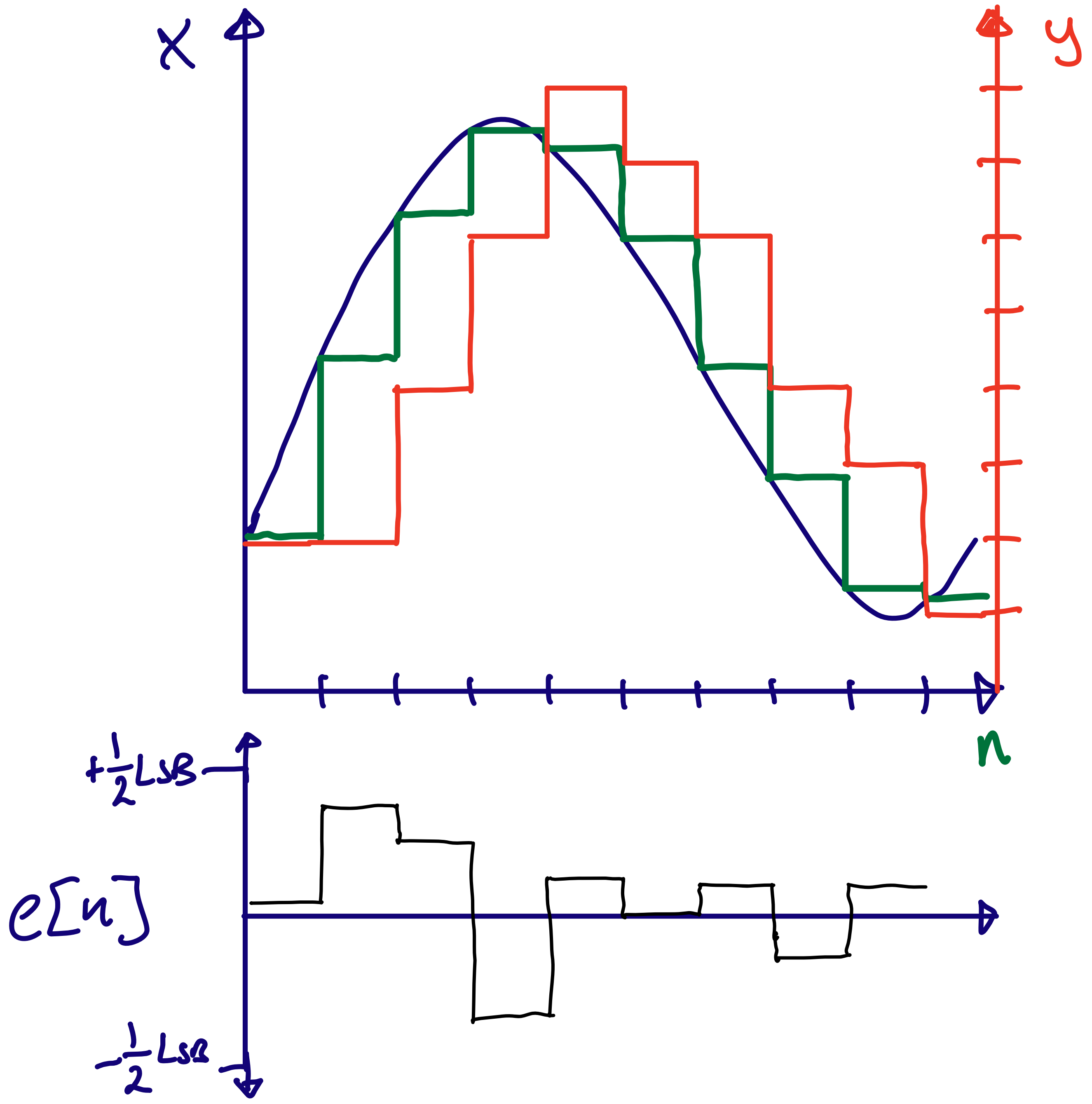
Above 180 dB is extreme



Quantization


e[n]
x[n]
y[n]


x
y
t



x
y
n
$+\frac{1}{2}$ LSB
e[n]
$-\frac{1}{2}$ LSB

See The intermodulation and distortion due to quantization of sinusoids for details

$$e_n(t) = \sum_{p=1}^{\infty} A_p \sin p\omega t$$

where p is the harmonic index, and

$$A_p = \begin{cases} \delta_{p1} A + \sum_{m=1}^{\infty} \frac{2}{m\pi} J_p(2m\pi A) & , p = \text{ odd} \\ 0 & , p = \text{ even} \end{cases}$$

$$\delta_{p1} \begin{cases} 1 & , p = 1 \\ 0 & , p \neq 1 \end{cases}$$

and $J_p(x)$ is a Bessel function of the first kind, A is the amplitude of the input signal.

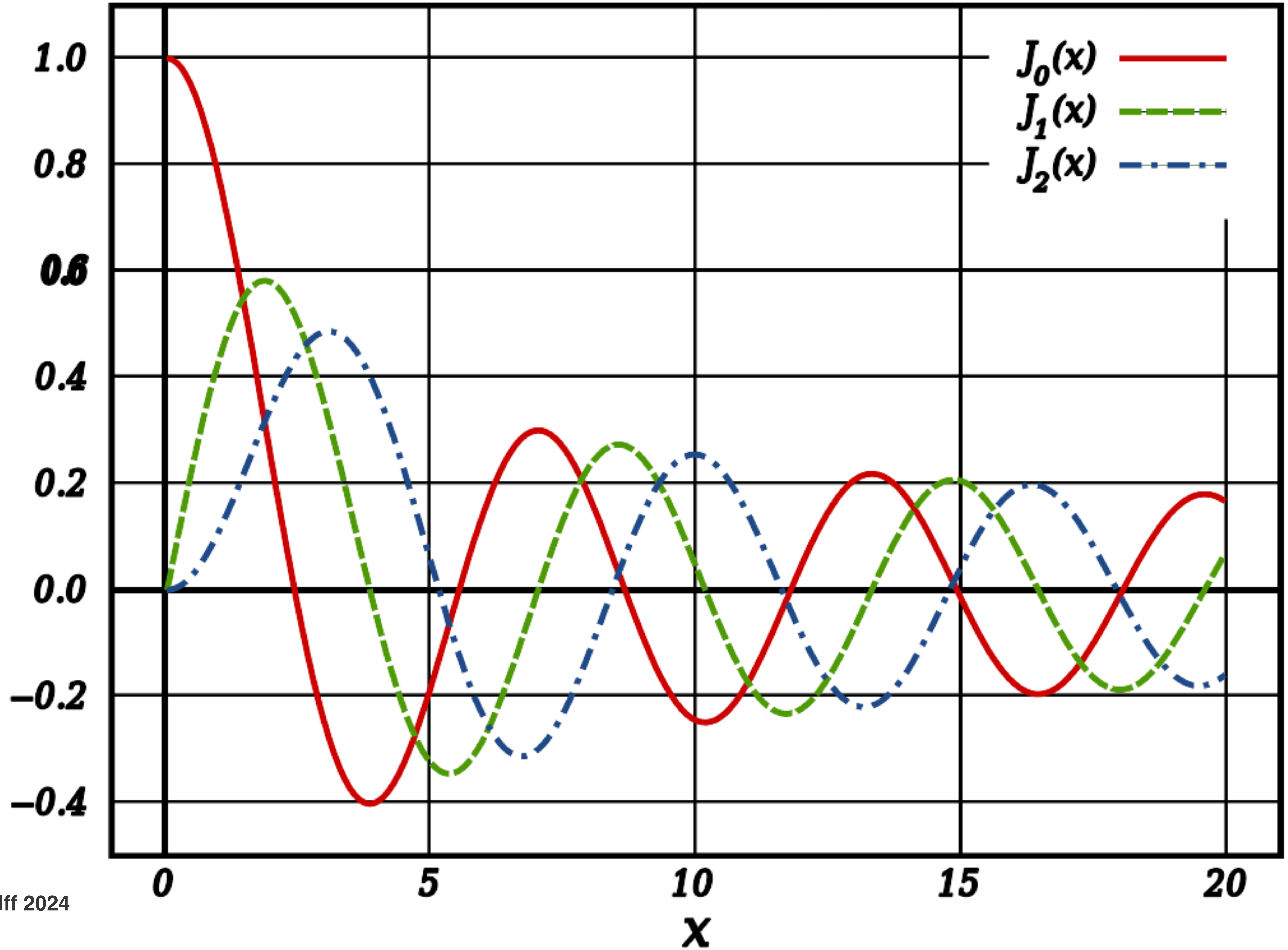
If we approximate the amplitude of the input signal as

$$A = \frac{2^n - 1}{2} \approx 2^{n-1}$$

where n is the number of bits, we can rewrite as

$$e_n(t) = \sum_{p=1}^{\infty} A_p \sin p\omega t$$

$$A_p = \delta_{p1} 2^{n-1} + \sum_{m=1}^{\infty} \frac{2}{m\pi} J_p(2m\pi 2^{n-1}), p = odd$$



