# EE288 Data Conversions/Analog Mixed-Signal ICs
# Spring 2018

## Lecture 3: ADC Architectures, Sampling

Prof. Sang-Soo Lee
sang-soo.lee@sjsu.edu
ENG-259

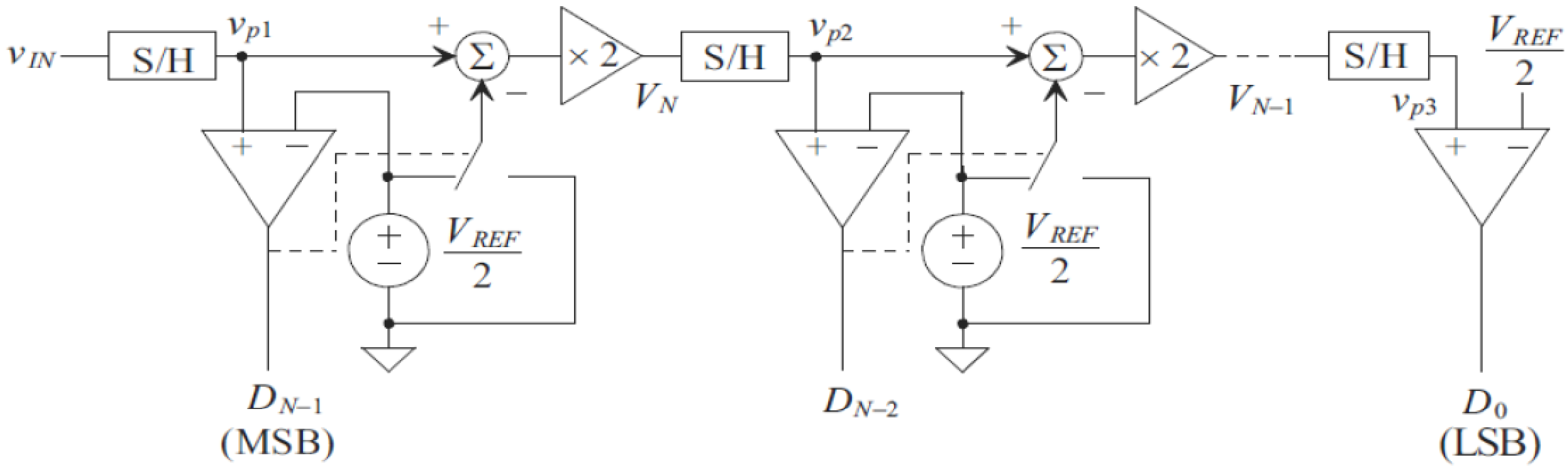
# Course Schedule – Subject to Change

| Date | Topics |
|---|---|
| 24-Jan | ~~Course introduction and ADC architectures~~ |
| 29-Jan | ~~Converter basics: AAF, Sampling, Quantization, Reconstruction~~ |
| 31-Jan | ~~ADC dynamic performance metrics, Spectrum analysis using FFT~~ |
| 5-Feb | ADC & DAC static performance metrics, INL and DNL |
| 7-Feb | OPAMP and bias circuits review |
| 12-Feb | SC circuits review |
| 14-Feb | Sample and Hold Amplifier - Reading materials |
| 19-Feb | Flash ADC and Comparators: Regenerative Latch |
| 21-Feb | Comparators: Latch offset, preamp, auto-zero |
| 26-Feb | Finish Flash ADC |
| 28-Feb | DAC Architectures - Resistor, R-2R |
| 5-Mar | DAC Architectures - Current steering, Segmented |
| 7-Mar | DAC Architectures - Capacitor-based |
| 12-Mar | SAR ADC with bottom plate sampling |
| 14-Mar | SAR ADC with top plate sampling |
| 19-Mar | Midterm Review |
| 21-Mar | Midterm exam |
| 26-Mar | Spring break |
| 28-Mar | Spring break |
| 2-Apr | Pipelined ADC stage - comparator, MDAC, x2 gain |
| 4-Apr | Pipelined ADC bit sync and alignment using Full adders |
| 9-Apr | Pipelined ADC 1.5bit vs multi-bit structures |
| 11-Apr | Fully-differential OPAMP and Switched-capacitor CMFB |
| 16-Apr | Single-slope ADC |
| 18-Apr | Oversampling & Delta-Sigma ADCs |
| 23-Apr | Second- and higher-order Delta-Sigma Modulator. |
| 25-Apr | Hybrid ADC - Pipelined SAR |
| 30-Apr | Hybrid ADC - Time-Interleaving |
| 2-May | ADC testing and FoM |
| 7-May | Project presentation 1 |
| 8-May | Project presentation 2 |
| 14-May | Final Review |
| 20-May | Project Report Due by 6 PM |

