Simulation Techniques in Data Converter Design

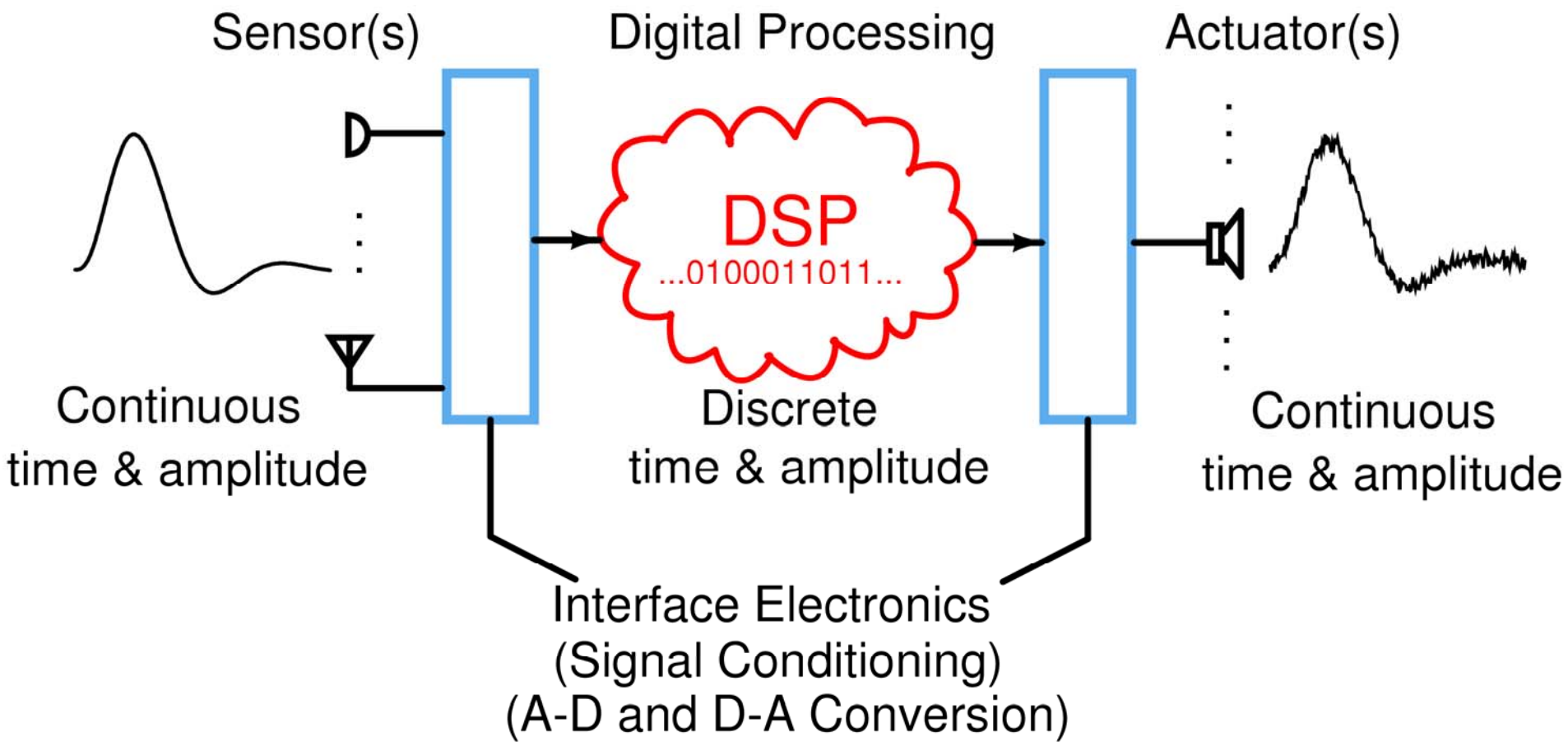
Shanthi Pavan

Indian Institute of Technology, Madras

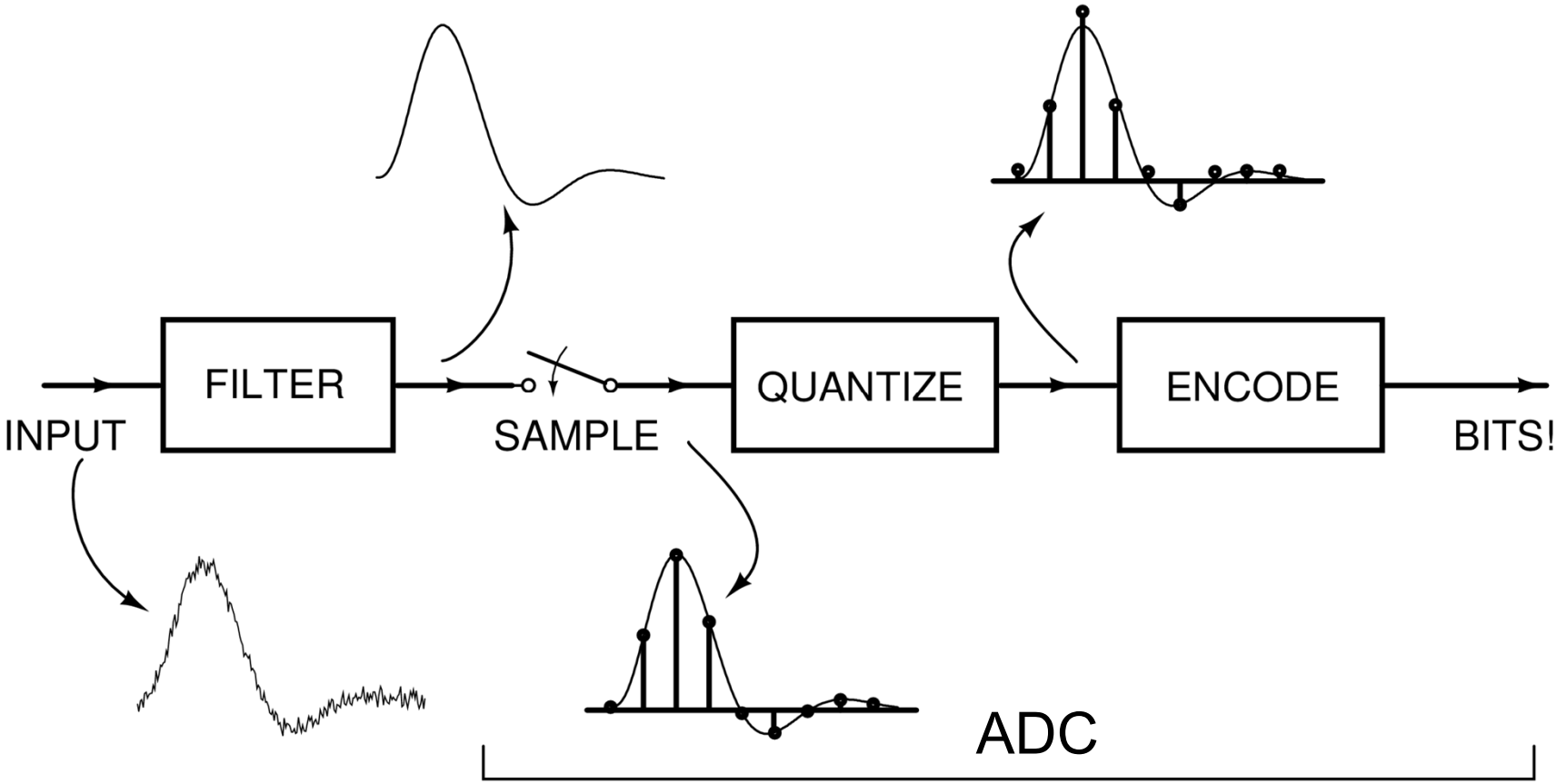
Outline

- Overview of selected Nyquist ADCs
- Creating and interpreting ADC spectra
- Oversampling ADCs
  - Simulation techniques
  - Use of RF analyses in oversampling converter designs
- Concluding Remarks
- References

Data Conversion in our World



The A/D Conversion Process



Simulation Strategies

- ADCs and DACs are complex mixed signal systems
- Can be simulated at several levels
  - ADC Functional Behavior
  - Models at the block level
  - Models at the circuit level
  - Mix and match

Simulation Strategies

- Functional behavior
  - Full chip or SoC functionality simulation
  - Second order effects not important/needed
- Block level models
  - Architectural exploration
  - What-if scenarios
  - Rapid estimates of mismatch effects
- Mix and match
  - Behavioral + transistor level
  - Helps isolate problems during debug

Simulation Strategies

- Functional behavior
  - Full chip or SoC functionality simulation
  - Second order effects not important/needed
- Bl
  -
  -
  - Rapid estimates of mismatch effects
- Mix and match
  - Behavioral + transistor level
  - Helps isolate problems during debug

Always run the full layout extracted design at the transistor level – What you did not model bites you most of the time

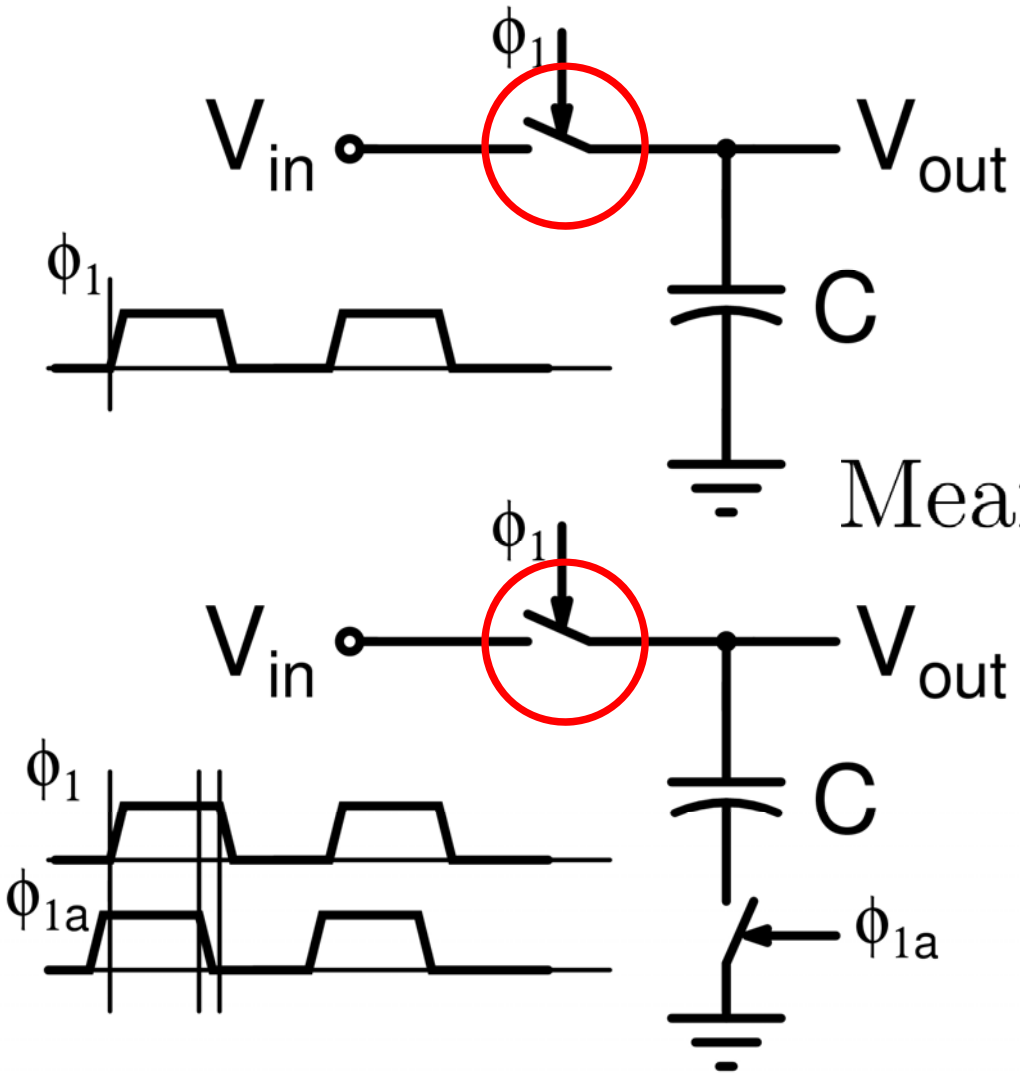
ADC Architectures

- Nyquist Rate
  - Flash
  - Folding
  - Two step
  - Pipeline and Algorithmic
  - Successive Approximation (SAR)
- Oversampling ($\Sigma\Delta$) ADCs
  - Discrete-time $\Sigma\Delta$
  - Continuous-time $\Sigma\Delta$

DAC Architectures

- Nyquist Rate
  - Current Steering
  - Resisitor String
  - Algorithmic
- Oversampling ($\Sigma\Delta$) DACs
- Not covered in this tutorial

ADC Operations : Sampling



Top Plate Sampling

Mean Square Noise : $\frac{kT}{C}$

Bottom Plate Sampling

Quantizer Basics

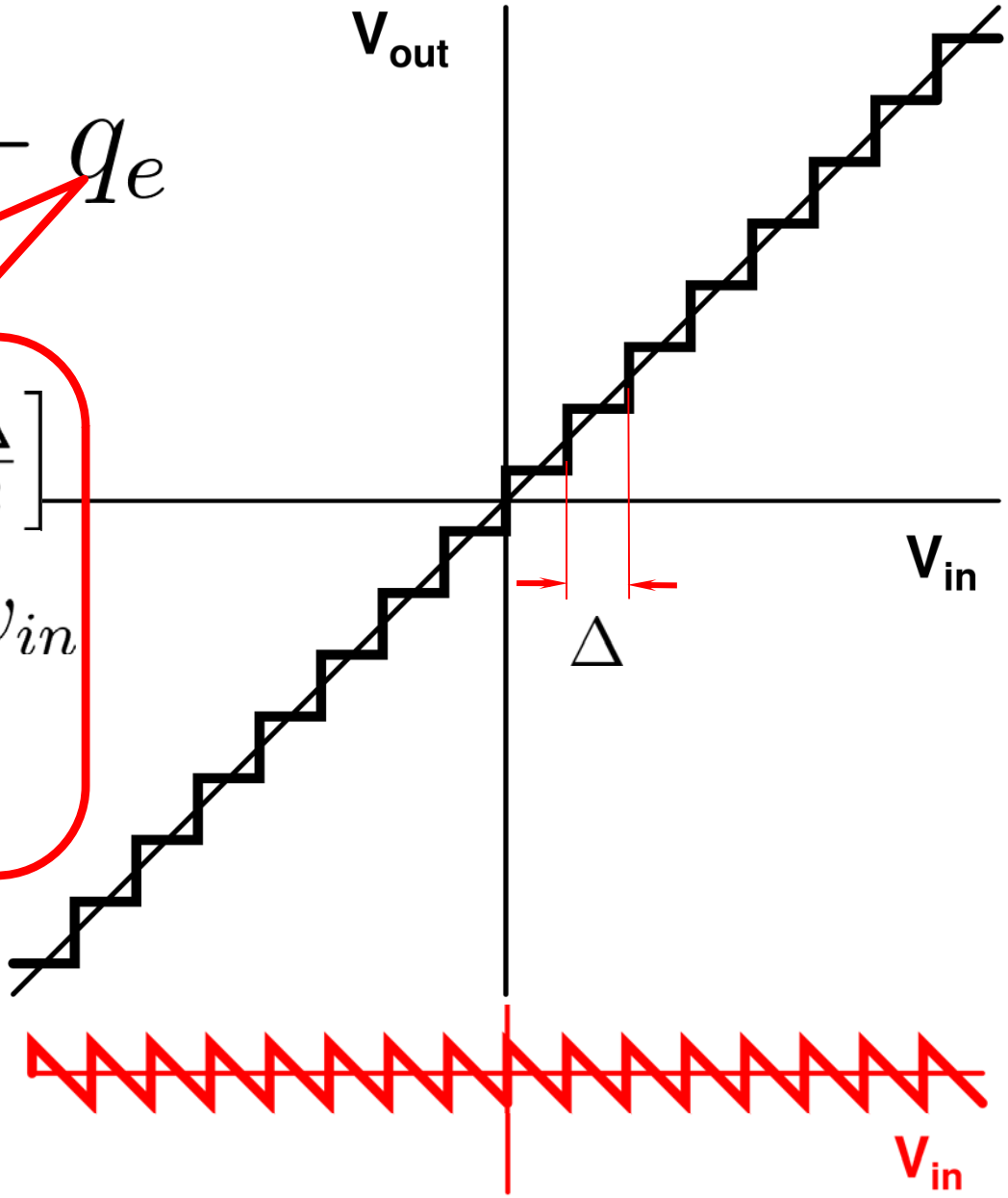
$$v_{out} = v_{in} + q_e$$

Assumptions

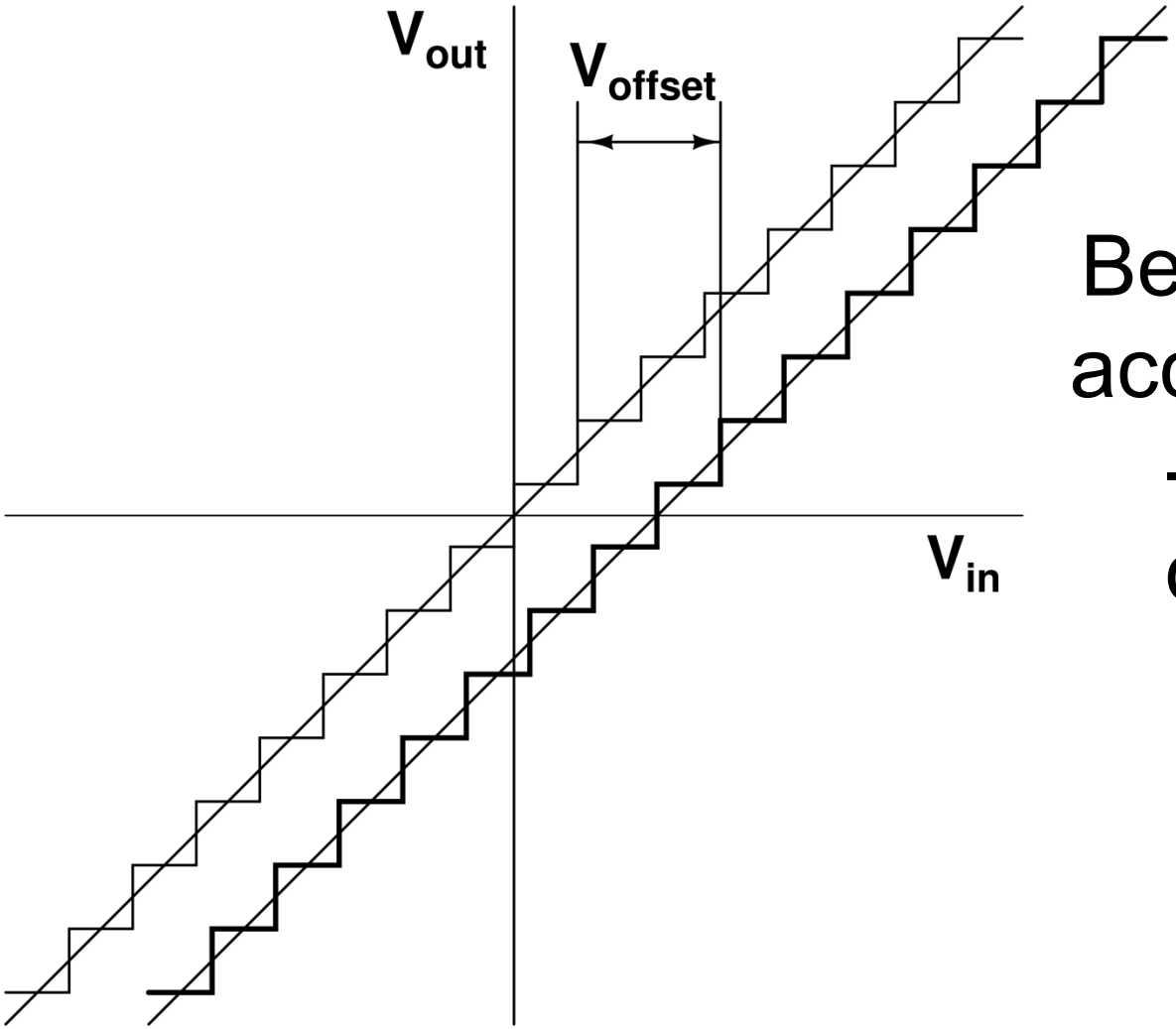
uniform in $\left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$

uncorrelated with v_{in}

spectrally white



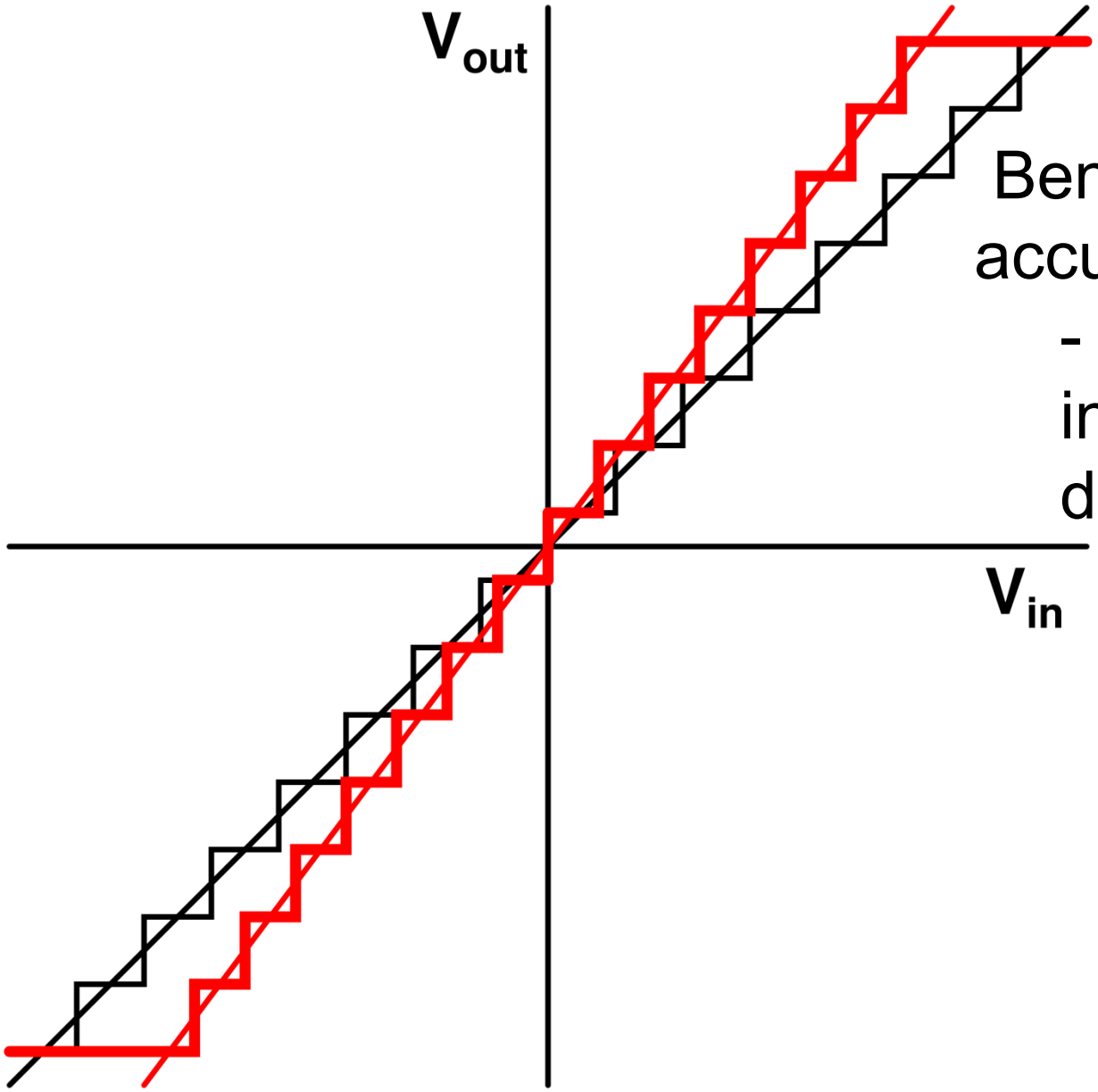
Offset Error



Benign when relative accuracy is desired

- Cancelled digitally

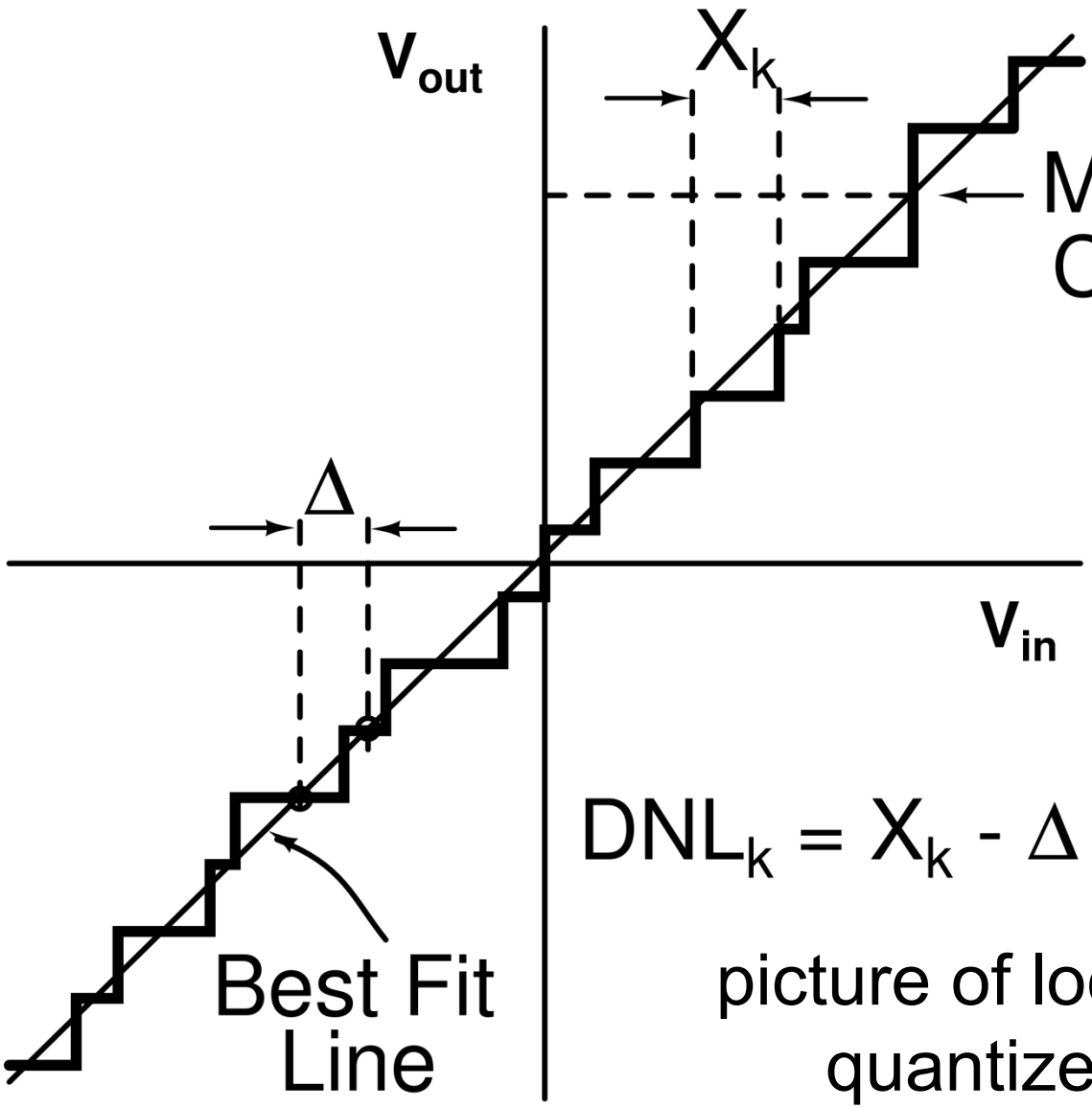
Gain Error



Benign when relative accuracy is desired

- Correct using AGC in the analog/digital domains

Differential Nonlinearity (DNL)



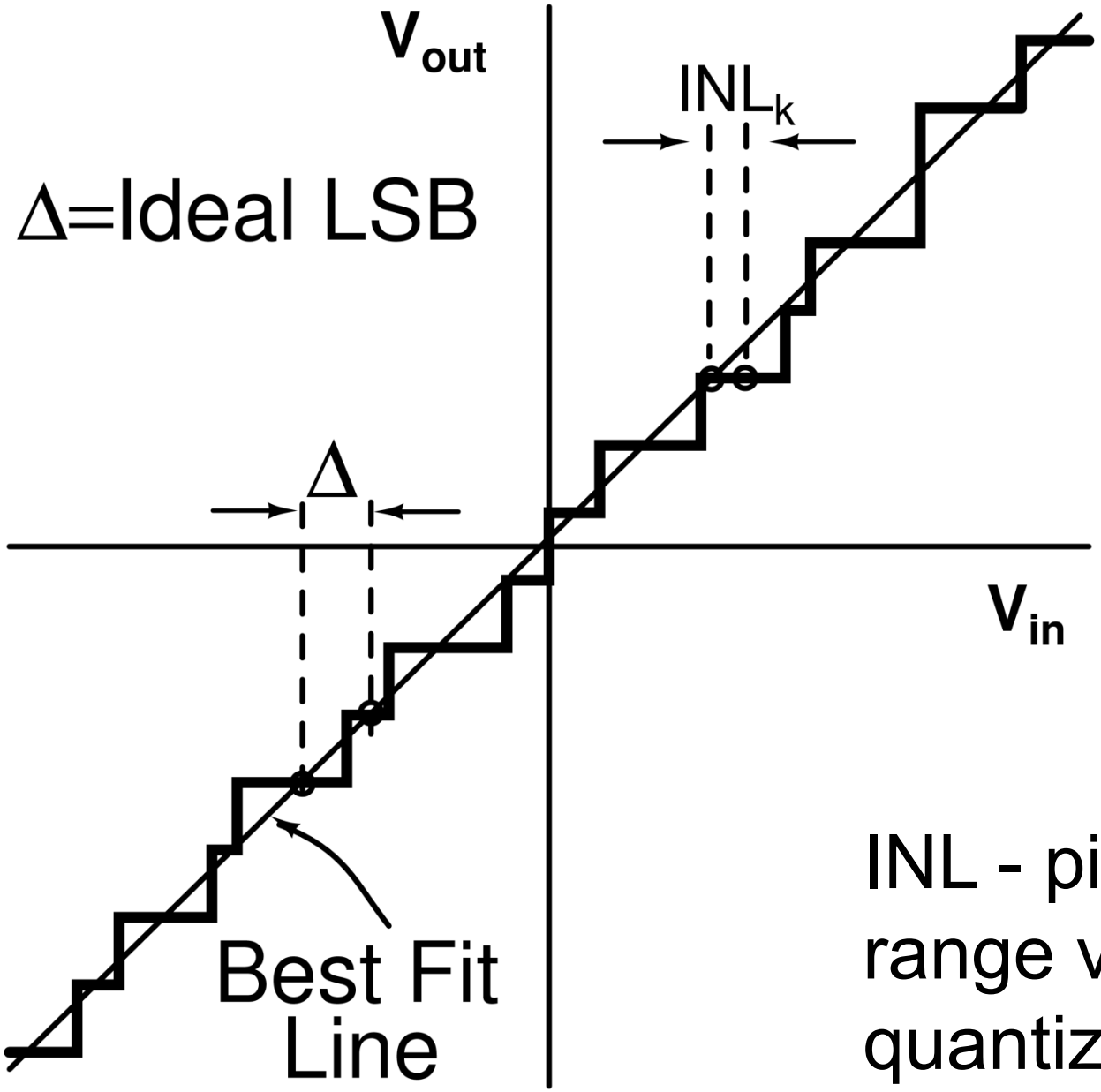
Nonmonotonicity & missing codes

Monotonic if $|DNL| < \Delta$

$DNL_k = X_k - \Delta$

picture of local variations in quantizer thresholds

Integral Nonlinearity (INL)



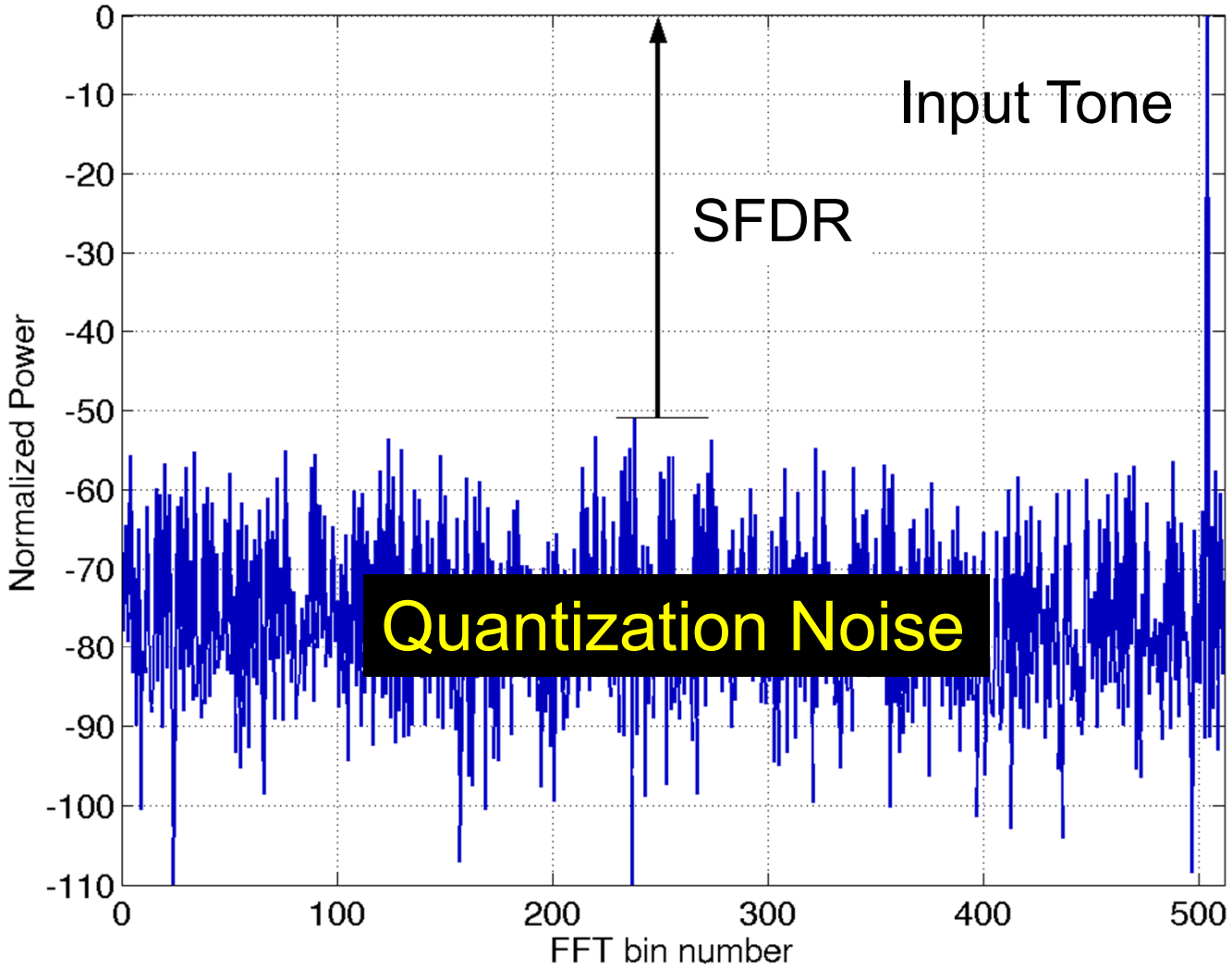
INL - picture of long range variations in quantizer thresholds