1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
x
$J_0(x)$
$J_1(x)$
$J_2(x)$

$$\overline{e_n(t)} = 0$$

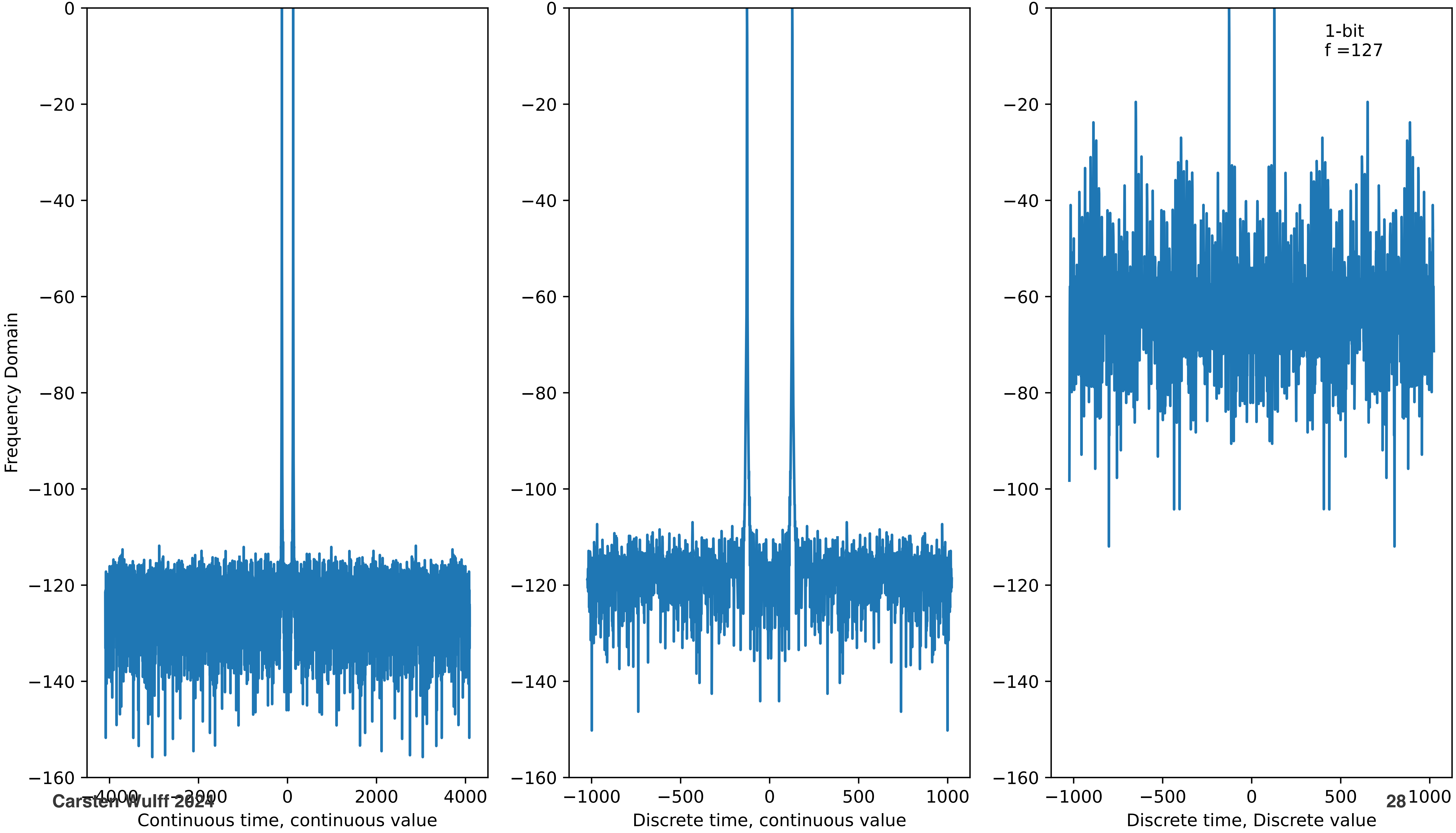
$$\overline{e_n(t)^2} = \frac{\Delta^2}{12}$$

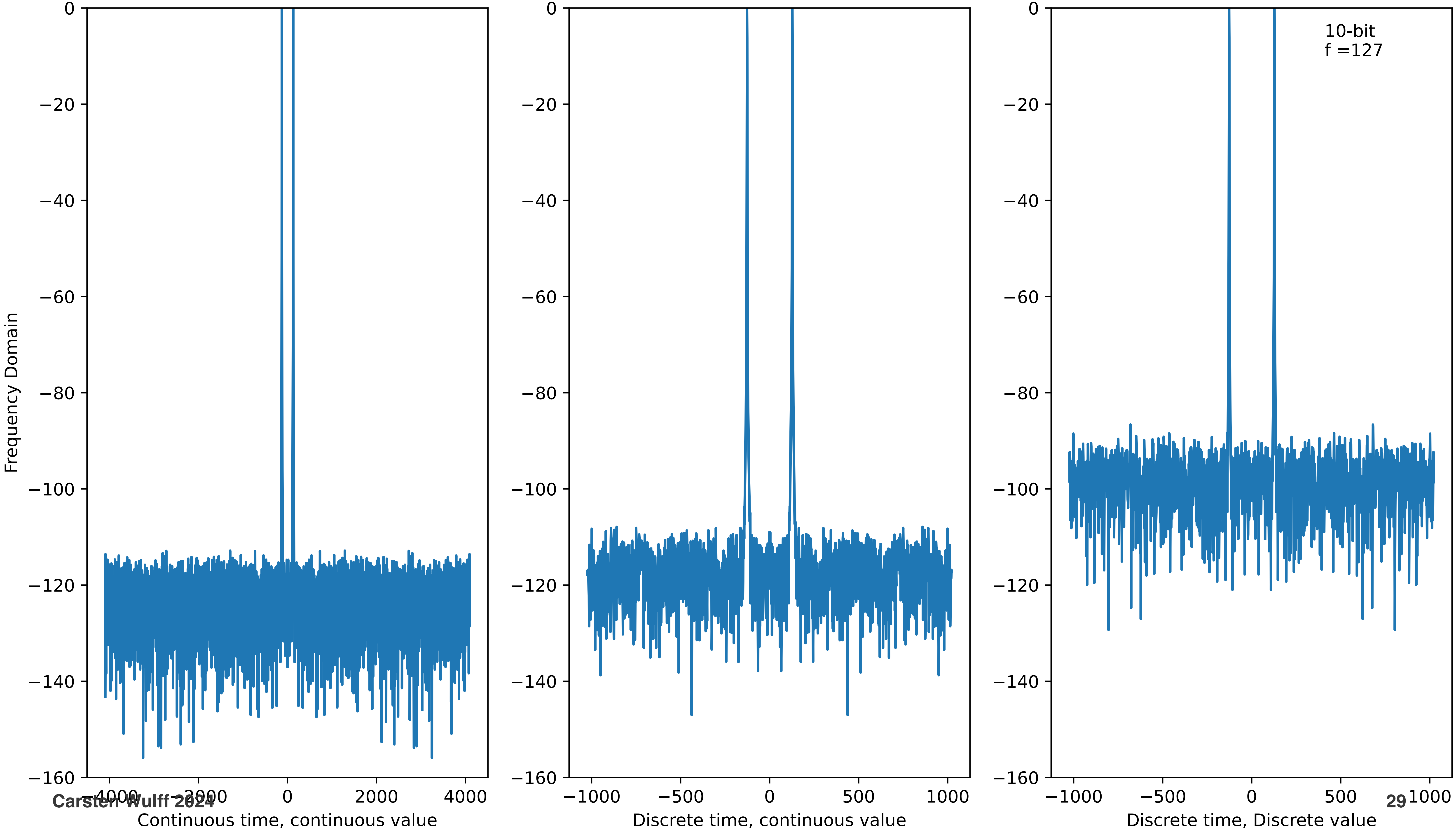
$$SQNR = 10\log\left(\frac{A^2/2}{\Delta^2/12}\right) = 10\log\left(\frac{6A^2}{\Delta^2}\right)$$

$$\Delta = \frac{2A}{2^B}$$

$$SQNR = 10\log\left(\frac{6A^2}{4A^2/2^B}\right) = 20B\log 2 + 10\log 6/4$$

$$SQNR \approx 6.02B + 1.76$$


Frequency Domain
Continuous time, continuous value
Discrete time, continuous value
Discrete time, Discrete value
1-bit
f =127


Frequency Domain
0
−20
−40
−60
−80
−100
−120
−140
−160
−4000
−2000
0
2000
4000
Continuous time, continuous value
−1000
−500
0
500
1000
Discrete time, continuous value
10-bit
f =127
Discrete time, Discrete value

Oversampling

in-band quantization noise for a oversampling ratio (OSR)

$$\overline{e_n(t)^2} = \frac{\Delta^2}{12OSR}$$

$$SQNR = 10\log\left(\frac{6A^2}{\Delta^2/OSR}\right) = 10\log\left(\frac{6A^2}{\Delta^2}\right) + 10\log(OSR)$$