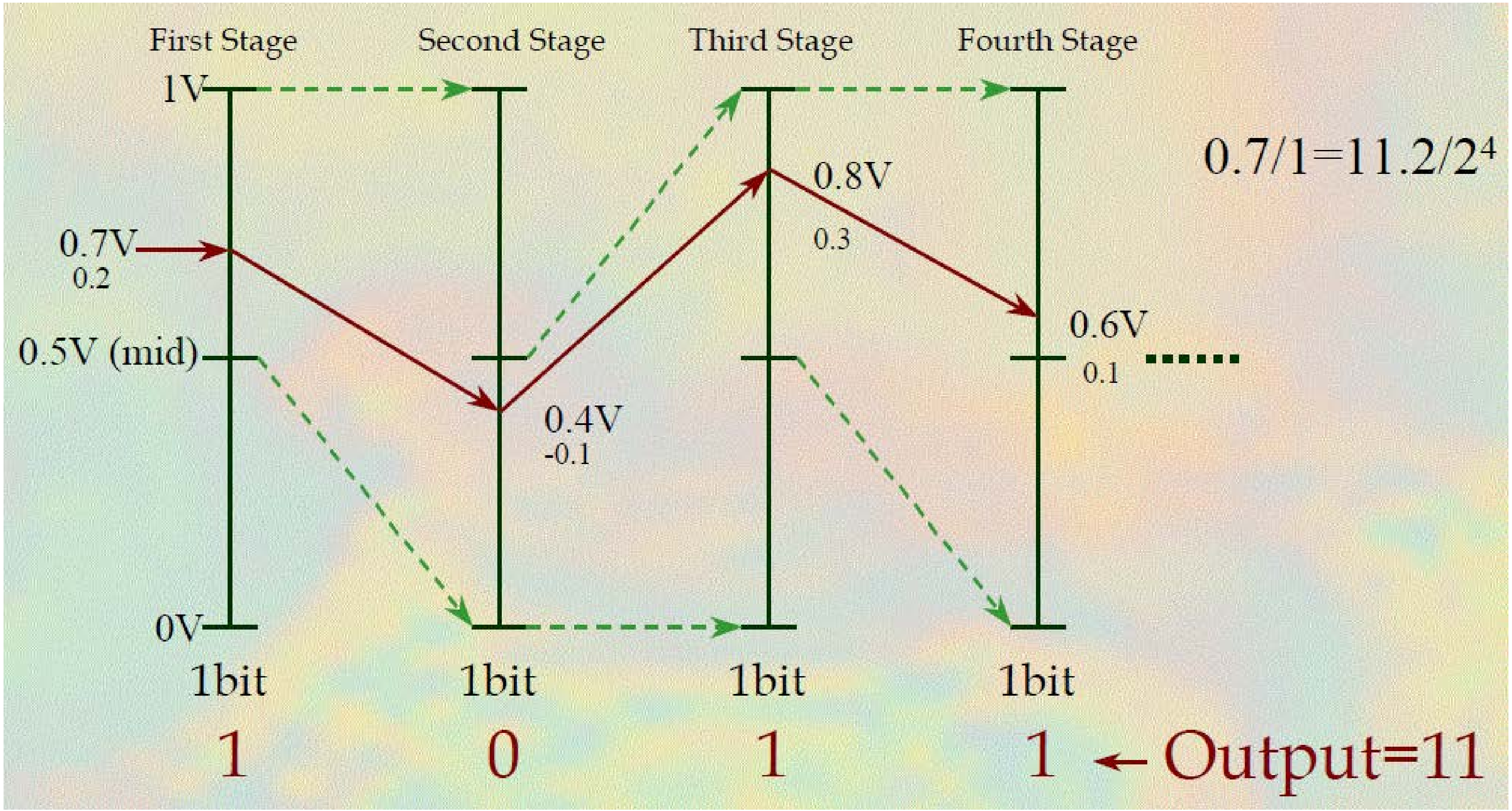
**Sampling**

***Midterm Exam dates are approximate and subject to change with reasonable notice.**

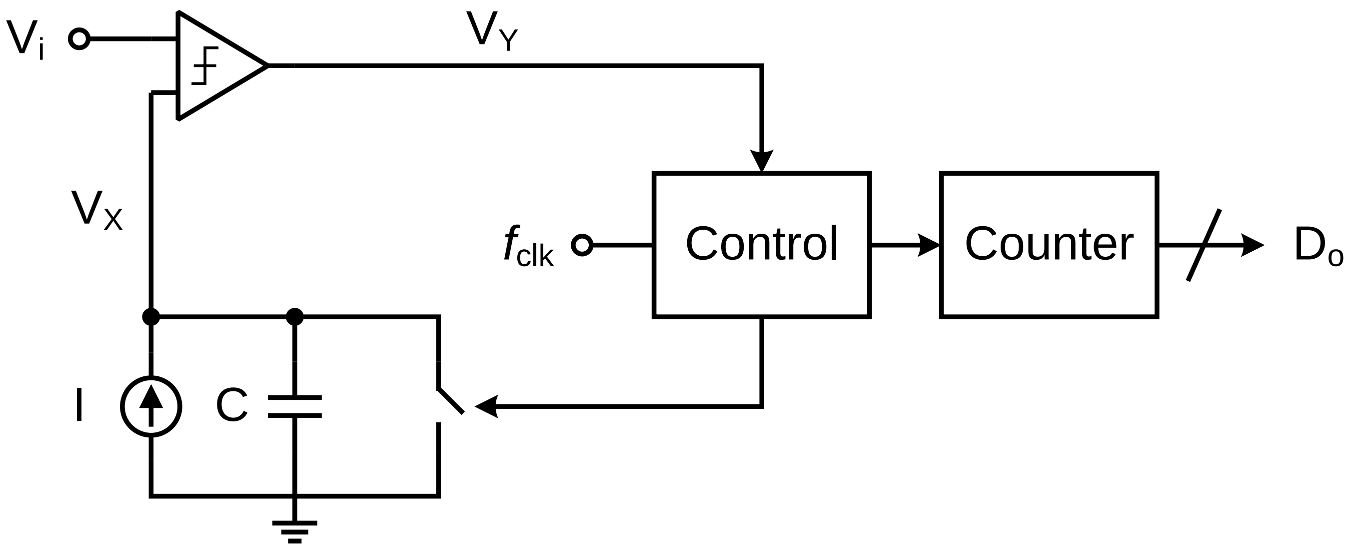
# 1-bit/stage Pipelined ADC



# 1-bit/stage Pipelined ADC

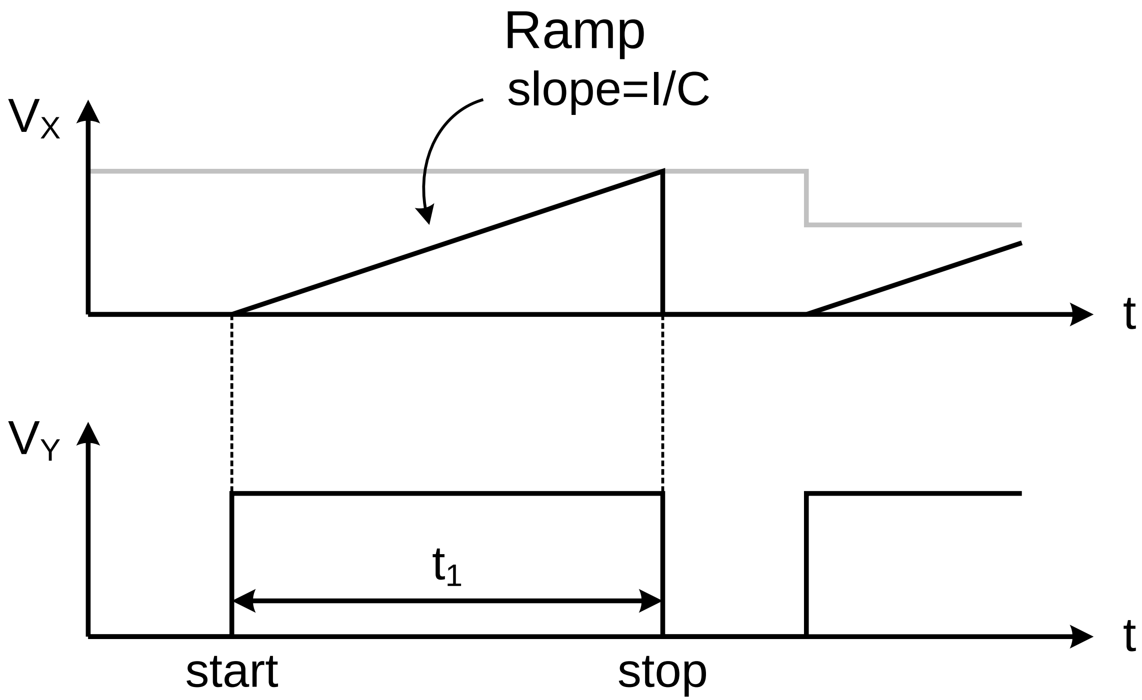


# Single-Slope Integration ADC



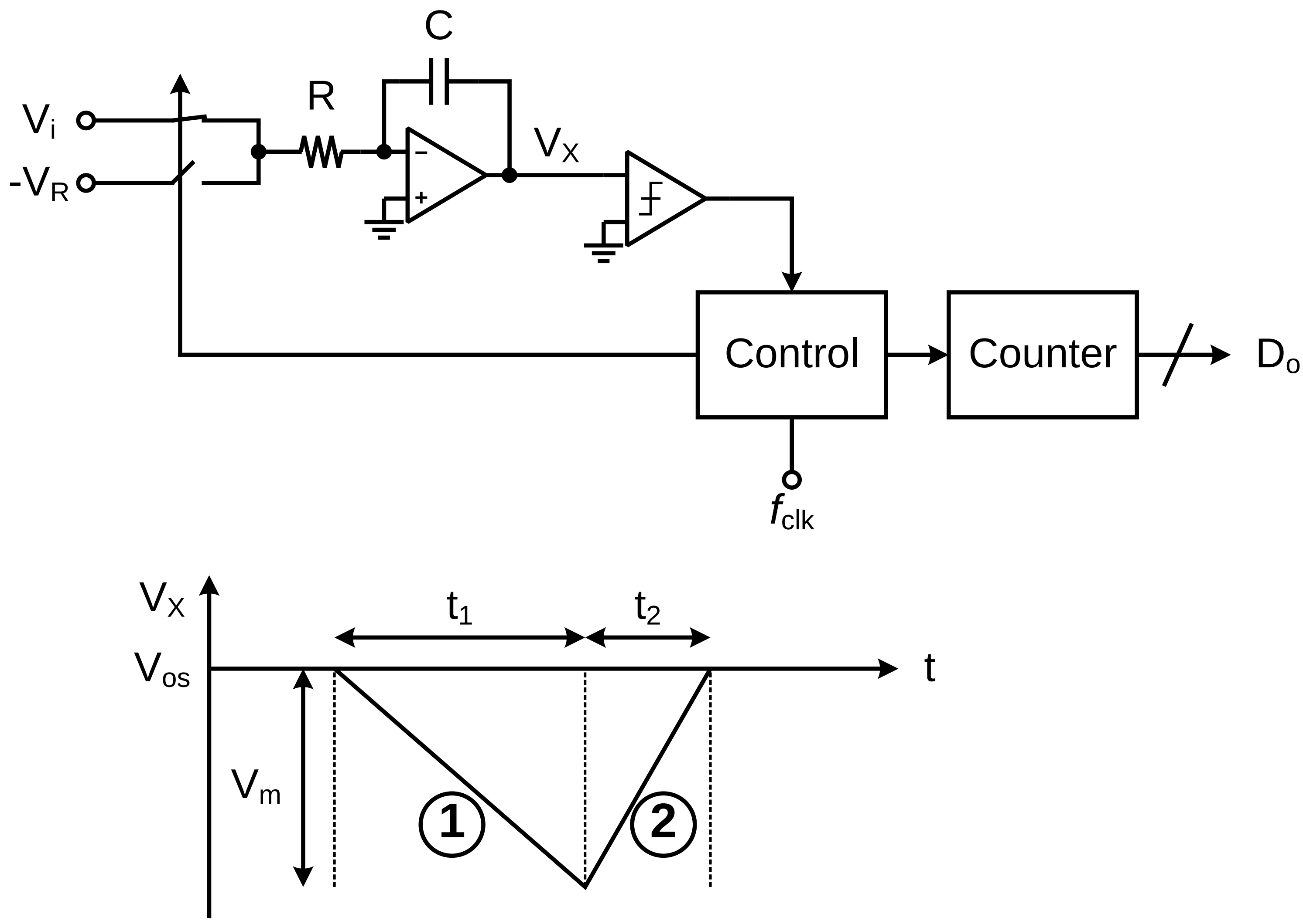

- Sampled-and-held input ($V_i$)
- Counter keeps counting until comparator output toggles
- Simple, inherently monotonic, but very slow ($2^N$*$T_{clk}$/sample)