Dynamic Performance Metrics

- Peak Signal to Noise Ratio (SNR) of an ideal quantizer
  - (6N + 1.76) dB for a full scale sine wave
  - 1/2 the step size means ¼ the noise power
- Signal to Noise + Distortion Ratio (SNDR)
  - Signal to everything else
- Effective Number of Bits (ENOB)

$$\mathrm{ENOB} = \frac{\mathrm{SNDR} - 1.76}{6.02}$$

Spurious Free Dynamic Range



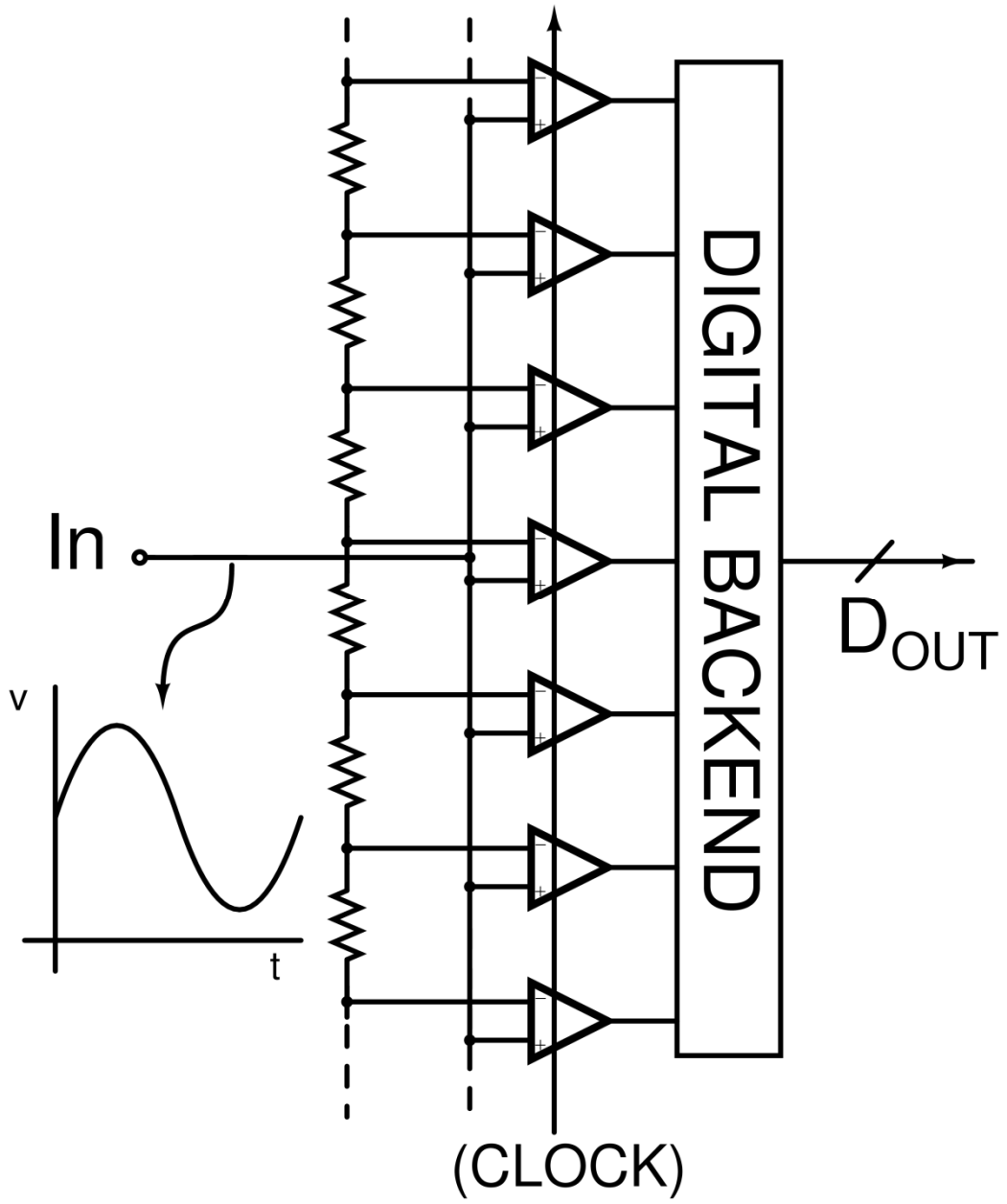
Flash A/D Conversion

(+) Parallel technique
→ Low latency

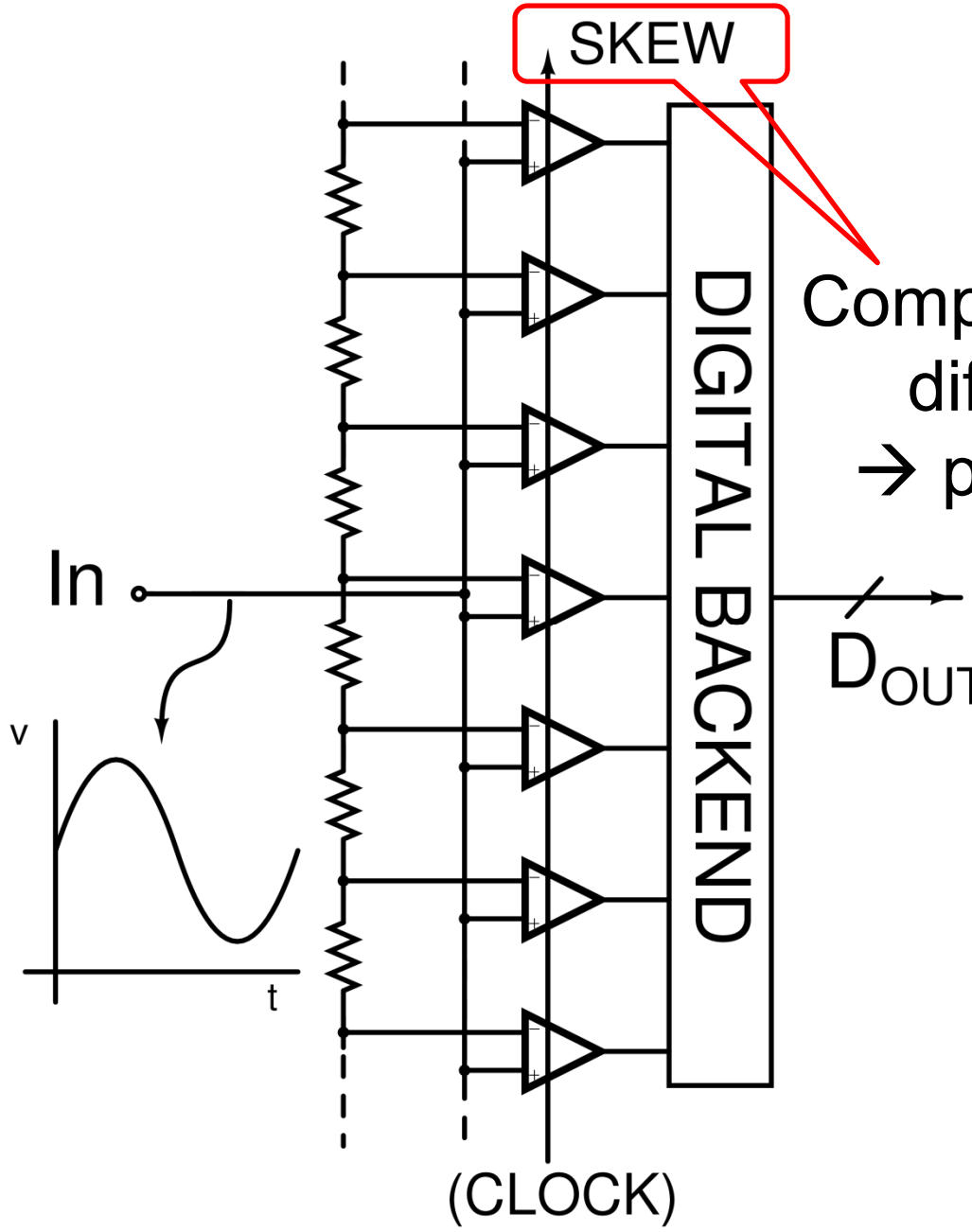
(+) References
– resistor ladder

(-) Complexity - $O(2^N)$

(-) Excessive power
and area for N > 6

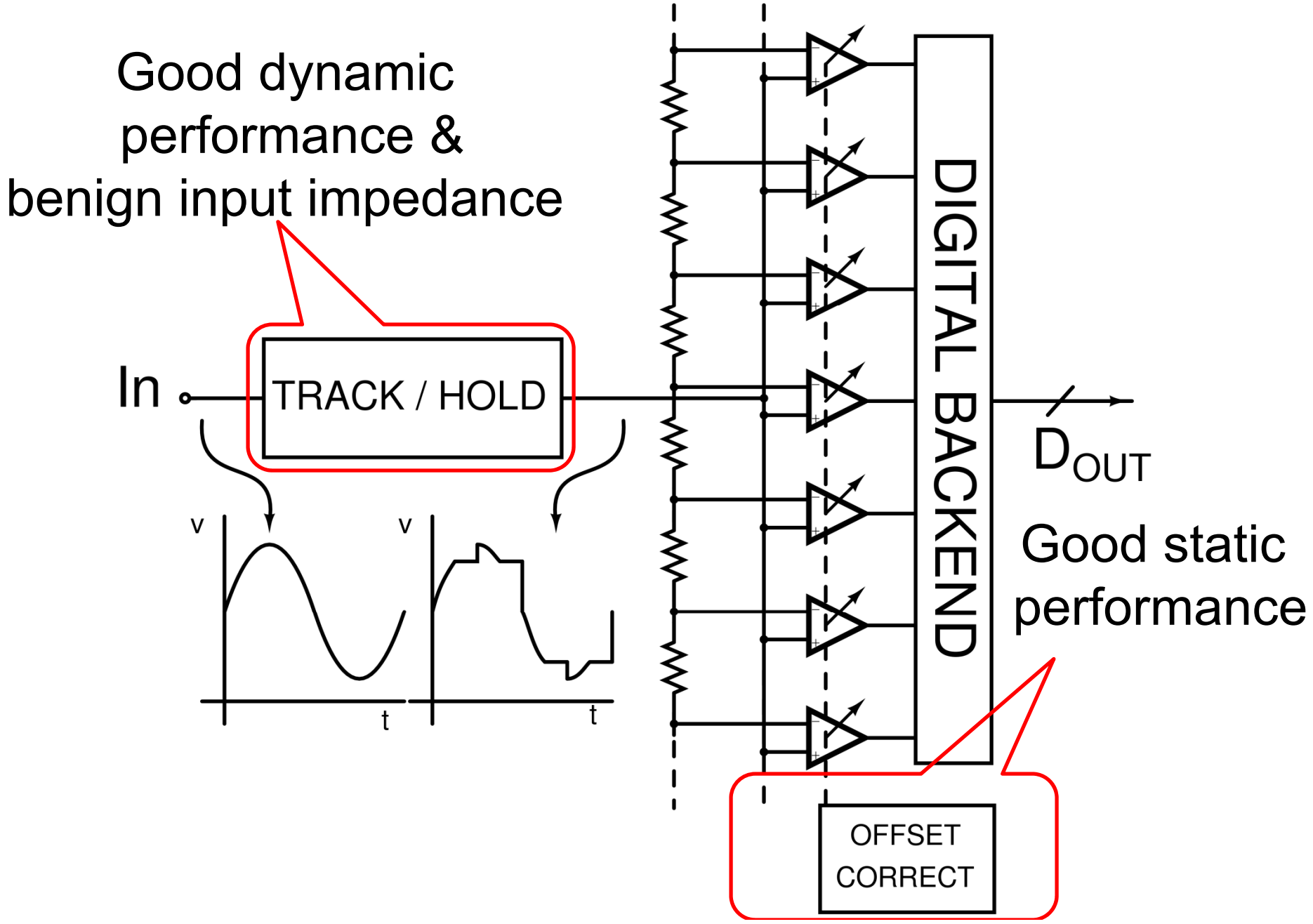


Flash ADCs : Clock skew

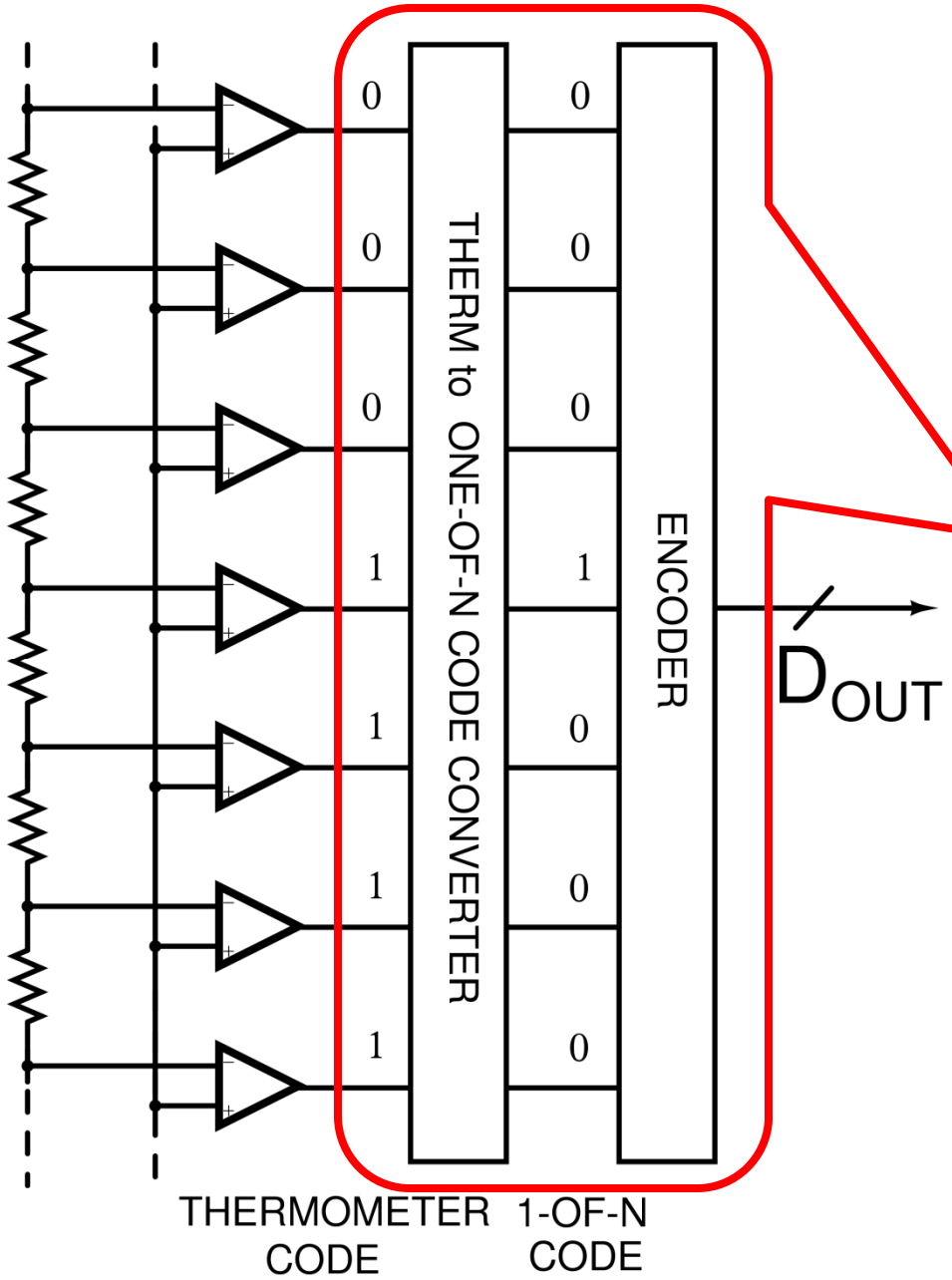


Comparators sample different inputs → poor SNDR for high f_{in}

Practical Flash ADC



Backend Logic

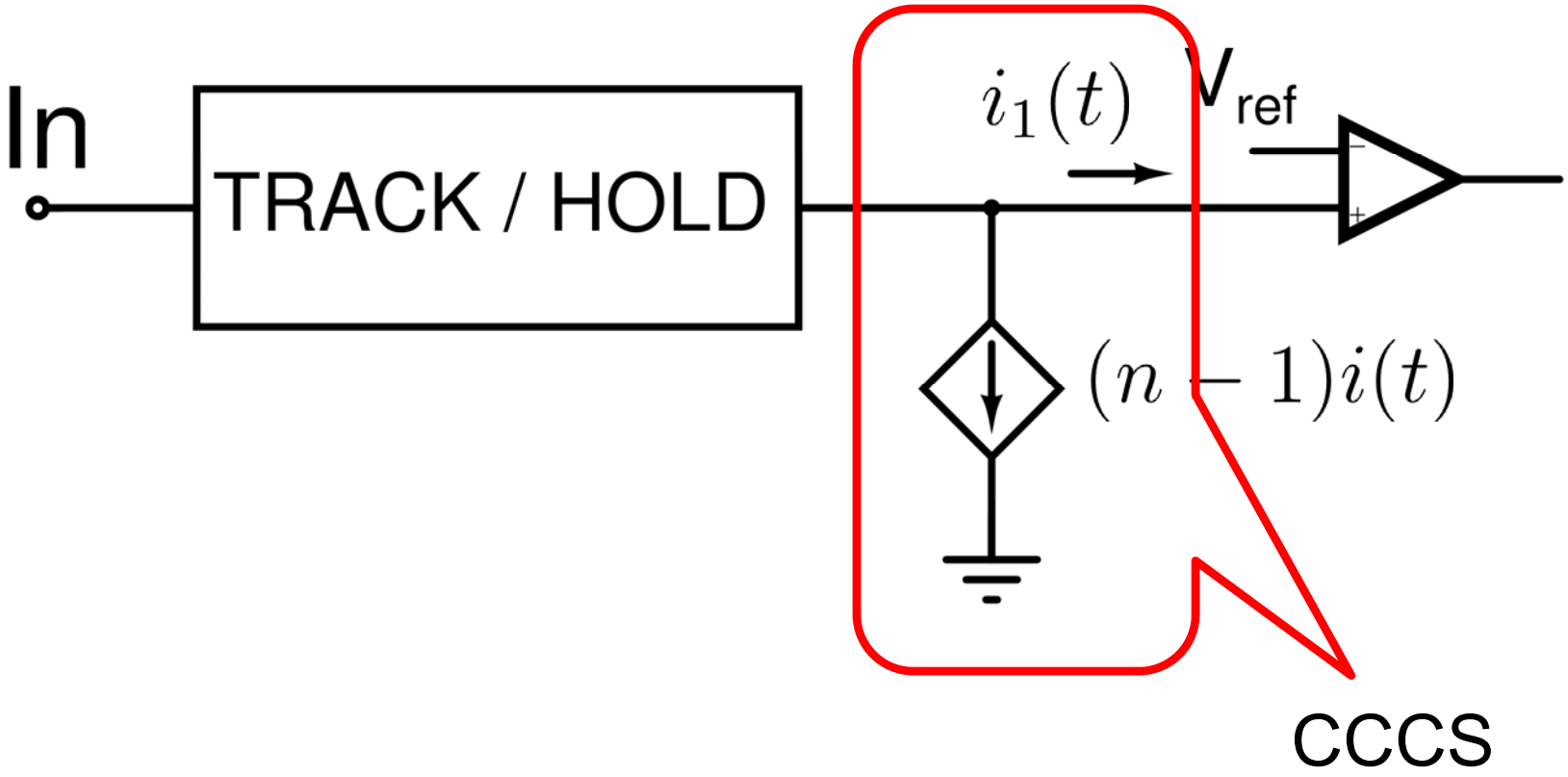


Ideal candidates for abstraction during the design phase

"Flash ADC Family" Design Steps

- Comparator design
  - Should have sufficiently low static and dynamic offset
  - Must regenerate sufficiently fast
- Track and Hold design
  - Tricky and depends on comparator architecture
- Back end logic

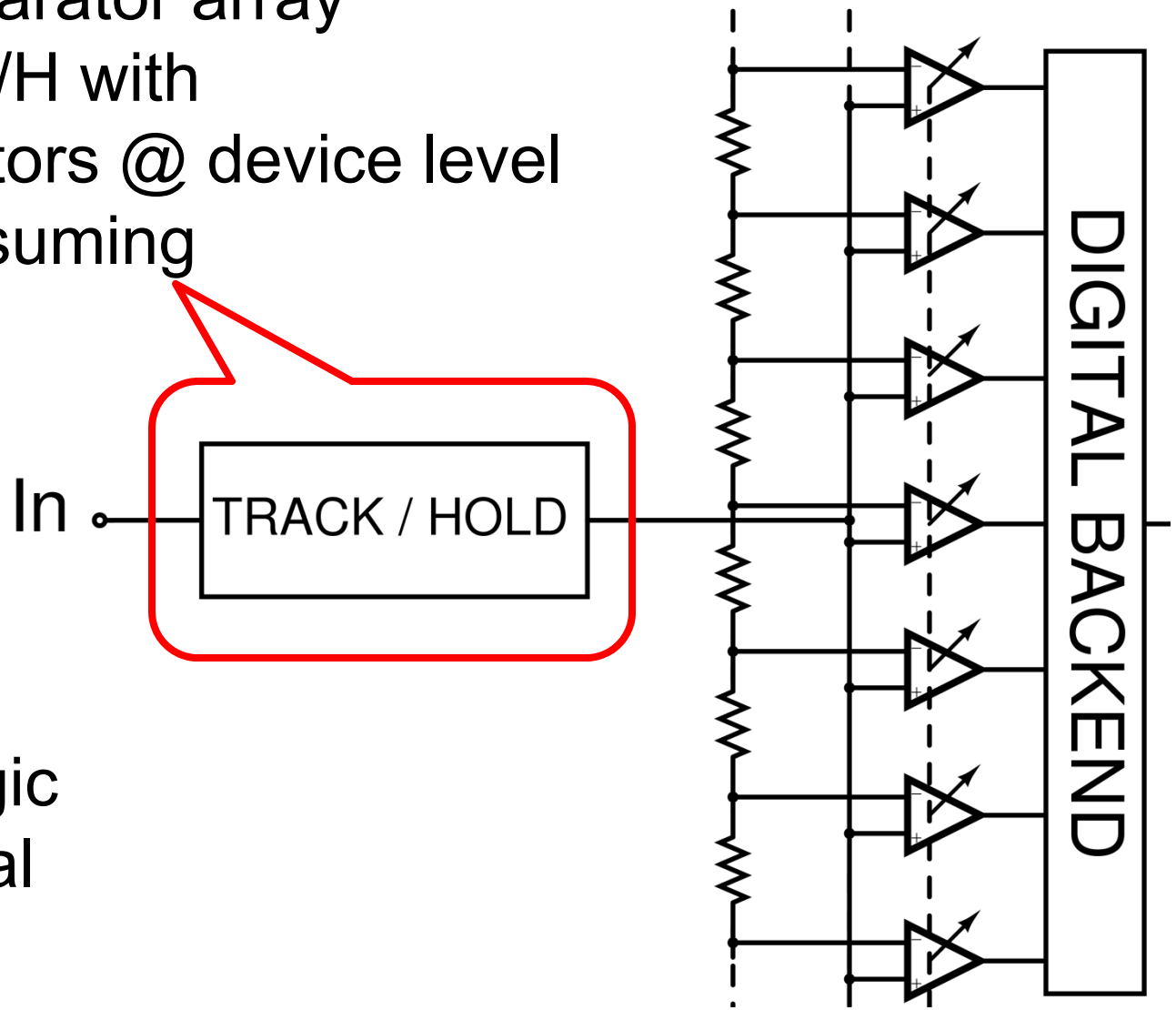
Flash ADC Simulations



- CCCS emulates the load of *n* comparators
  - with only one comparator instantiation
- Convenient during design iteration

Flash ADC Simulations

Drives comparator array
Simulating T/H with
all comparators @ device level
→ time consuming

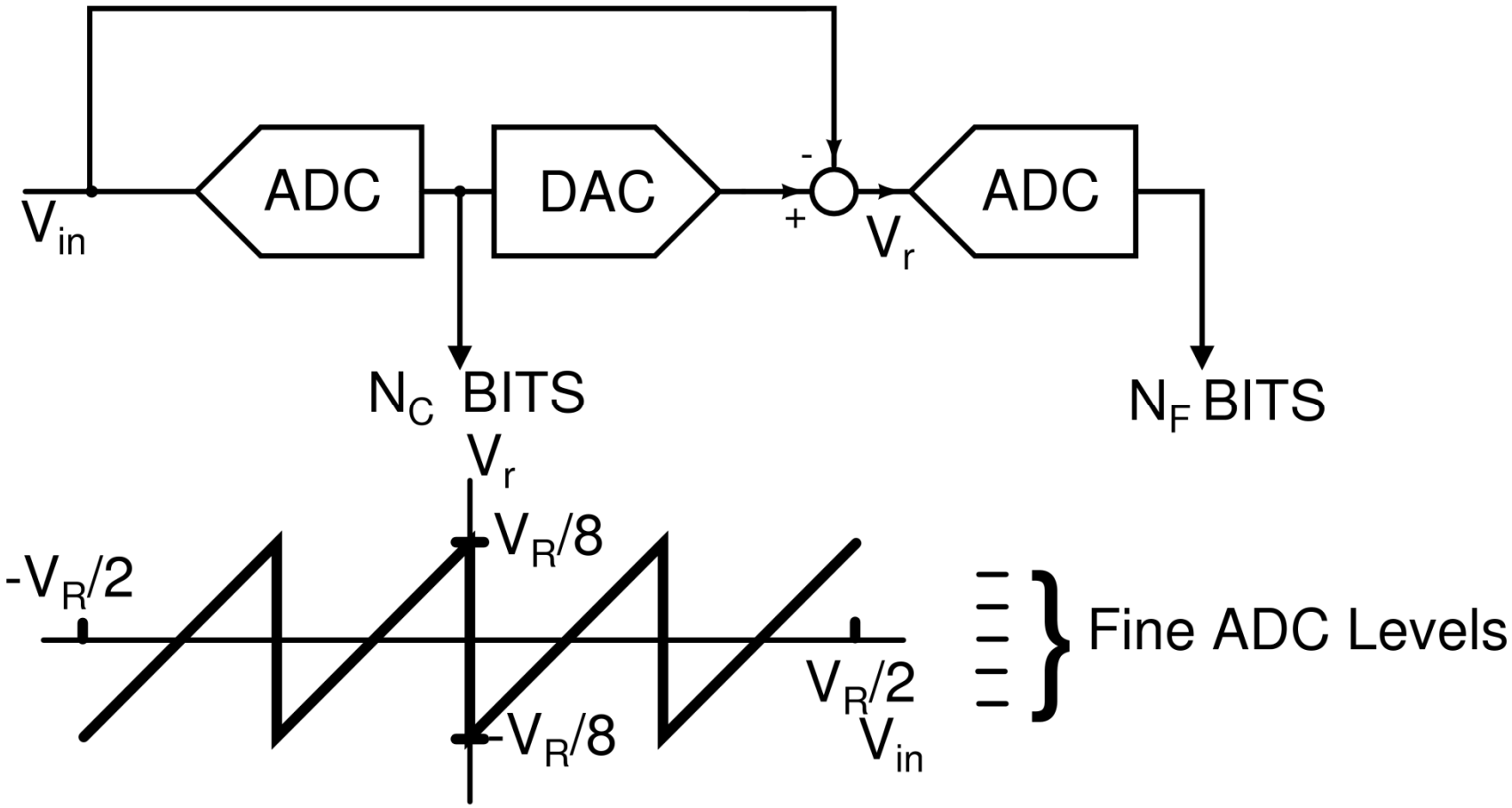


Back end logic
→ behavioral

Two Step Flash ADC

- Motivation:
  - Reduce number of comparators in a flash ADC
- Idea:
  - In a flash ADC, only comparators “near” the input give useful information
  - Use a coarse ADC to estimate where the signal is, then use a fine ADC placed “around” the coarse estimate for better accuracy

Two Step Flash ADC



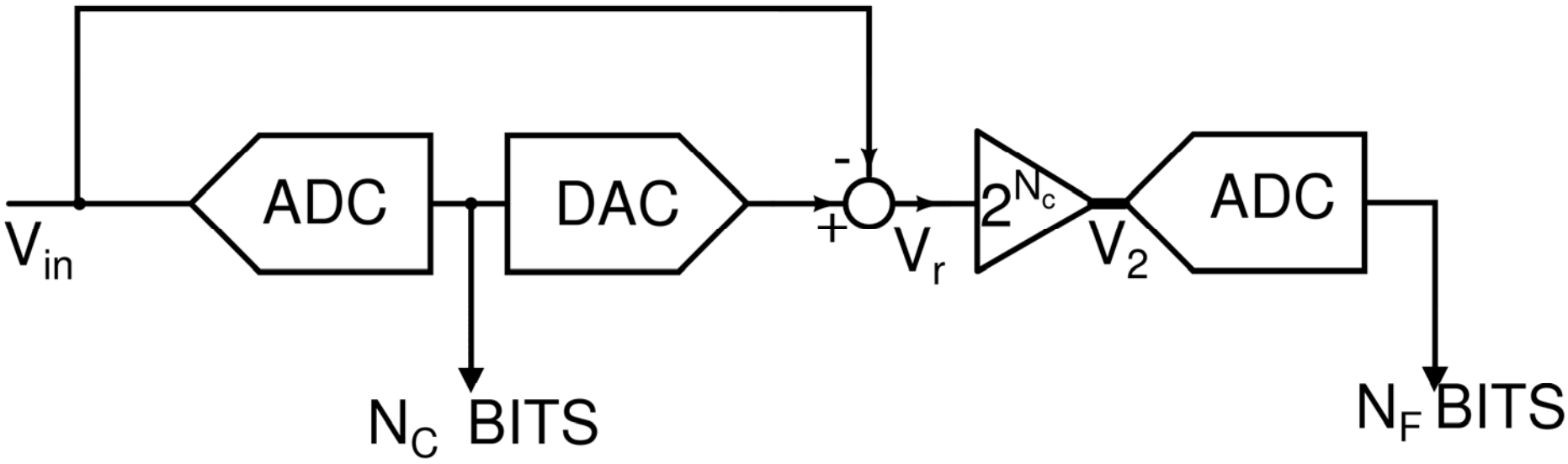
Number of comparators : $(2^{N_c} + 2^{N_f} - 2)$

Resolution $N_c + N_f$ bits

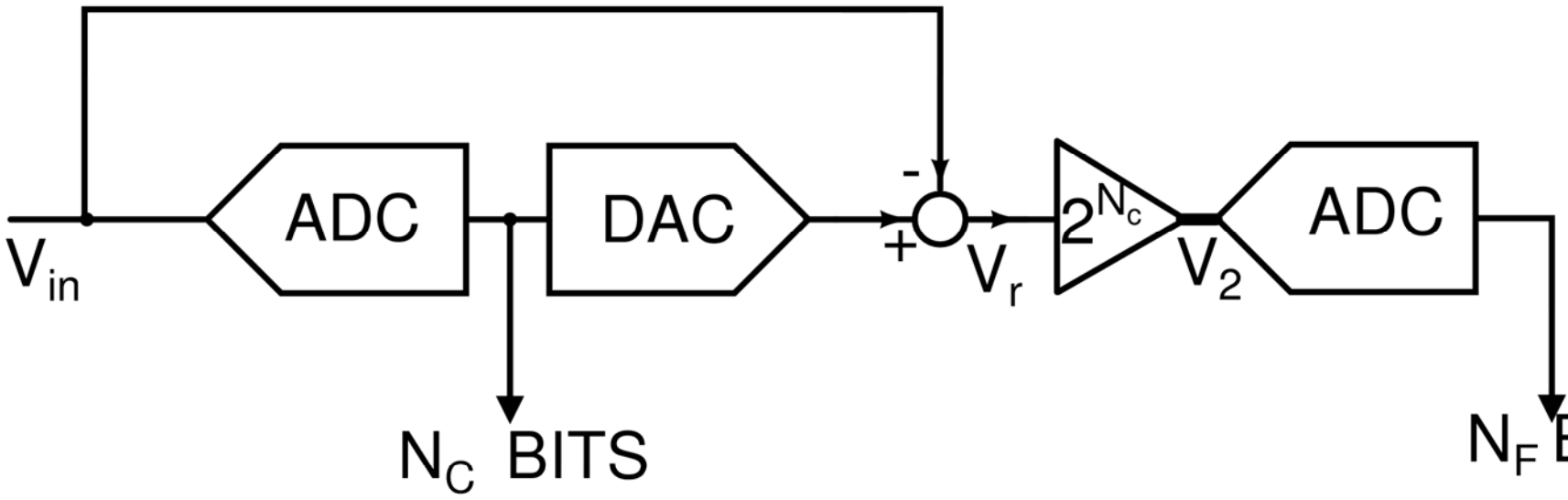
ADCs & DAC must be good to $(N_c + N_f)$ bits

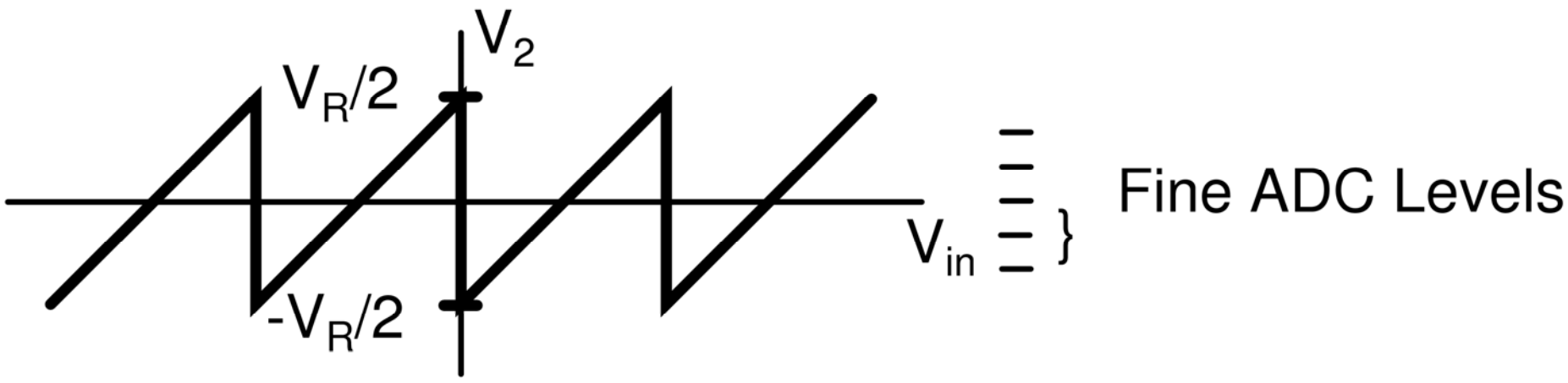
Improved Two Step Flash ADCs

- The fine ADC operates on a small input
  - Offset requirements for the fine ADC can be relaxed if the input signal swing was larger
  - Amplify the input to the fine ADC

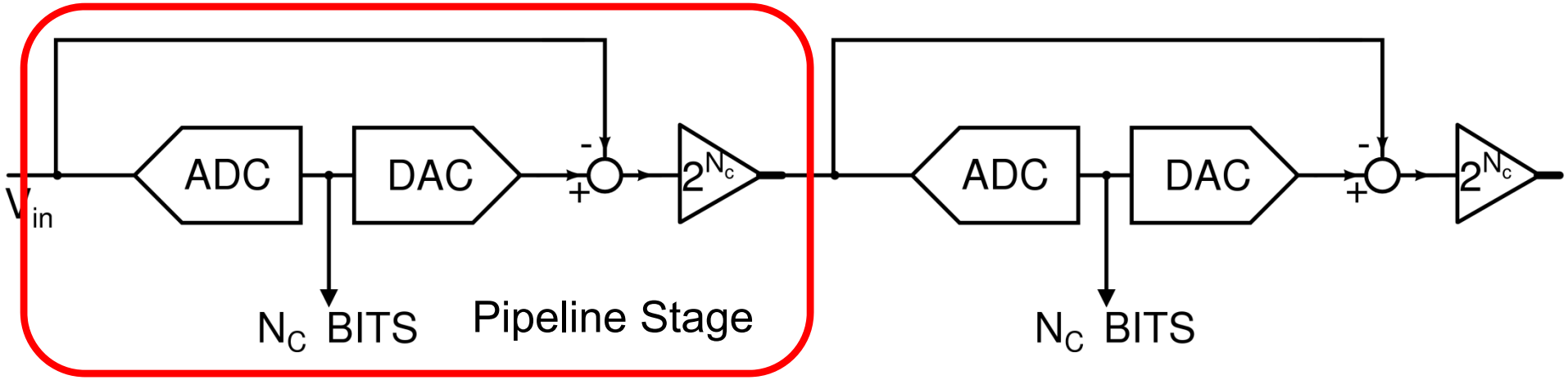


Pipeline ADC Principle





Pipeline ADC Principle

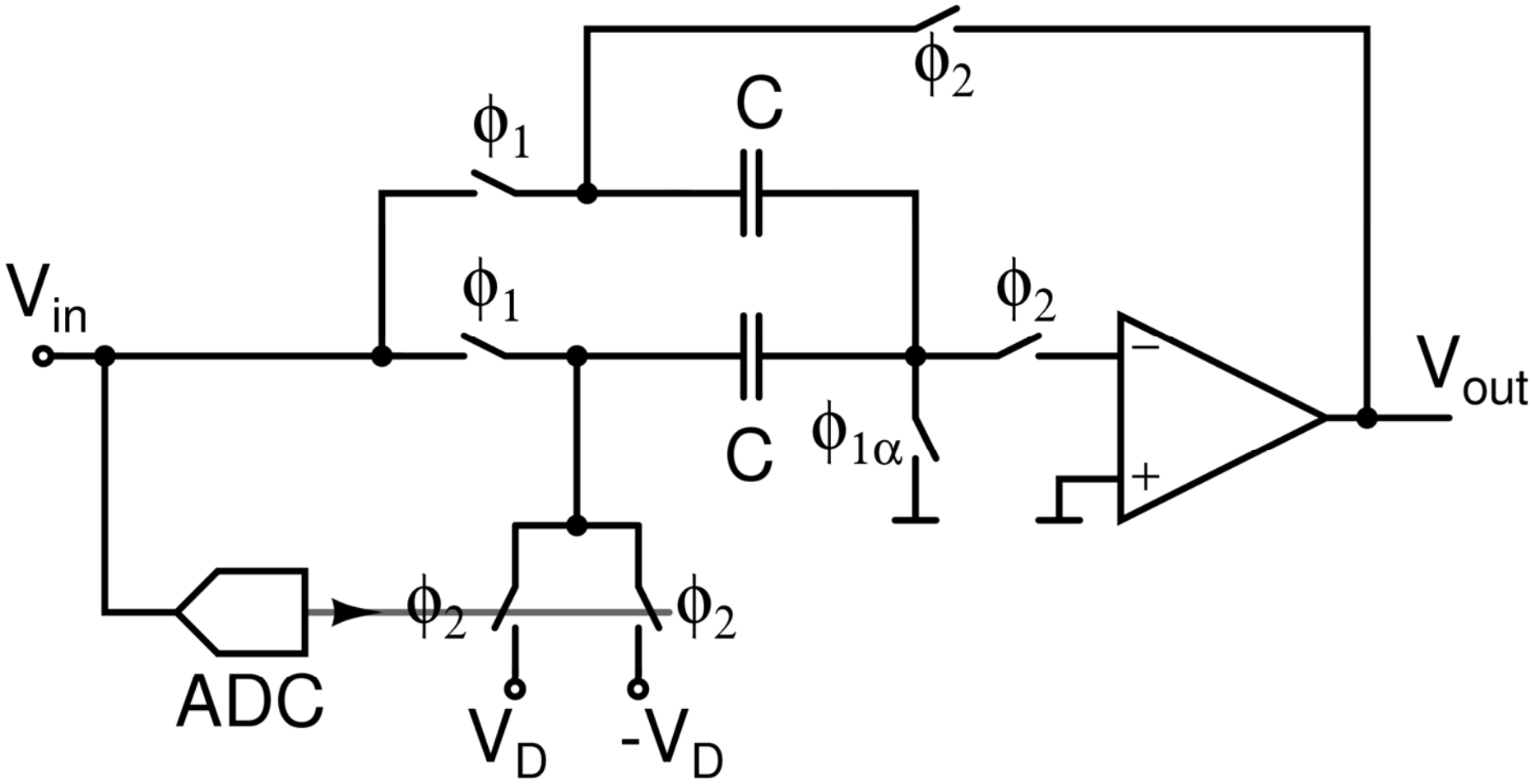


- Recursive implementation of the fine ADC
- Each stage implemented using switched capacitor techniques
- State of the art pipelines based on sophisticated digital correction and calibration algorithms

Pipeline ADC Design Flow

- High level design exploration at the behavioral level
  - Bits per stage
  - Opamp gain, linearity and bandwidth requirements
  - Capacitor matching studies
  - Digital calibration algorithms
- Stage Design
  - Opamp
  - Coarse ADC & DAC

Example Pipeline ADC Stage

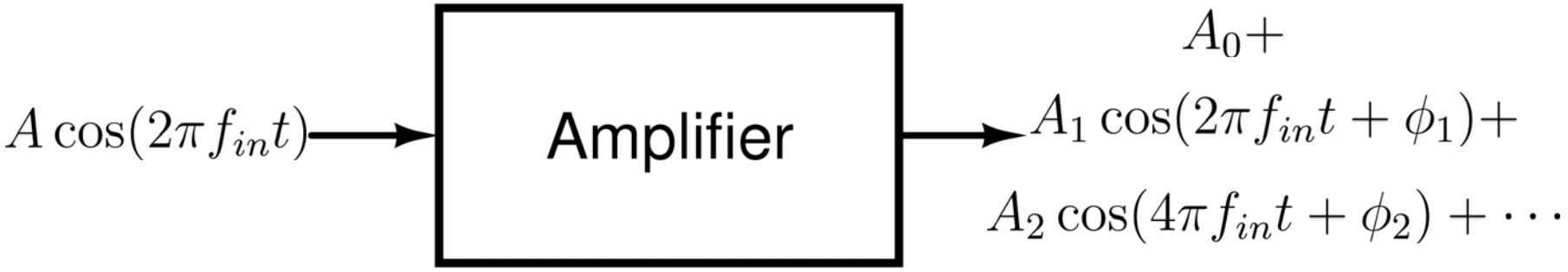


DAC, Subtraction and Amplifier using
one switched capacitor stage : SPICE it carefully