$$SQNR \approx 6.02B + 1.76 + 10\log(OSR)$$

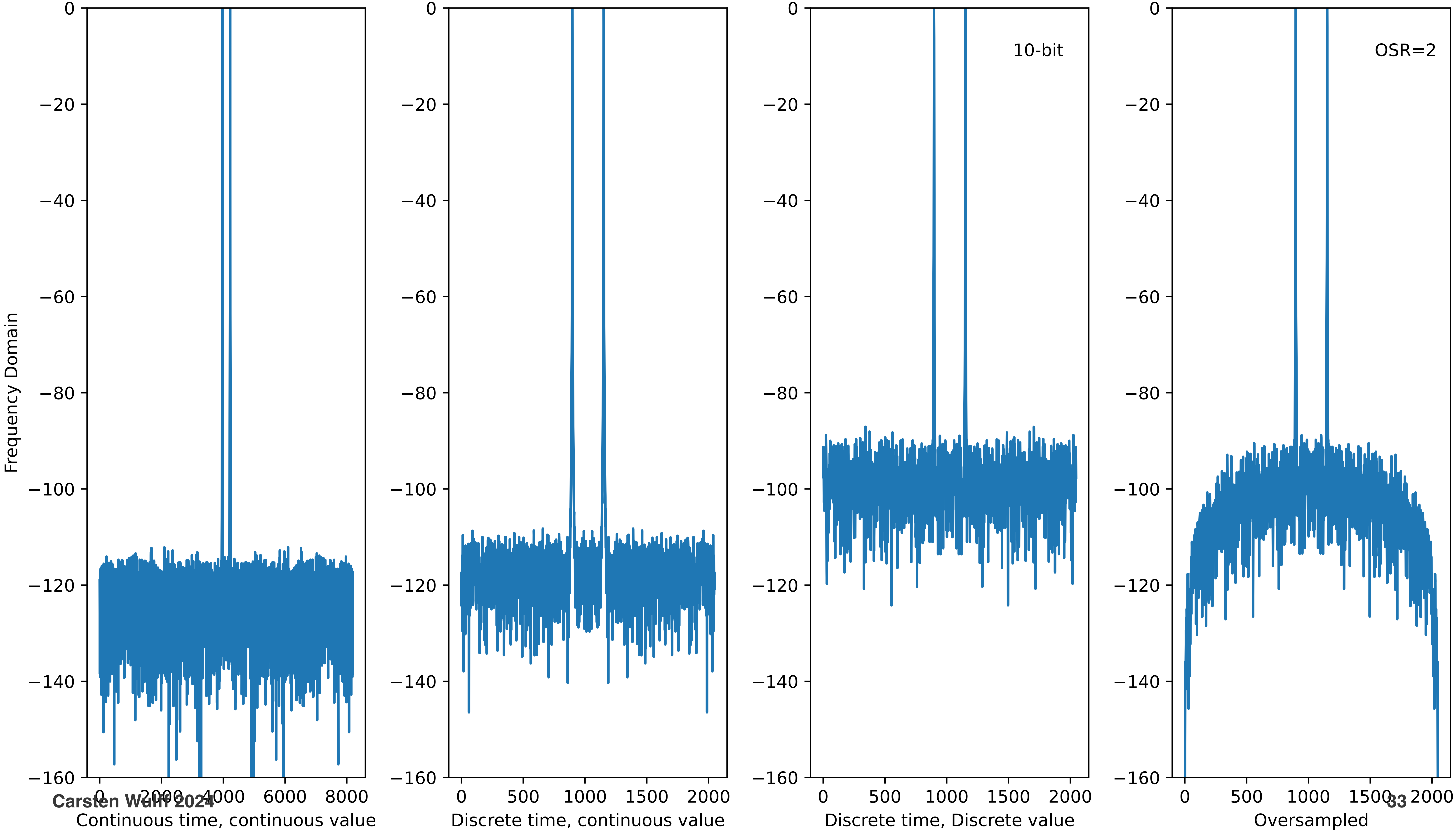
$$10\log(2) \approx 3dB$$

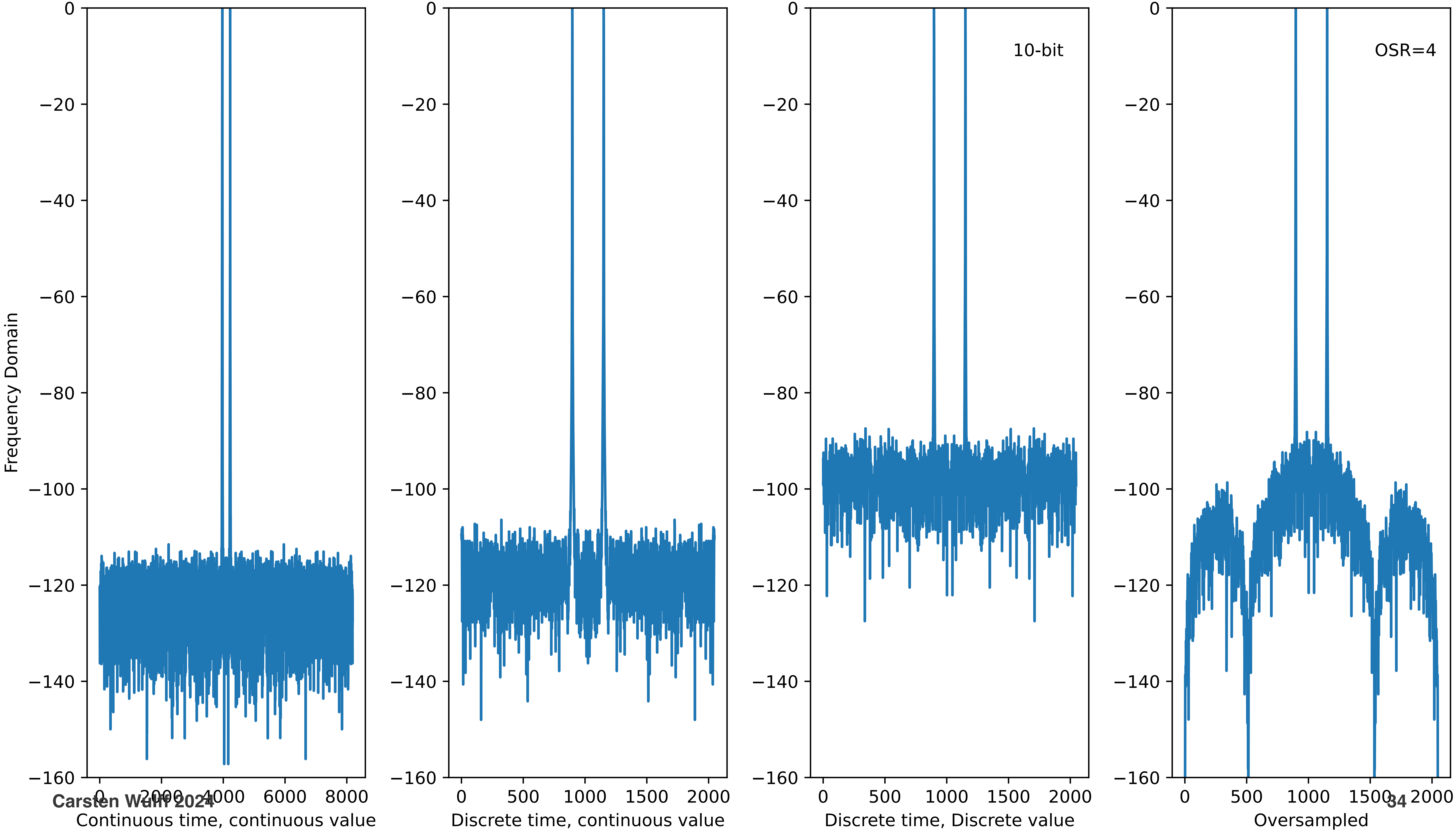
$$10\log(4) \approx 6dB$$

0.5-bit per doubling of OSR

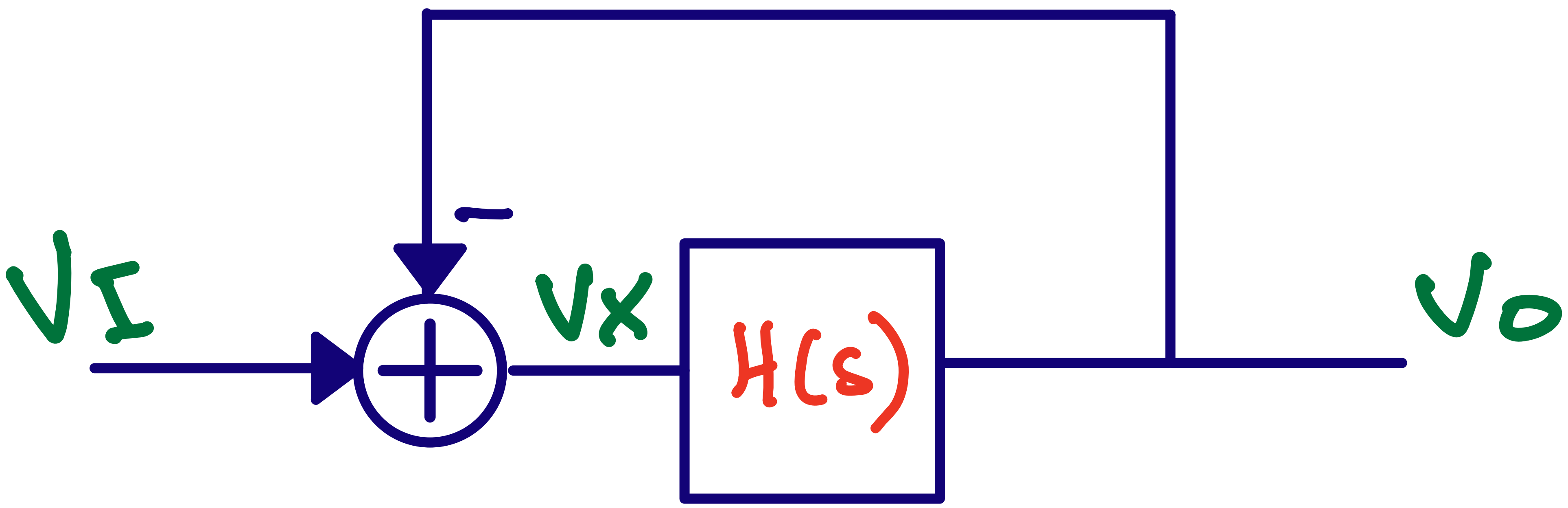
```python
def oversample(x,OSR):
    N = len(x)
    y = np.zeros(N)

    for n in range(0,N):
        for k in range(0,OSR):
            m = n+k
            if (m < N):
                y[n] += x[m]
    return y
```


Frequency Domain
0
−20
−40
−60
−80
−100
−120
−140
−160
10-bit
OSR=2
0 2000 4000 6000 8000
Continuous time, continuous value
0 500 1000 1500 2000
Discrete time, continuous value
Discrete time, Discrete value
Oversampled