# Single-Slope Integration ADC

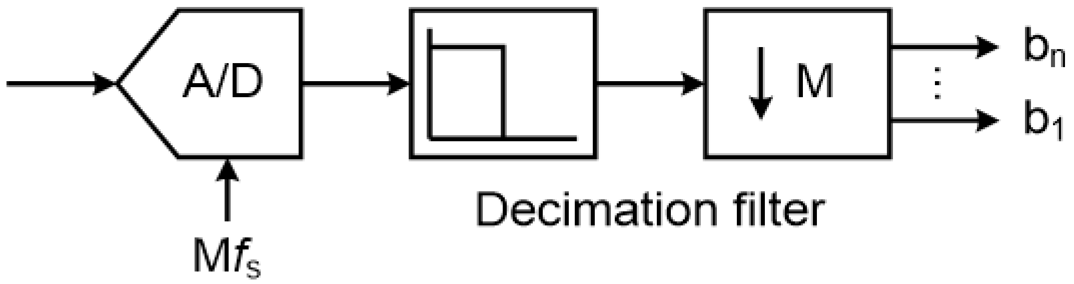


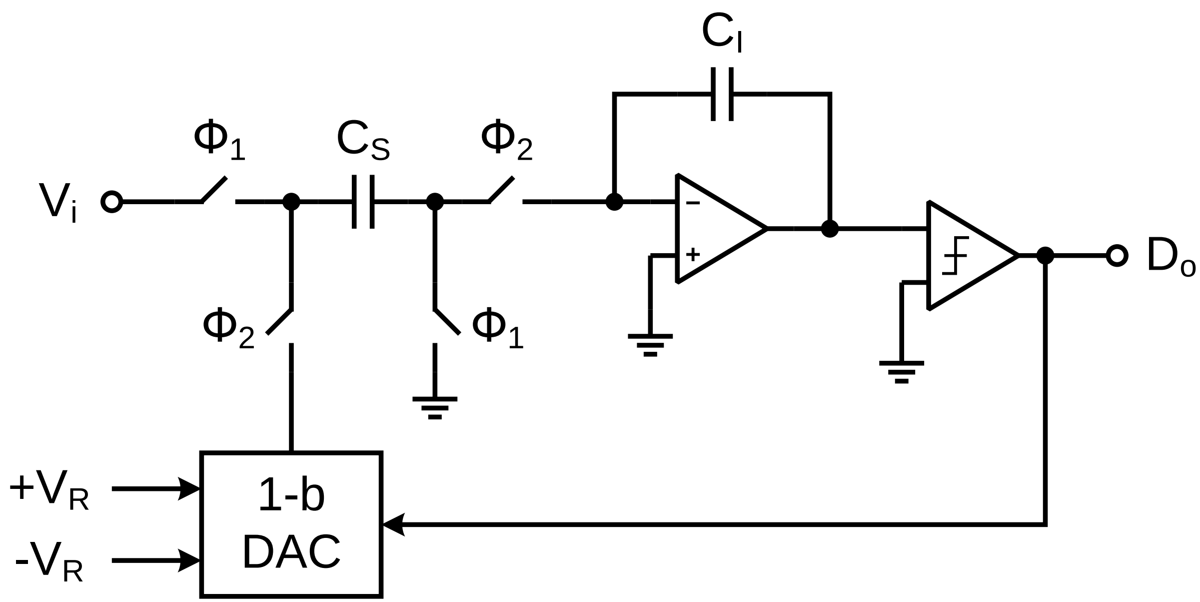

- INL depends on the linearity of the ramp signal
- Precision capacitor (C), current source (I), and clock ($T_{clk}$) required
- Comparator must handle wide input range of [0, $V_{FS}$]

# Dual Slope Integration ADC



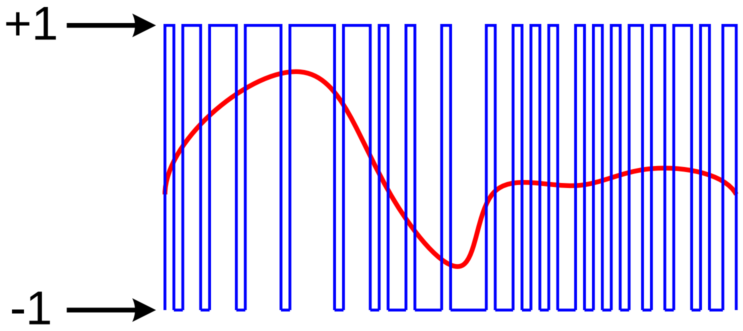
# Oversampling ADC



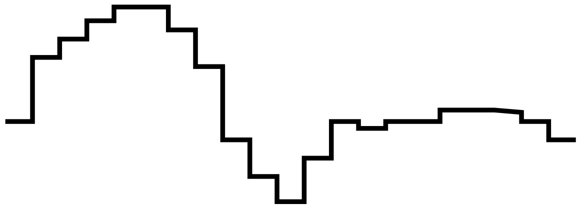



# Sigma Delta vs. Nyquist ADC

<u>ΣΔ ADC output (1-bit)</u>




<u>Nyquist ADC output</u>



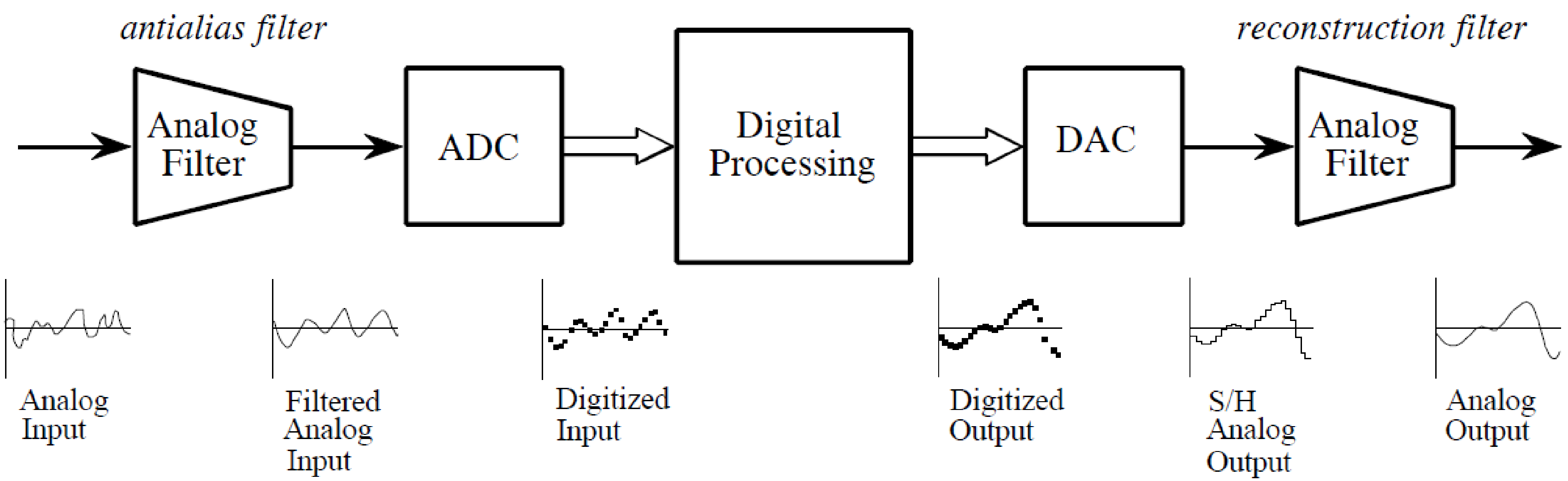
- ΣΔ ADC behaves quite differently from Nyquist converters
- Digital codes only display an “average” impression of the input
- INL, DNL, monotonicity, missing code, etc. do not directly apply in ΣΔ converters → use SNR, SNDR, SFDR instead