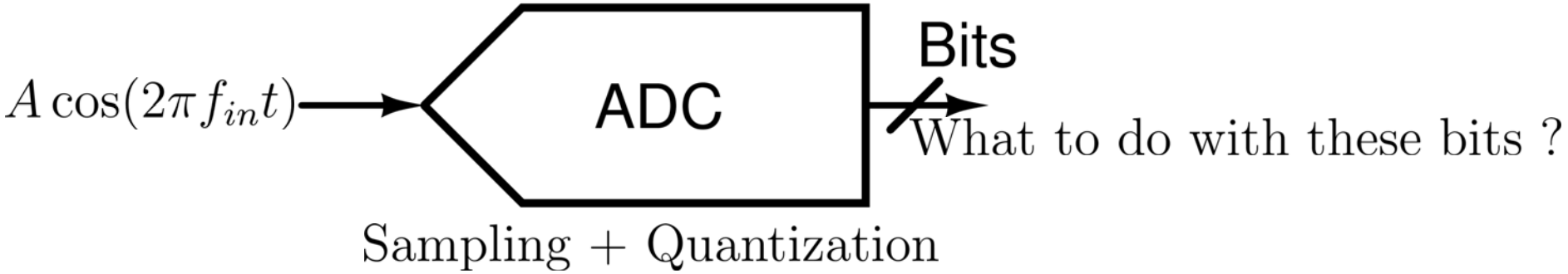
Spectral Analysis in ADC Design

© 2013 IEEE

ADC Simulation: Black Box Testing



Fourier Analysis



Basics

A discrete-time signal is periodic with period *P* if

$$s[n] = s[n+P]$$

It can be expanded into the *discrete* Fourier Series

$$s[n] = \sum_{k=0}^{k=P-1} A_k \cos\left[2\pi \frac{k}{P} n + \phi_k\right]$$

A discrete-time periodic signal with period *P* has only *P* coefficients in its Fourier expansion

A_k, ϕ_k can be rapidly computed using the FFT

Basics

$$s[n] = \sum_{k=0}^{k=P-1} A_k \cos\left[2\pi \frac{k}{P} n + \phi_k\right]$$

$s(t)$ sampled at f_s

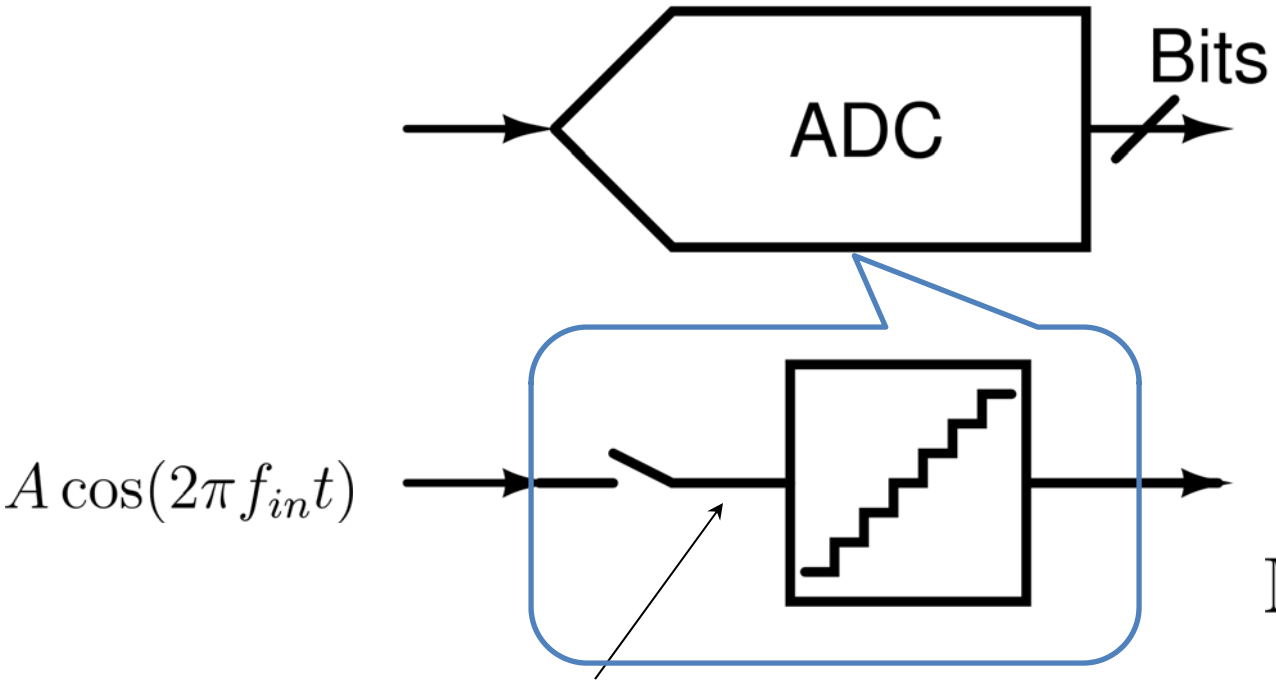
Jargon

P is the FFT record length

$\frac{f_s}{P}$ is the FFT bin width

© 2013 IEEE

Choice of f_{in}



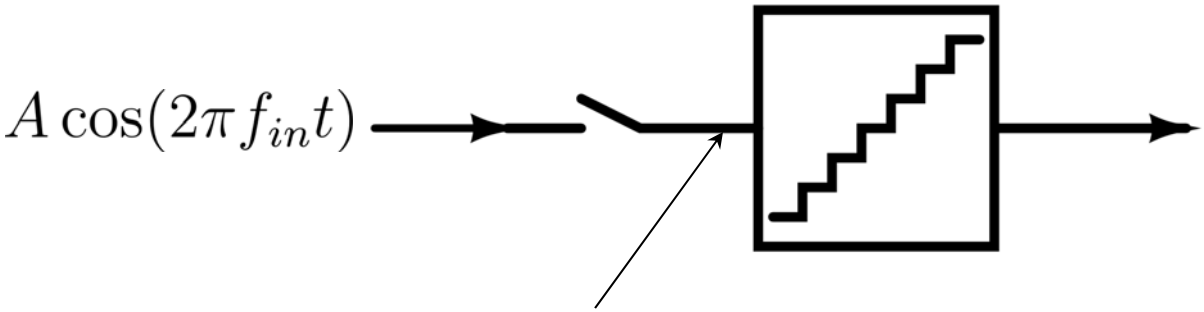
Multiple of 2π

$$A\cos\left[2\pi\frac{f_{in}}{f_s}n\right] = A\cos\left[2\pi\frac{f_{in}}{f_s}(n+P)\right] = A\cos\left[2\pi\frac{f_{in}}{f_s}n + 2\pi\frac{f_{in}}{f_s}P\right]$$

$$\text{periodic only if } \frac{f_{in}}{f_s} = \frac{m}{P}$$

Input and sampling frequencies must be rationally related

Choice of f_{in}



Multiple of 2π

$$A\cos\left[2\pi\frac{f_{in}}{f_s}n\right] = A\cos\left[2\pi\frac{f_{in}}{f_s}(n+P)\right] = A\cos\left[2\pi\frac{f_{in}}{f_s}n + 2\pi f_{in}\frac{P}{f_s}\right]$$

$$\text{only if } \frac{P}{f_s} = \frac{m}{f_{in}}$$

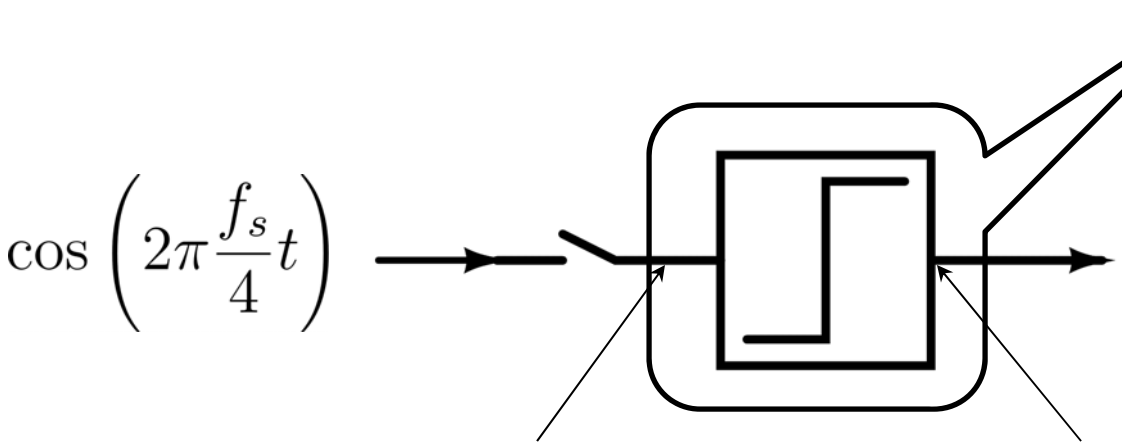
Integral number of input cycles in the length of a record

$$\frac{f_{in}}{f_s} = \frac{m}{P} \quad \text{Is } \mathbf{any} \ \frac{m}{P} \text{ good ?}$$

Choice of f_{in}

$$\text{Example : choose } \frac{f_{in}}{f_s} = \frac{m}{P} = \frac{1}{4}$$

Very bad quantizer !

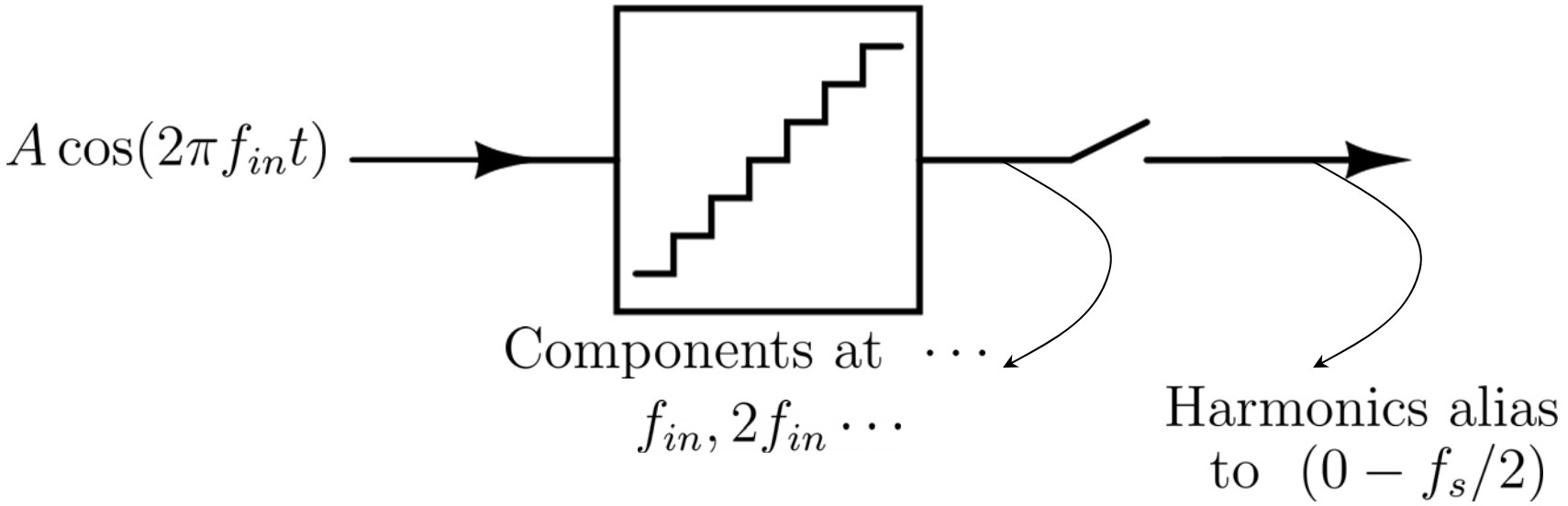


$$\cos\left[2\pi\frac{f_s}{4}\frac{n}{f_s}\right] = \cos\left[\frac{\pi}{2}n\right] = 1, 0, -1, 0\cdots \quad 1, 0, -1, 0\cdots = \cos\left[\frac{\pi}{2}n\right]$$

Predicts zero distortion !

Why ?

Choice of f_{in}



For a wrongly chosen *m*, multiple harmonics alias to the same frequencies after sampling

$$\text{If } f_{in} = f_s/4 \text{ all odd harmonics alias to } f_s/4$$

Moral : Choose f_{in} so that its significant harmonics do not alias to the same locations

Choice of f_{in}

- (f_{in} / f_s) must be rational (m/P)
- *Choose m and P to be relatively prime*
- Convenient for FFT : choose $P = 2^Q$

$$f_{in} = \frac{m}{2^Q} f_s$$

Jargon : Input tone “lies on a bin”

© 2013 IEEE

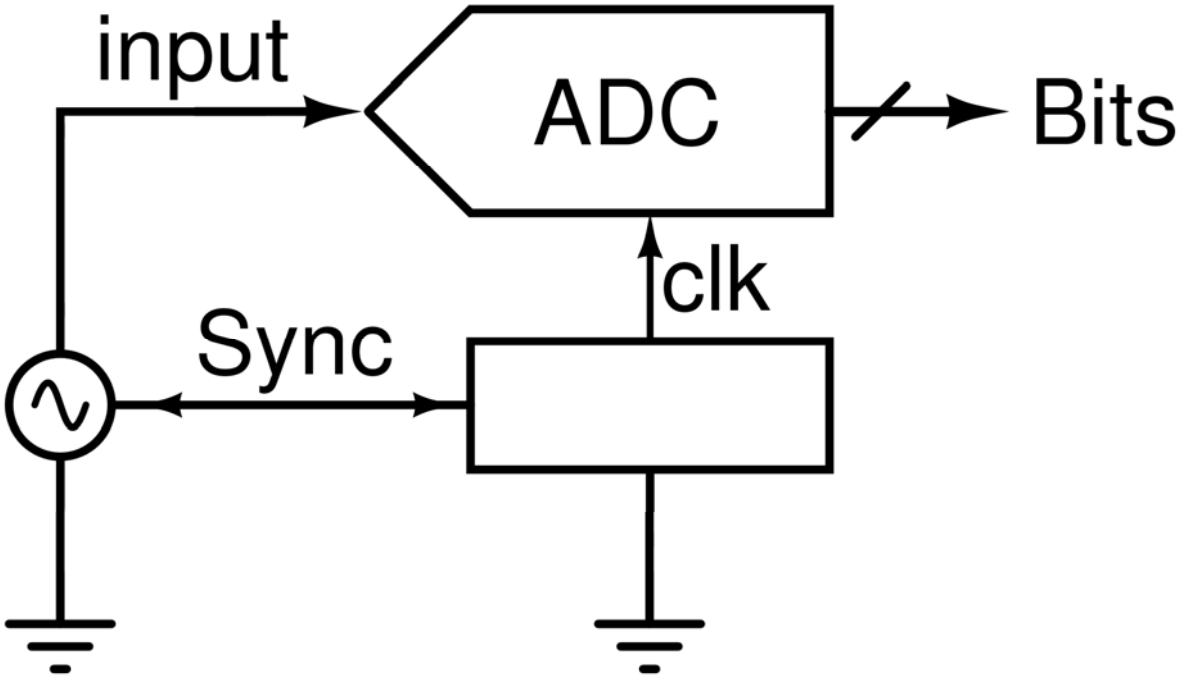
Example : Choice of f_{in}

- 6-bit ADC sampling rate (f_s) = 1 GS/s
- Want to test around 250 MHz
- Choose record length = 1024

$$f_{in} = \frac{253}{1024} f_s = 247.0703125\,\mathrm{MHz}$$

- Do not round off !

Test Setup for a Nyquist ADC



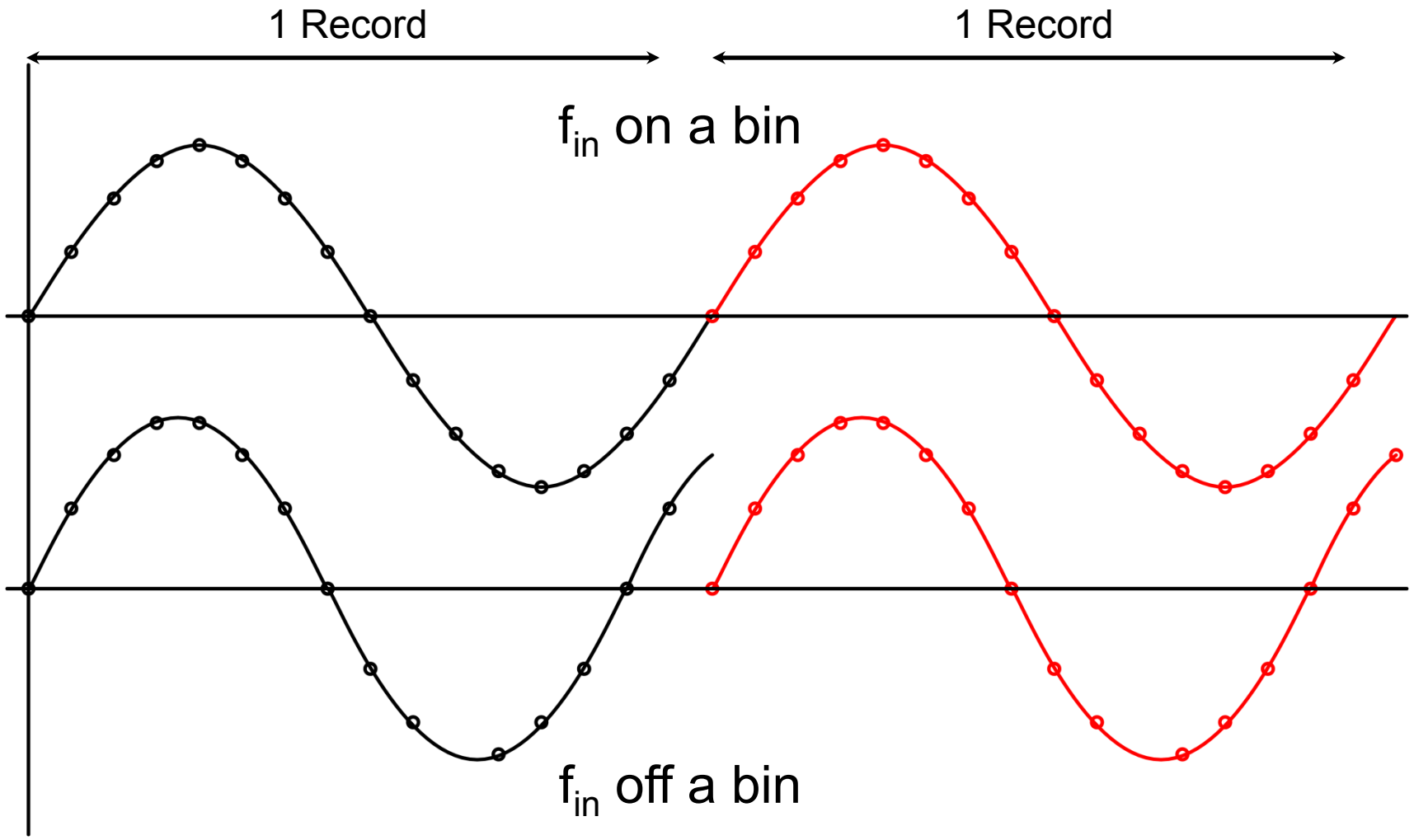
ADC input and clock source must be synchronous

- What happens when f_{in} & f_s are not rationally related ? (a.k.a f_{in} is “not on a bin”)

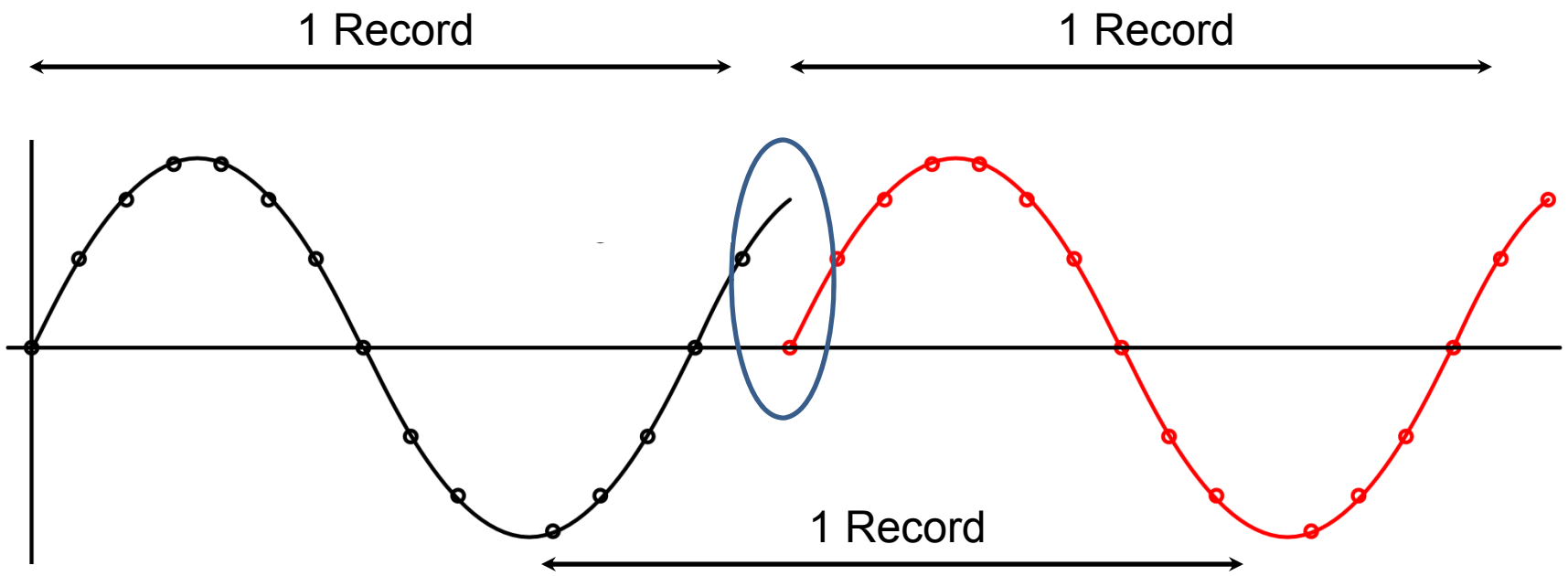
Aside

- What are the implications of computing the FFT of an *P* point sequence ?
- You assume that the sequence is **one period** of a periodic discrete-time sequence with period *P*
- You are computing the coefficients of the discrete Fourier series of this periodic sequence

FFT Leakage

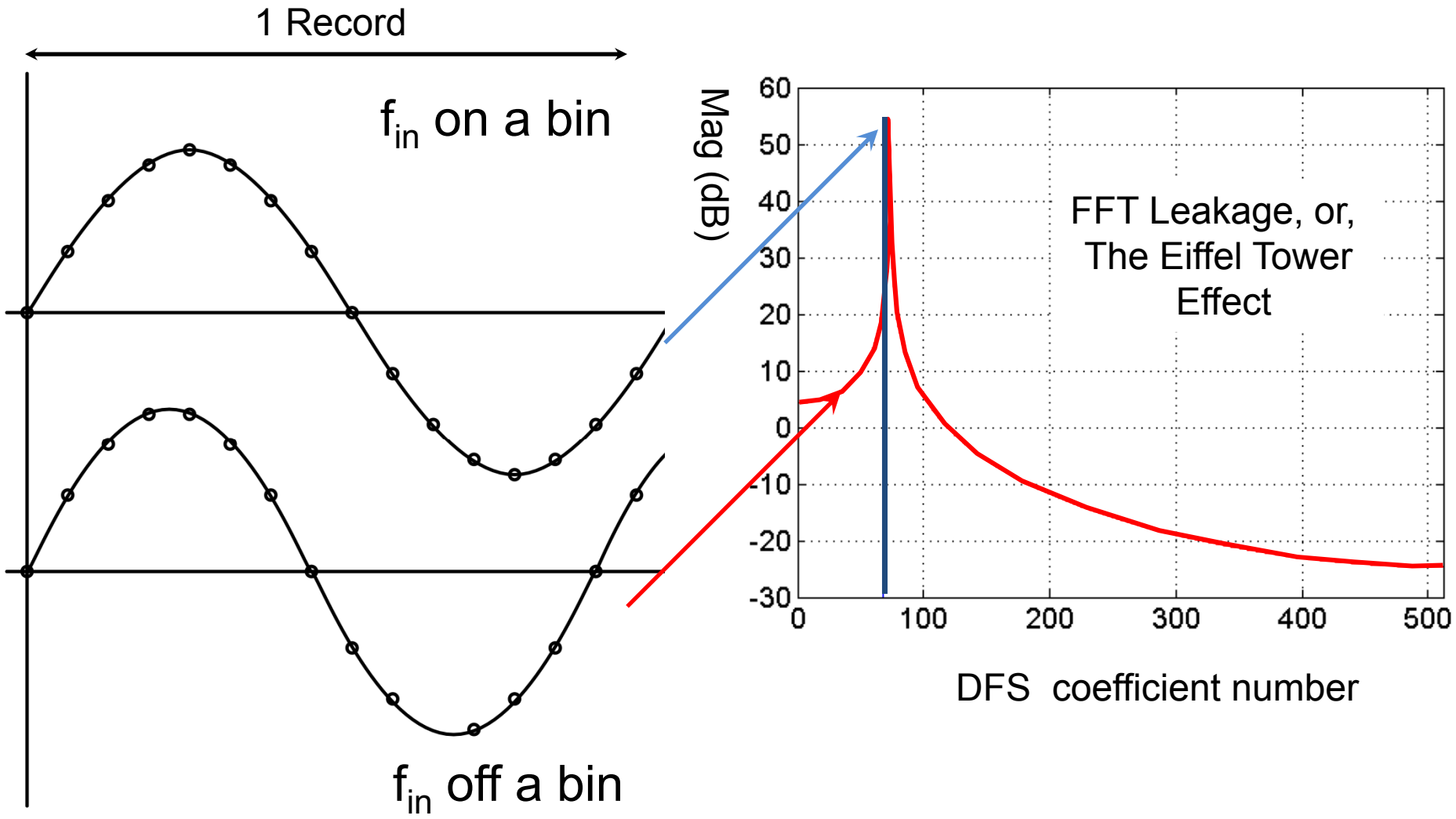


FFT Leakage

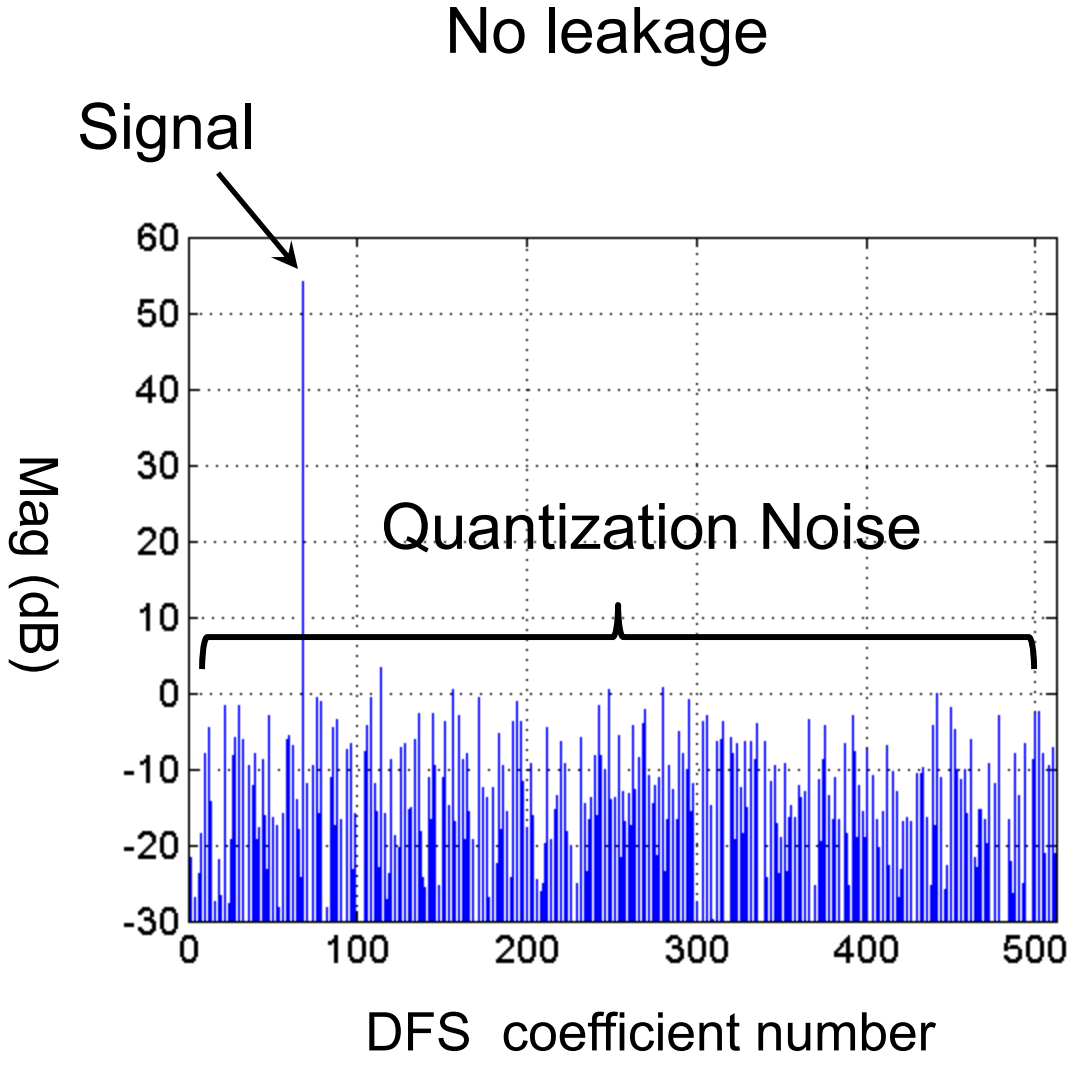


Jump here – impulse – lots of spectral energy everywhere
Called FFT “leakage”

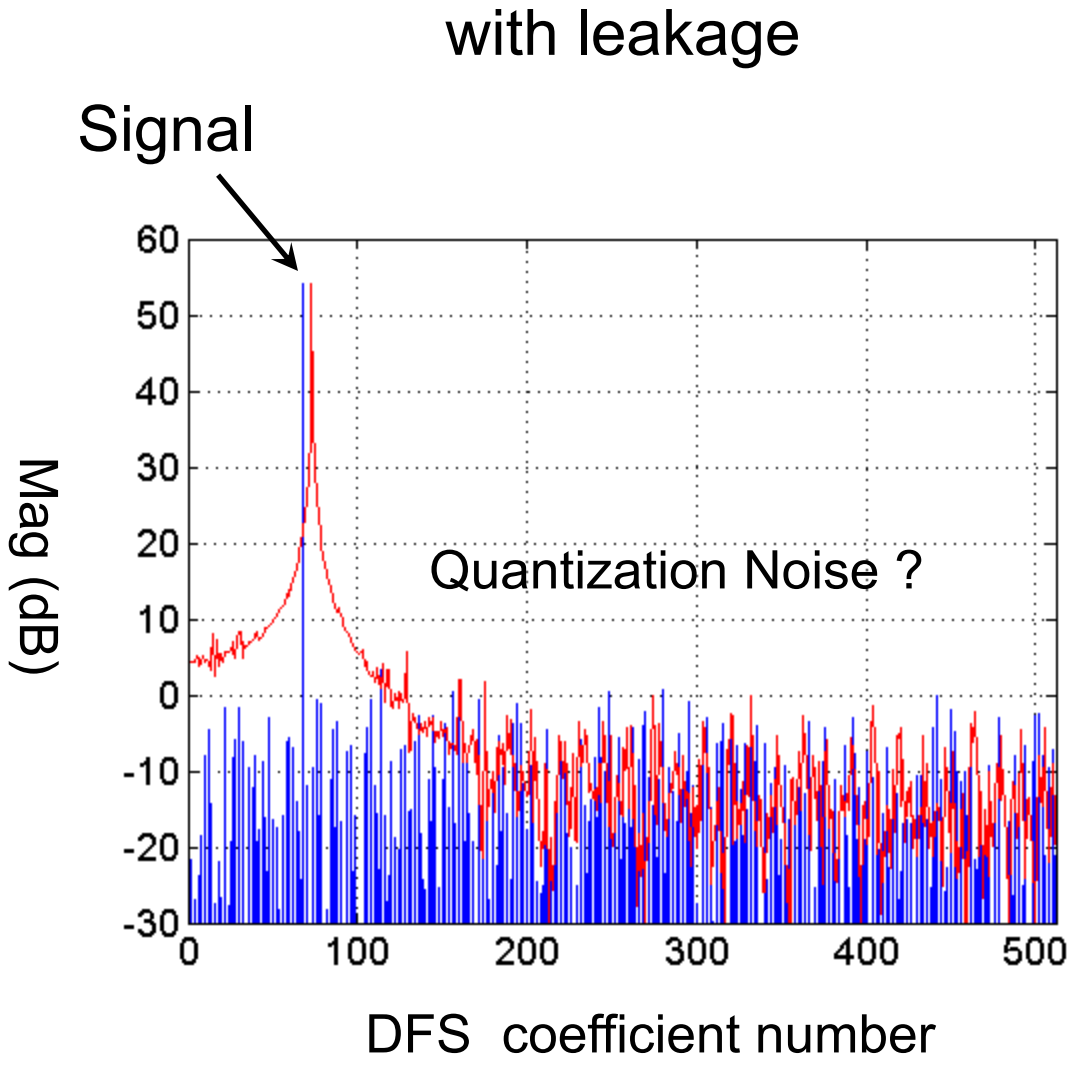
FFT Leakage



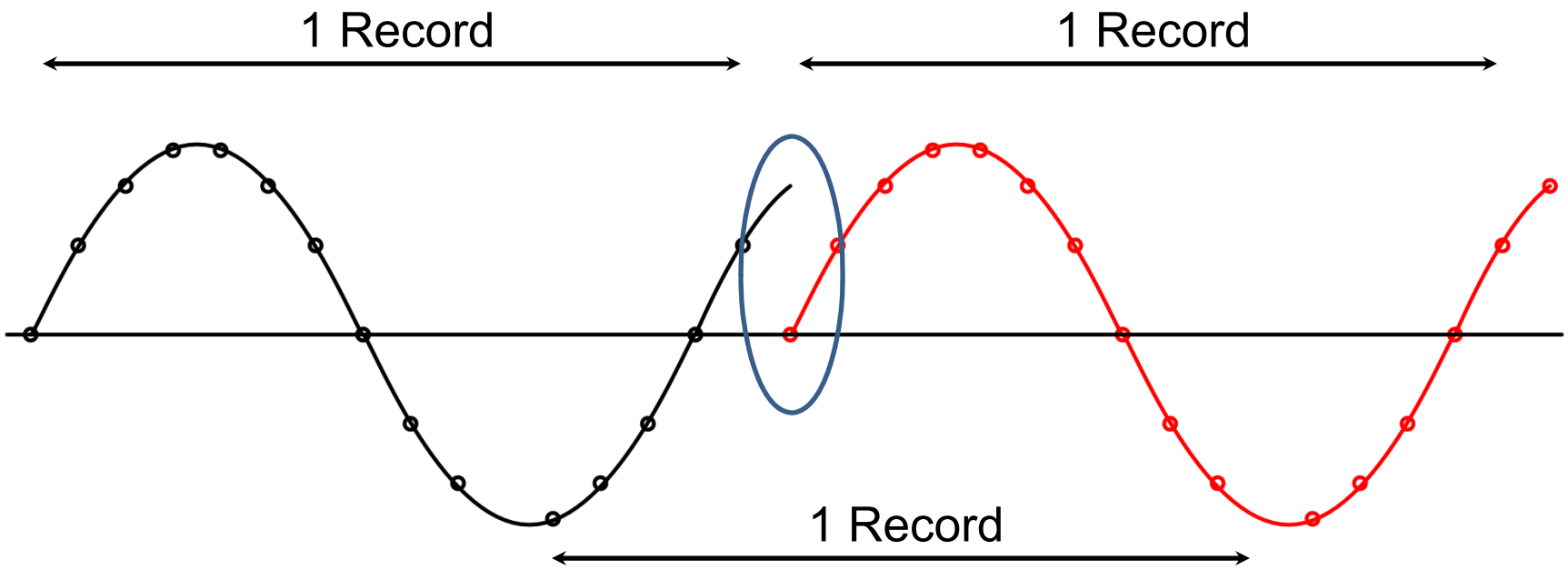
Why is Leakage Problematic ?



Why is Leakage Problematic ?