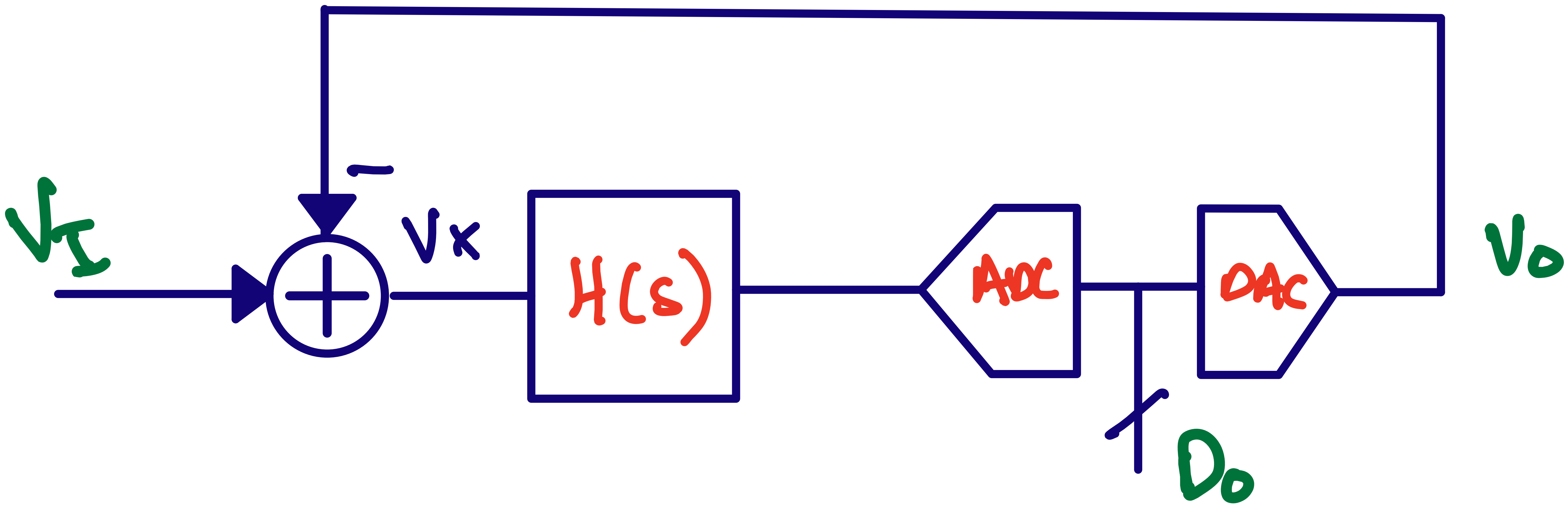

Frequency Domain
10-bit
OSR=4
Continuous time, continuous value
Discrete time, continuous value
Discrete time, Discrete value
Oversampled

Noise Shaping

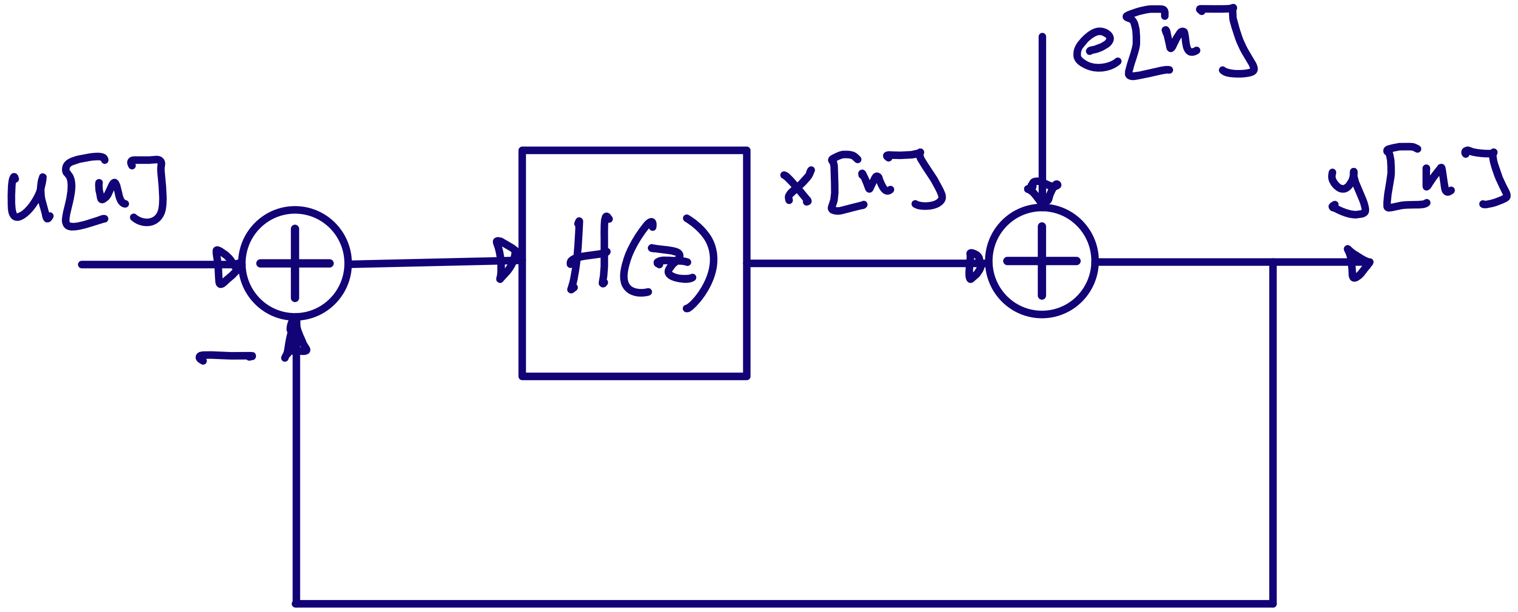


$$V_I - V_o = V_x \qquad V_o = V_x H(s)$$

$$V_I = V_o + \frac{V_o}{H(s)} \quad \xrightarrow{H(s)=\infty} \quad V_o = V_I$$



V_I
−
V_x
H(s)
ADC
DAC
V_o
D_o



Sample domain

$$y[n] = e[n] + h * (u[n] - y[n])$$

Z-Domain

$$Y(z) = E(z) + H(z)\left[U(z) - Y(z)\right]$$

Signal transfer function

Assume U and E are uncorrelated, and E is zero

$$Y = HU - HY$$

$$STF = \frac{Y}{U} = \frac{H}{1+H} = \frac{1}{1+\frac{1}{H}}$$

Noise transfer function

Assume U is zero

$$Y = E + HY \rightarrow NTF = \frac{1}{1+H}$$

Combined transfer function

$$Y(z) = STF(z)U(z) + NTF(z)E(z)$$

First-Order Noise-Shaping

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

$$STF = \frac{1/(z-1)}{1+1/(z-1)} = \frac{1}{z} = z^{-1}$$

$$NFT = \frac{1}{1+1/(z-1)} = \frac{z-1}{z} = 1 - z^{-1}$$

$$z = e^{sT} \overset{s=j\omega}{\rightarrow} e^{j\omega T} = e^{j2\pi f/f_s}$$

$$NTF(f) = 1 - e^{-j2\pi f/f_s}$$

$$= \frac{e^{j\pi f/f_s} - e^{-j\pi f/f_s}}{2j} \times 2j \times e^{-j\pi f/f_s}$$

$$= \sin\frac{\pi f}{f_s} \times 2j \times e^{-j\pi f/f_s}$$

$$|NFT(f)| = \left|2\sin\left(\frac{\pi f}{f_s}\right)\right|$$

$$P_s = A^2/2$$

$$P_n = \int_{-f_0}^{f_0} \frac{\Delta^2}{12} \frac{1}{f_s} \left[2\sin\left(\frac{\pi f}{f_s}\right)\right]^2 dt$$