# Building Blocks for ADCs

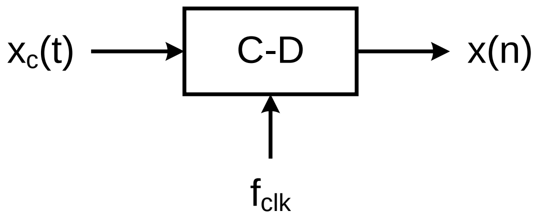
- Comparators (Preamplifier and Latch)
- Operational Amplifier
- Sample-and-Hold (Track-and-Hold) Amplifier
- Switched-Capacitor Amplifiers, Integrators, and Filters
- Voltage and Current DAC’s
- Current Sources
- Voltage/Current/Bandgap References

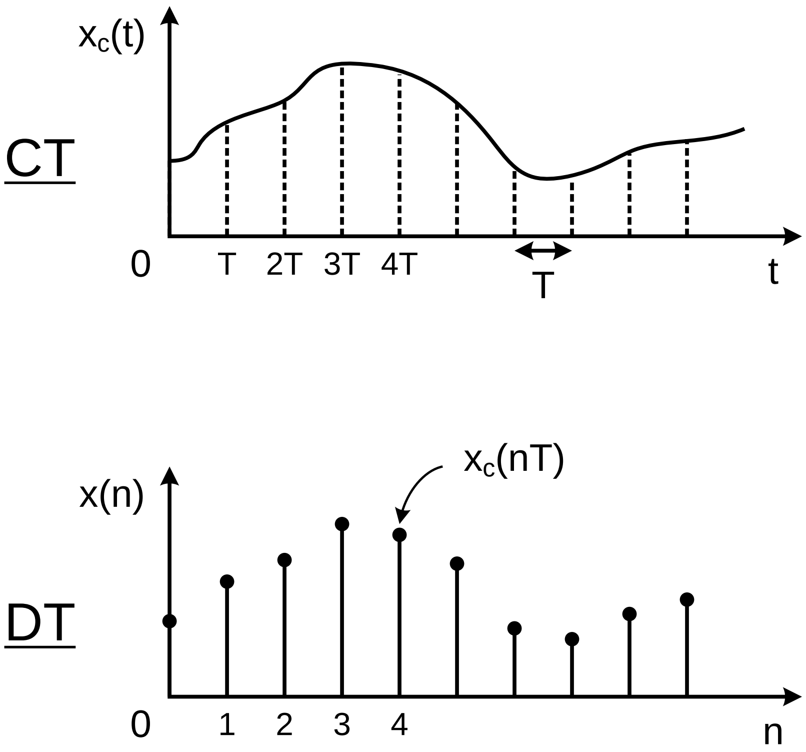
# Sampling, Quantization, Spectrum Analysis

# Block Diagram of a DSP System



# Sampling

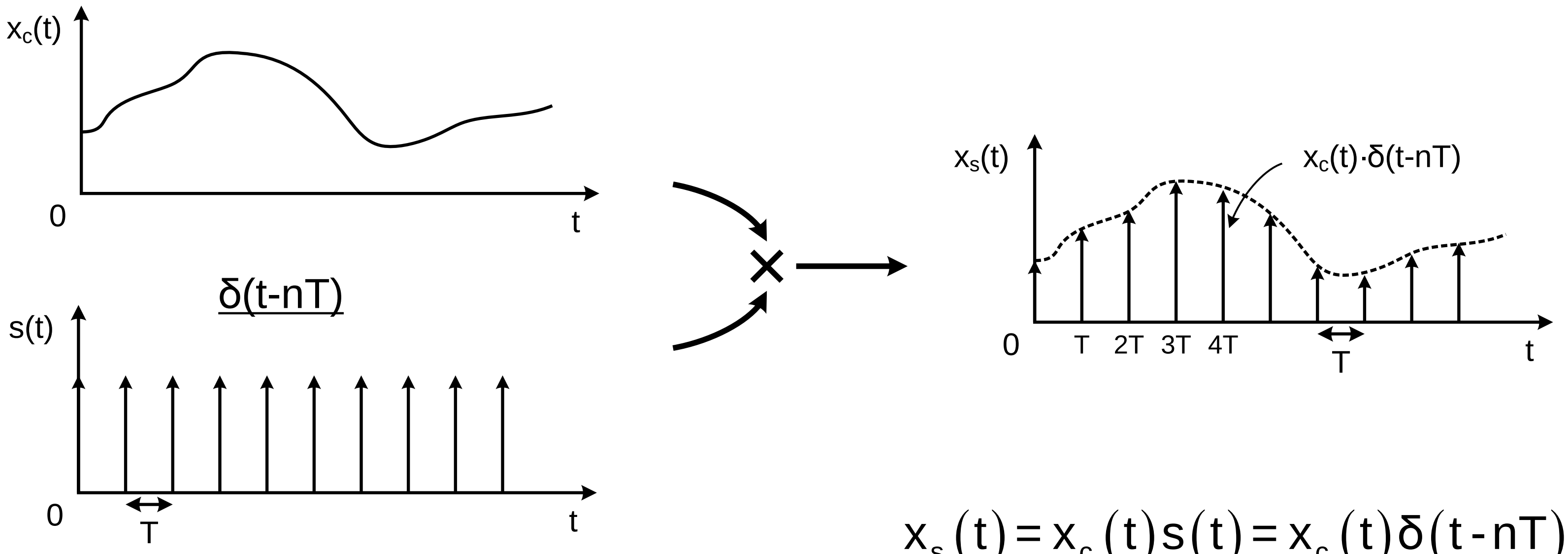



$$x(n) = x_c(t = nT)$$

$$x_c(t) \overset{FT}{\Leftrightarrow} X_c(j\Omega)$$

# Sampling




$$x_s(t) = x_c(t)s(t) = x_c(t)\delta(t-nT)$$

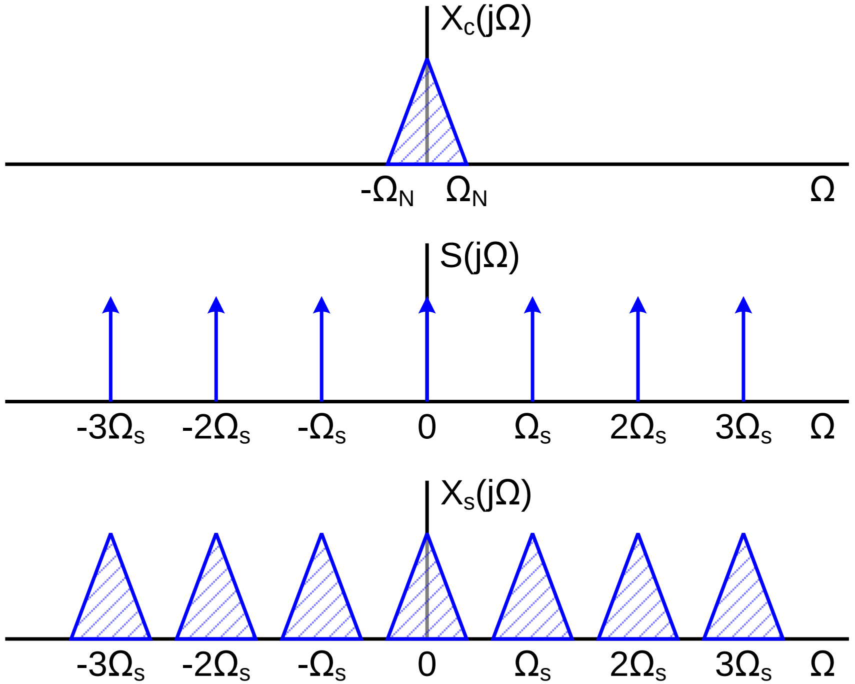
$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) \otimes S(j\Omega)$$