FFT Leakage : Solution

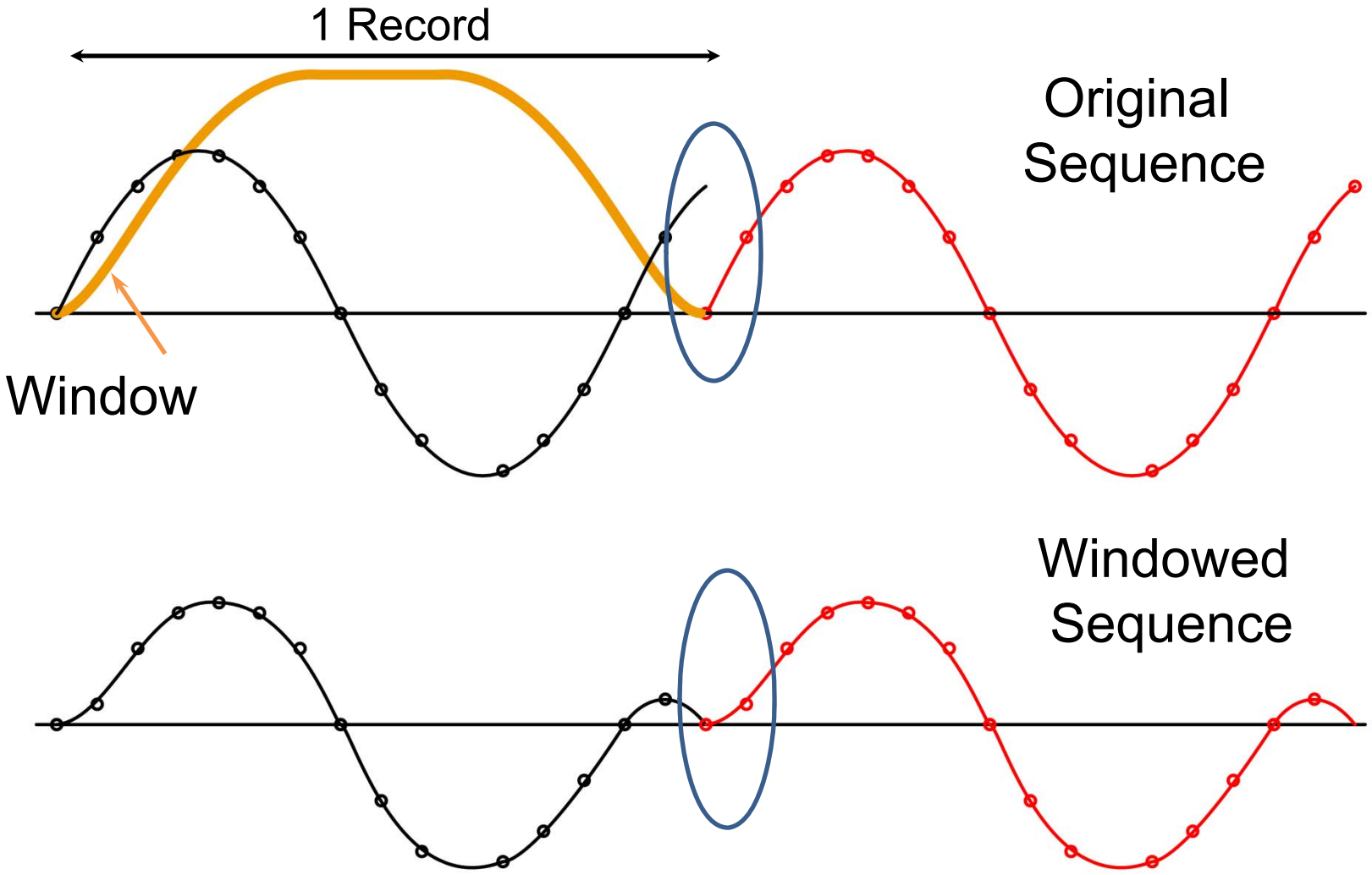


Basic Idea :

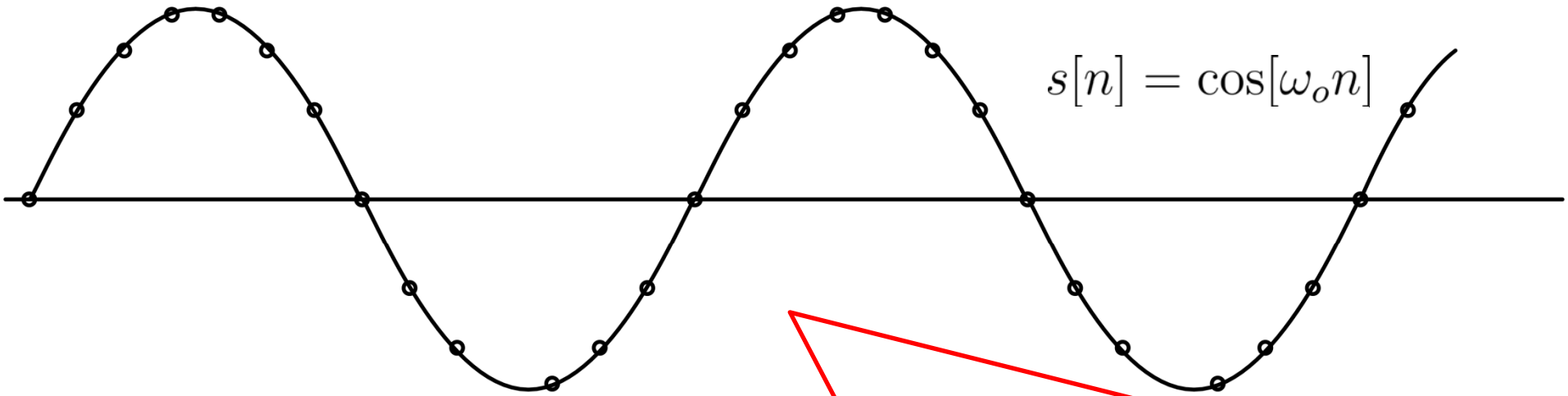
Data at the ends of the record are problematic

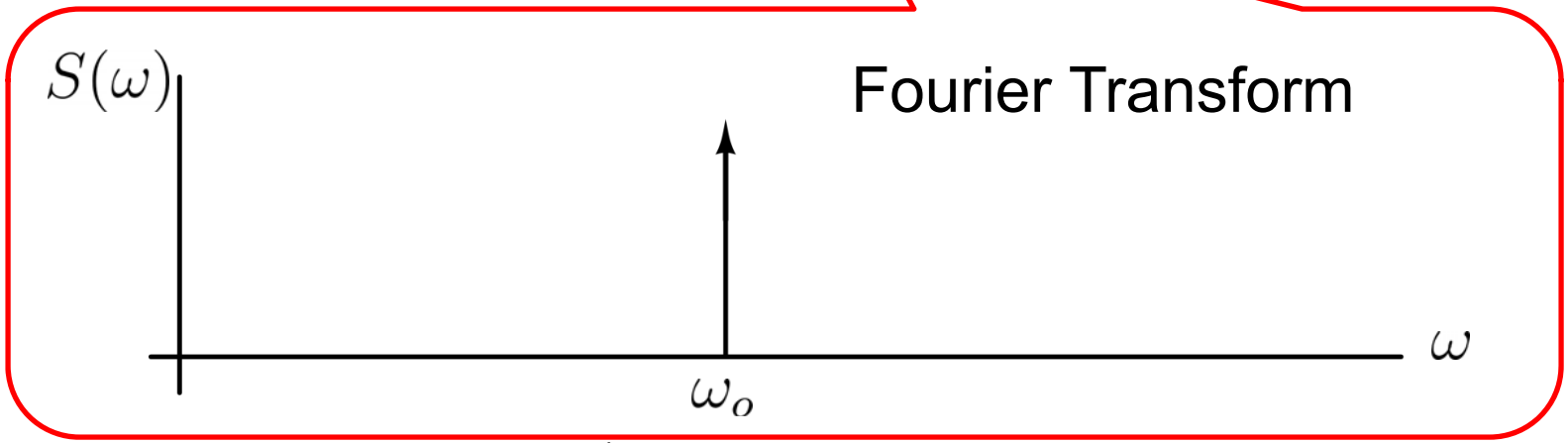
→Pay less attention to the ends of the record

Windowing



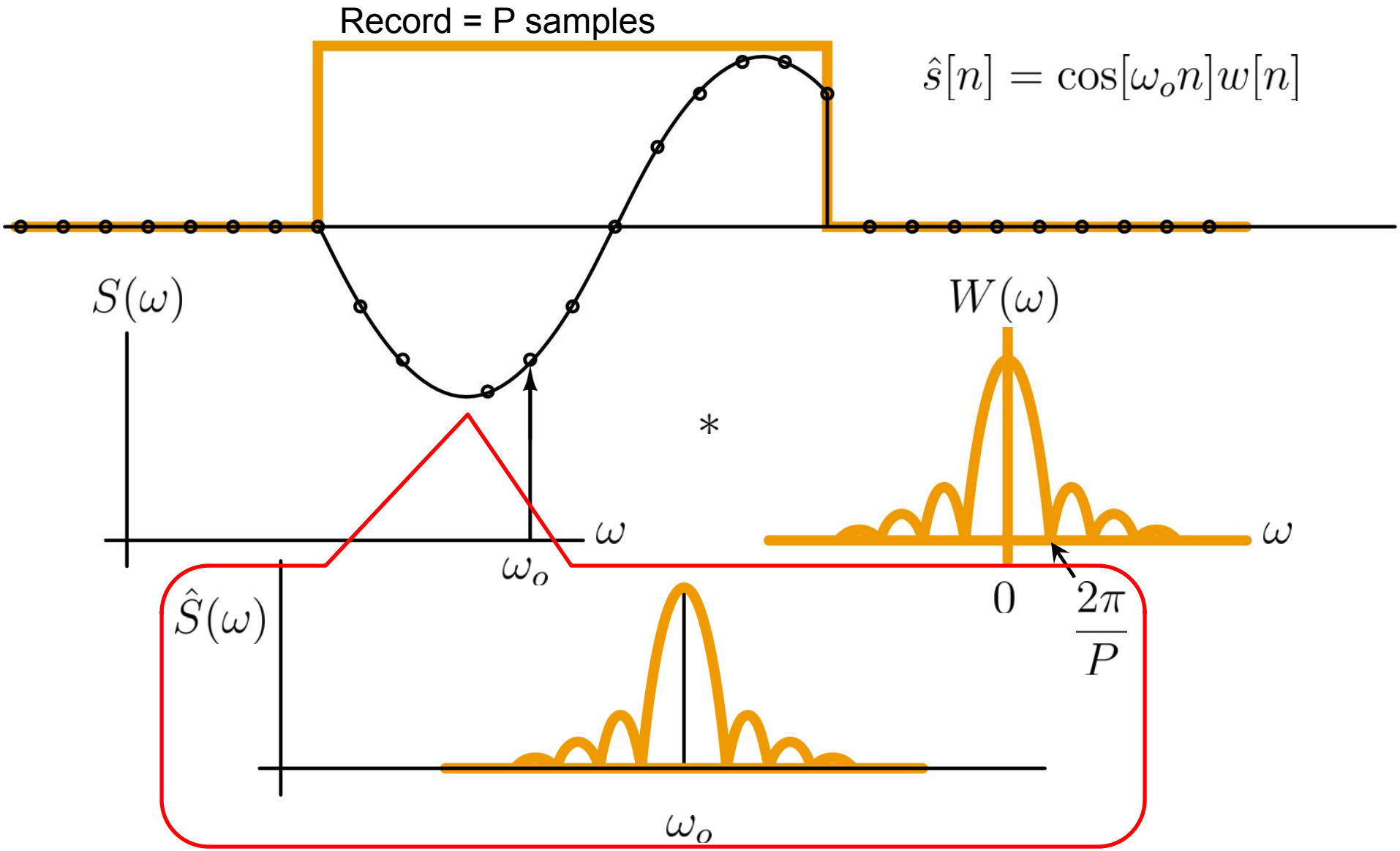
Leakage : A spectral view



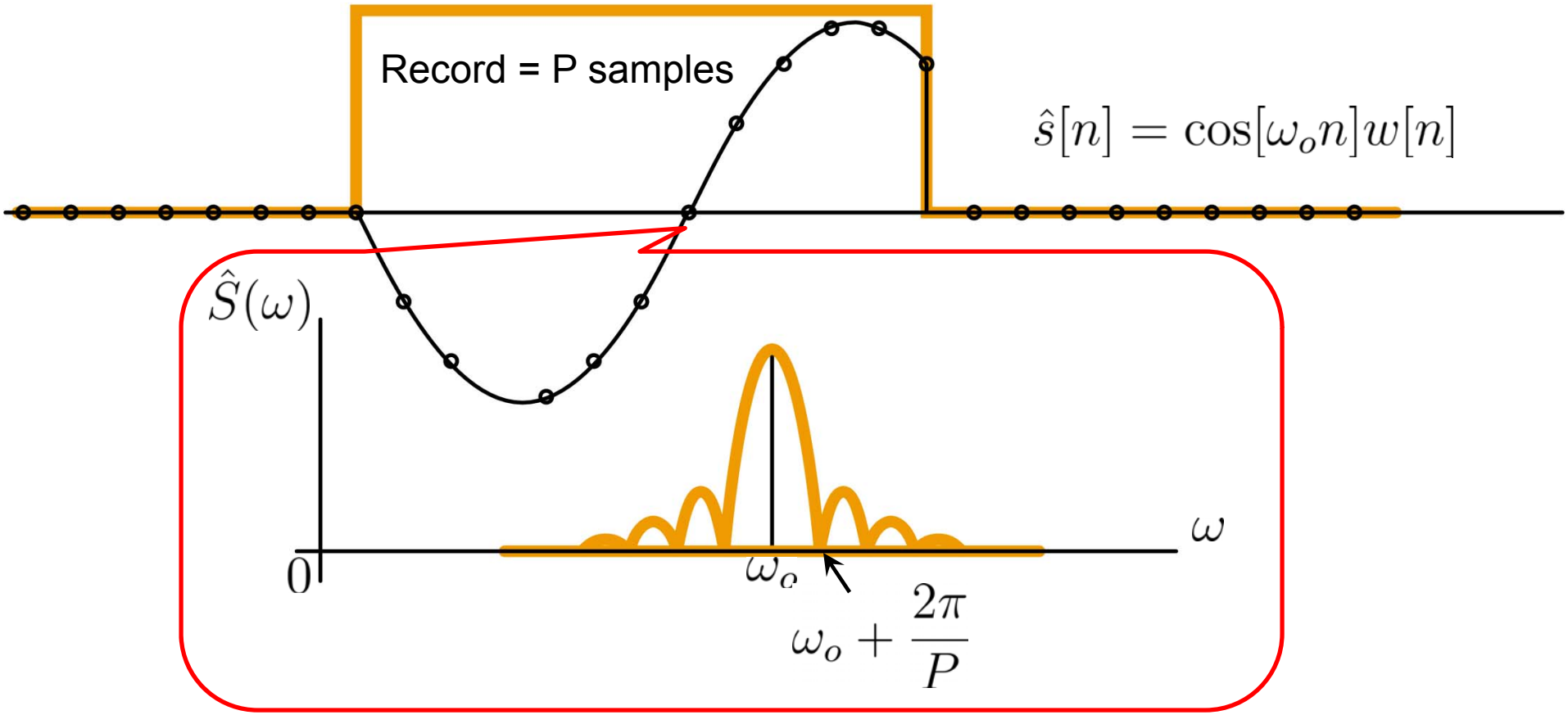


$$S[\omega] = \frac{1}{2}\left(\delta(\omega - \omega_o) + \delta(\omega + \omega_o)\right)$$

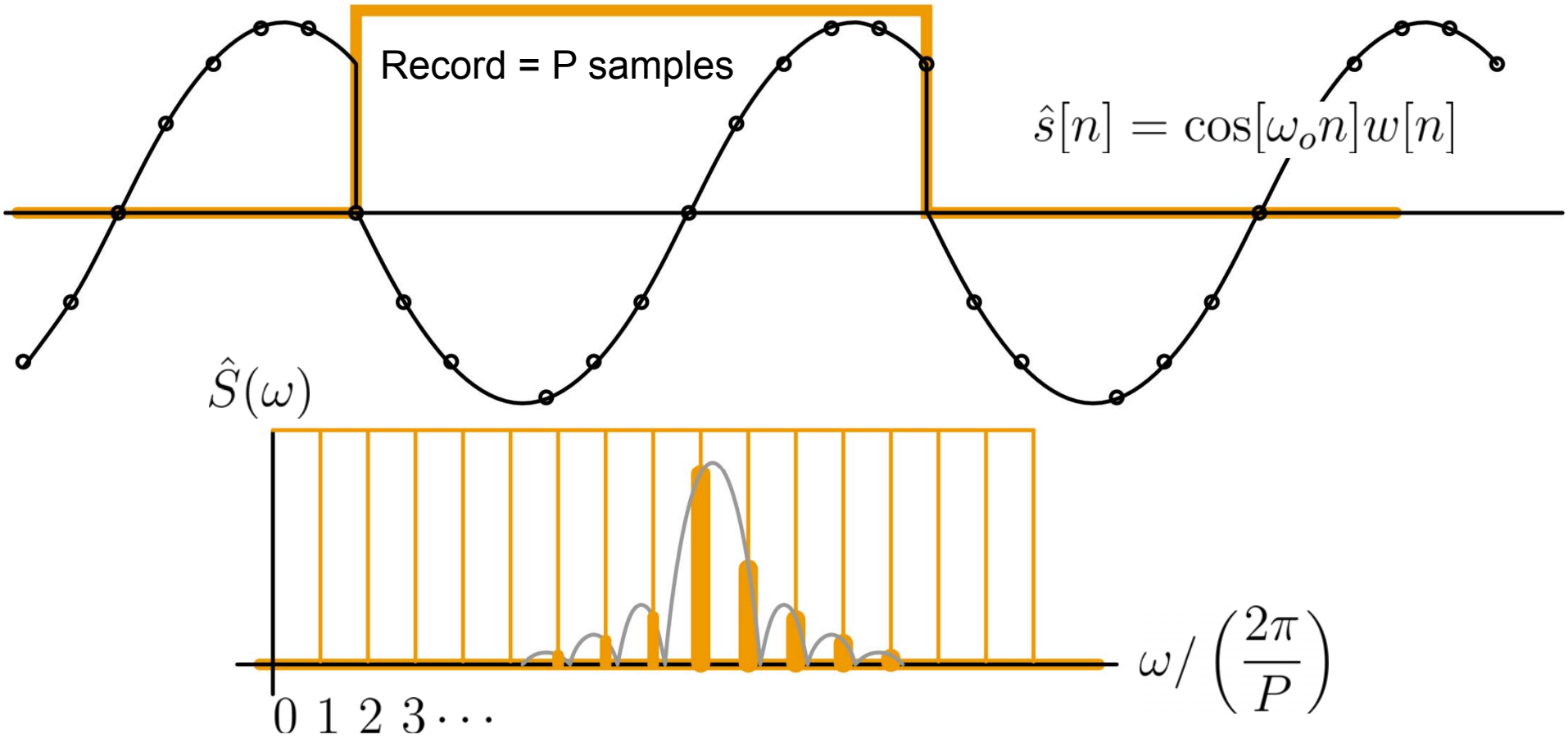
Leakage : A spectral view



Leakage : A spectral view

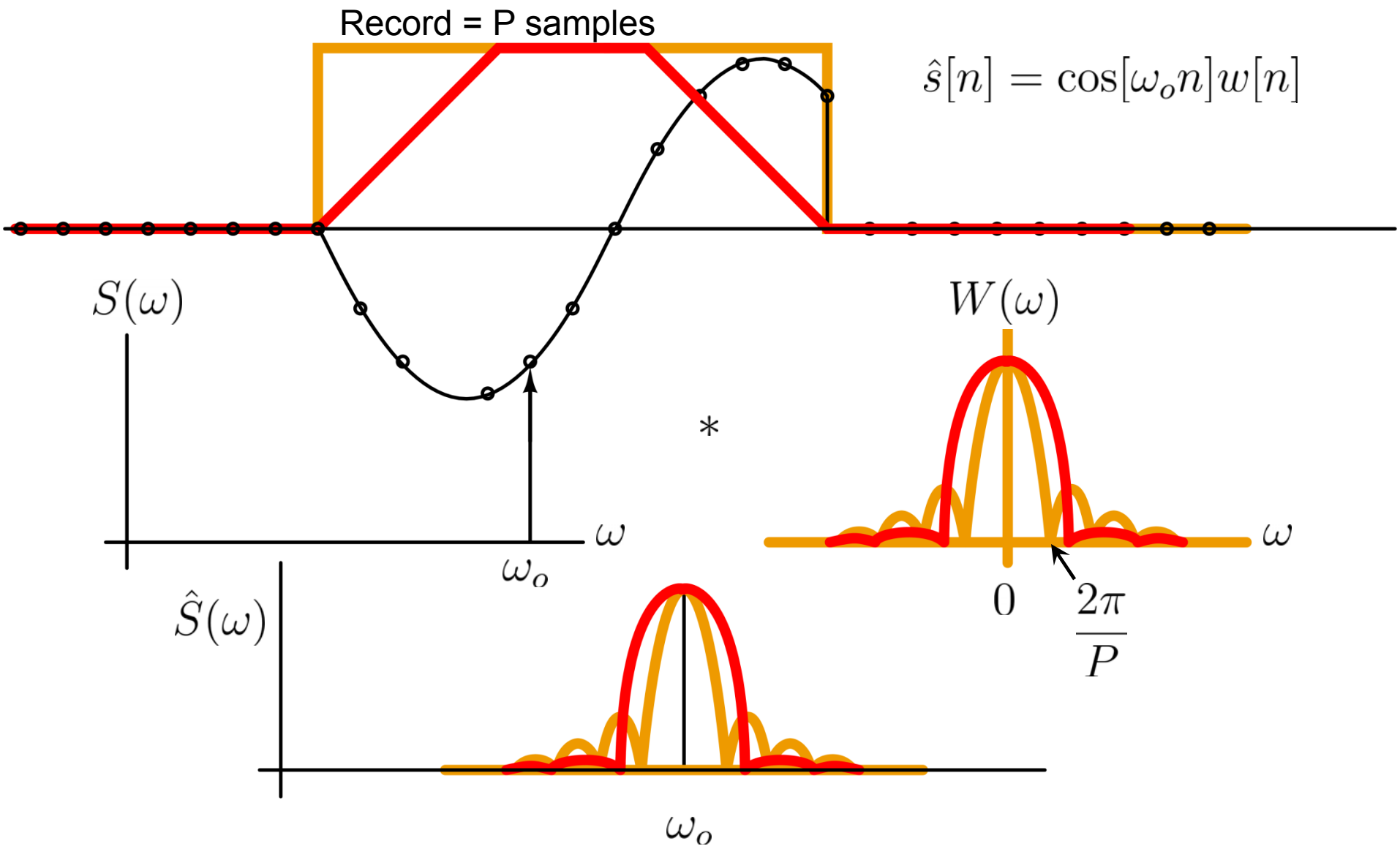


Leakage : A spectral view

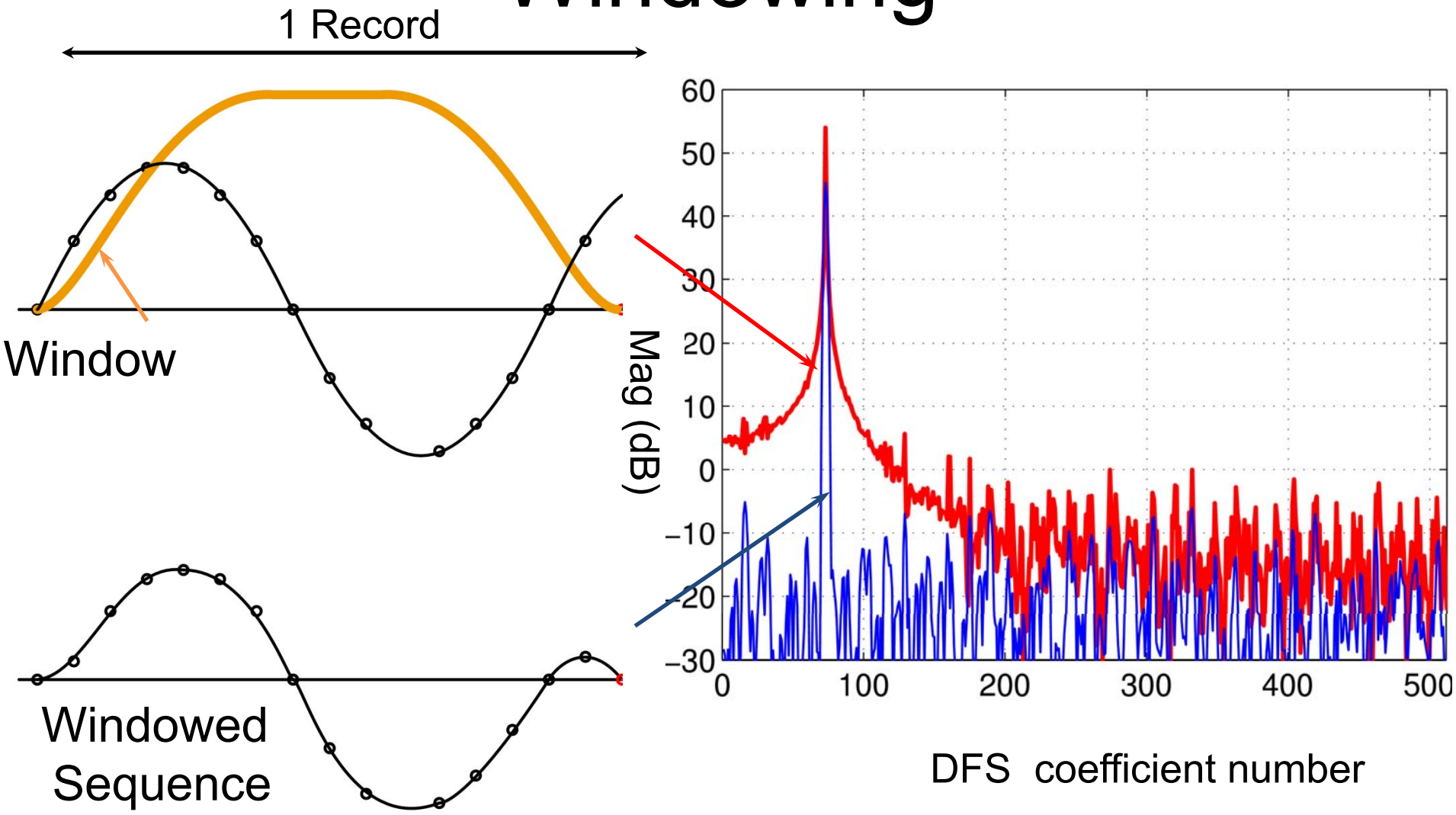


Discrete Fourier Series is obtained by sampling the Discrete Fourier Transform

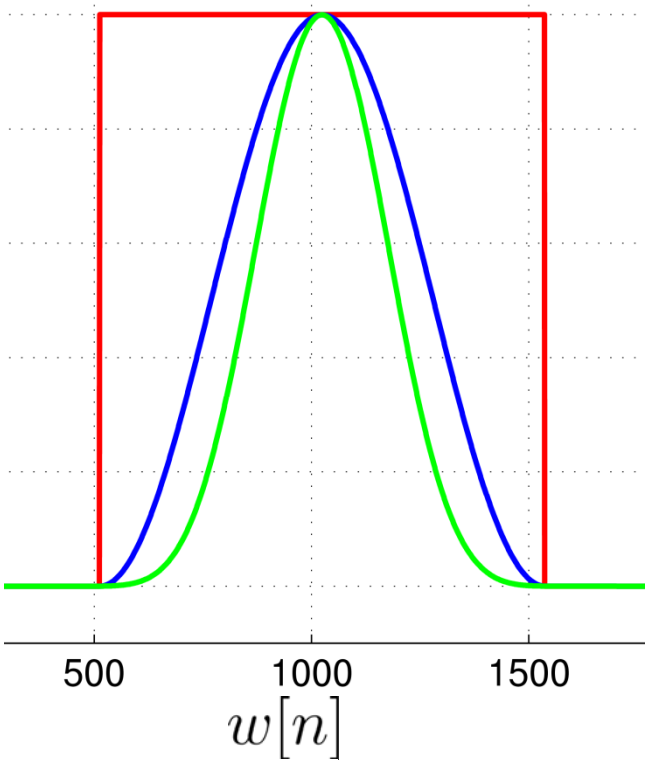
Leakage : with windowing

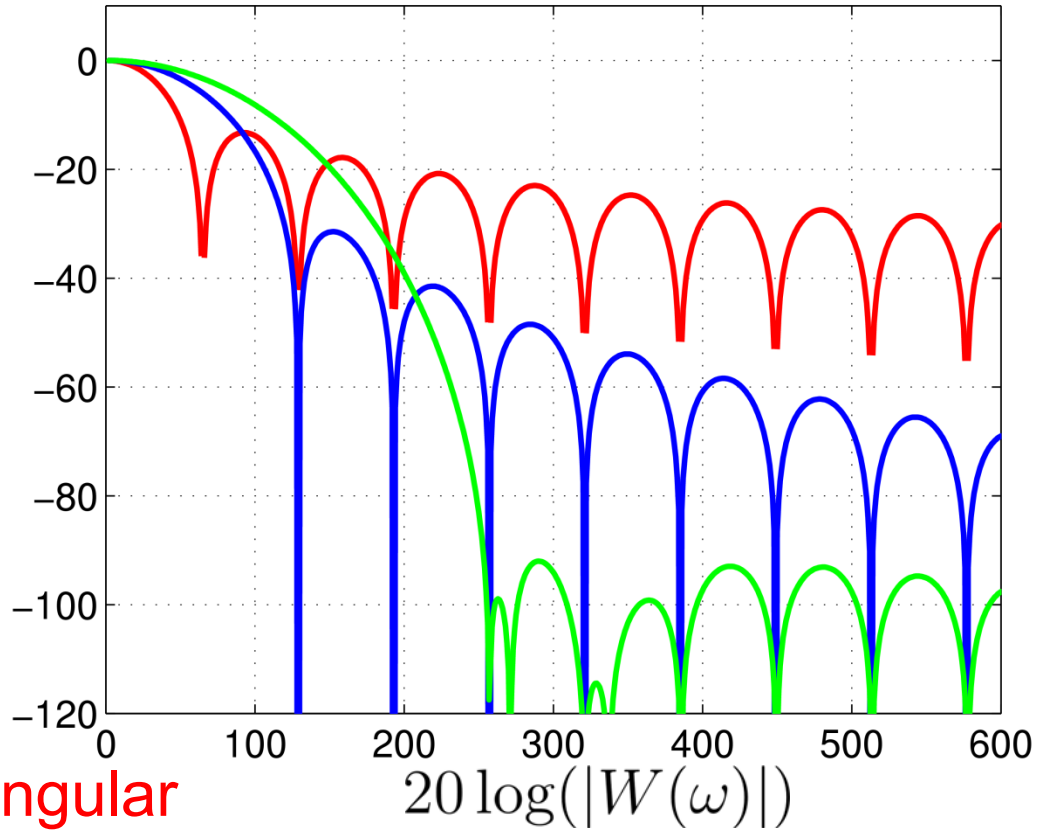


Windowing


1 Record
Window
Windowed
Sequence
Mag (dB)
60
50
40
30
20
10
0
−10
−20
−30
0
100
200
300
400
500
DFS coefficient number

Commonly used windows





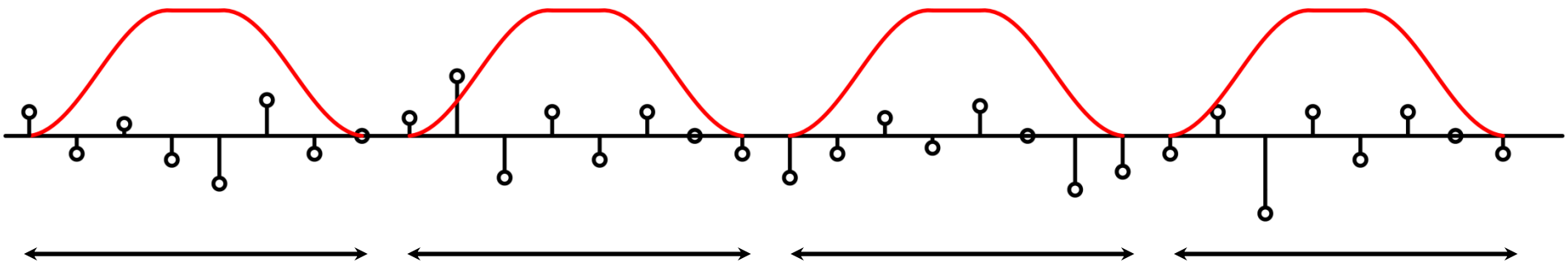
Rectangular

Hann

Blackman-Harris

Trade off between main lobe width and side lobe suppression

PSD & FFT



N_{fft} N_{fft} N_{fft} N_{fft}

$|FFT_1|^2$ $|FFT_2|^2$ $|FFT_3|^2$ $|FFT_4|^2$

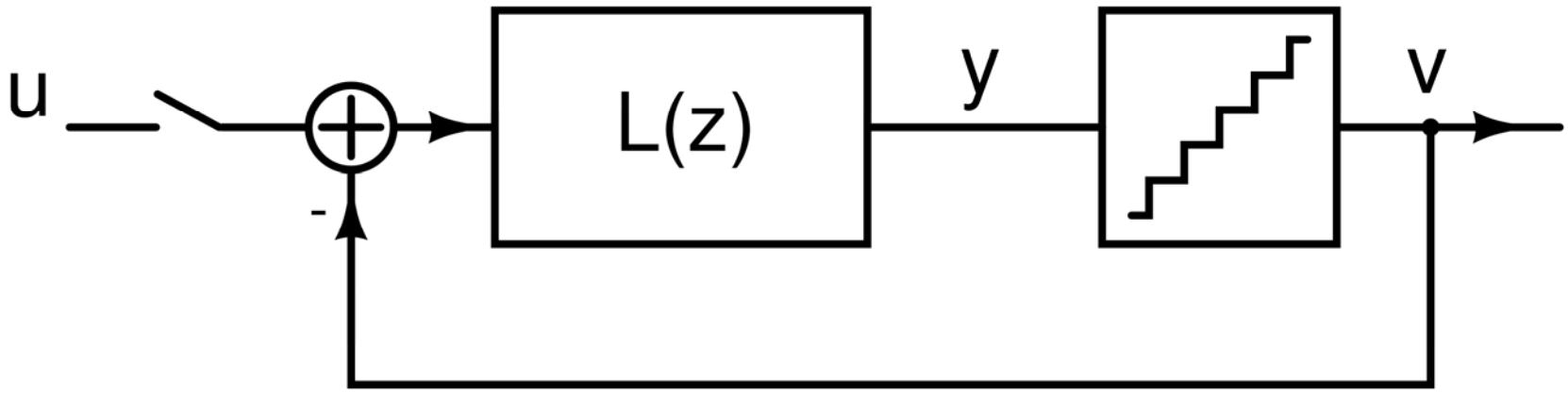
$$PSD = \frac{1}{N}\sum_{k=1}^{N}|FFT_k|^2$$

© 2013 IEEE

Simulation of $\Delta\Sigma$ ADCs

- Stiff
  - Low frequency input, high frequency clock
- No “one-to-one” correspondence between input and output
  - Debug is not trivial
- Computing SNDR
  - More involved than the Nyquist case
- Simulating device noise
  - Computationally expensive

Delta Sigma ADCs



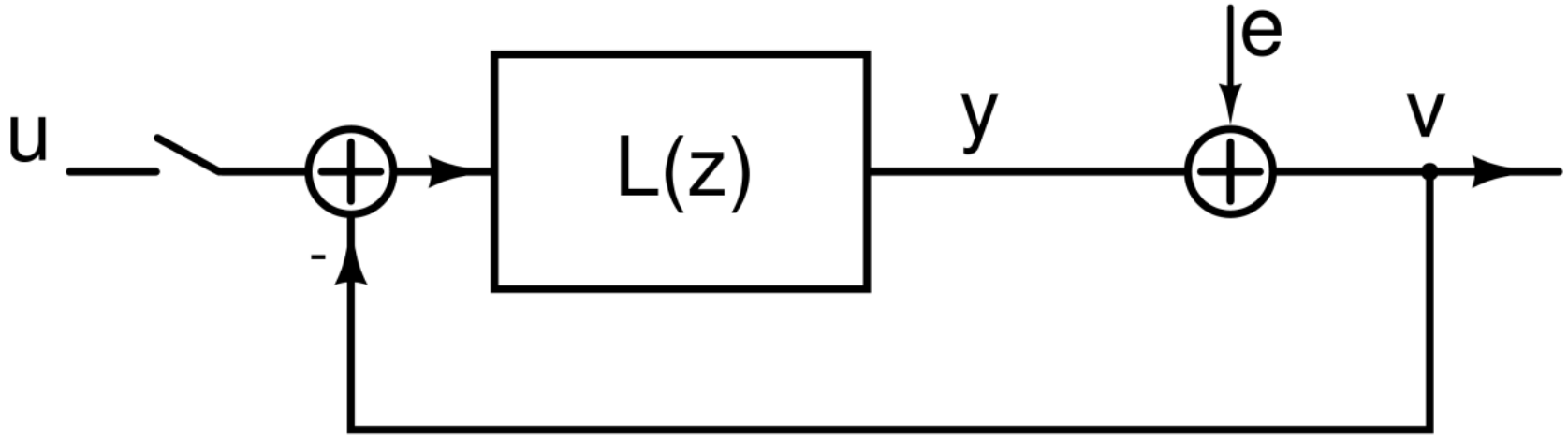
Basic Idea

Embed a coarse quantizer in a negative feedback loop

$$L(z) \text{ has high gain @ low frequency}$$

$$\Rightarrow v \approx u \text{ @ low frequency}$$

Oversampling Converters



$$V(z) = \frac{L(z)}{1+L(z)}U(z) + \frac{1}{1+L(z)}E(z)$$

Signal Transfer Function STF

Noise Transfer Function NTF

Oversampling Converters

$$V(z) = \frac{L(z)}{1+L(z)}U(z) + \frac{1}{1+L(z)}E(z)$$

Signal Transfer Function STF

Noise Transfer Function NTF

$$L(z) \to \infty \text{ as } z = e^{j\omega} \to 1$$

$$STF(z) \to 1 \text{ and } NTF(z) \to 0$$

Quantization Noise is high pass filtered

$\Rightarrow$ High inband SNR

© 2013 IEEE

Oversampling Converters

$$V(z) = \frac{L(z)}{1+L(z)}U(z) + \frac{1}{1+L(z)}E(z)$$

Noise Transfer Function
NTF

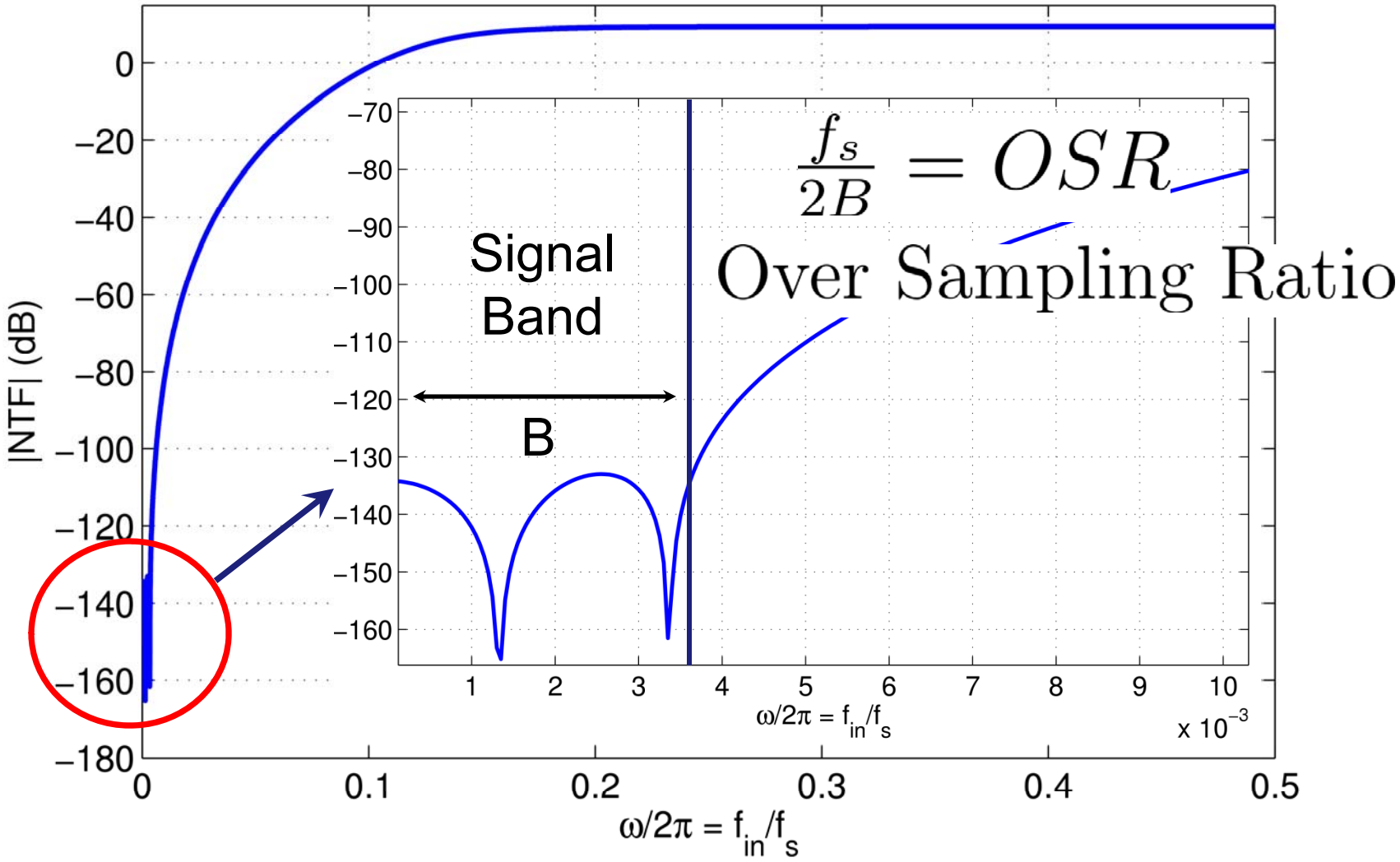
Choosing an appropriate L(z) is the first step in the design