$$\vdots$$

$$SQNR = 6.02B + 1.76 - 5.17 + 30\log(OSR)$$

SQNR and ENOB

$$SQNR_{nyquist} \approx 6.02B + 1.76$$

$$SQNR_{oversample} \approx 6.02B + 1.76 + 10\log(OSR)$$

$$SQNR_{\Sigma\Delta1} \approx 6.02B + 1.76 - 5.17 + 30\log(OSR)$$

$$SQNR_{\Sigma\Delta2} \approx 6.02B + 1.76 - 12.9 + 50\log(OSR)$$

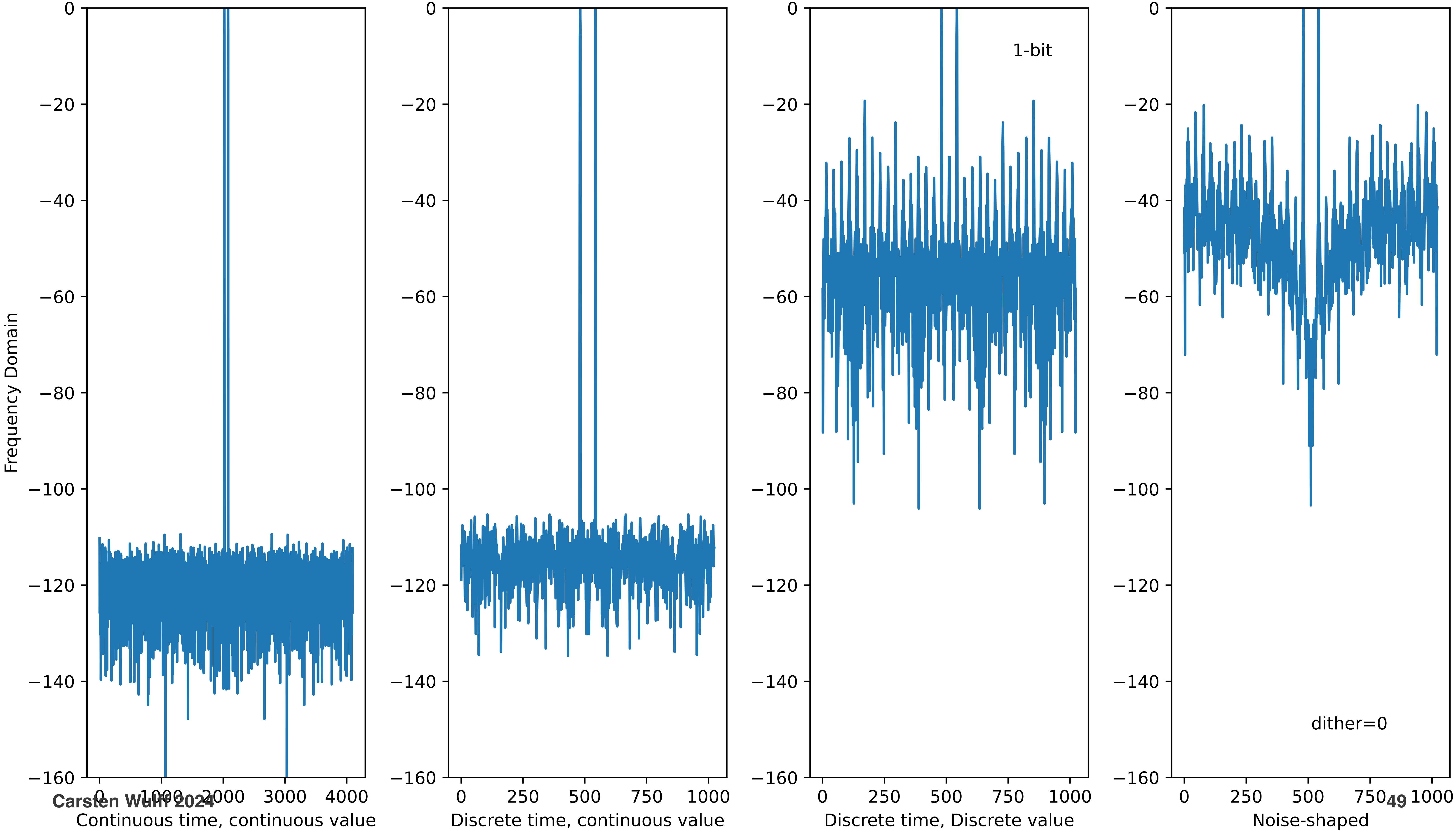
$$ENOB = (SQNR - 1.76)/6.02$$

Assume 1-bit quantizer, what would be the maximum ENOB?

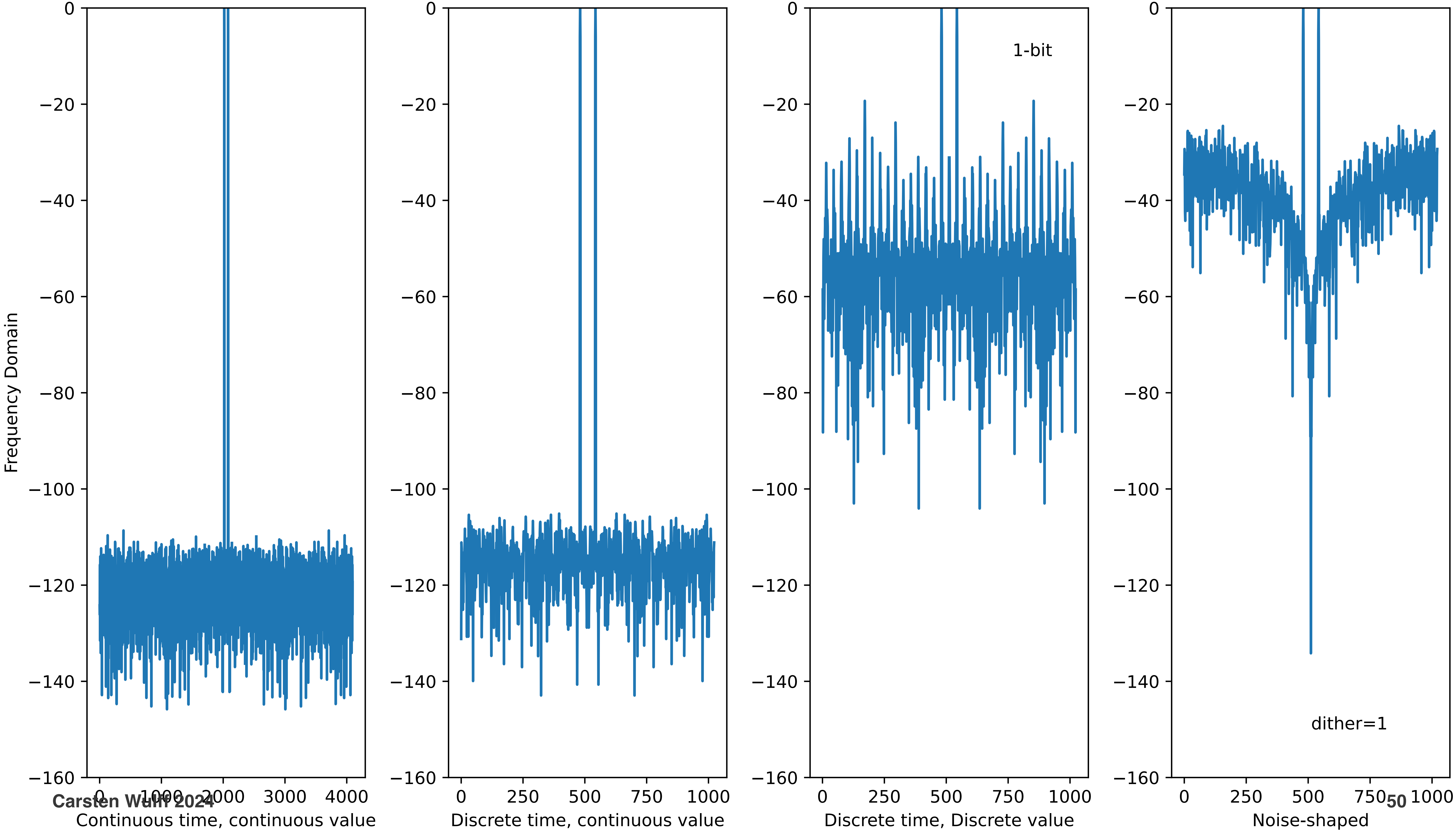
OSR	Oversampling	First-Order	Second Order
4	2	3.1	3.9
64	4	9.1	13.9
1024	6	15.1	23.9

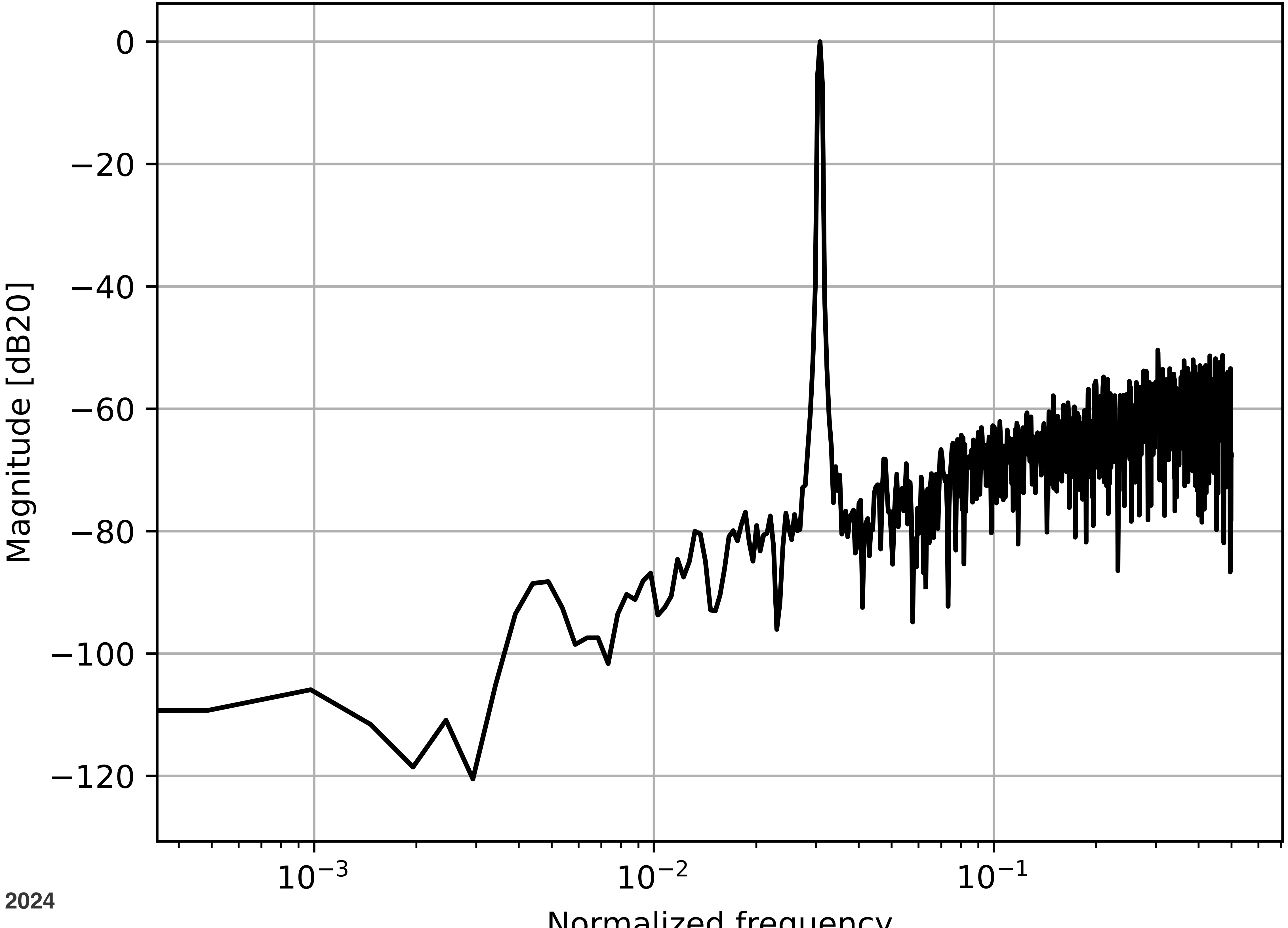
Examples

```
# u is discrete time, continuous value input
M = len(u)
y_sd = np.zeros(M)
x = np.zeros(M)
for n in range(1,M):
    x[n] = x[n-1] + (u[n]-y_sd[n-1])
    y_sd[n] = np.round(x[n]*2**bits
    + dither*np.random.randn()/4)/2**bits
```