$$s(t) \overset{FT}{\Leftrightarrow} \frac{2\pi}{T}\sum_k \delta(\Omega - k\Omega_s), \quad \Omega_s = \frac{2\pi}{T}$$

$$X_s(j\Omega) = \frac{1}{T}\sum_k X_c(\Omega - k\Omega_s)$$

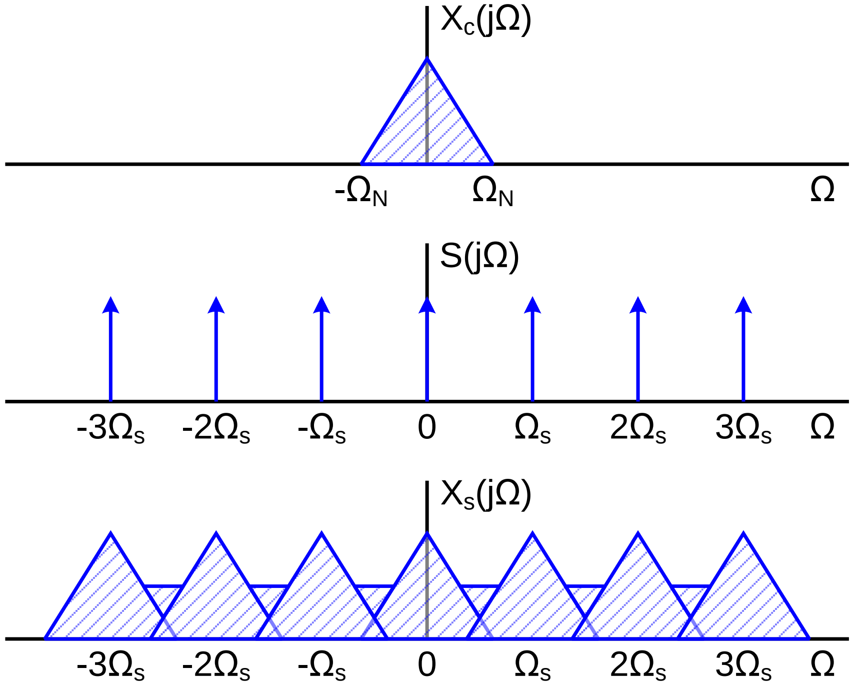
$$\boxed{\begin{aligned} X\left(e^{j\omega}\right) &= \sum_{n=-\infty}^{\infty} x(n) z^{-n} \\ &= X_s(j\Omega)\Big|_{\Omega=\frac{\omega}{T}} = \frac{1}{T}\sum_k X_c(\Omega - k\Omega_s)\Big|_{\Omega=\frac{\omega}{T}} \end{aligned}}$$

# Spectrum of Sampled Signal ($\Omega_s > 2\Omega_N$)



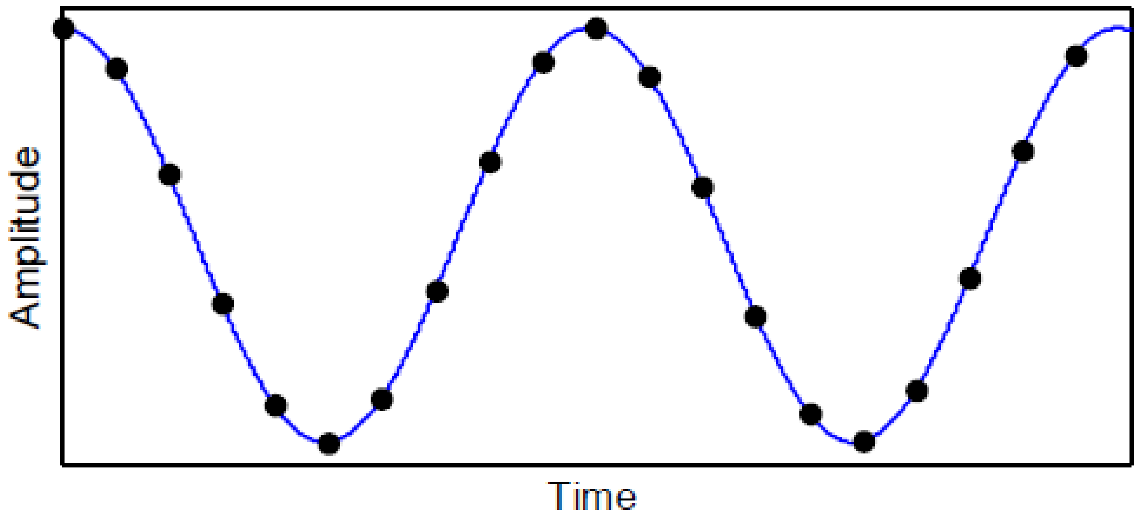

The spectrum of the sampled signal is periodic in $\Omega_s = 2\pi/T$.

# Spectrum of Sampled Signal ($\Omega_s < 2\Omega_N$)




- Aliasing (folding) results in irreversible signal distortion.
- Can only be avoided by using sufficiently high sample rate, or band-limit the input signal with a coarse, continuous-time filter –> AAF.

# Aliasing Example




$$f_S = \frac{1}{T_S} = 1000kHz$$

$$f_{sig} = 101kHz$$

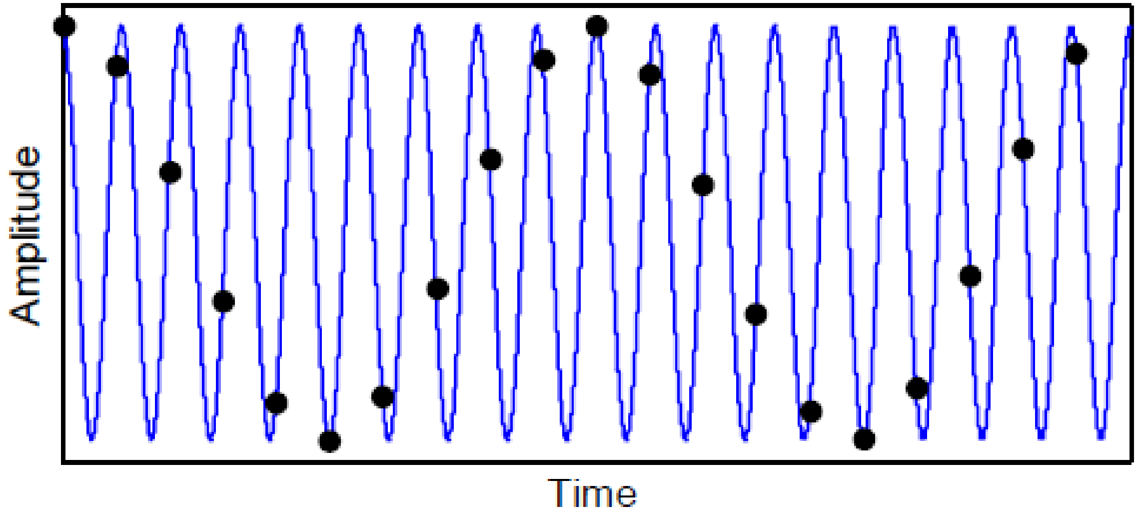
$$v_{sig}(t) = cos(2\pi \cdot f_{in} \cdot t)$$