Too little gain : Not enough noise suppression

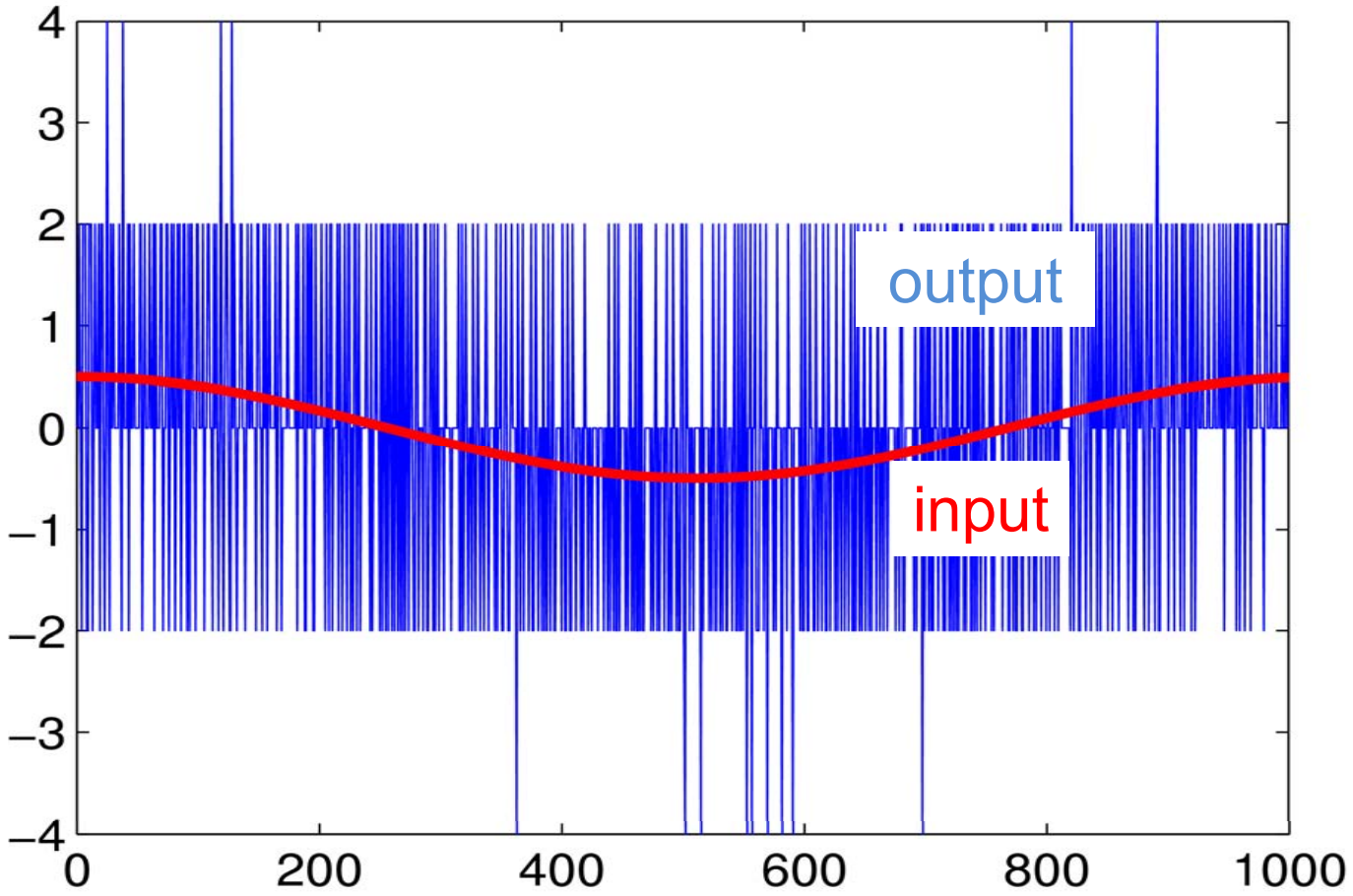
Gain is too high : Loop becomes unstable

© 2013 IEEE

NTF = High Pass Filter Design

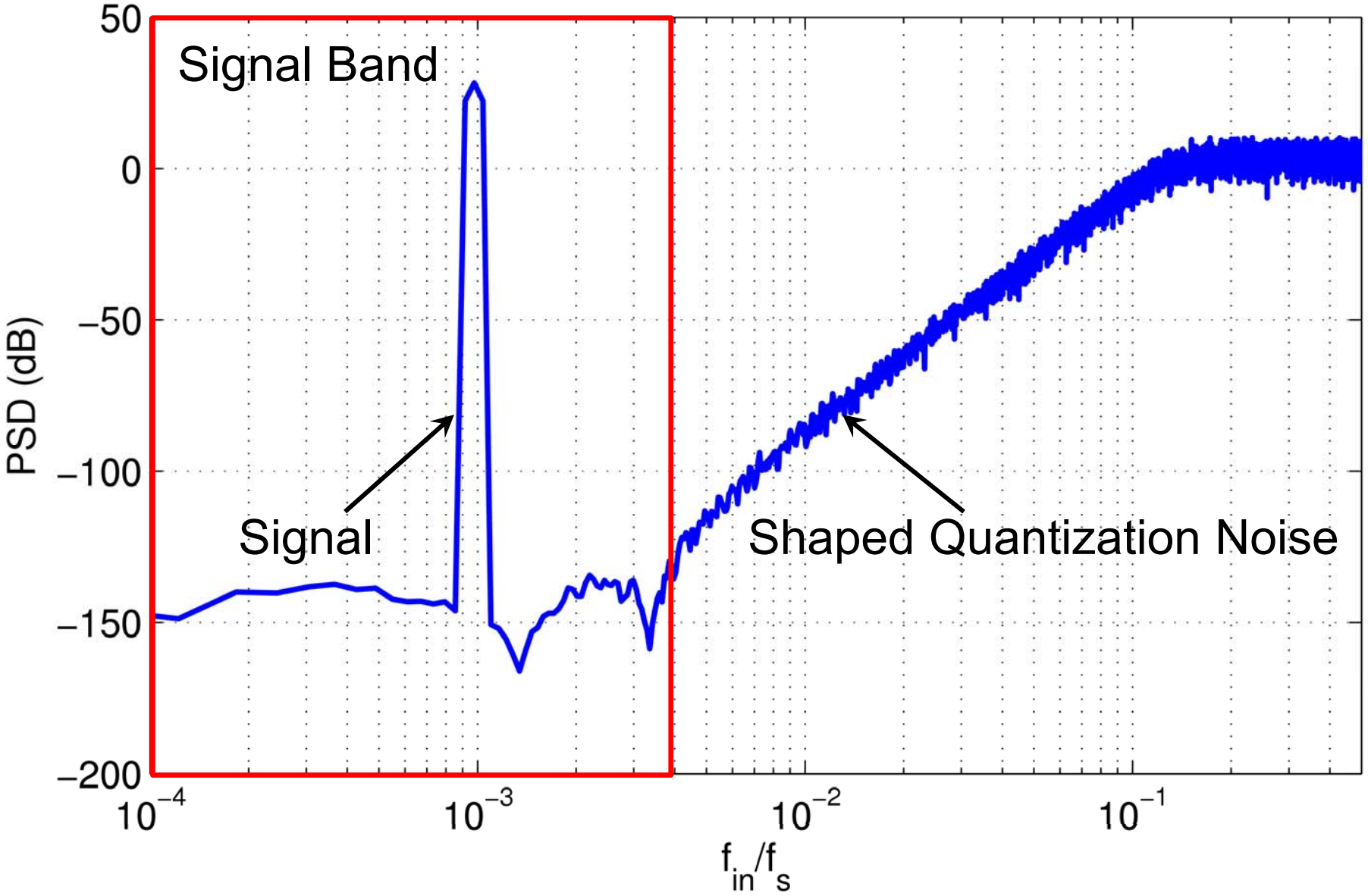


Example ΔΣ Converter Waveforms



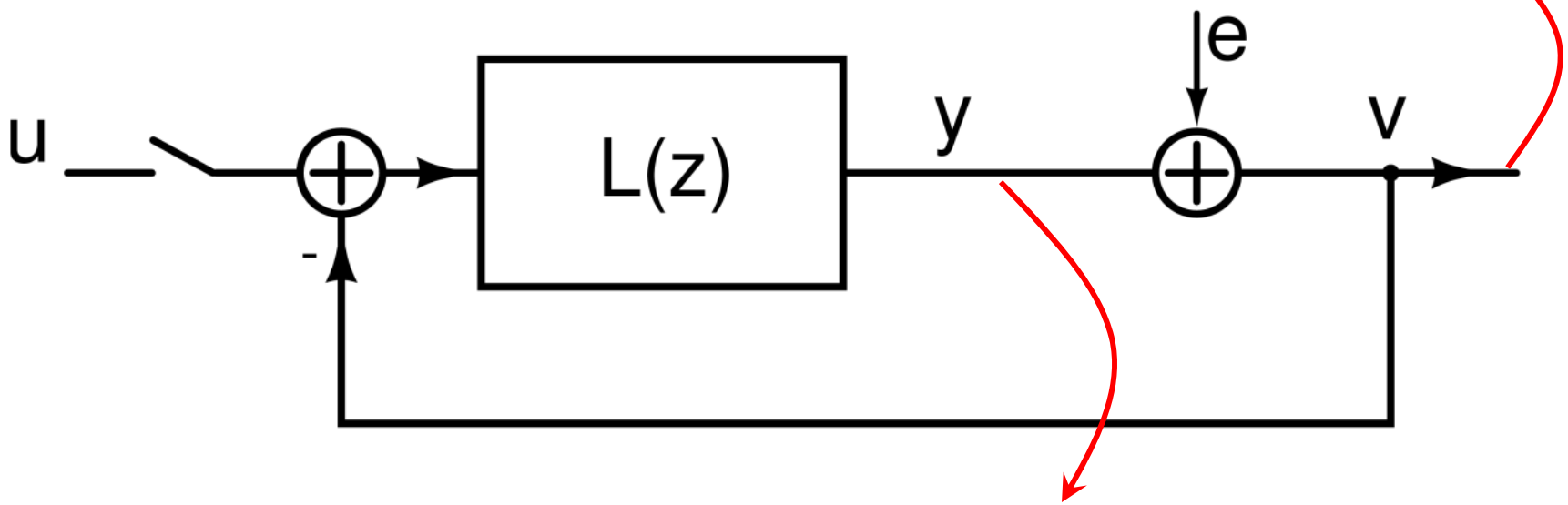
© 2013 IEEE

Example ΔΣ Converter Spectrum



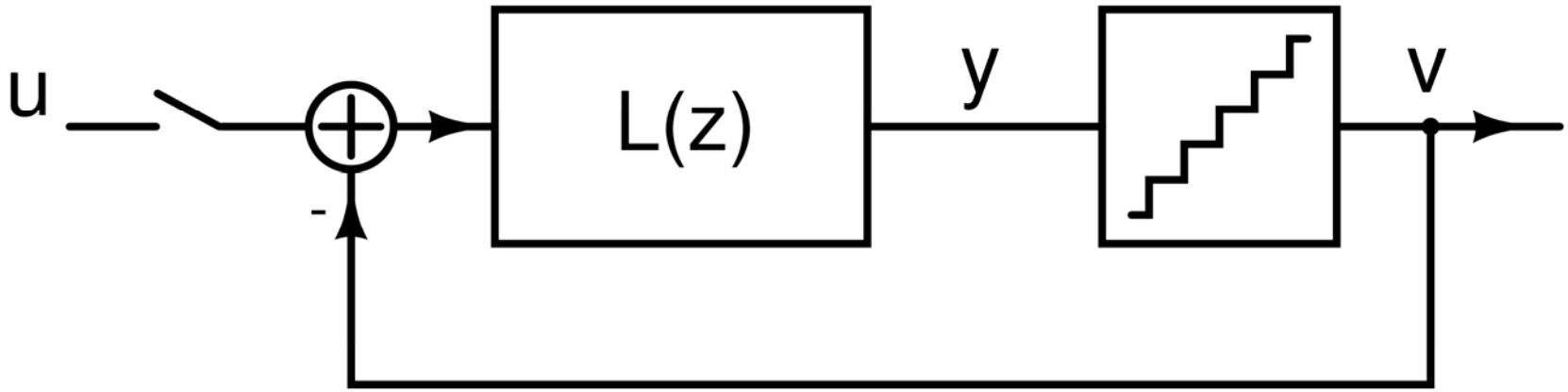
Signal Dependent Stability

$$V(z) = STF(z)\,U + NTF(z)\,E$$



$$\begin{aligned} Y(z) &= STF(z)\,U + (NTF(z) - 1)\,E \\ &\approx U + (NTF(z) - 1)\,E \end{aligned}$$

Stability



$$Y(z) \approx U + (NTF(z) - 1)\,E$$

Shaped Quantization Noise

u → close to quantizer saturation limits

- Shaped noise saturates the quantizer → e increases
- This saturates the quantizer more
  → e increases even more
- … until y becomes unbounded
  → modulator becomes unstable

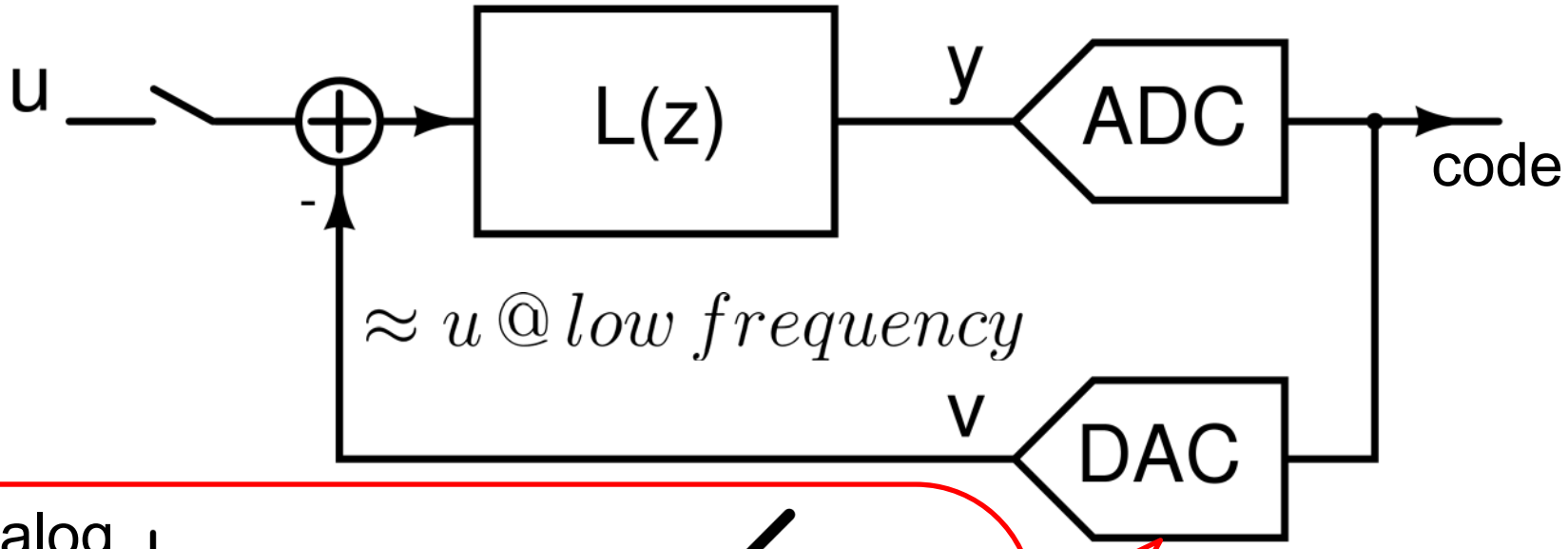
Stability

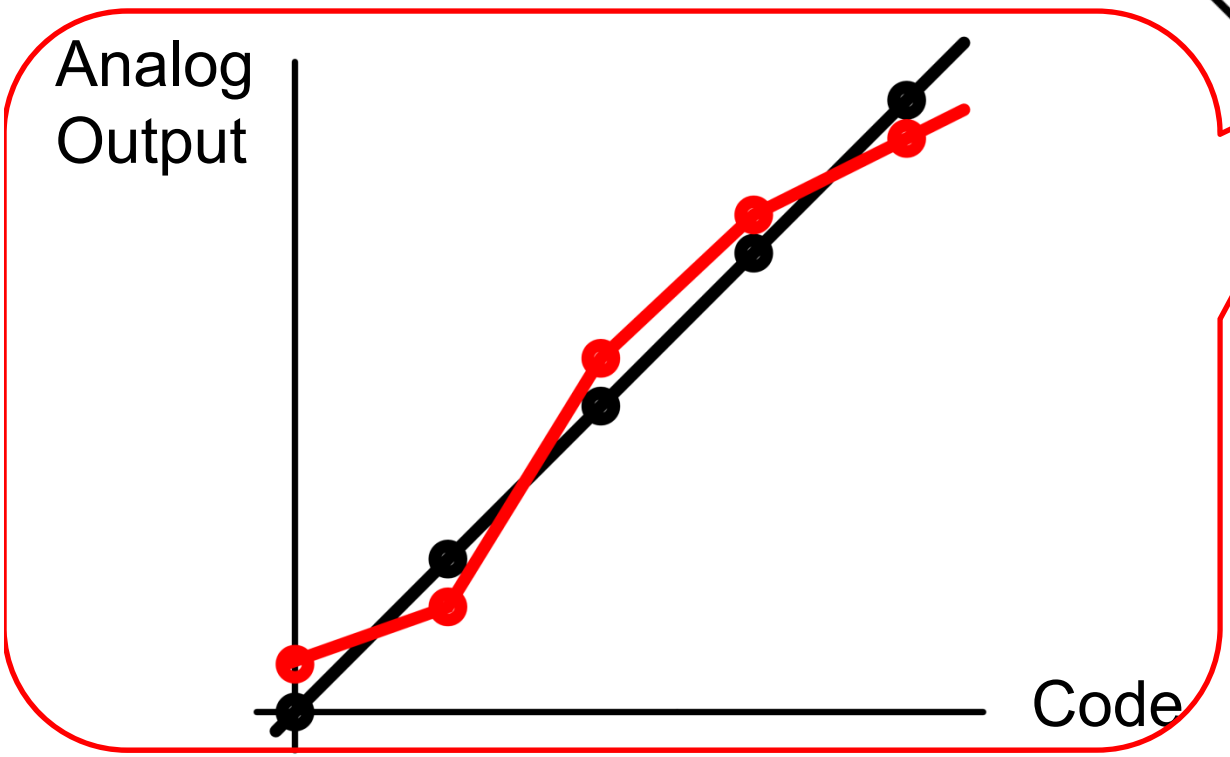
- Signal dependent stability to be **expected**

$$\frac{\text{Stable Input Range}}{\text{Quantizer Range}} = \textbf{M}\text{aximum } \textbf{S}\text{table } \textbf{A}\text{mplitude (MSA)}$$

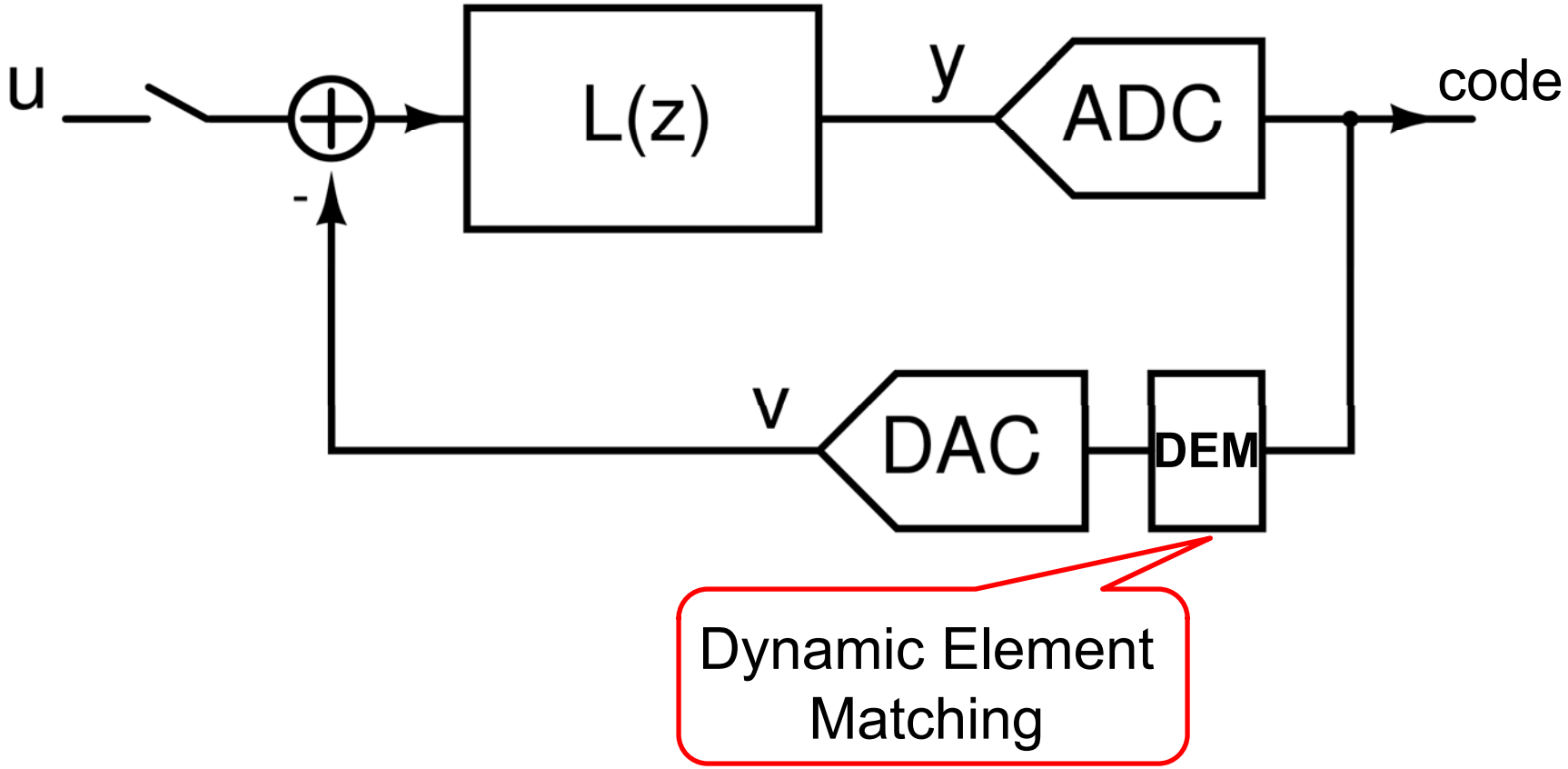
- MSA depends on
  - NTF
  - Number of quantizer levels
  - Input frequency

DAC Nonidealities



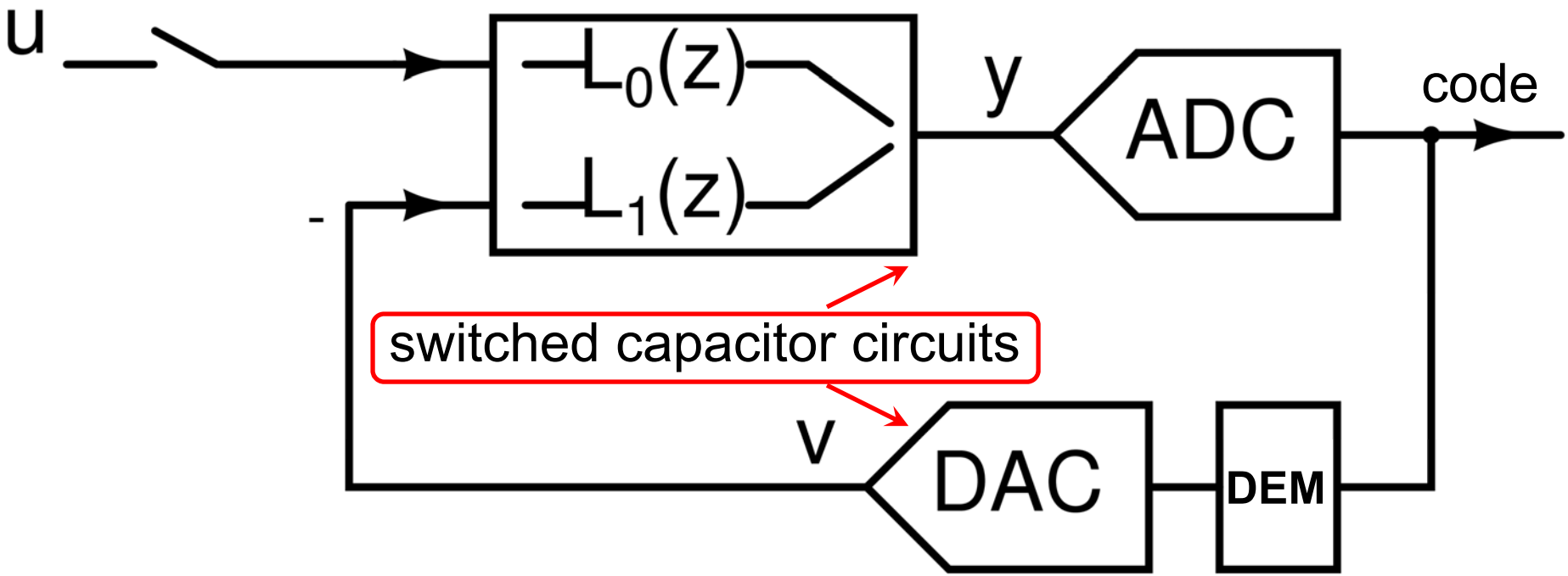


DAC Nonidealities



DEM : Magically shapes mismatch noise out of the signal band

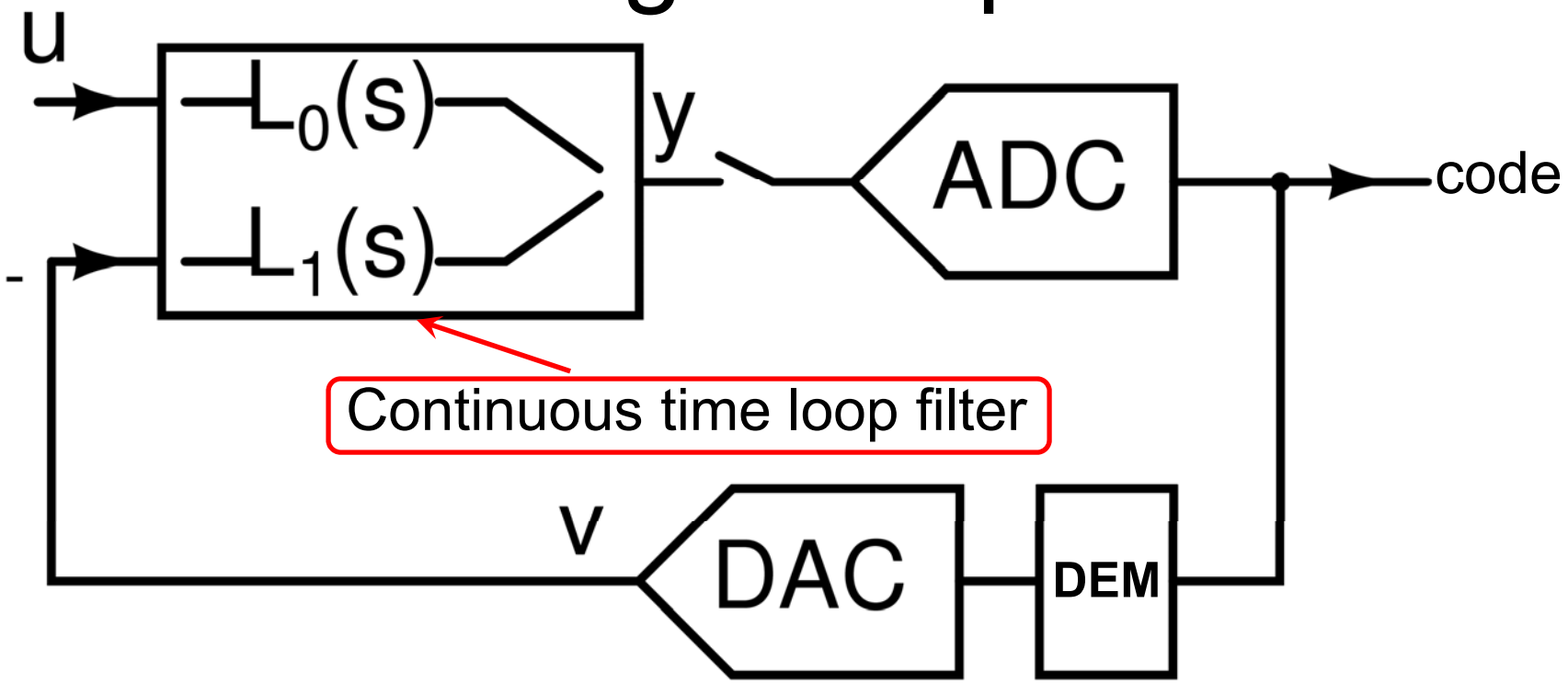
Generic Single Loop DT-DSM



$$NTF(z) = \frac{1}{1 + L_1(z)}$$

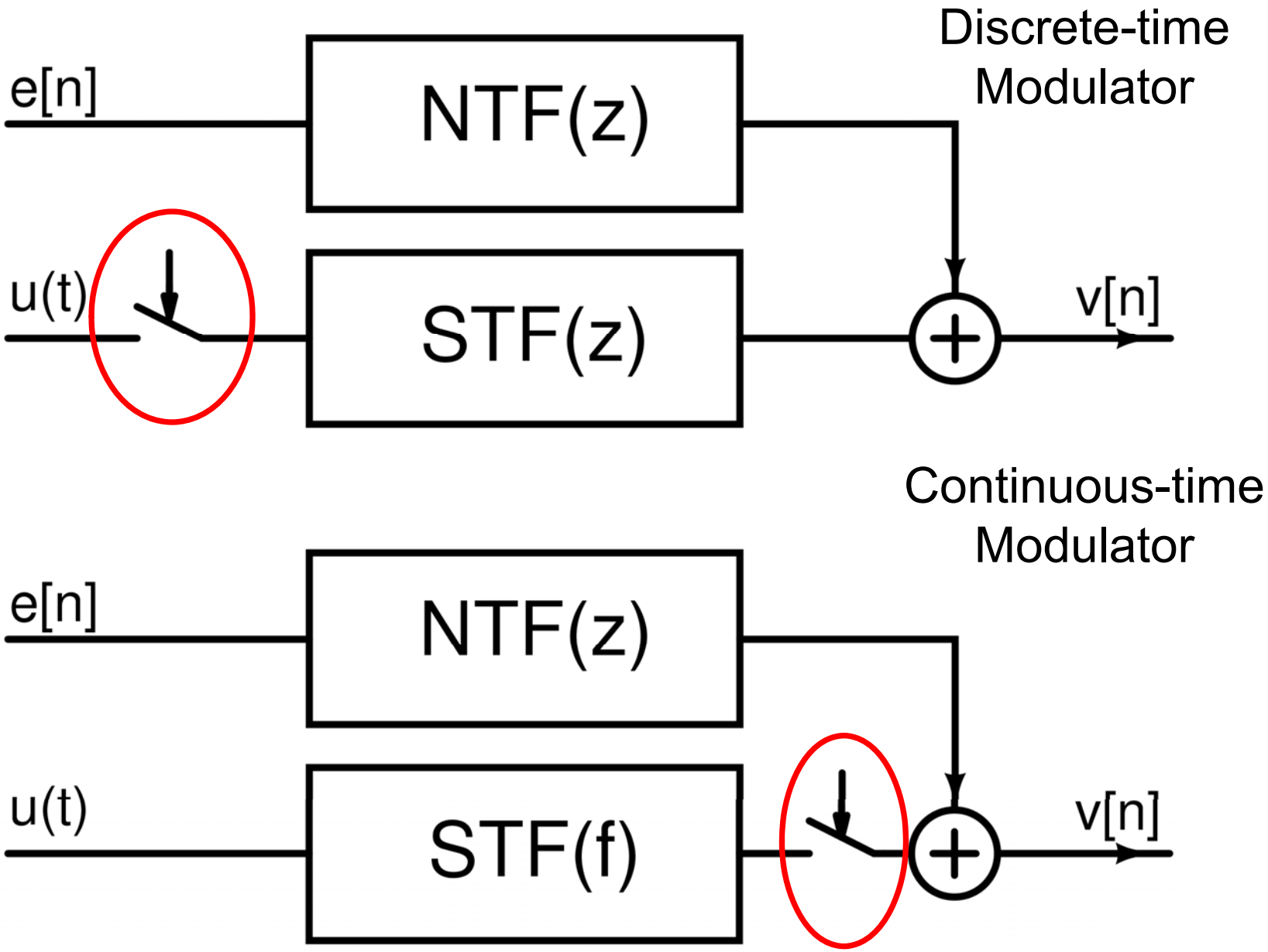
$$STF(z) = \frac{L_0(z)}{1 + L_1(z)}$$

Generic Single Loop CT-DSM



- Input sampled at the loop filter output
  - Implicit anti-aliasing if the loop filter is time invariant
- Resistive input impedance → easy to drive
- Reduced slew requirements of opamps

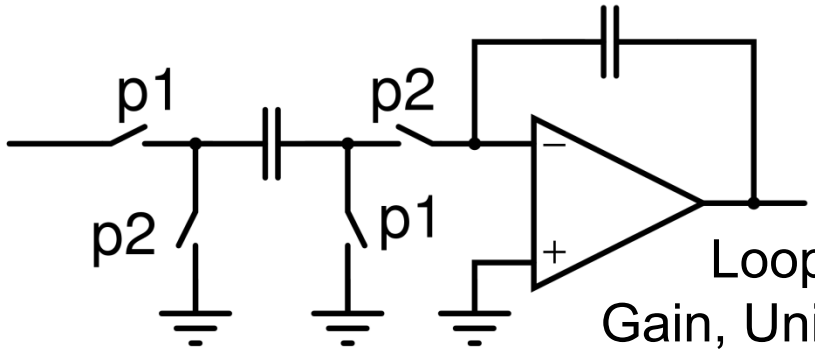
Linear Models : DT & CT DSMs



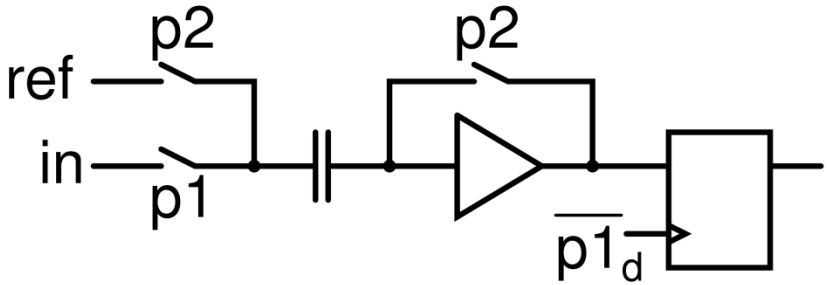
ΔΣ ADC : Design Flow

- Choose OSR, quantizer levels, NTF and architecture
  - Determine MSA (use simulation)
- Determine component values
- Design building blocks
  - Loop filter, ADC, DAC, DEM logic

Building Block Design - DT

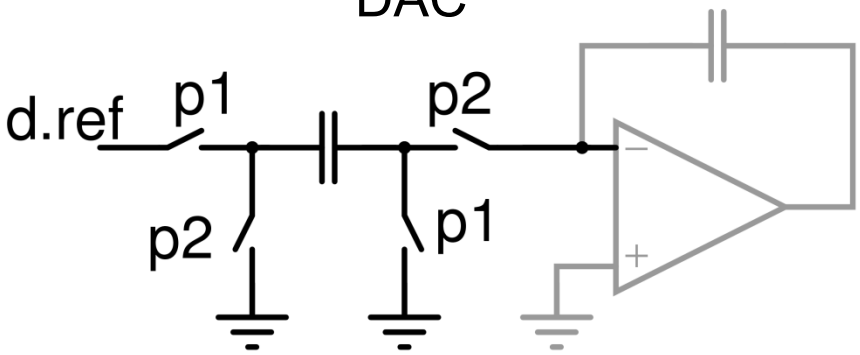


Loop Filter :
Gain, Unity gain freq.
Parasitic poles

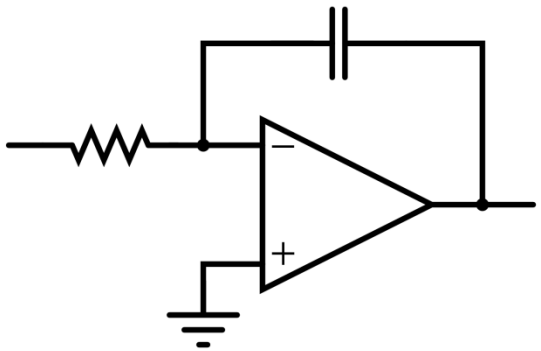


Comparator :
Offset, Delay

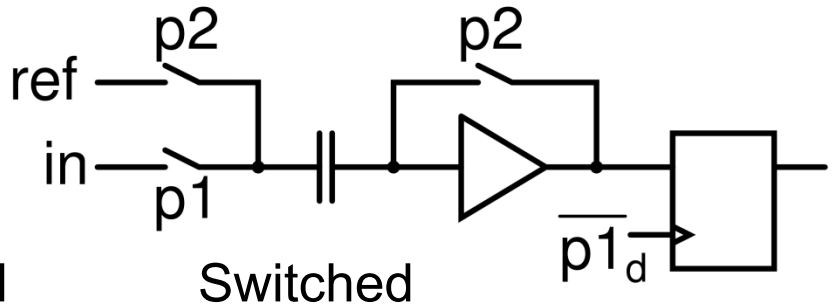
DAC



Building Block Design - CT

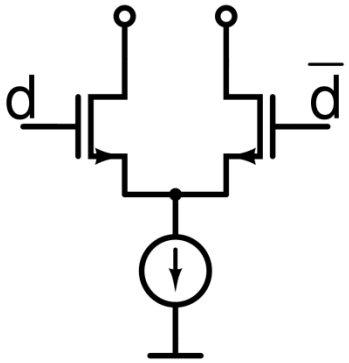


Loop Filter :
Gain, Unity gain freq.
Parasitic poles



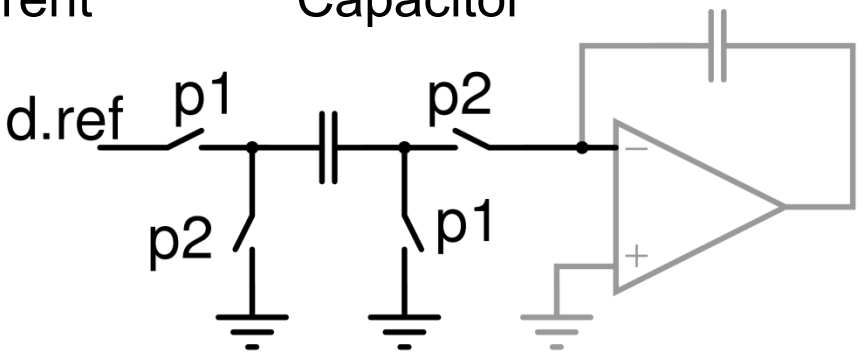
Comparator :
Offset, Delay

DACs

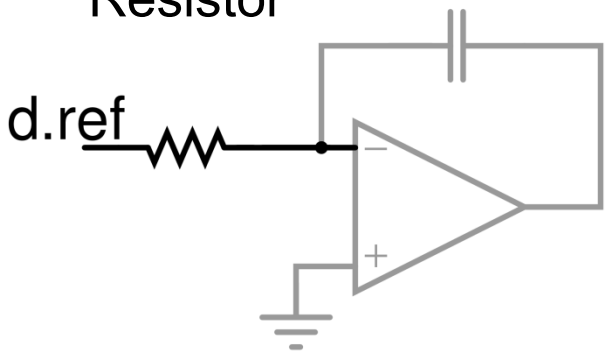


Switched
Current

Switched
Capacitor



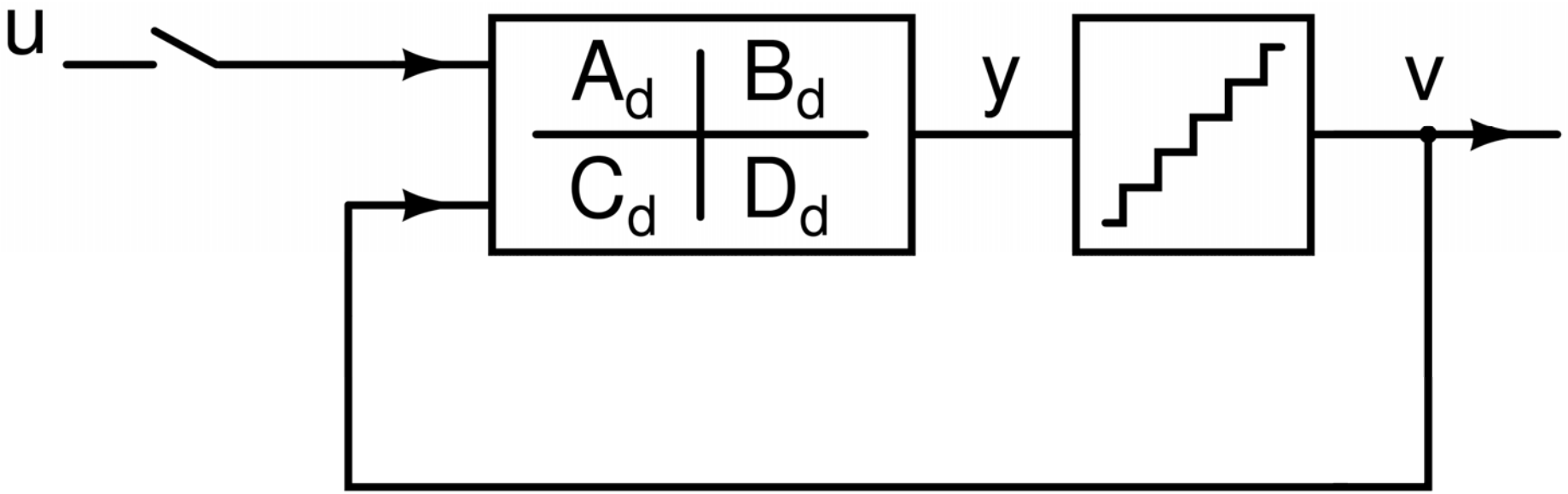
Switched
Resistor



High Level Design Aids

- $\Delta\Sigma$ Toolbox for MATLAB
  - Richard Schreier of Analog Devices
- Neat set of routines that enable high level design and rapid simulation
  - Find SN(D)R, MSA for a given NTF
  - Simulate a DT modulator given loop filter description in state space form

ΔΣ Toolbox for MATLAB



- Support for multiple inputs/quantizers
- Can be combined with the power of MATLAB to design/simulate CTDSMs as well

ΔΣ Toolbox for MATLAB

- All linear effects can be modeled with the tool box
  - Finite gain and parasitic poles in the loop filter
  - Excess loop delay
- With minor modifications
  - Flash ADC offsets
- Great design and educational value
- Highly recommended !