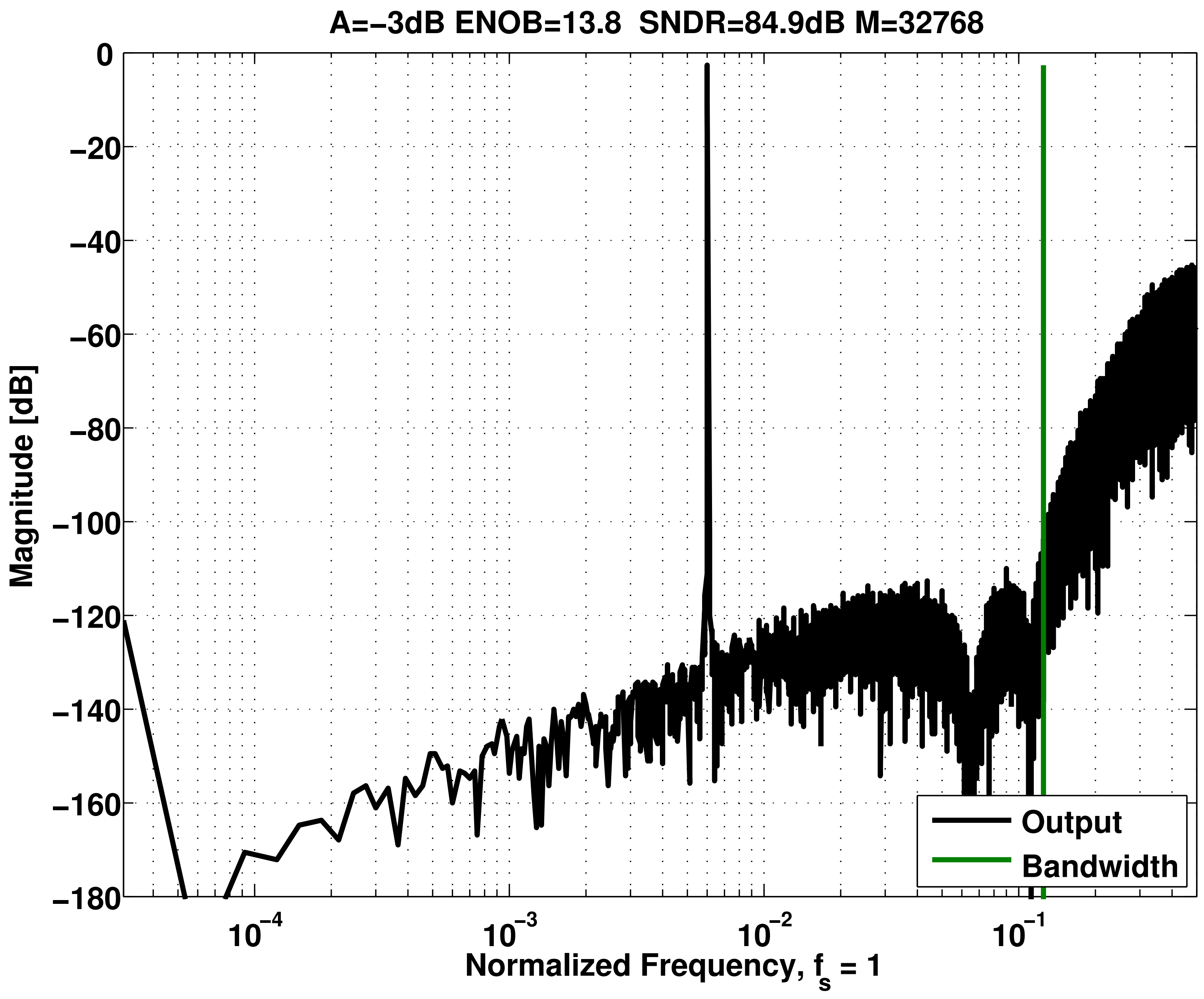
Frequency Domain
0
−20
−40
−60
−80
−100
−120
−140
−160
0 1000 2000 3000 4000
Continuous time, continuous value
0 250 500 750 1000
Discrete time, continuous value
1-bit
Discrete time, Discrete value
dither=0
Noise-shaped


Frequency Domain
0
−20
−40
−60
−80
−100
−120
−140
−160
0 1000 2000 3000 4000
Continuous time, continuous value
0 250 500 750 1000
Discrete time, continuous value
1-bit
Discrete time, Discrete value
dither=1
Noise-shaped

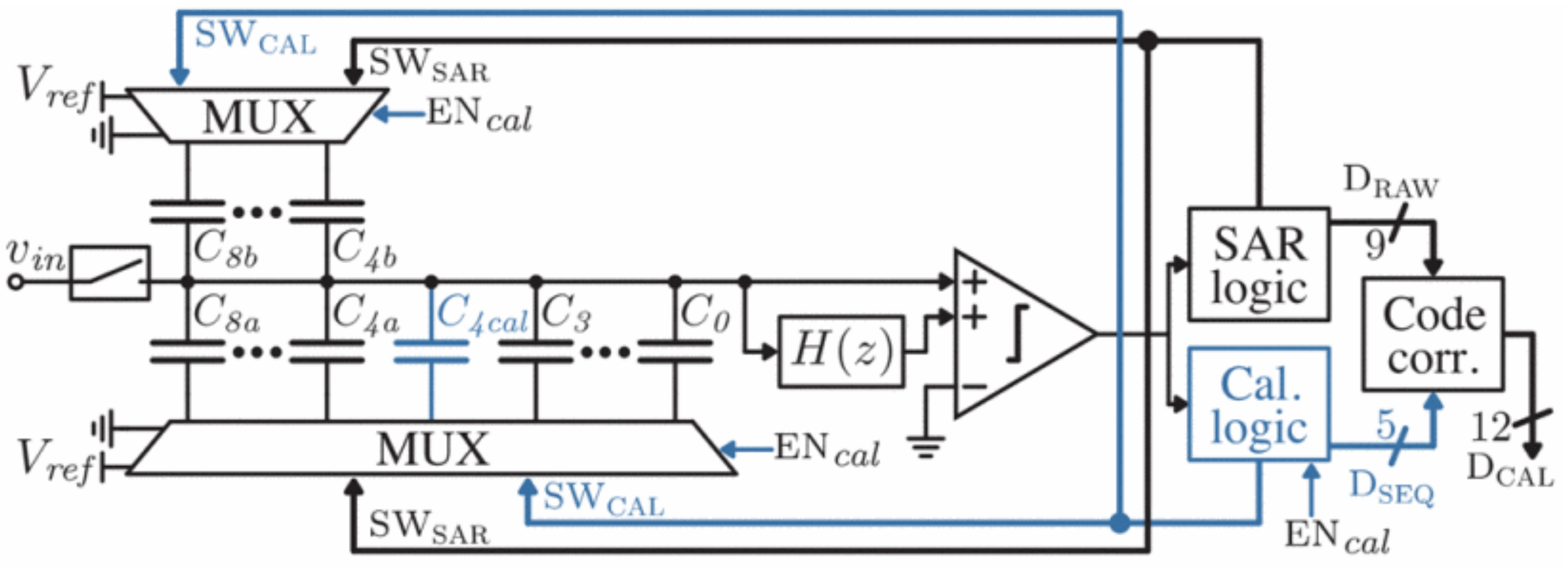


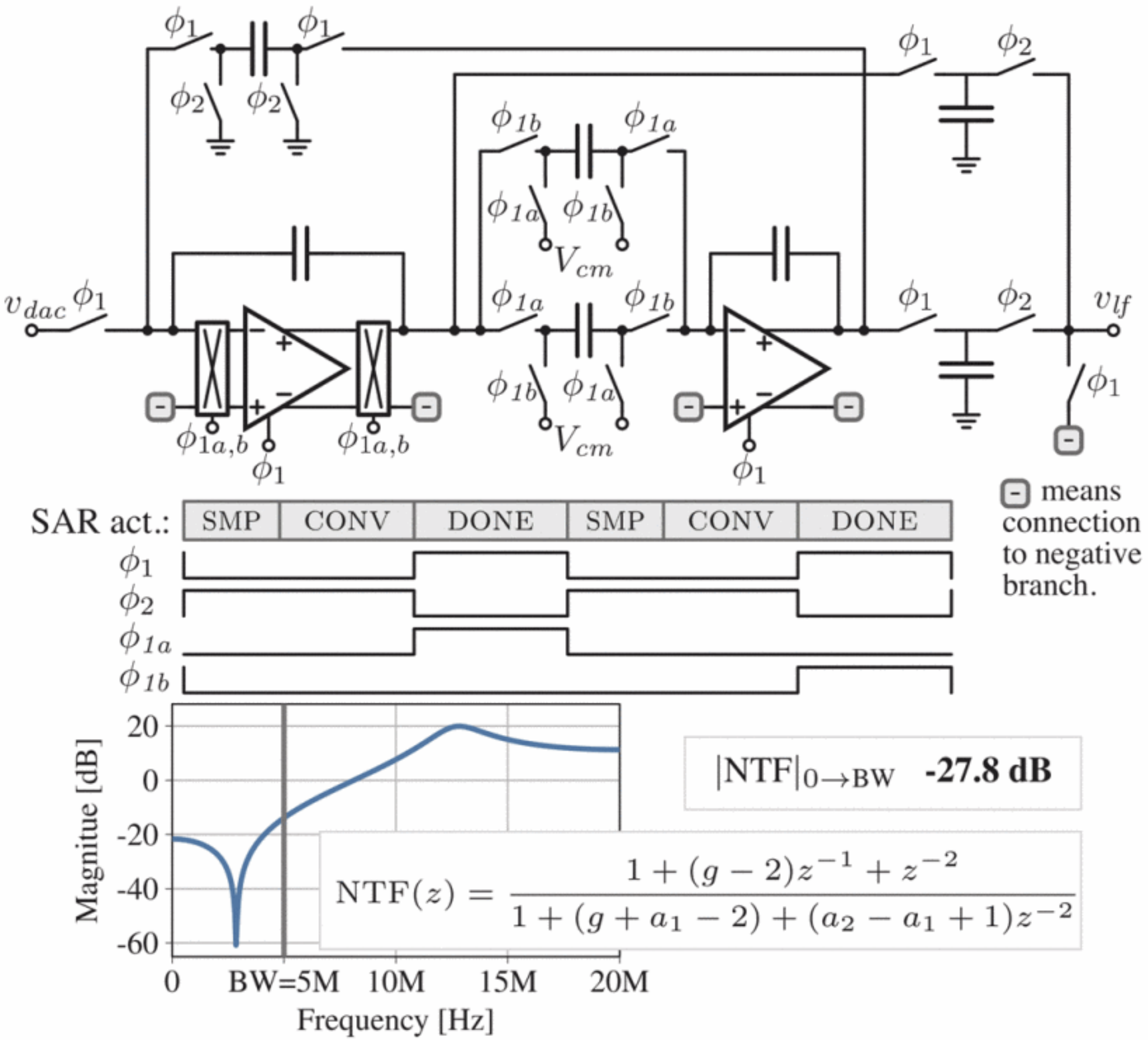
0
−20
−40
−60
−80
−100
−120
Magnitude [dB20]
10^{-3}
10^{-2}
10^{-1}
Normalized frequency