$$\boxed{t \to n \cdot T_S = \frac{n}{f_S}}$$

$$v_{sig}(n) = cos\left(2\pi \cdot \frac{f_{in}}{f_S} \cdot n\right)$$

$$= cos\left(2\pi \cdot \frac{101}{1000} \cdot n\right)$$

# Aliasing Example

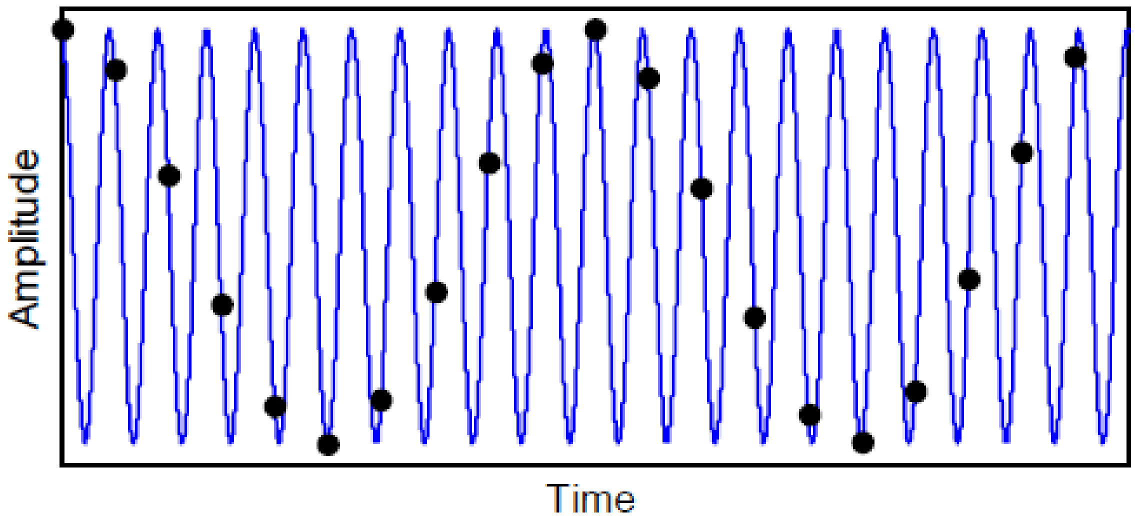



$$f_s = \frac{1}{T_s} = 1000kHz$$

$$f_{sig} = 899kHz$$

$$v_{sig}(n) = cos\left(2\pi \cdot \frac{899}{1000} \cdot n\right) = cos\left(2\pi \cdot \left[\frac{899}{1000} - 1\right] \cdot n\right) = cos\left(2\pi \cdot \frac{101}{1000} \cdot n\right)$$

# Aliasing Example

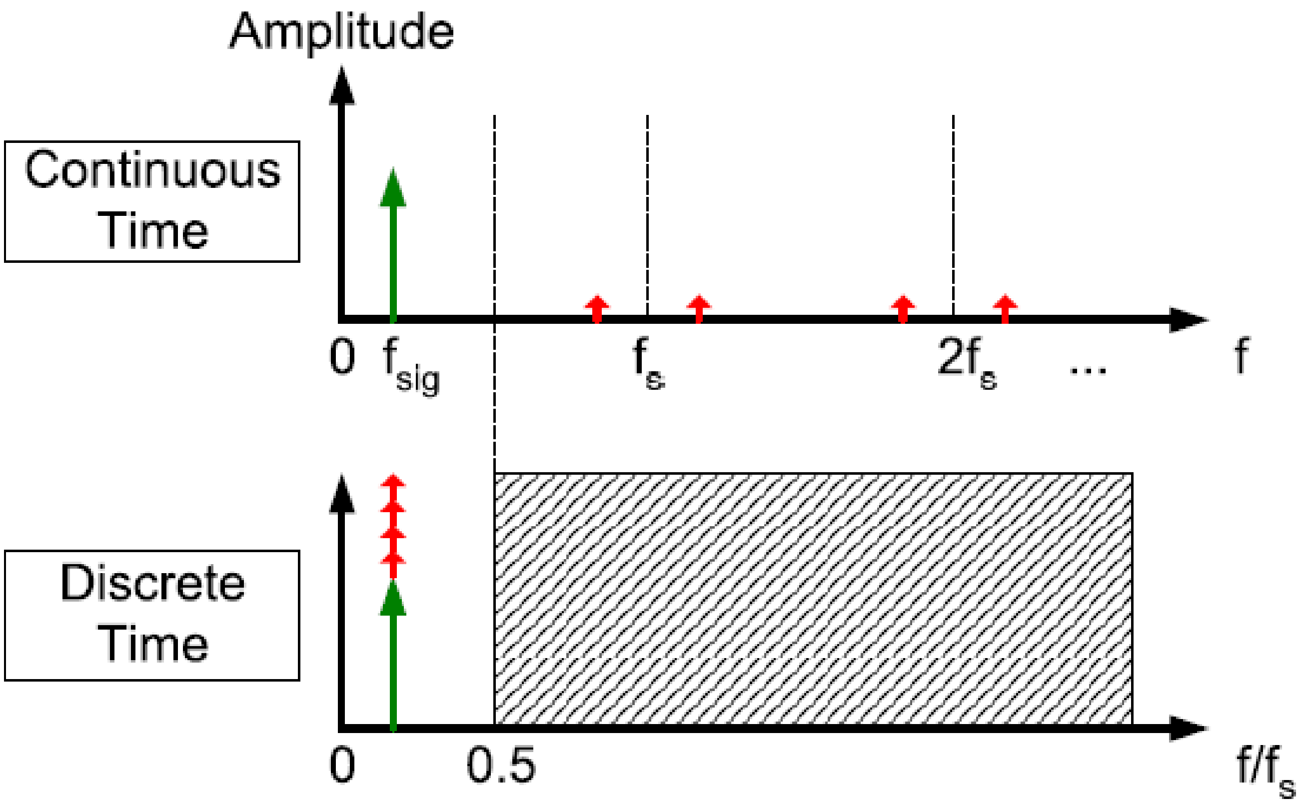



$$f_s = \frac{1}{T_s} = 1000kHz$$

$$f_{sig} = 1101kHz$$

$$v_{sig}(n) = cos\left(2\pi \cdot \frac{1101}{1000} \cdot n\right) = cos\left(2\pi \cdot \left[\frac{1101}{1000} - 1\right] \cdot n\right) = cos\left(2\pi \cdot \frac{101}{1000} \cdot n\right)$$

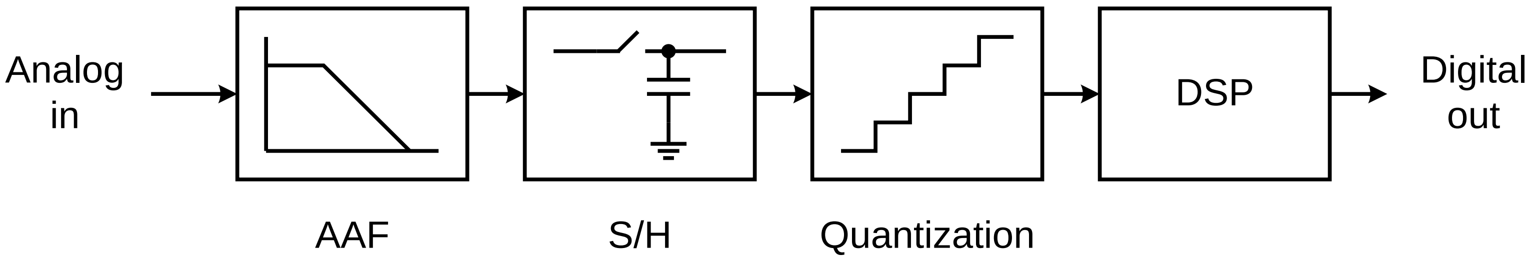
# Aliasing Consequence



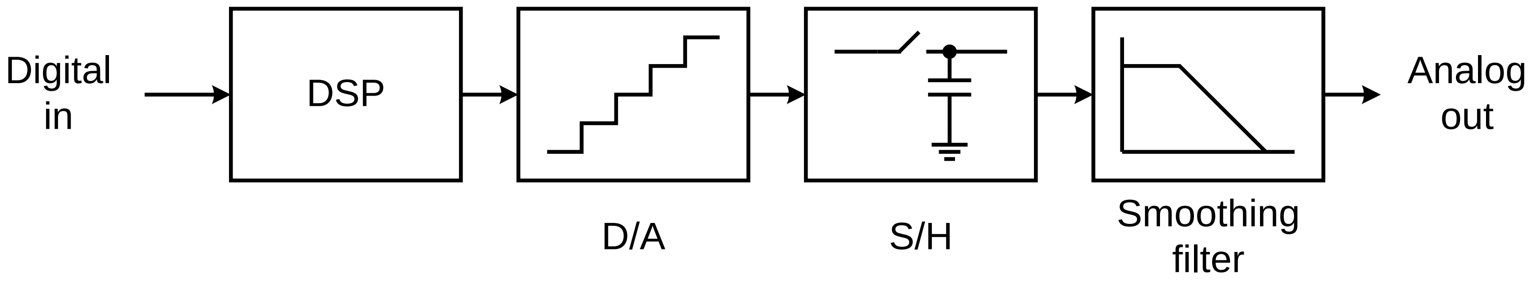

- The frequencies $f_{sig}$ and $N \cdot f_s \pm f_{sig}$ (N integer), are indistinguishable in the discrete time domain