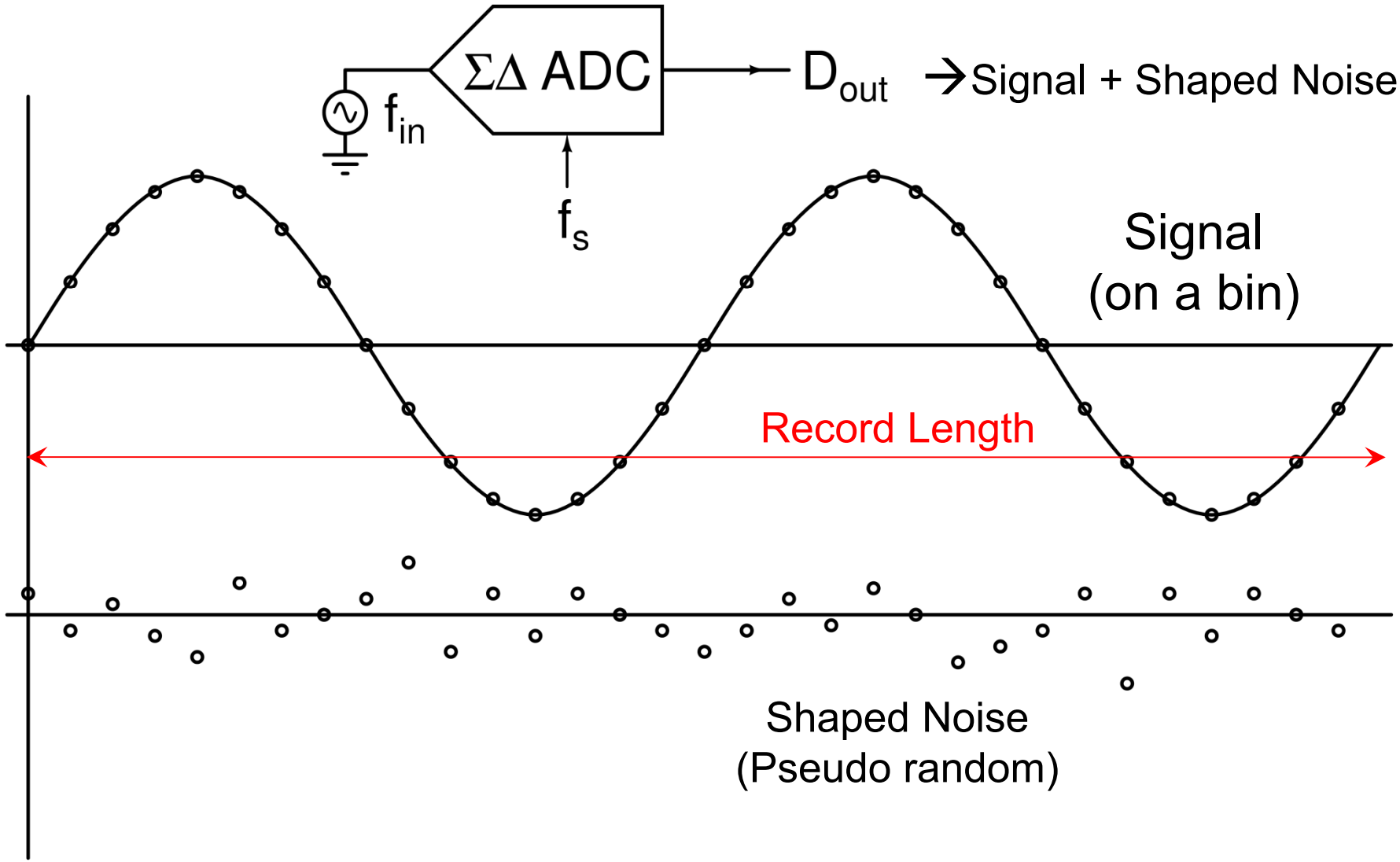
Circuit Simulation Strategies

- Keep one block real – all others behavioral
  - e.g loop filter real but Verilog-A ADC, DAC,DEM
  - Isolates ``gross" problems
- Two blocks real – others Verilog-A
  - Enables debug of ``interface" issues
- MUST run the complete design at the transistor level
  - Many problems are often subtle and only show up at this level

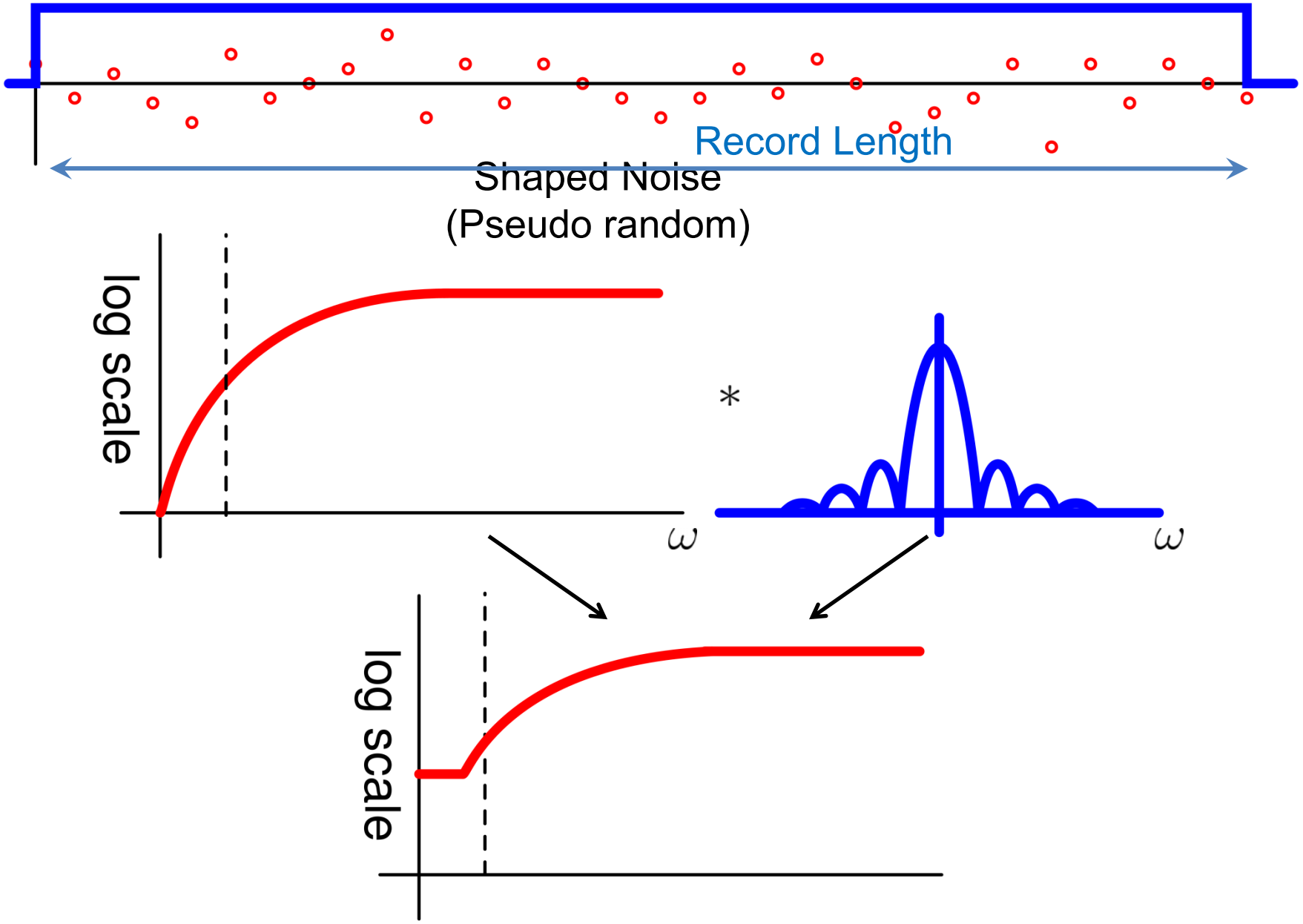
ΔΣ ADC : Simulations

- Inband SNDR
  - How to calculate this ?
  - How long should I simulate ?
- Realized NTF
  - As opposed to intended NTF
- Realized STF
  - Especially in CT modulators
- Device Noise

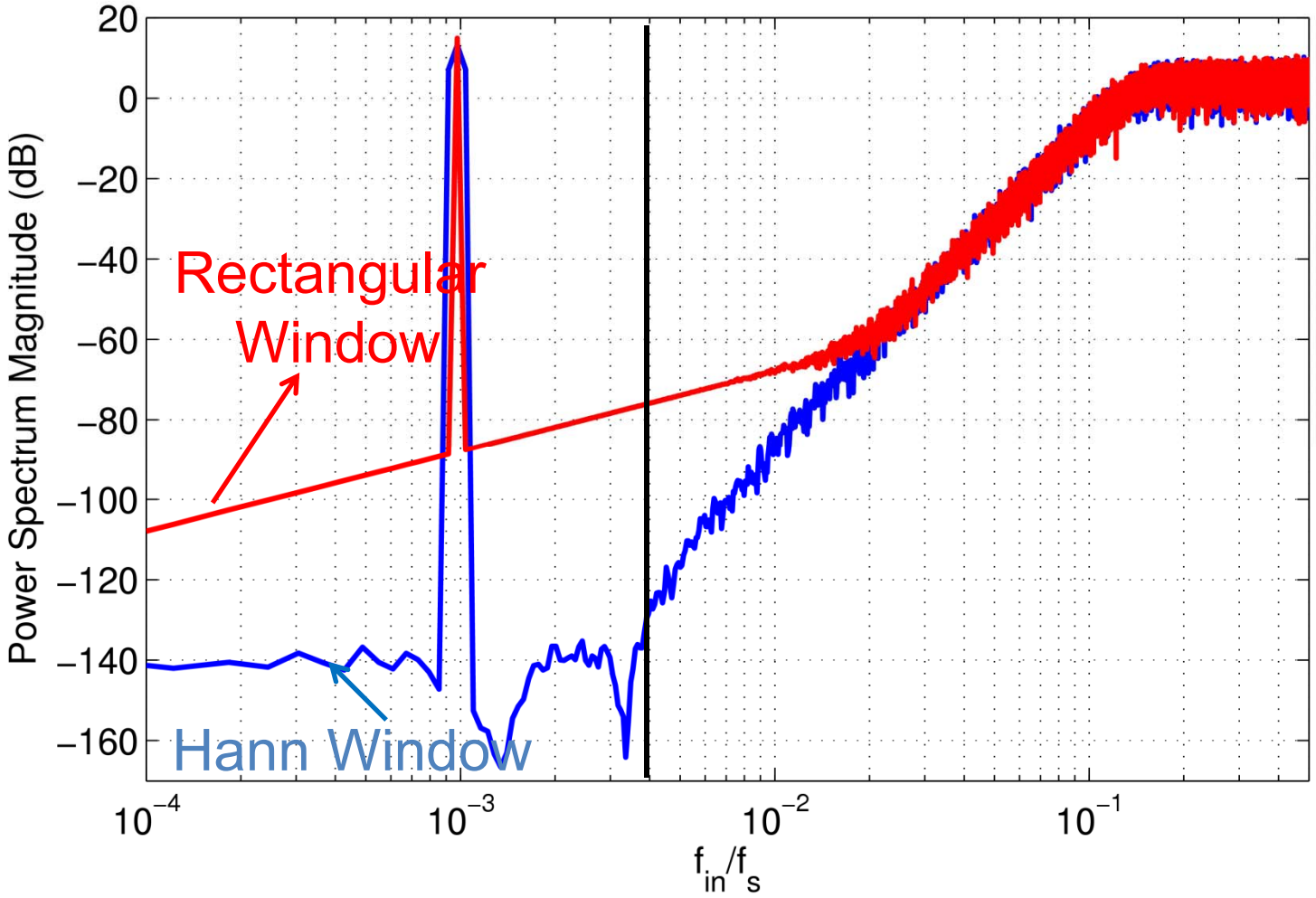
Simulating Inband SNDR in ΣΔ ADCs



Simulating SNDR



ΣΔ Example Spectra



Moral : Must window data even if input tone lies on a bin

How many samples do I need ?

- Record Length = Number of FFT bins = N
- Number of bins in the signal band : $\frac{N}{2\,OSR}$

- Input signal + 2 harmonics (Hann): 9 bins
- DC Offset : 2 bins

$$\text{Estimating noise from } (\frac{N}{2\,OSR} - 11) \approx 15 \text{ bins}$$

$$\Rightarrow N \approx 52\,\text{OSR samples}$$

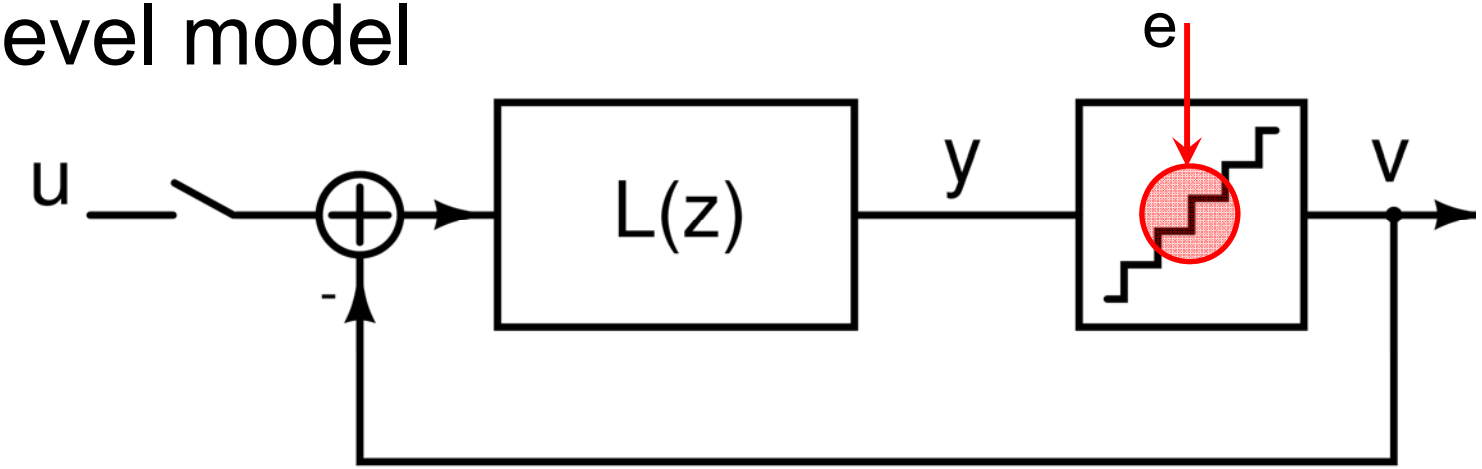
$$\text{Thumbrule : Choose N to be } \mathbf{atleast}\ 64\,\text{OSR}$$

ΔΣ ADC : Simulations

- Inband SNDR
  - How to calculate this ?
  - How long should I simulate ?
- Realized NTF
  - As opposed to intended NTF
- Realized STF
  - Especially in CT modulators
- Device Noise

What NTF do I have ?

- NTF is the all important loop parameter
  - Deviates too much from the designed → trouble
- NTF → easy to determine MSA using high level model



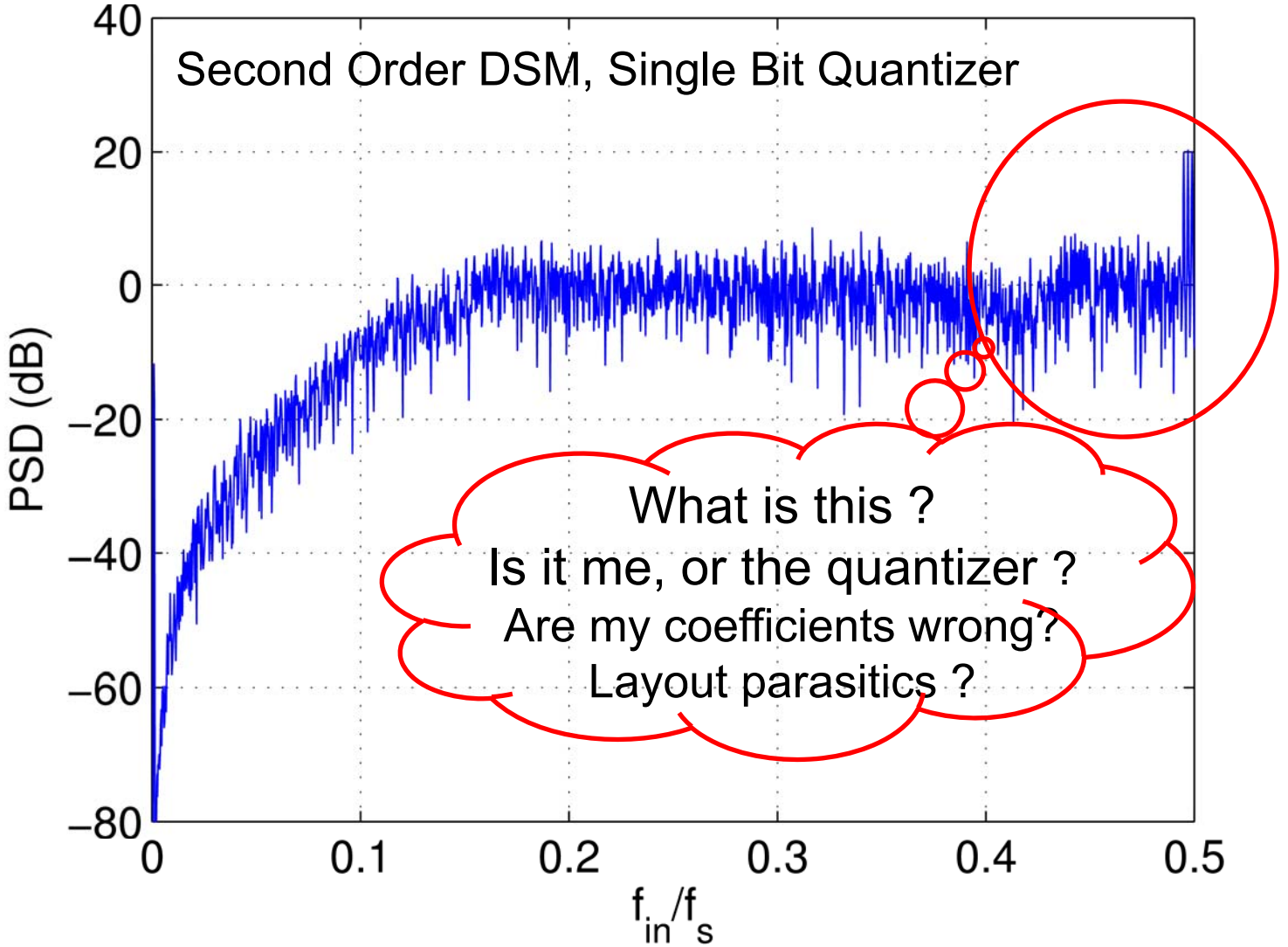
$$V(z) \approx U(z) + NTF(z)E(z)$$

$$PSD(v) = PSD(u) + |NTF|^2 PSD(e)$$

© 2013 IEEE

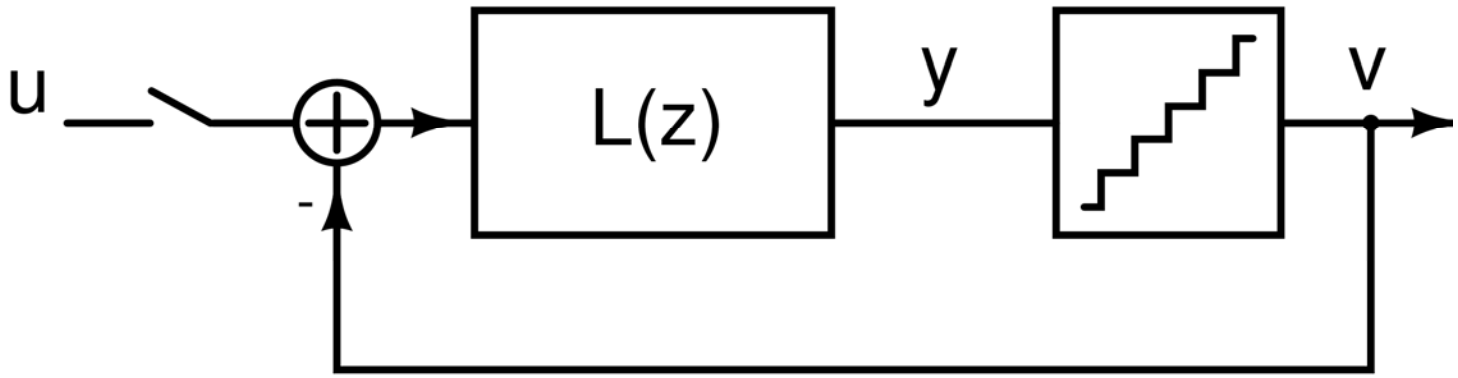
Output PSD

$$PSD(v) = PSD(u) + |NTF|^2 PSD(e)$$



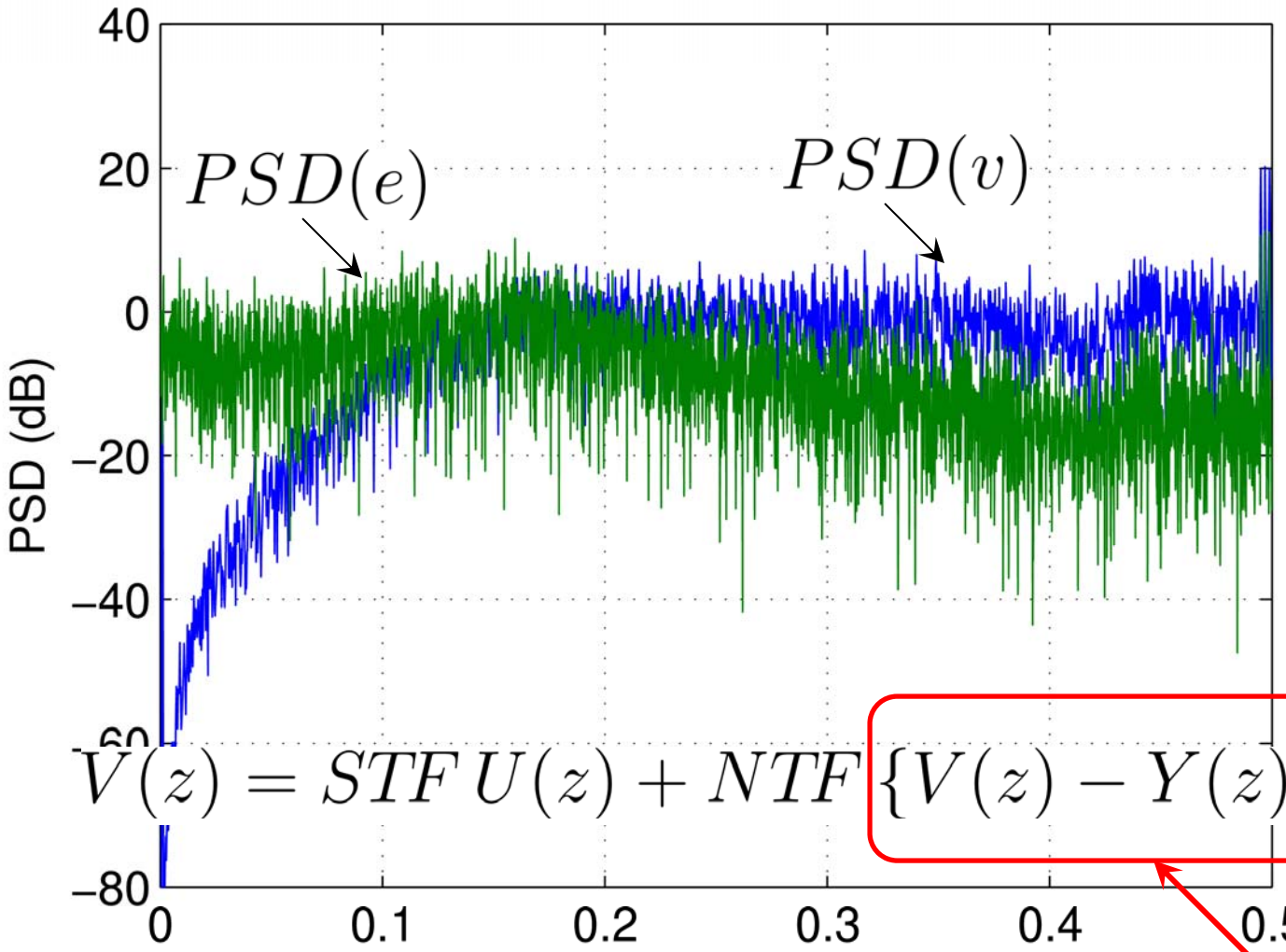
Estimating NTF

- Eyeballing the PSD → misleading



$$V(z) = STF\,U(z) + NTF\,\{V(z) - Y(z)\}\ \textbf{exactly}$$

"Error" is in thinking of this as white noise



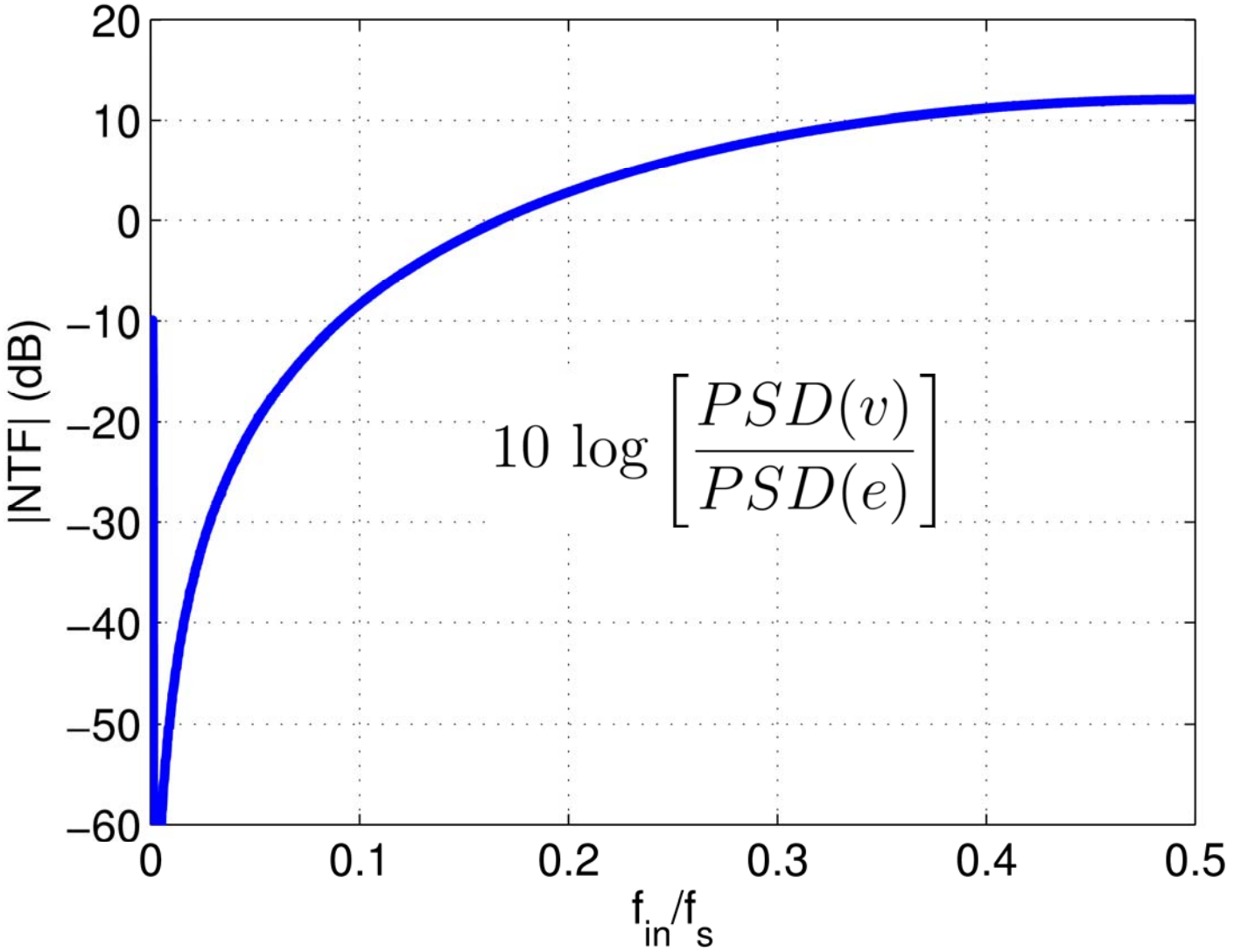
$$V(z) = STF\,U(z) + NTF\,\{V(z) - Y(z)\}$$

known in simulation

$$\frac{PSD(v)}{PSD(e)} = |NTF|^2$$

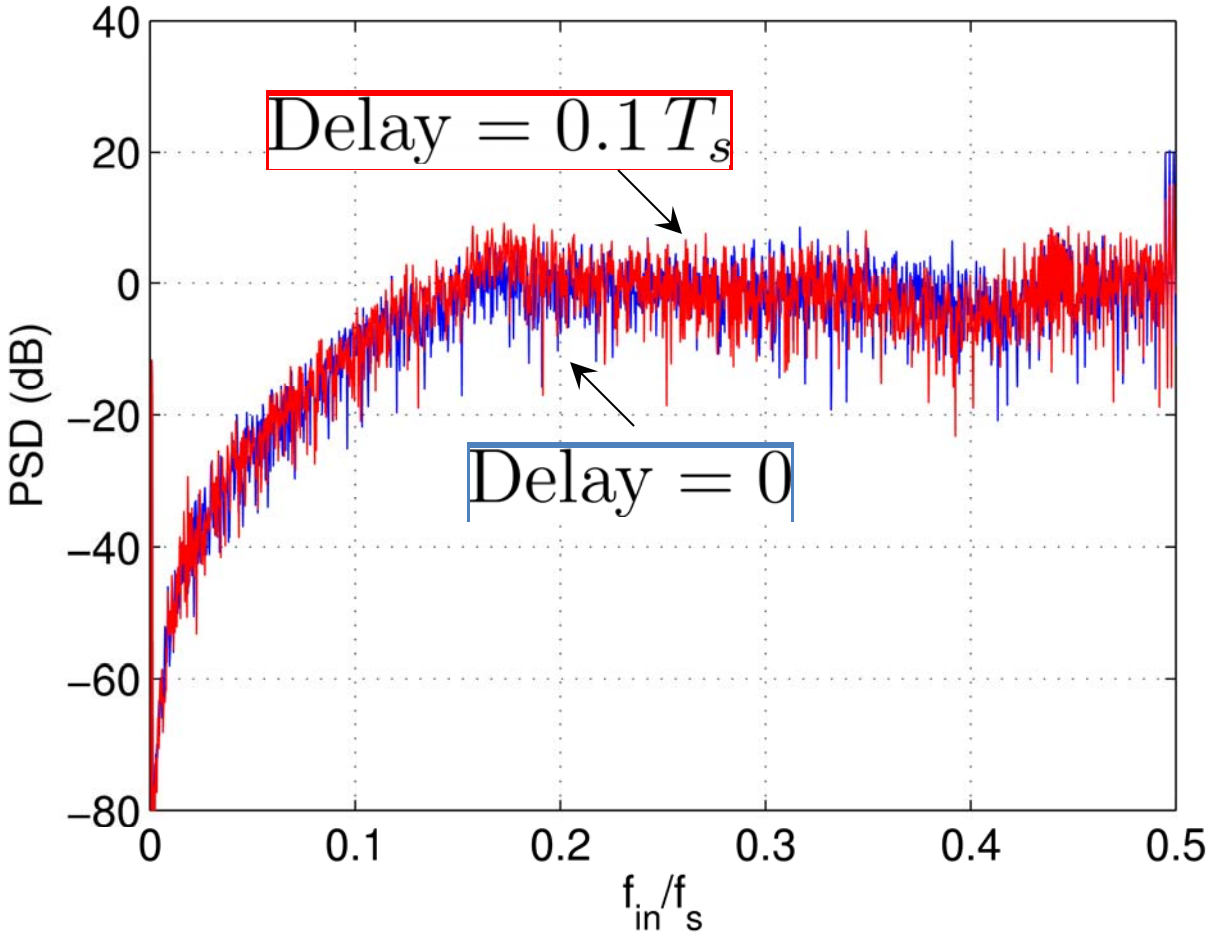
© 2013 IEEE

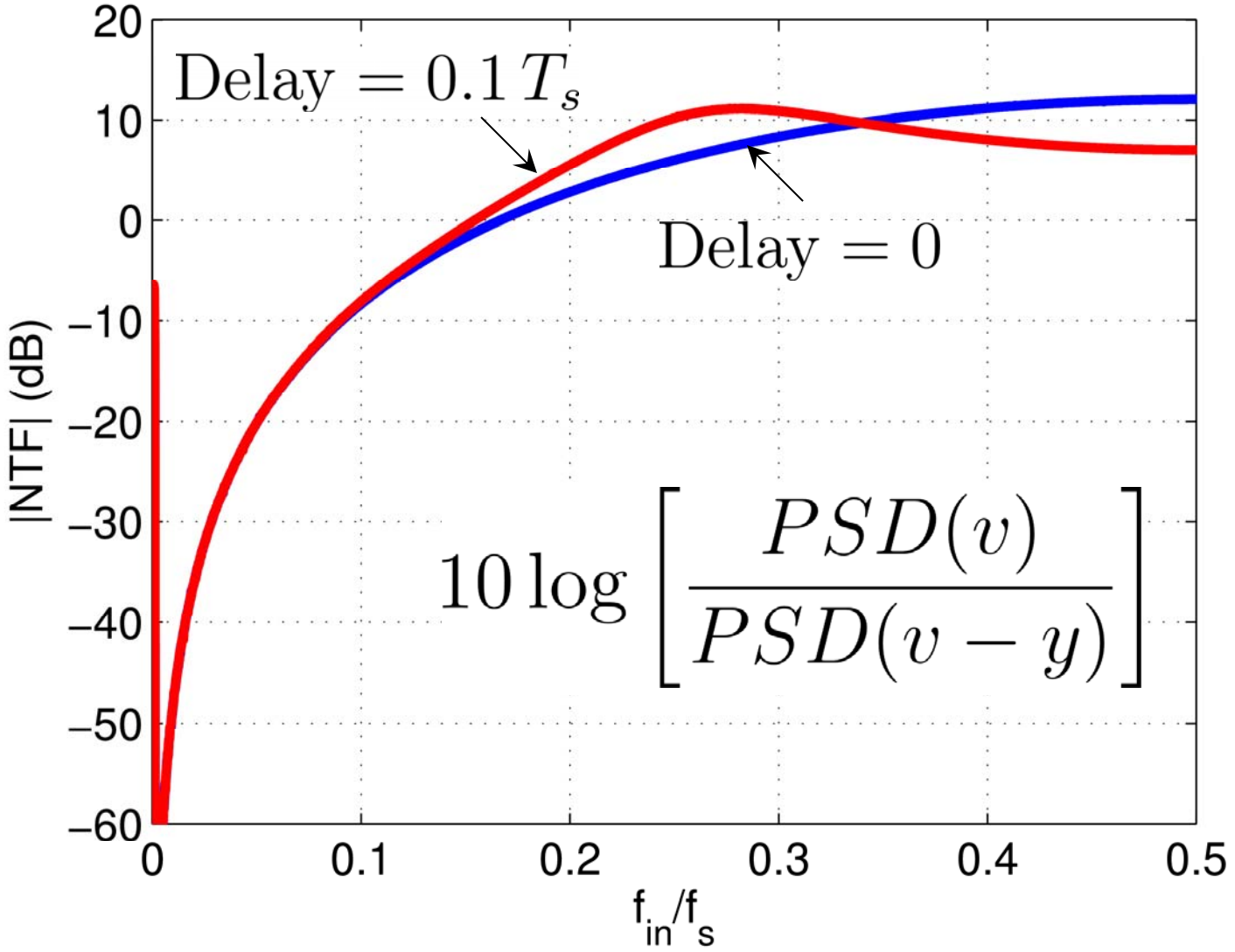
Estimated NTF



Another Example

- Two second order single bit CTDSMs
  - Nominal NTF = $(1 - z^{-1})^2$
  - One with unintended feedback delay of $0.1T_s$





ΔΣ ADC : Simulations

- Inband SNDR
  - How to calculate this ?
  - How long should I simulate ?
- Realized NTF
  - As opposed to intended NTF
- Realized STF
  - Especially in CT modulators
- Device Noise

RF Analyses in ADC Designs ?