Resonators in Open-Loop Sigma-Delta Modulators



A 68 dB SNDR Compiled Noise-Shaping SAR ADC With On-Chip CDAC Calibration





v_{dac}
ϕ_1
ϕ_2
ϕ_{1a}
ϕ_{1b}
$\phi_{1a,b}$
V_{cm}
v_{lf}
means connection to negative branch.
SAR act.:
SMP
CONV
DONE
SMP
CONV
DONE
Magnitue [dB]
Frequency [Hz]
0
BW=5M
10M
15M
20M
20
0
-20
-40
-60
$|\mathrm{NTF}|_{0\rightarrow \mathrm{BW}}$ -27.8 dB
$\mathrm{NTF}(z) = \frac{1 + (g-2)z^{-1} + z^{-2}}{1 + (g + a_1 - 2) + (a_2 - a_1 + 1)z^{-2}}$

Design Considerations for a Low-Power Control-Bounded A/D Converter

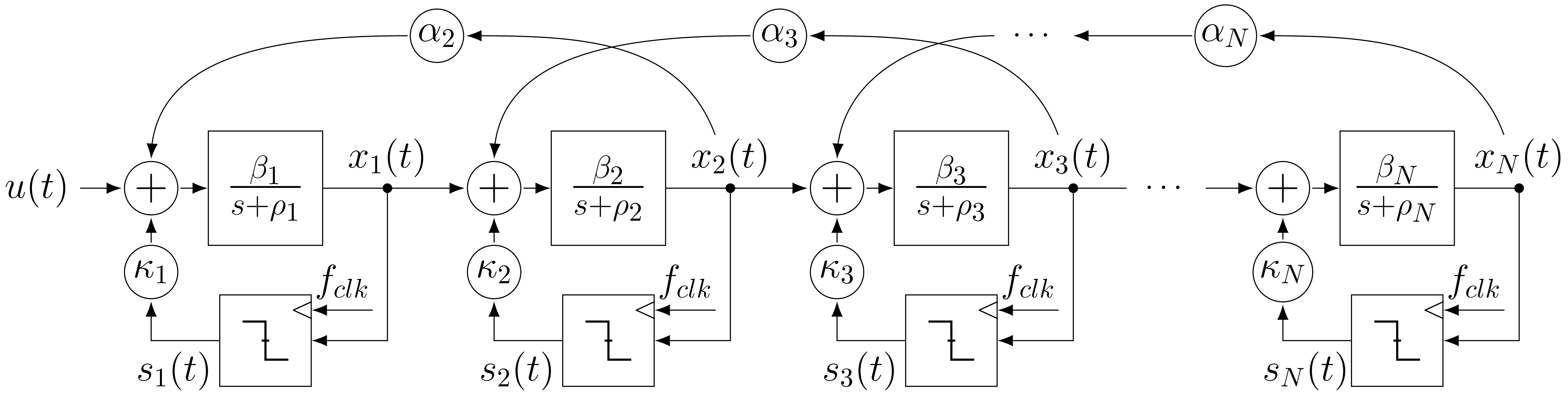
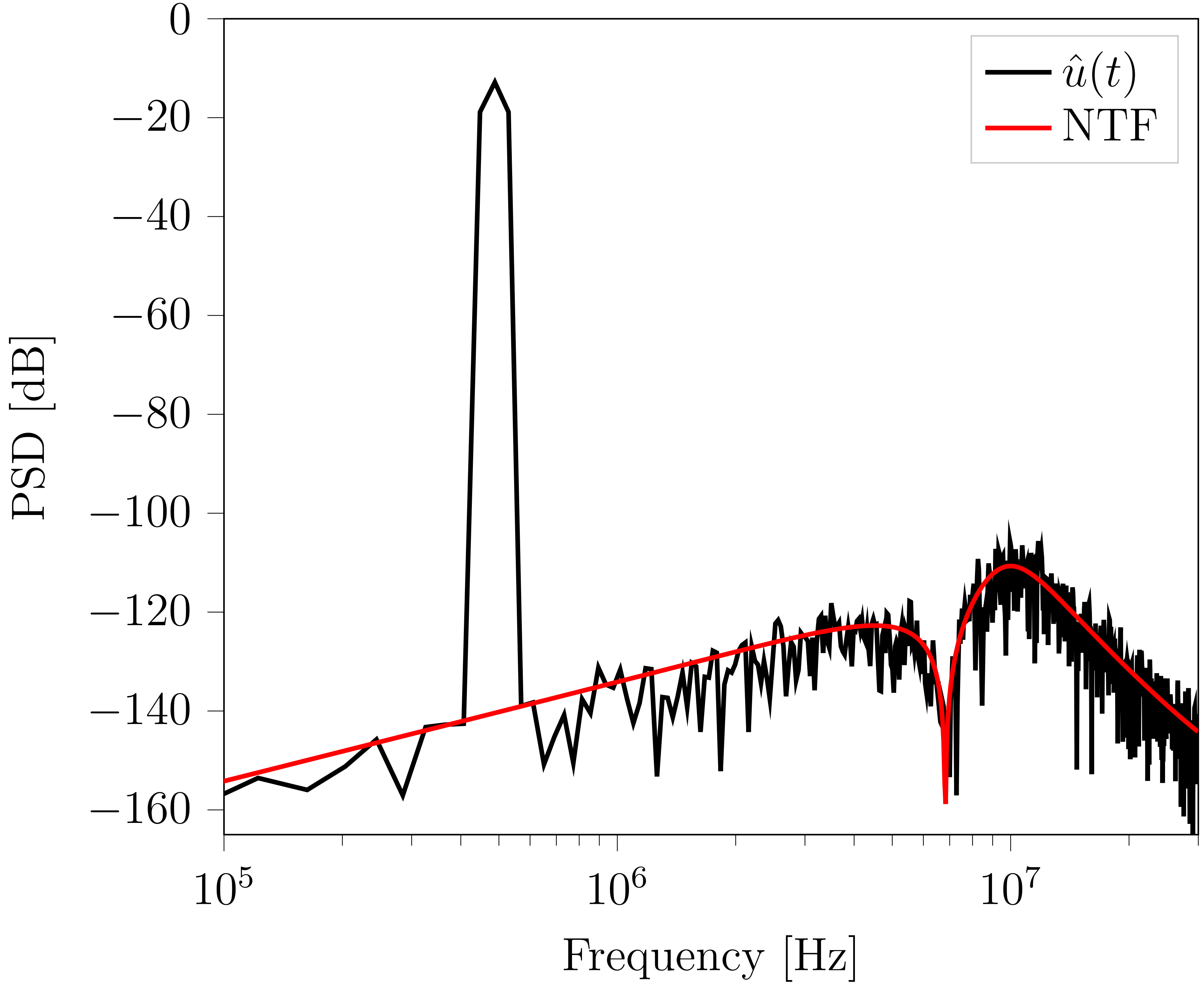
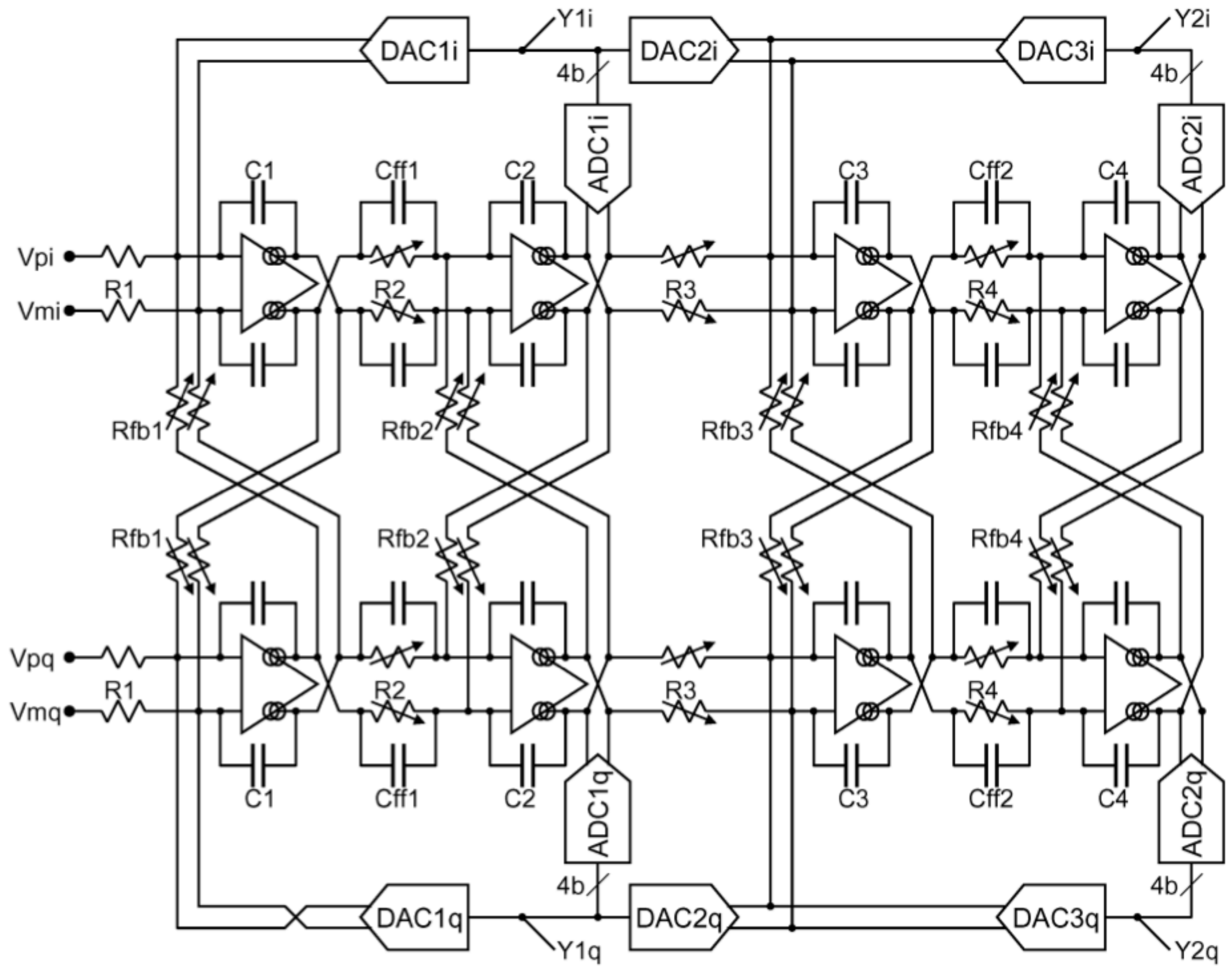