# Recap A/D and D/A Signal Path

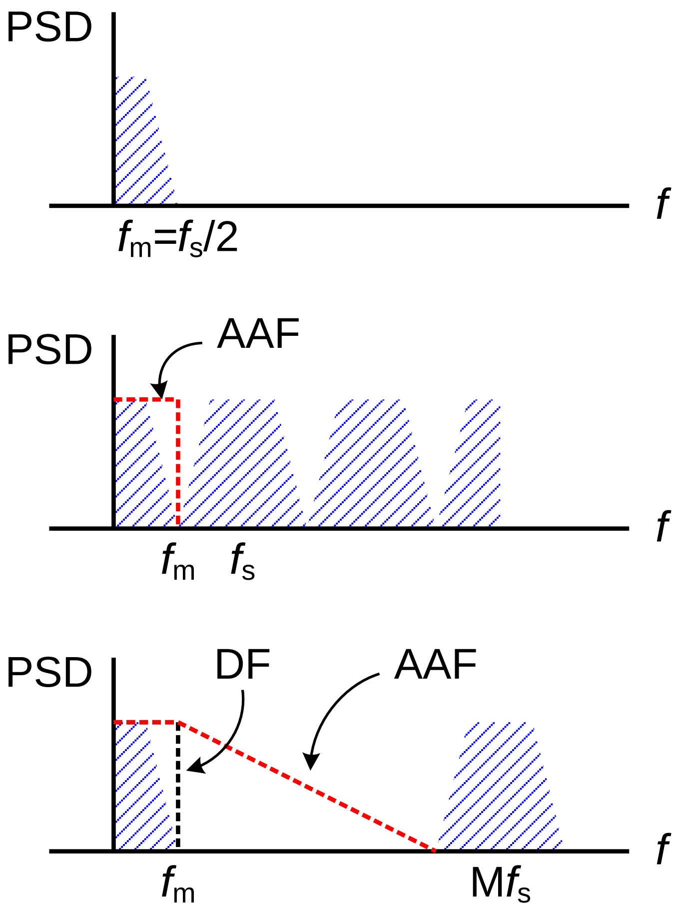
## A/D Conversion




## D/A Conversion



# Anti-Aliasing Filter (AAF)




- Input signal must be band-limited prior to sampling
- Nyquist sampling places stringent requirement on the roll-off characteristic of AAF
- Often some oversampling is employed to relax the AAF design (better phase response too)
- Decimation filter (digital) can be linear-phase