- Typical RF designer's arsenal
  - Periodic Steady State (PSS)
  - Periodic AC (PAC)
  - Periodic Transfer Function (PXF)
  - Periodic Noise (Pnoise)
- Why/where are they relevant in ADC design ?

RF Analyses in ADC Designs ?

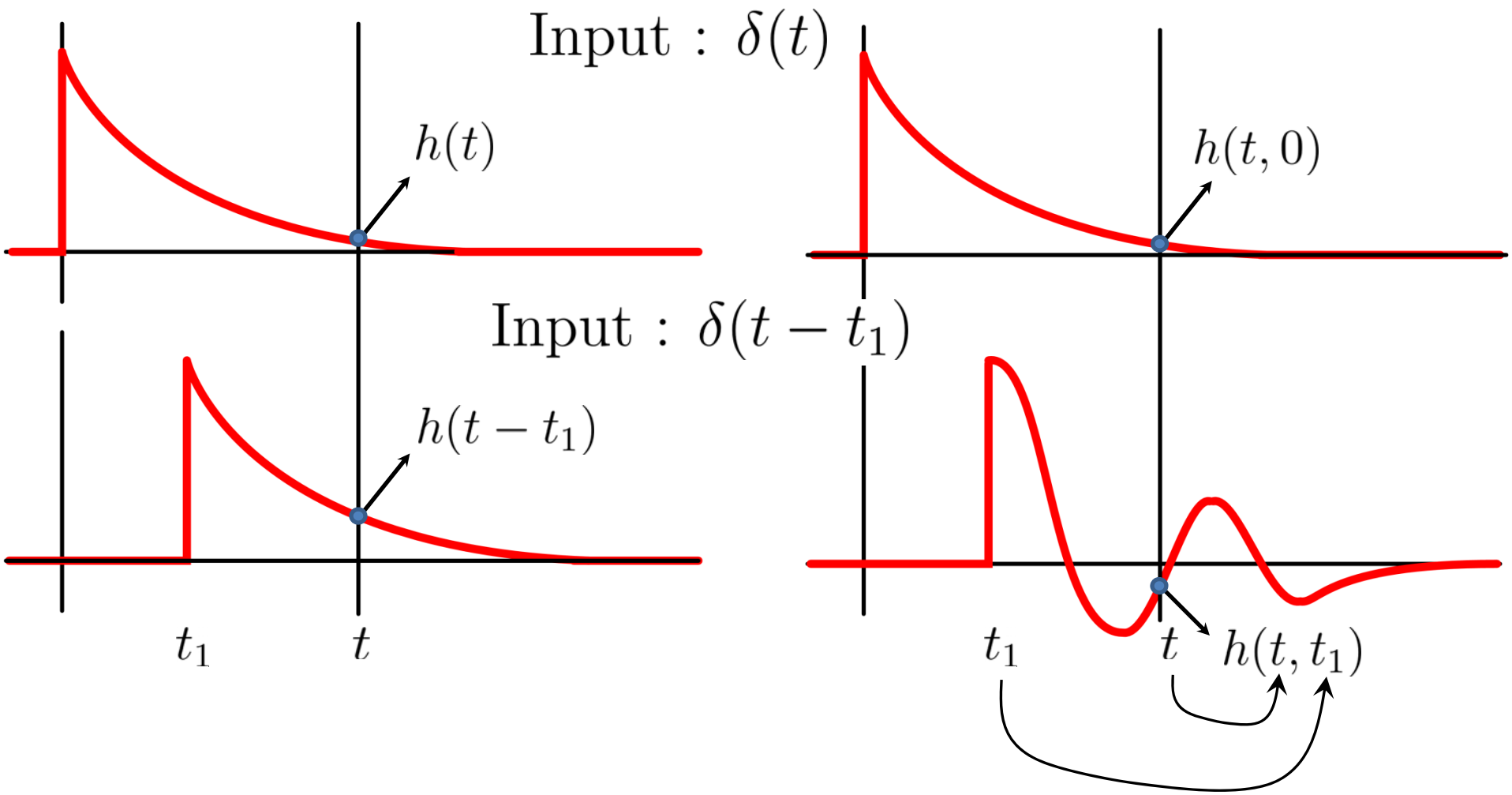
- PSS, PAC, PXF, Pnoise
  - Pertain to systems described by linear differential equations with periodically time varying coefficients
  - Linear networks with periodic switching
- Why are they important ?
  - All ADCs aim to be Linear Periodically Time Varying (LPTV) (if quantization noise is zero)

LPTV Systems

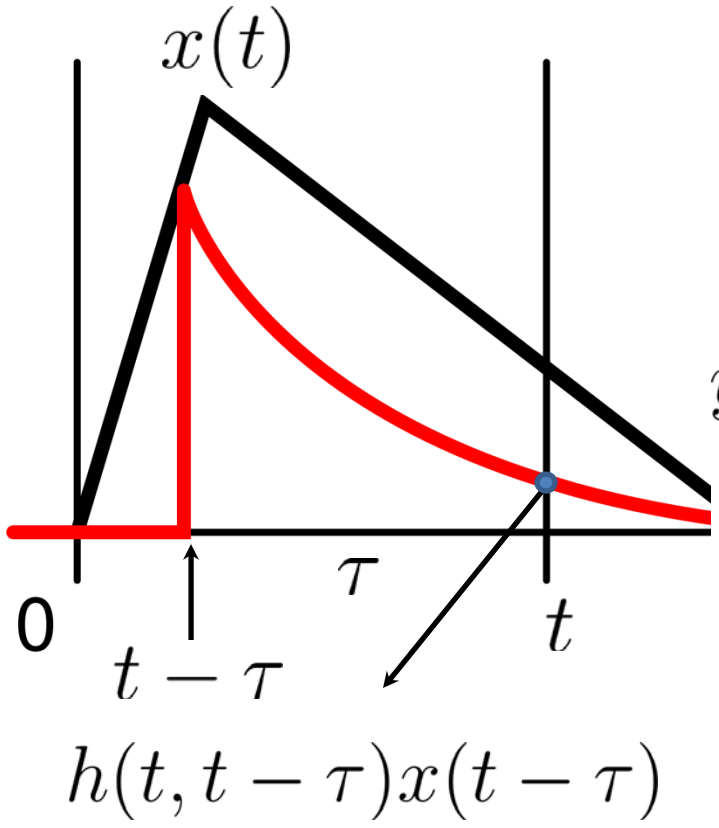
- Time domain analysis time consuming
  - Especially with noise
- Frequency domain analysis of LPTV systems
  - Yields certain kinds of insight
  - Now supported by several commercial simulators
  - Powerful and fast analysis tool
  - Helps to know “what is going on” behind these methods

Linear Time Invariant

Linear Time Variant



LTV System



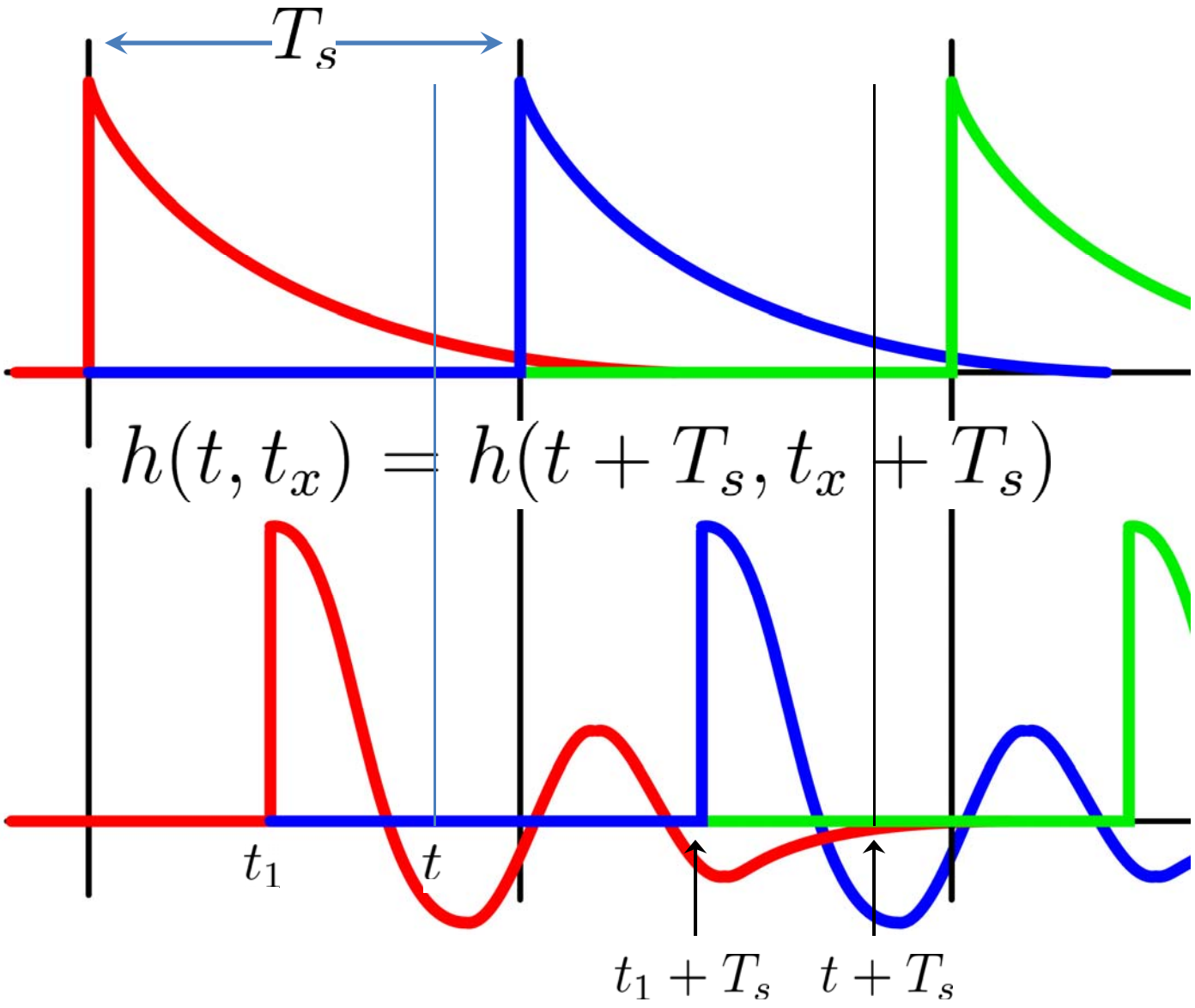
$$y(t) = \int_0^\infty h(t, t-\tau)x(t-\tau)d\tau$$

$$x(t) = e^{j\omega t}$$

$$\begin{aligned} y(t) &= \int_0^\infty h(t, t-\tau)e^{j\omega(t-\tau)}d\tau \\ &= e^{j\omega t}\int_0^\infty h(t, t-\tau)e^{-j\omega\tau}d\tau \\ &= e^{j\omega t}H(j\omega; t) \end{aligned}$$

Time Dependent
Frequency Response

L-Periodically-TV Systems



LPTV Systems

$$H(j\omega;t) = \int_0^\infty h(t,t-\tau)e^{-j\omega\tau}d\tau$$

$$H(j\omega;t+T_s) = \int_0^\infty h(t+T_s,t+T_s-\tau)e^{-j\omega\tau}d\tau$$

$$= \int_0^\infty h(t,t-\tau)e^{-j\omega\tau}d\tau = H(j\omega;t)$$

$$H(j\omega;t+T_s) = H(j\omega;t) = \sum_{l=-\infty}^{l=\infty} H_l(j\omega)e^{jl\omega_s t}$$

$\frac{2\pi}{T_s}$

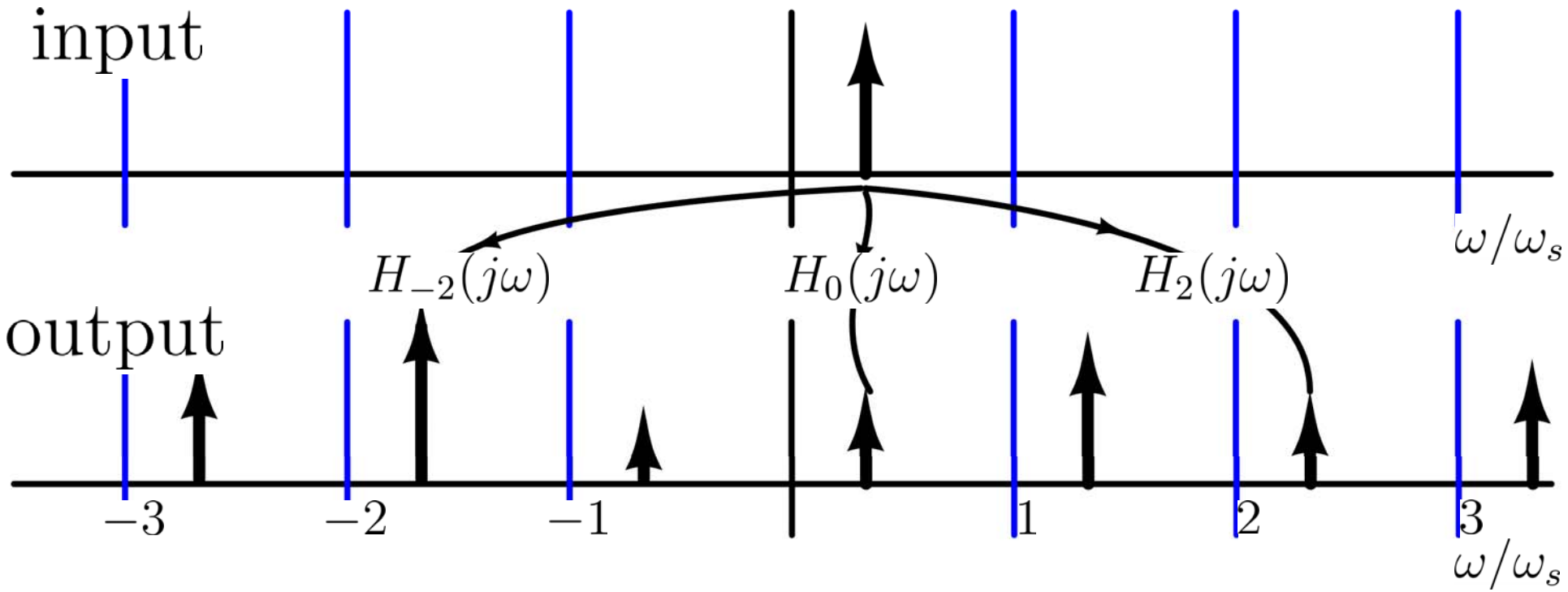
$$\text{Input}: x(t) = e^{j\omega t}$$

$$y(t) = \sum_{l=-\infty}^{l=\infty} H_l(j\omega)e^{j(\omega+l\omega_s)t}$$

© 2013 IEEE

$$y(t) = \sum_{l=\infty}^{l=\infty} H_l(j\omega)e^{j(\omega+l\omega_s)t}$$

sideband

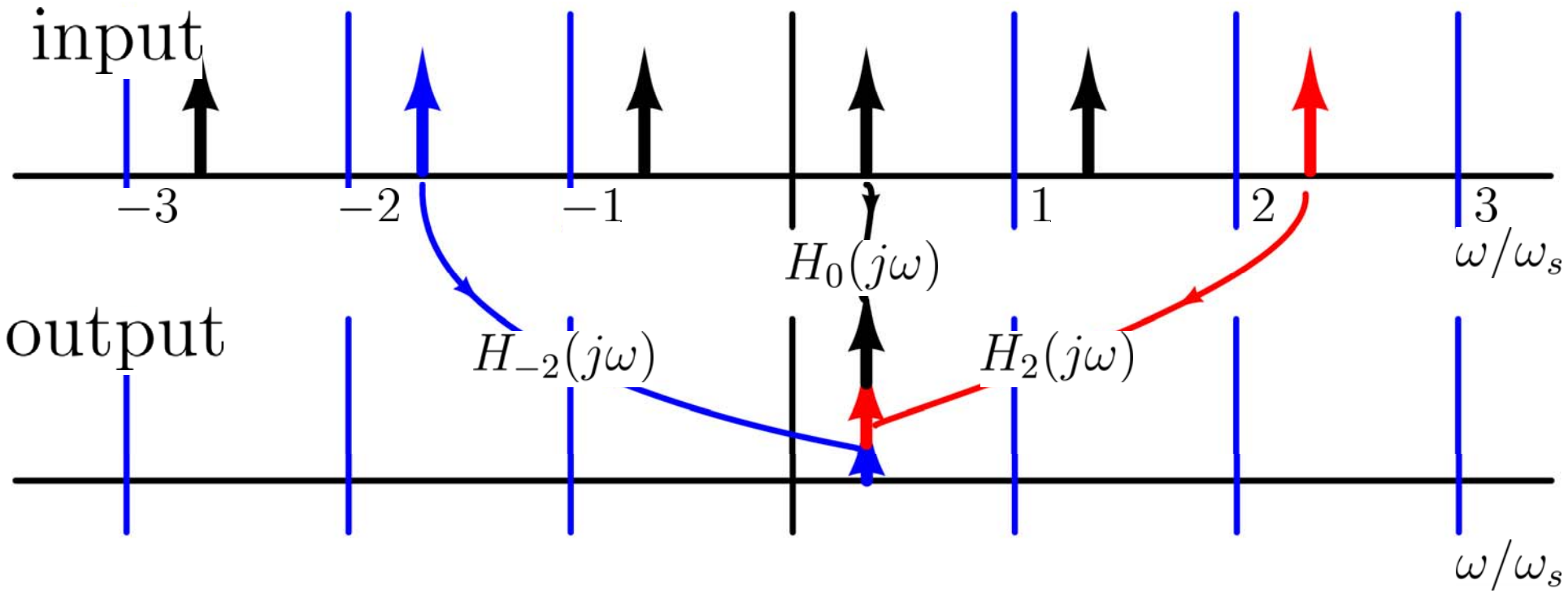


Input at ω causes components at $\omega \pm (0, 1, 2, \cdots)\omega_s$

Periodic AC Analysis yields $H_l(j\omega)$ for any desired l

PAC Analysis

Periodic XF Analysis



Output at ω due to components at $\omega \pm (0, 1, 2, \cdots)\omega_s$

Periodic XF Analysis yields $H_l(j\omega)$ for any desired l

PXF Analysis

Noise in LPTV Systems

- Frequency translation is an inherent property of LPTV systems
- Same applies to noise

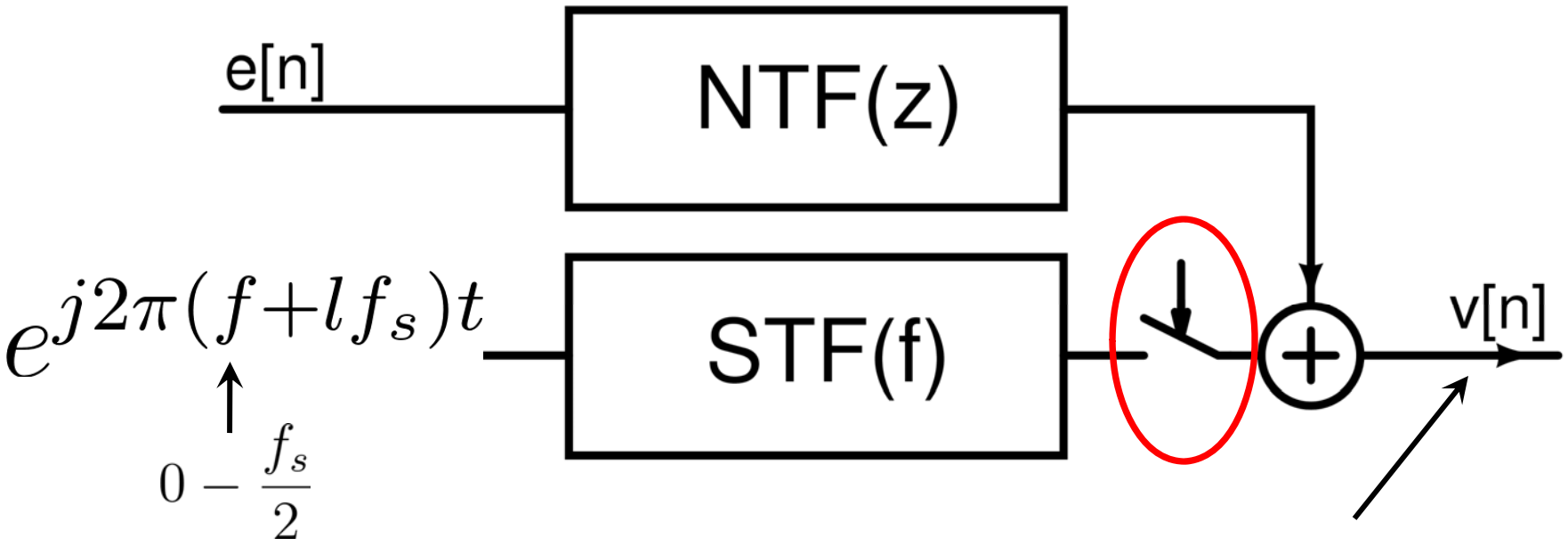
Noise at ω due to components at $\omega \pm (0, 1, 2, \cdots)\omega_s$

PNoise finds noise accounting for freq translation

ΔΣ ADC : Simulations

- Inband SNDR
  - How to calculate this ?
  - How long should I simulate ?
- Realized NTF
  - As opposed to intended NTF
- Realized STF
  - Especially in CT modulators
- Device Noise

Linear Model : CT DSMs



$$STF\left[j2\pi(f+lf_s)\right]e^{j2\pi fT_s n}$$

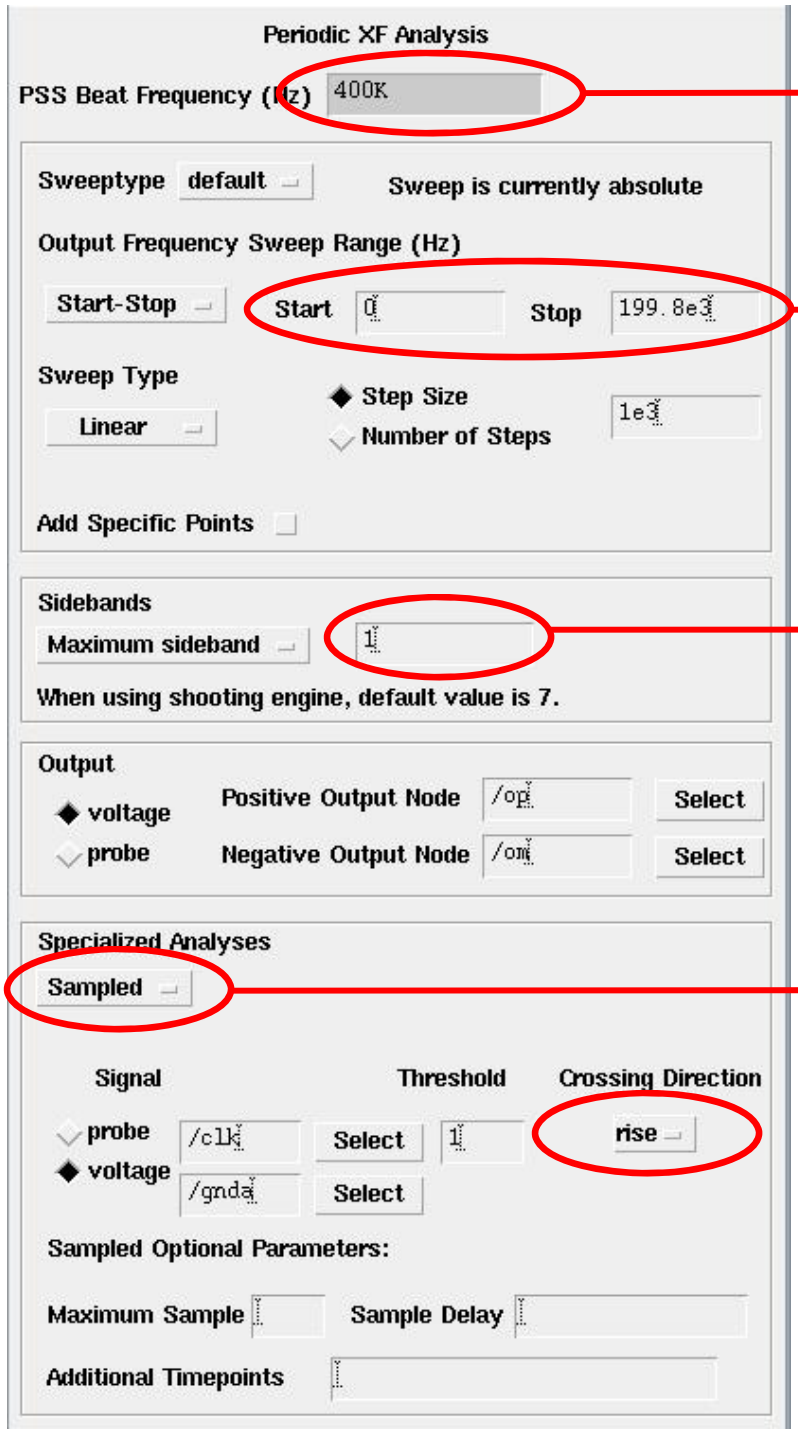
Determining the STF

Atleast two ways of doing this

- **A :** Put in a sinewave, run the modulator, measure the strength of the output sinewave, repeat for another frequency
  - Takes very long, if high resolution of frequency is needed
  - Amplitude of the tone has to be carefully chosen
  - The “true” STF, accounting for the quantizer gain

Determining the STF

- **B:** Periodic Transfer Function (PXF) analysis
  - Replace quantizer by a sampler
  - The system is LPTV
  - Run PSS/PXF
  - Yields the STF assuming unity quantizer gain



Sampling Rate

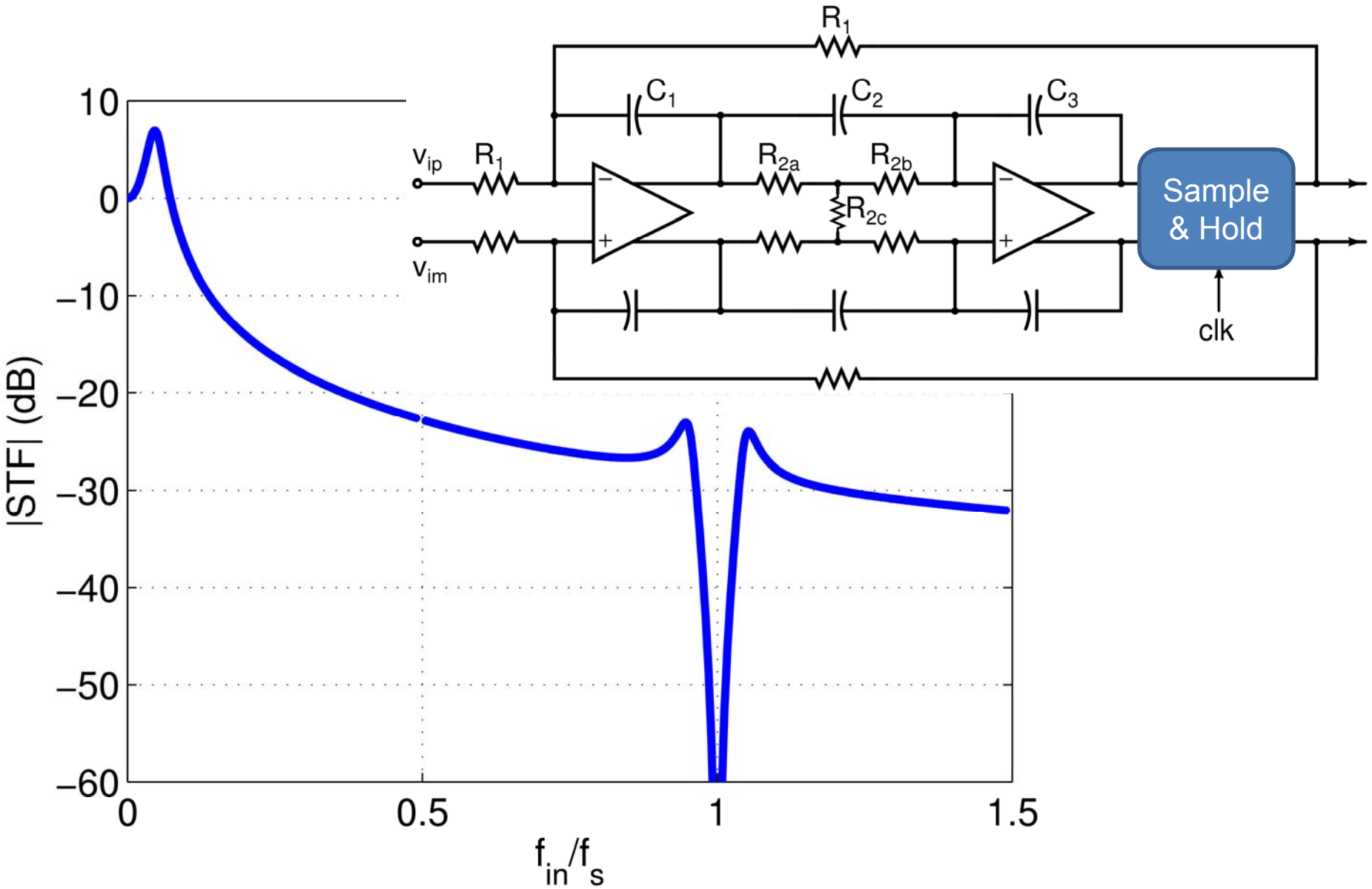
$0 - f_s/2$

To find STF in the range

$$\pm\left([0-\frac{f_s}{2}]+\{\mathbf{0},\mathbf{1}\}f_s\right)$$

Only the output samples at clock rising edge are relevant

STF from PXF Analysis

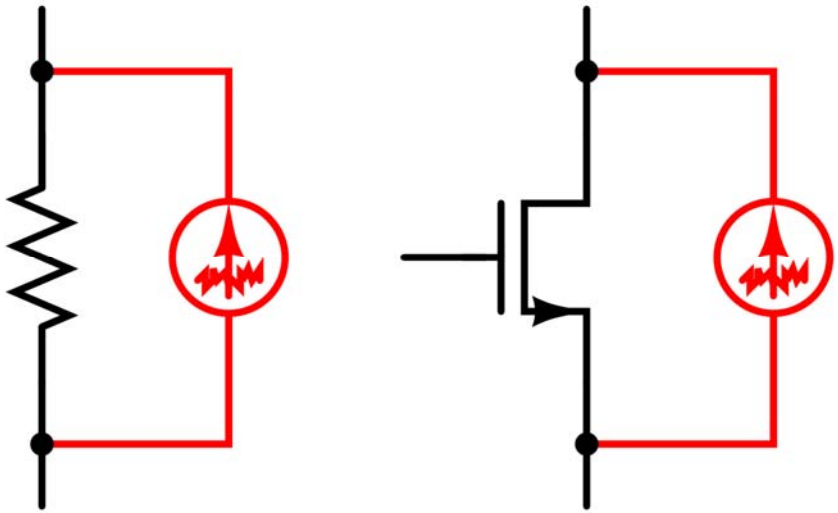


ΔΣ ADC : Simulations

- Inband SNDR
  - How to calculate this ?
  - How long should I simulate ?
- Realized NTF
  - As opposed to intended NTF
- Realized STF
  - Especially in CT modulators
- Device Noise

Transient Noise Analysis

- Conceptually straightforward

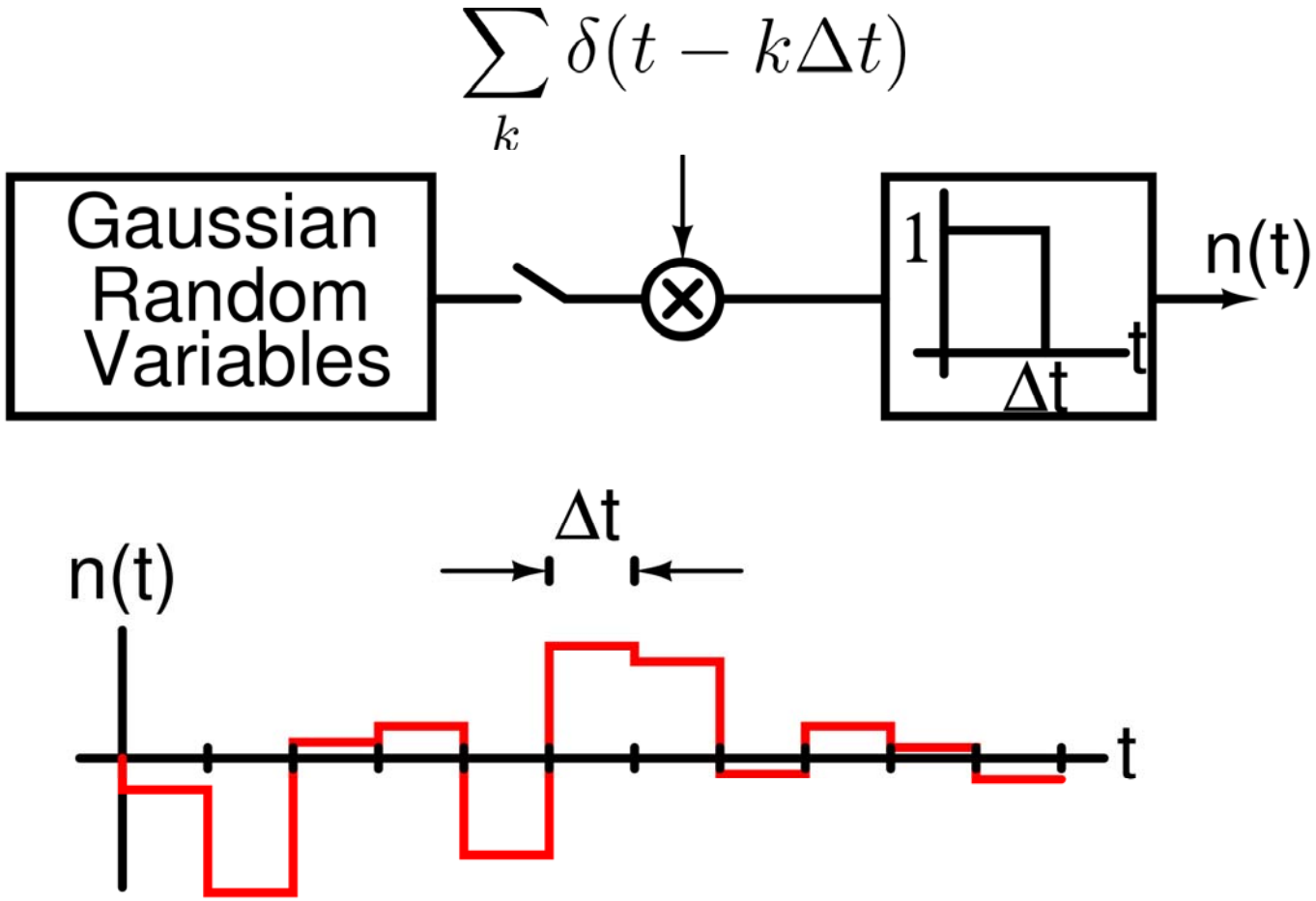


- Noise currents added in parallel with device currents
  - Spectral density can be bias dependent