Figure 3.1: The general structure of the Leapfrog ADC



0
−20
−40
−60
−80
−100
−120
−140
−160
PSD [dB]
10^5
10^6
10^7
Frequency [Hz]
$\hat{u}(t)$
NTF

A 56 mW Continuous-Time Quadrature Cascaded Sigma-Delta Modulator With 77 dB DR in a Near Zero-IF 20 MHz Band



sigma-delta modulator design.

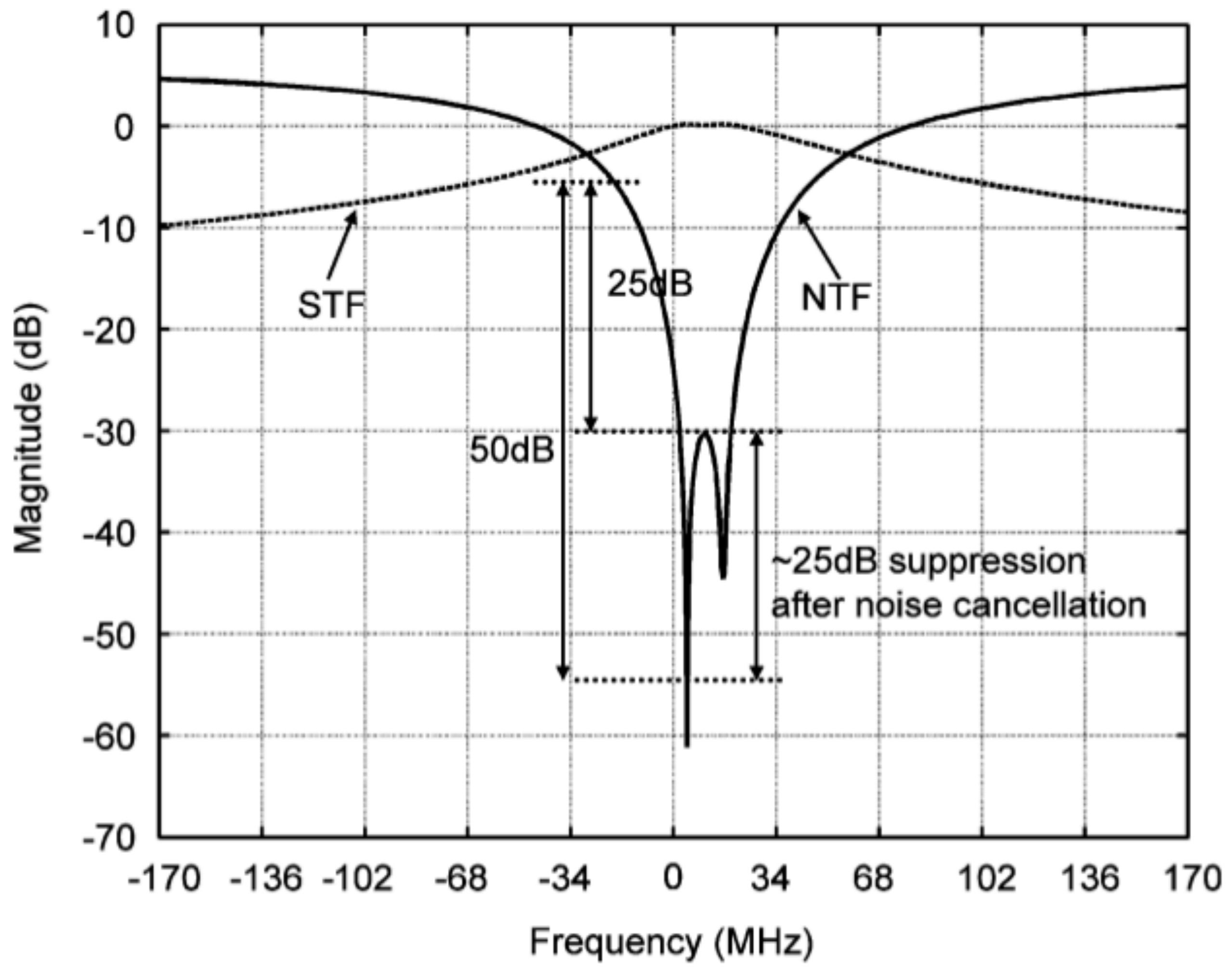
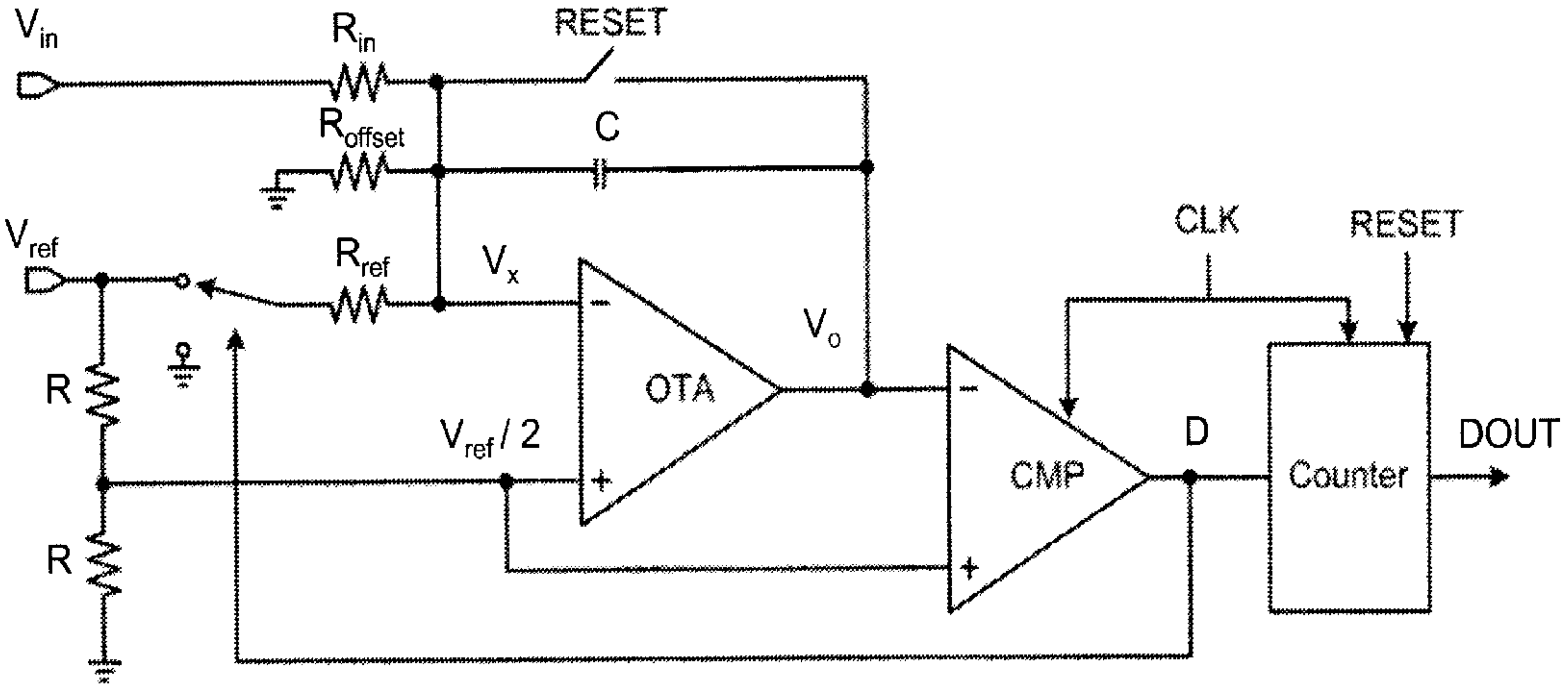


Fig. 9. NTF and STF of first stage.

Analogue-to-digital converter



Thanks!