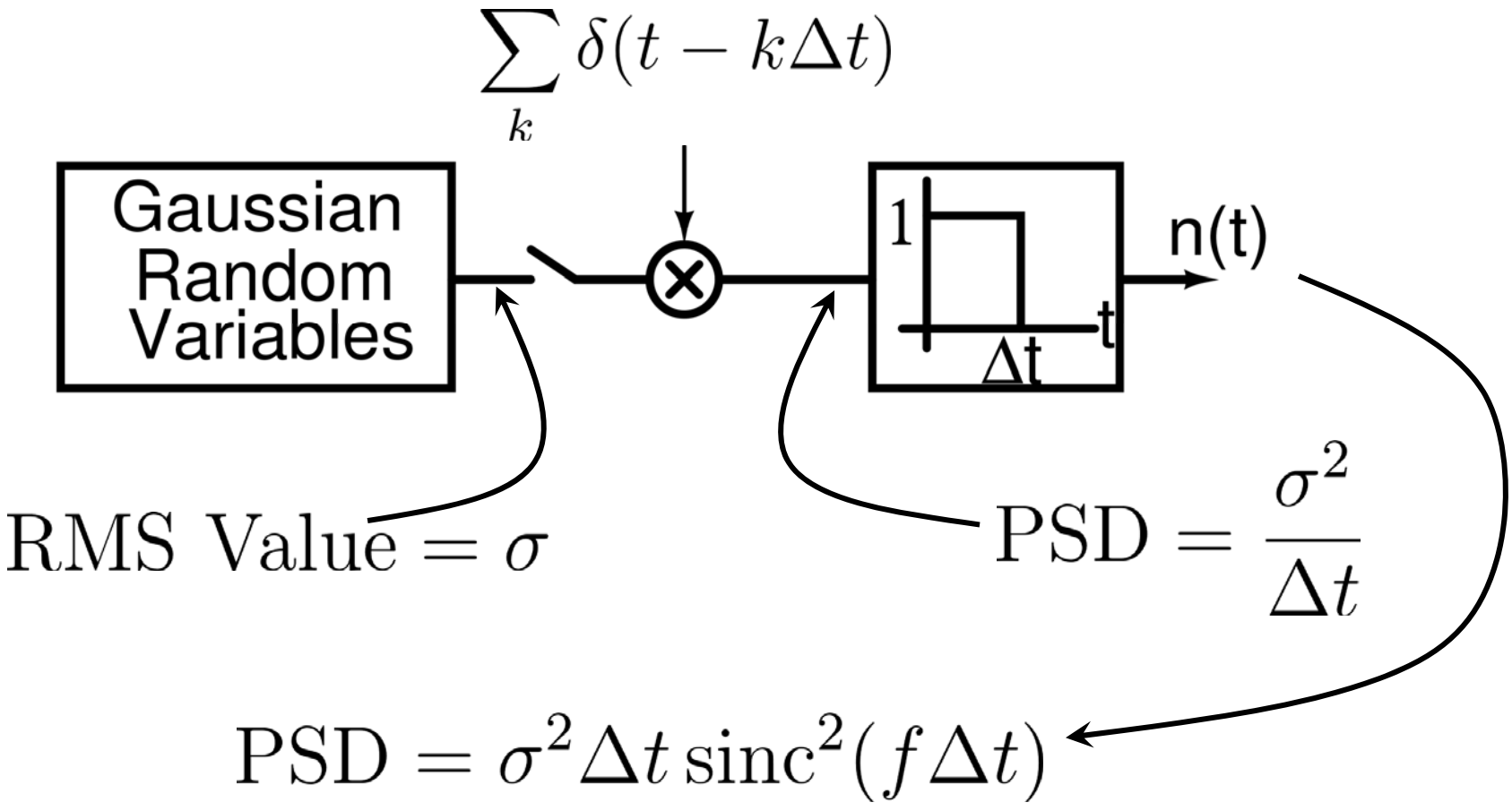
Transient Noise Analysis

- Noise updated at every simulation step
  - White and colored noise can be modeled
- Results need careful interpretation
- Very expensive with respect to simulation time
  - Can run many instances simultaneously
  - Use only as needed

Generating White Noise

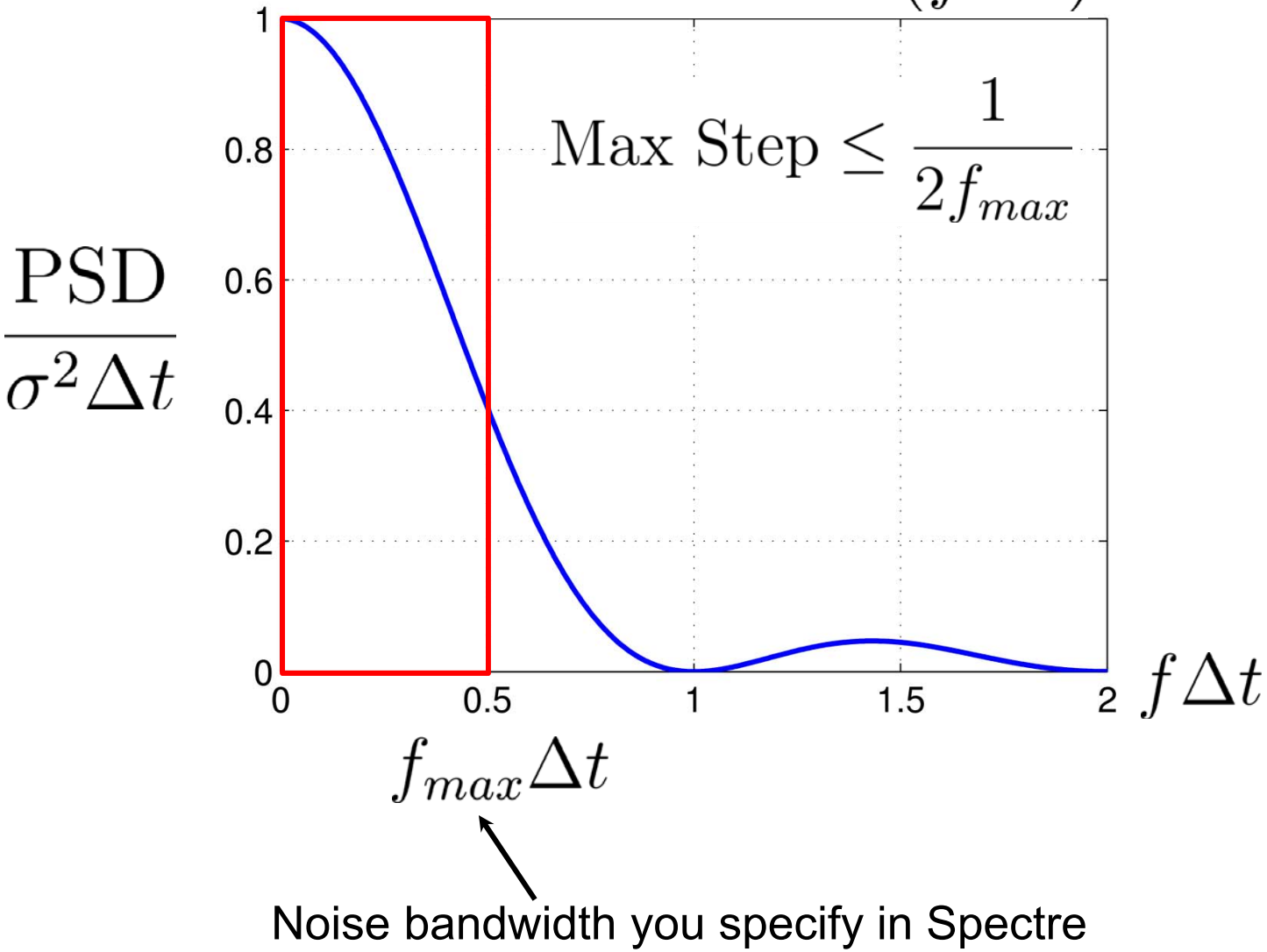


Generating White Noise



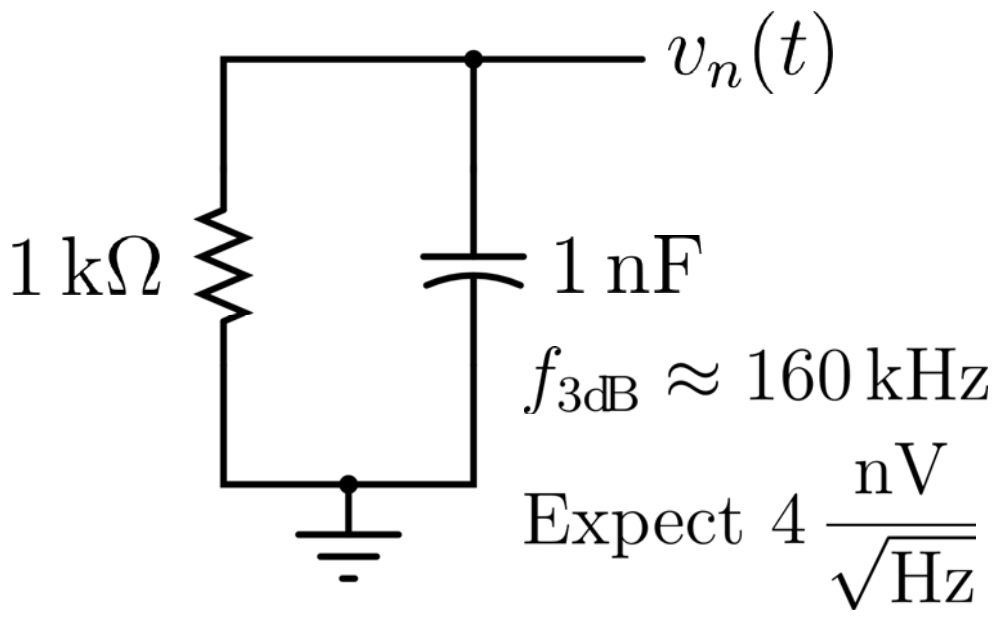
Generating White Noise

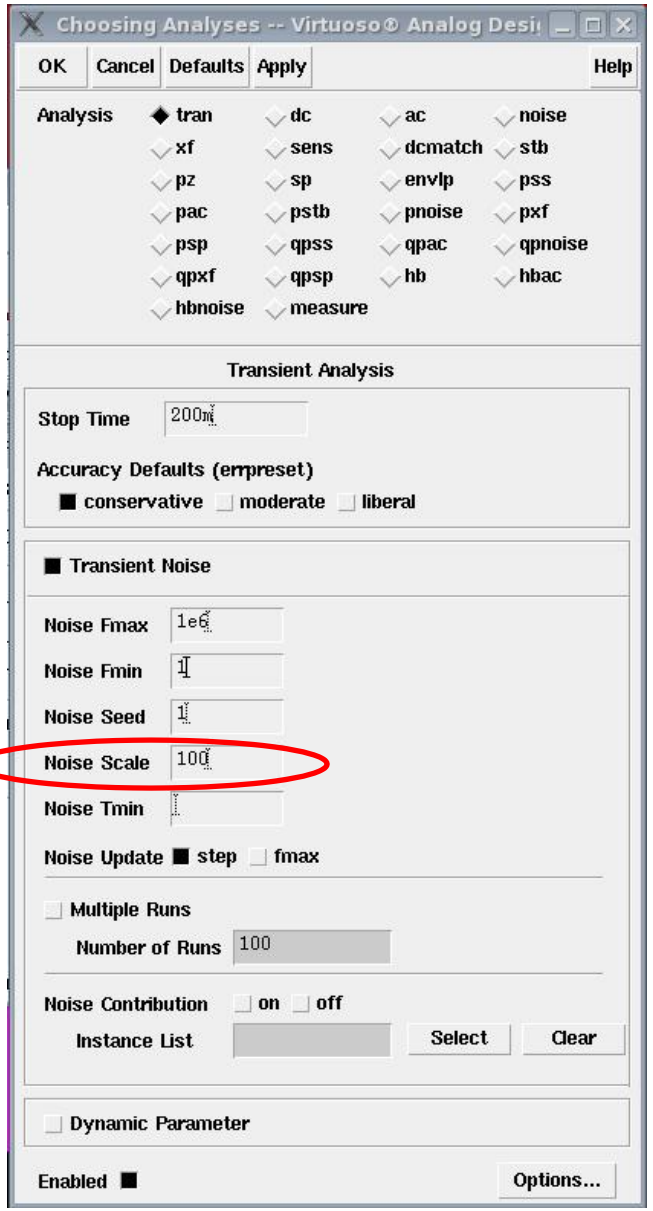
$$\mathrm{PSD} = \sigma^2 \Delta t \, \mathrm{sinc}^2(f \Delta t)$$



Noise bandwidth you specify in Spectre

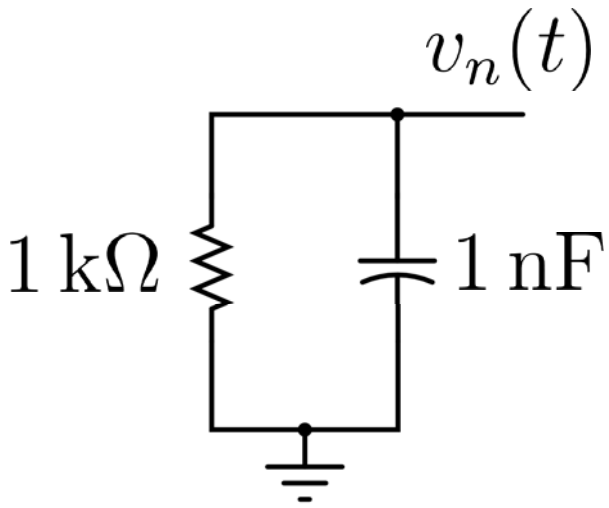
Simple Example



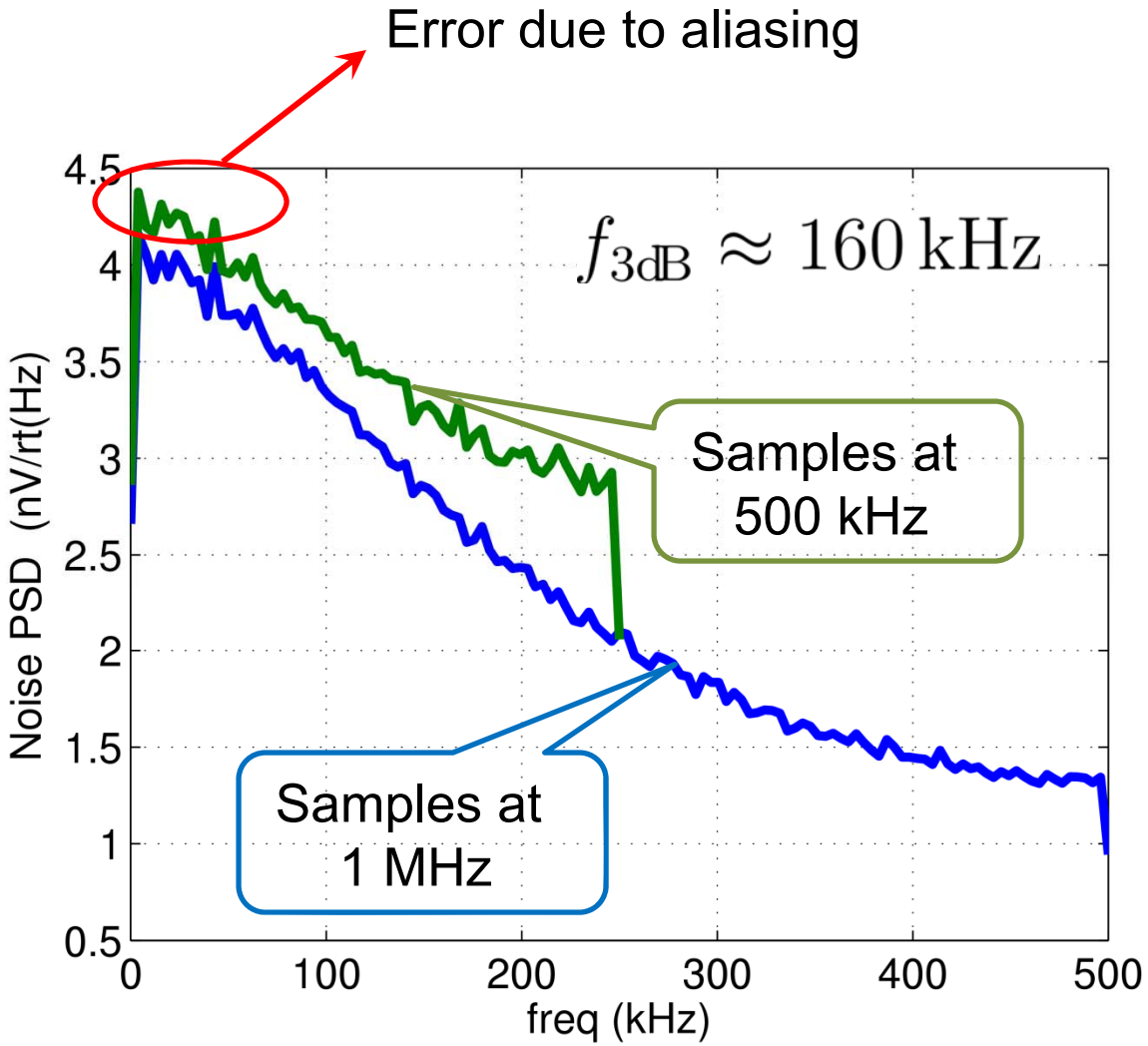


Scales noise voltages and currents by 100

Simple Example

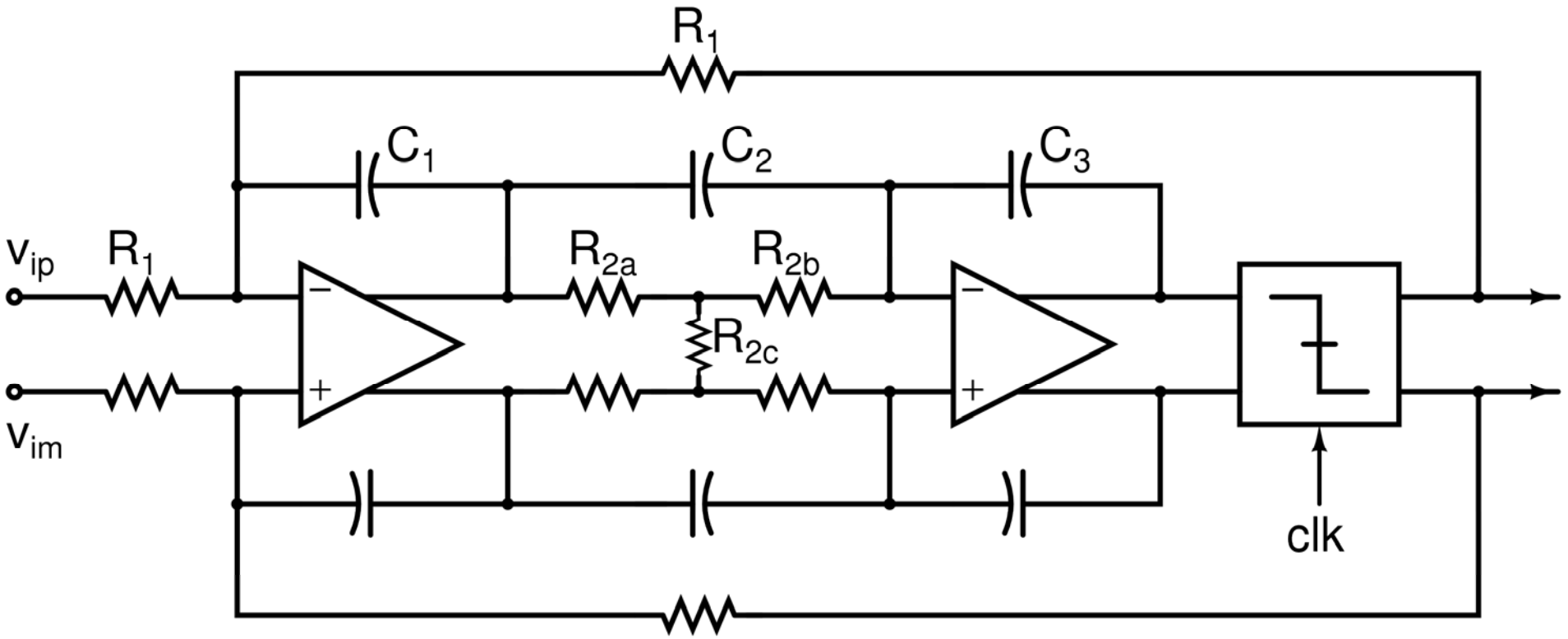


Expect $4\ \frac{\text{nV}}{\sqrt{\text{Hz}}}$




```
[P, f] = pwelch(0.01*vn,rectwin(Nfft),0,Nfft,fs)
```

Second Order CTDSM Example



400 kHz sampling rate

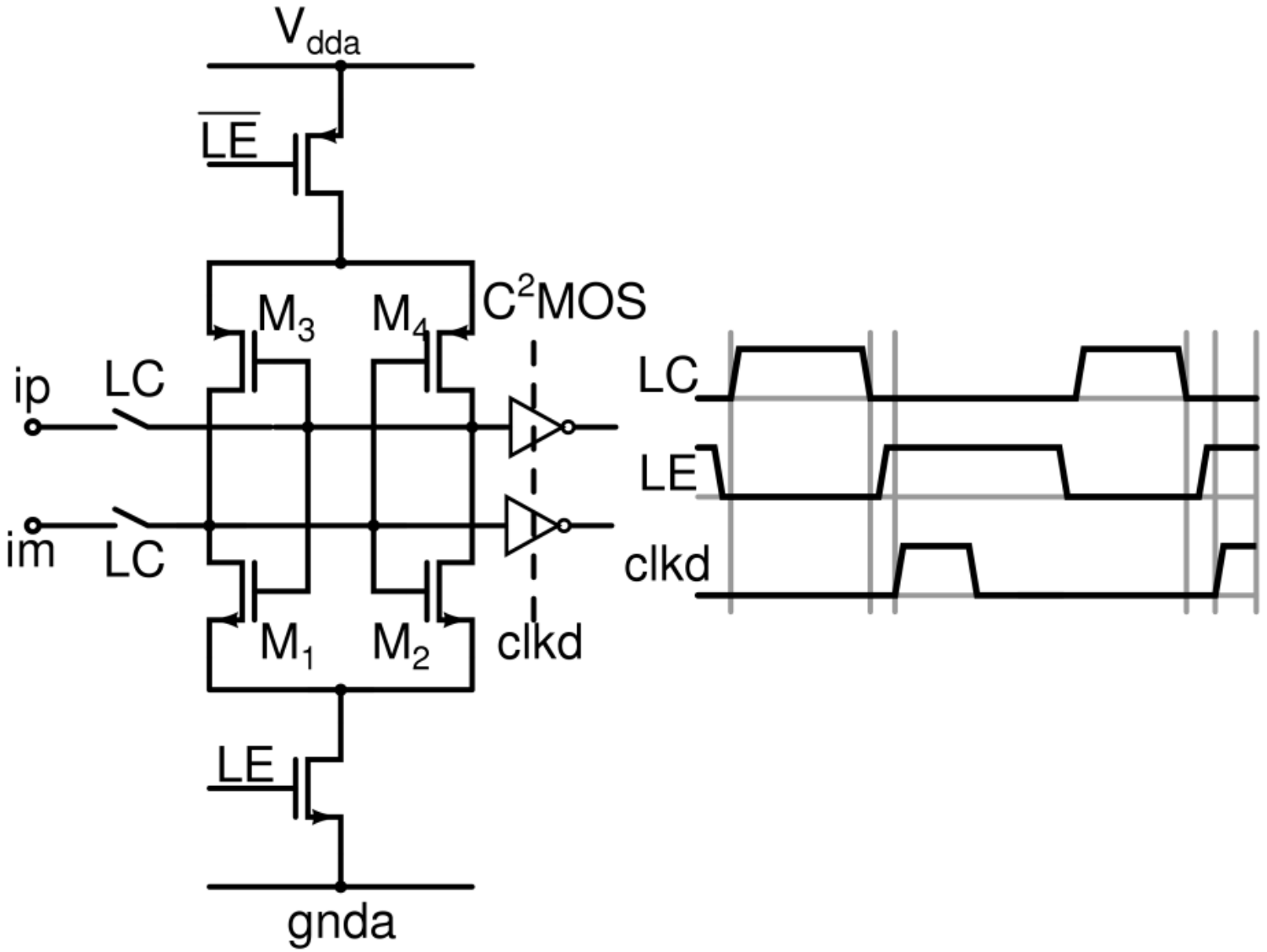
Not amenable to periodic noise analysis

Delta Sigma Modulator Noise

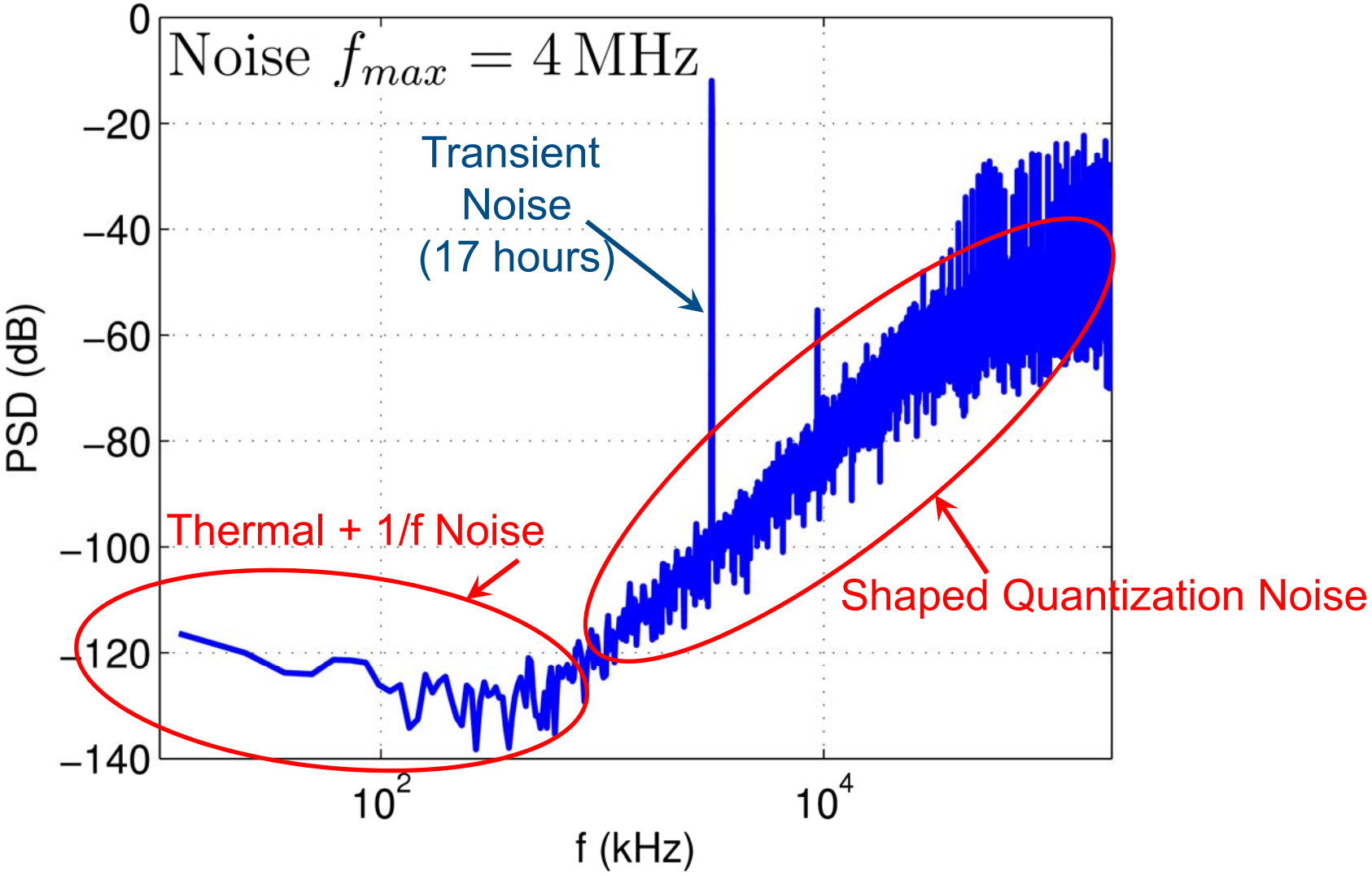
- In-band noise dominated by thermal and 1-by-f noise
- Quantization noise should be made lower by at least 12-15 dB
- Virtually all noise should come from the loop filter

Opamp Schematic

Comparator and Clock Waveforms



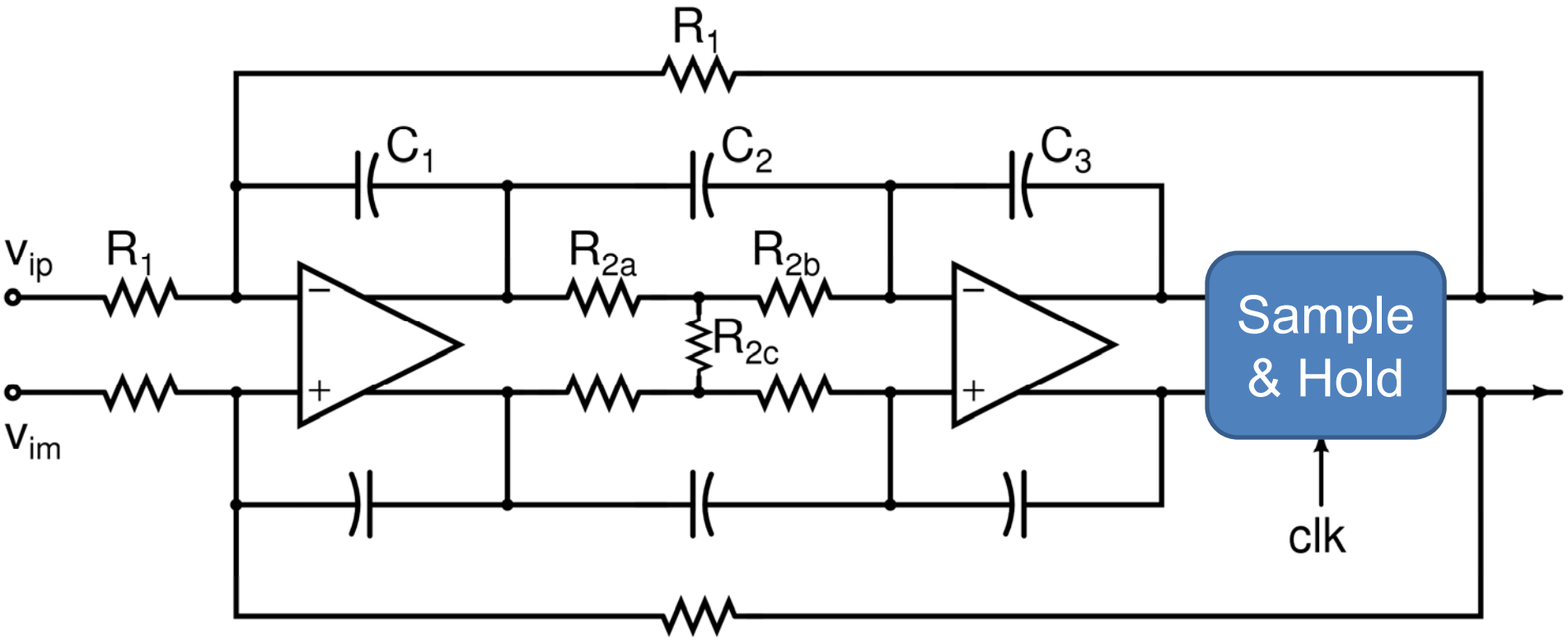
Delta Sigma Example



ΔΣ Noise Quick Simulation

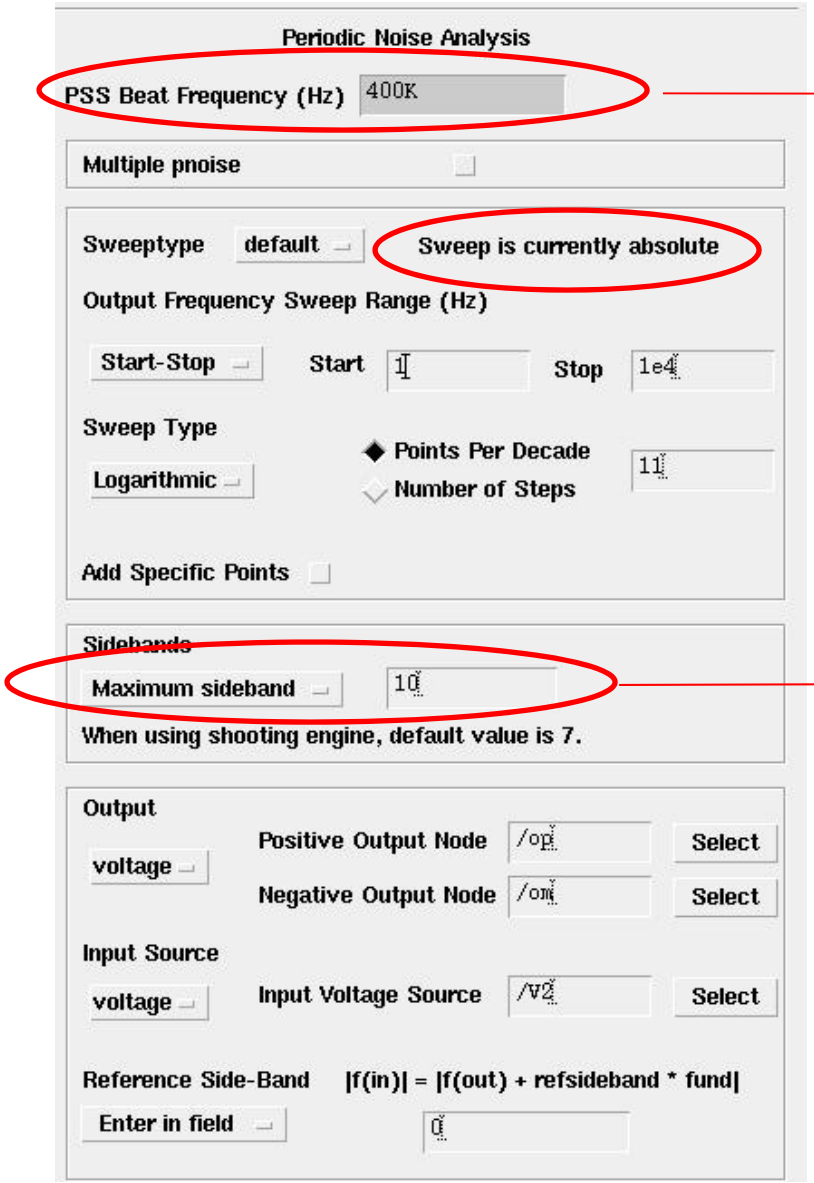
- Transient noise
  - Computationally expensive
- Periodic Noise is very fast
  - Delta Sigma ADCs do not lend themselves easily to periodic noise analysis (PNOISE)
  - PSS difficulties due to the quantizer

CTDSM : Using PNoise



Replace comparator with Sample & Hold
CTDSM becomes LPTV

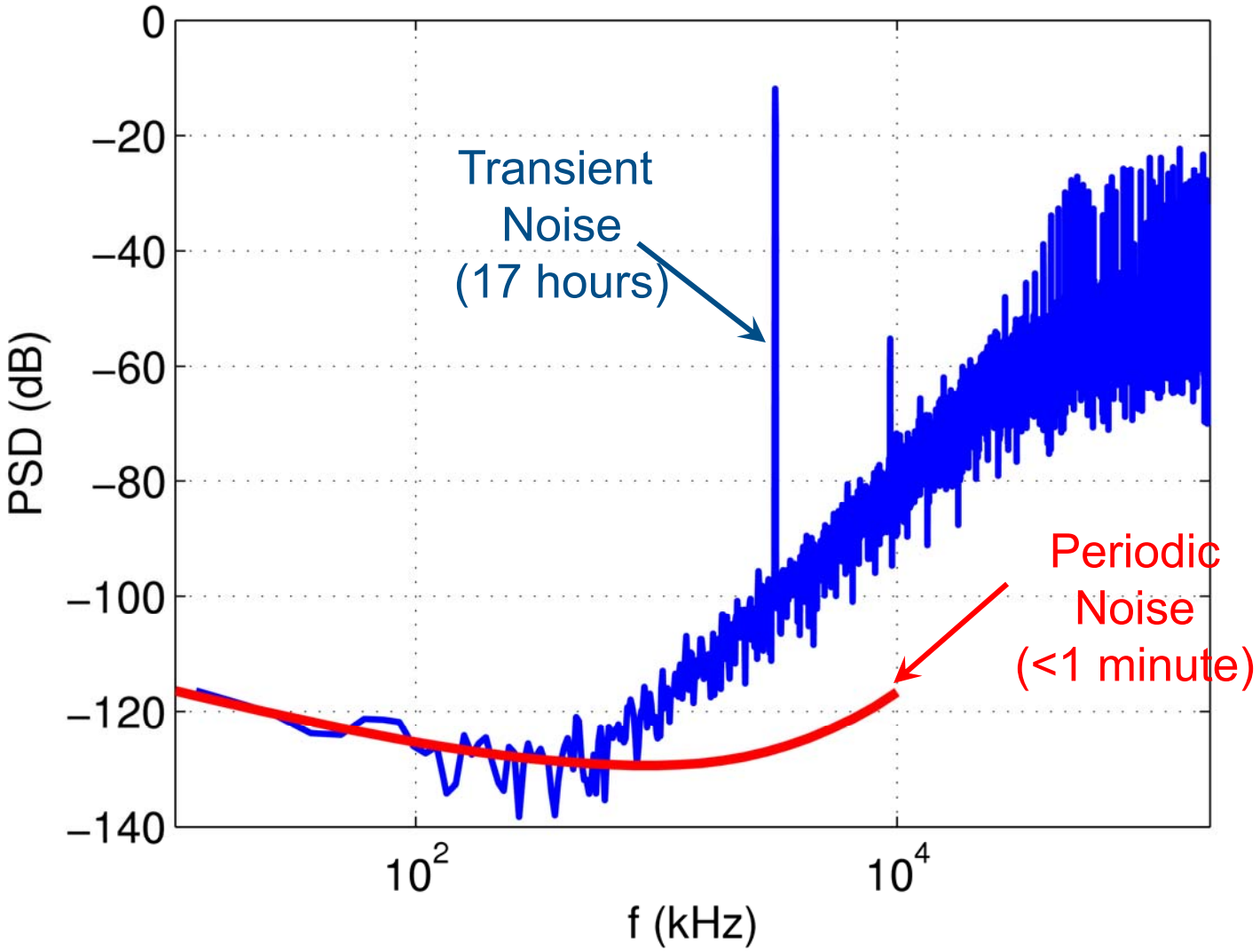
Pnoise Setup



LPTV fundamental frequency

Aliases noise from 10 sidebands

Delta Sigma Example with PNoise



© 2013 IEEE

Summary

- Data Converter Overview
  - Simulation in Flash and Pipeline ADCs
- Spectral analysis in ADCs
  - Windowing
- Simulation Techniques for $\Delta\Sigma$ ADCs
  - In-band SNR estimation
  - Determining NTF
- Use of RF analyses in data converter simulations
  - STF and Noise simulations

Closing Remarks

- Every ADC/DAC architecture
  → own design **and simulation** challenges
- Analysis and design iteration
  → behavioral models/simulation
- Design debug
  → mixture of tx level and macromodels
- Innovation
  →e.g : exploit RF analyses of commercial tools to reduce simulation times

References

- General Data Converter Reading
  - M. Gustavsson, J. Wikner & N. Tan, “CMOS Data Converters for Communications”, *Springer*
  - R. Jacob Baker, “CMOS : Mixed Signal Circuit Design”, *IEEE Press*
- Flash and Pipeline ADCs
  - R.van de Plassche : “CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters”, *Springer*
- Delta Sigma Data Converters
  - R.Schreier, S.Norsworthy and G.Temes, “Delta-Sigma Data Converters : Theory, Design and Simulation”, IEEE Press
  - R.Schreier and G.Temes, “Understanding Delta-Sigma Data Converters”, *Wiley*
  - J.Cherry, “Theory, Practice, and Fundamental Performance Limits of High-Speed Data Conversion Using Continuous-Time Delta-Sigma Modulators”, PhD Dissertation http://www.dissonance.com

References

- RF analyses background
  - K.Kundert, “Introduction to RF Simulation and its Applications”, *IEEE Journal of Solid State Circuits, Sept 1999.*
- Verilog-A Macromodels
  - White papers at http://www.designers-guide.org
- Macromodeling Tools
  - MATLAB and Simulink
  - The Schreier Delta-Sigma Toolbox
  - CPPSim http://cppsim.com