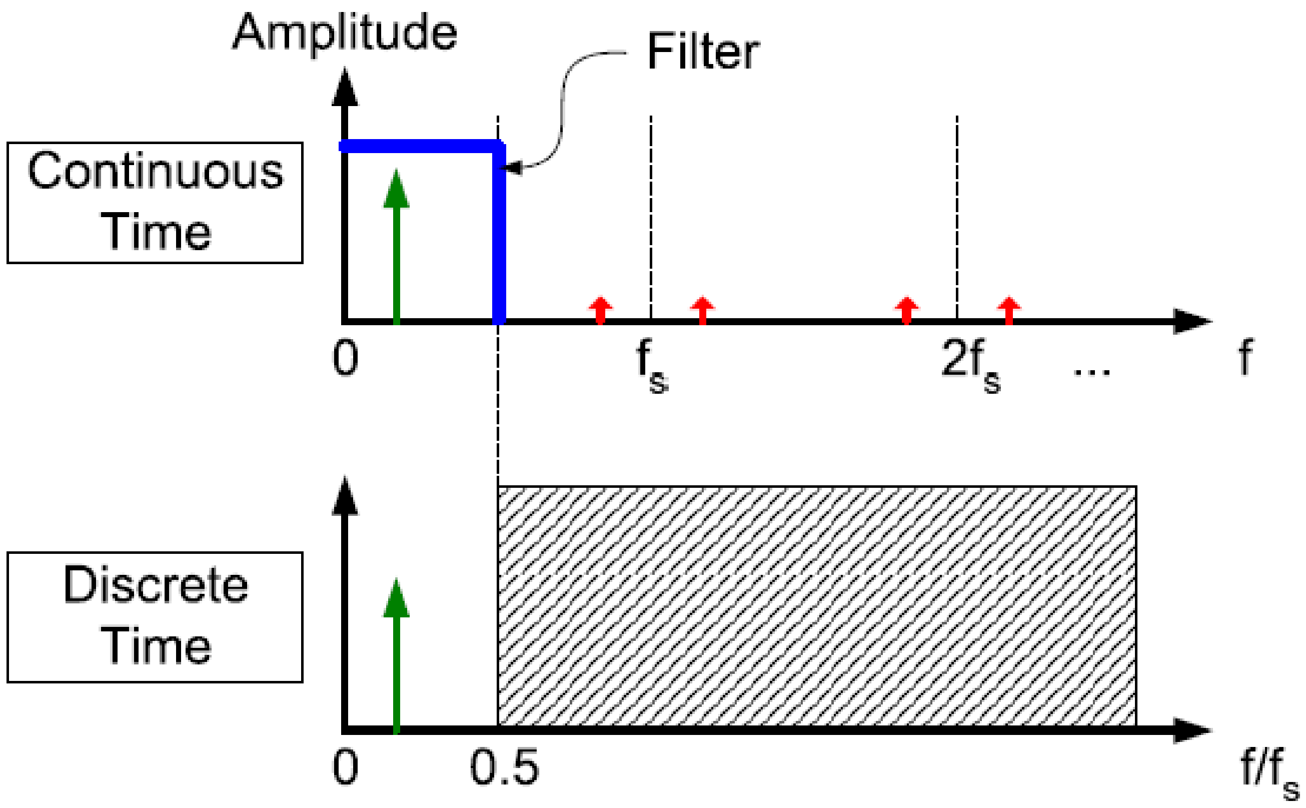
# Sampling Theorem

- In order to prevent aliasing, we need

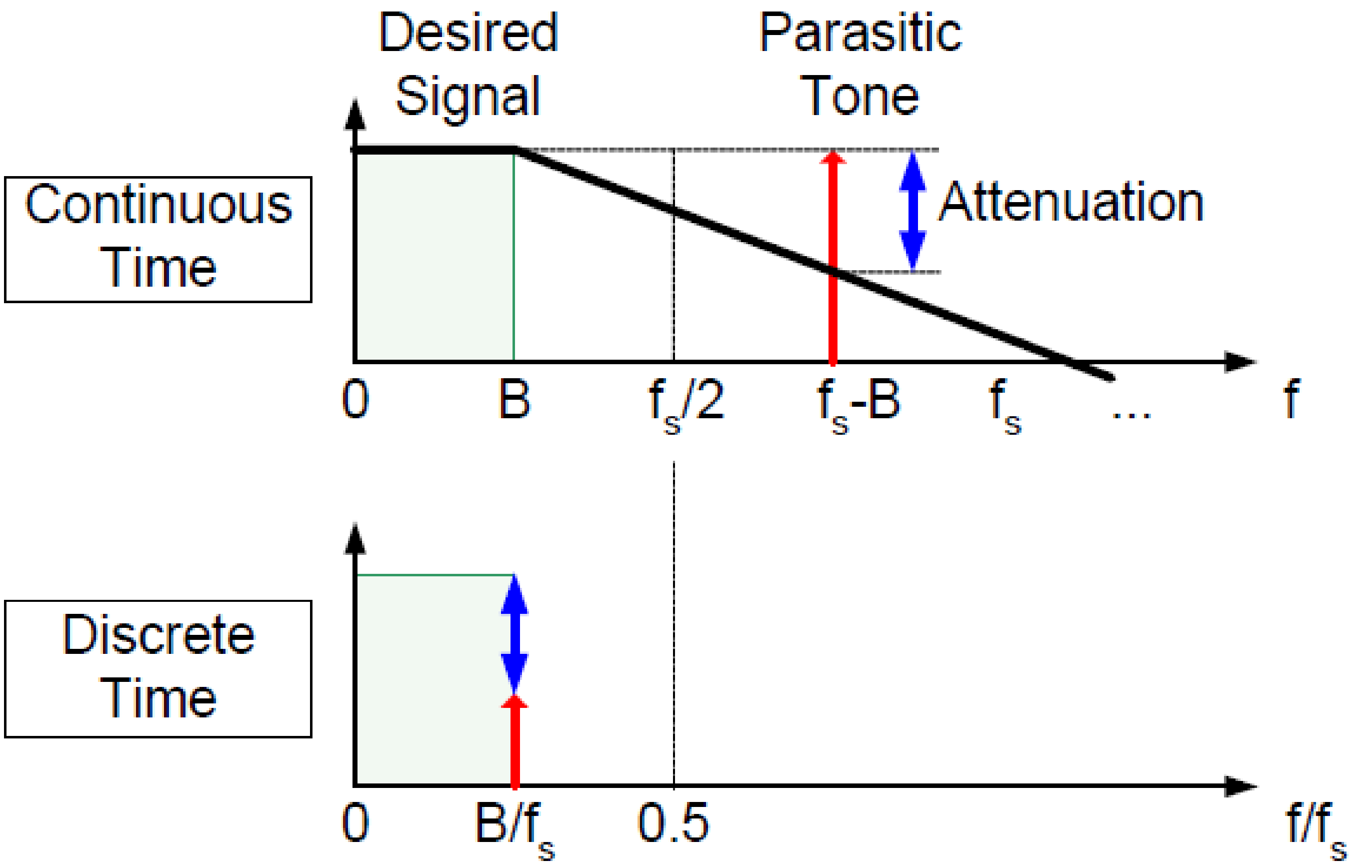
$$f_{sig,max} < \frac{f_S}{2}$$

- The sampling rate $f_s = 2 \cdot f_{sig,max}$ is called the Nyquist rate
- To avoid aliasing
  - Sample fast enough to cover all spectral components, including "parasitic" ones outside band of interest
  - Limit $f_{sig,max}$ through filtering

# Brick Wall Anti-Alias Filter

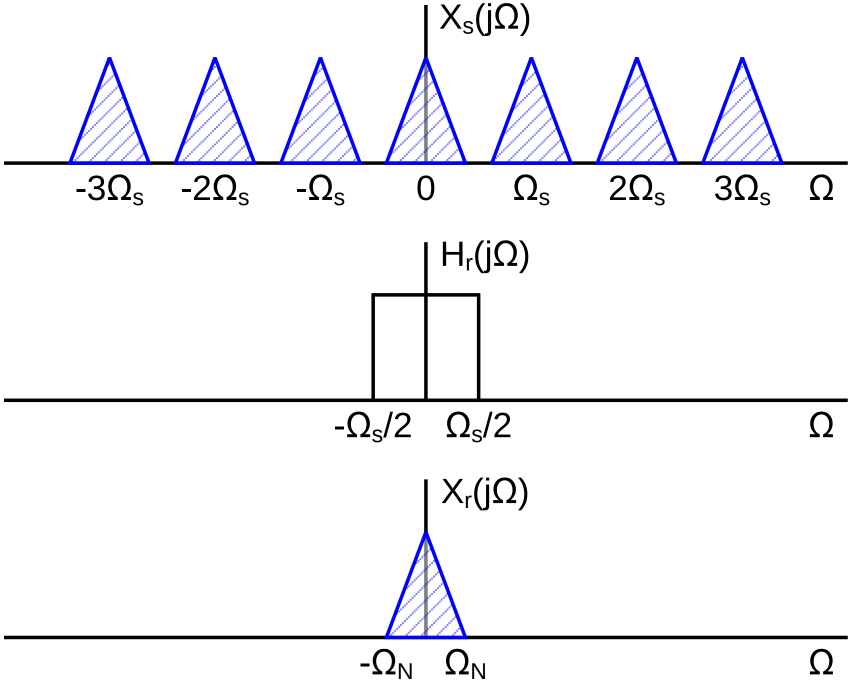


# Practical Anti-Alias Filter




- Need to sample faster than Nyquist rate to get good attenuation
  - "Oversampling"

# Reconstruction Filter (Nyquist)



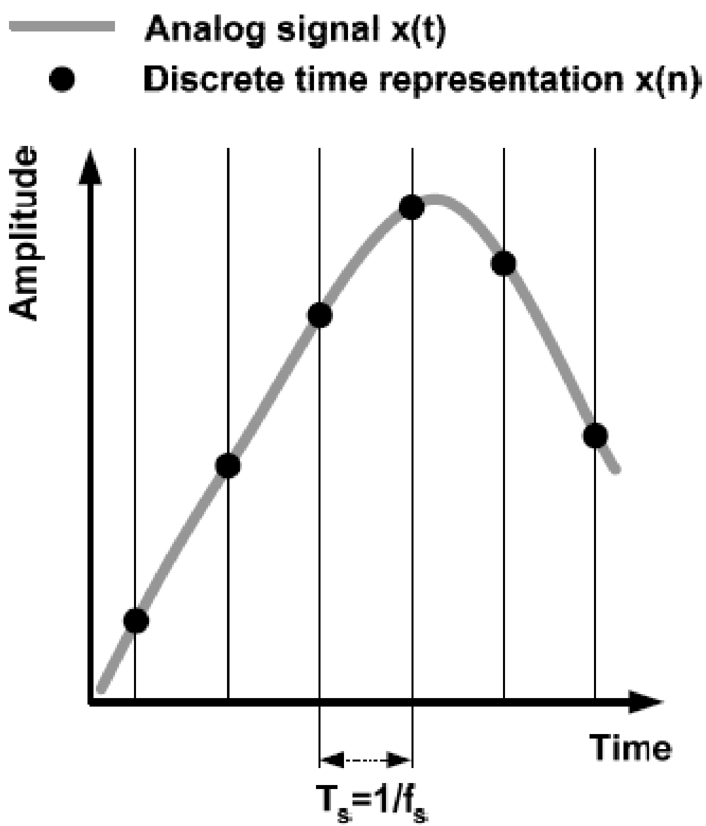

$$h_r(t) = \frac{\sin(\pi t / T)}{\pi t / T}$$

$$x_r(t) = \sum_{n=-\infty}^{\infty} x(n) h_r(t - nT)$$

$$= \sum_{n=-\infty}^{\infty} x(n) \frac{\sin\left[\pi(t - nT)/T\right]}{\pi(t - nT)/T}$$

Reconstruction filter = “smoothing” filter = “interpolation” filter

# Signal Reconstruction



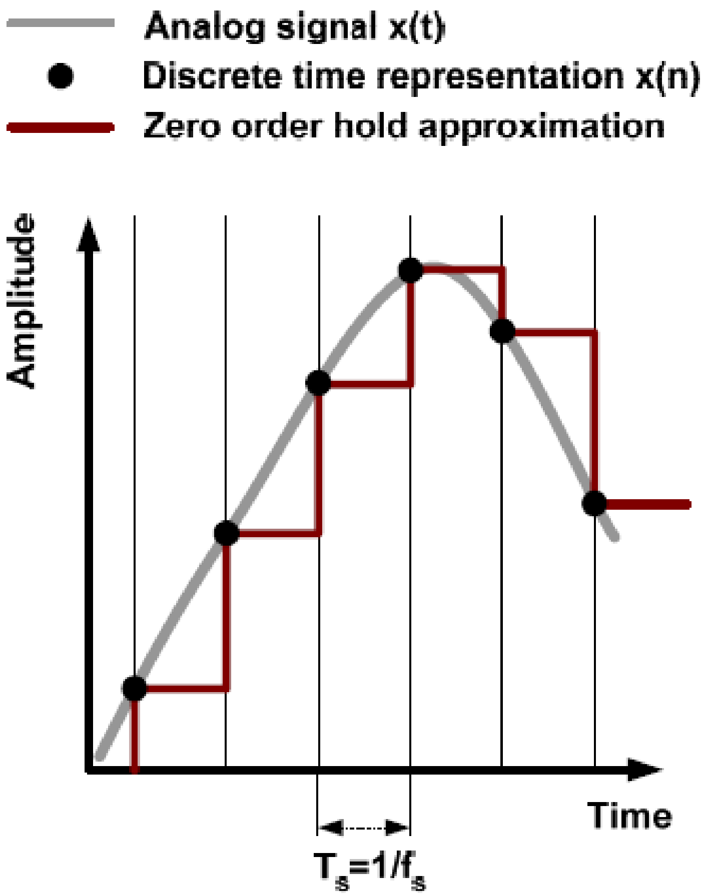

- As long as we sample fast enough, x(n) contains all information about x(t)
  - $f_s > 2 \cdot f_{sig,max}$
- How to reconstruct x(t) from x(n)?
- Ideal interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} x(n) \cdot g(t - nT_S)$$

$$g(t) = \frac{\sin(\pi f_S t)}{\pi f_S t}$$

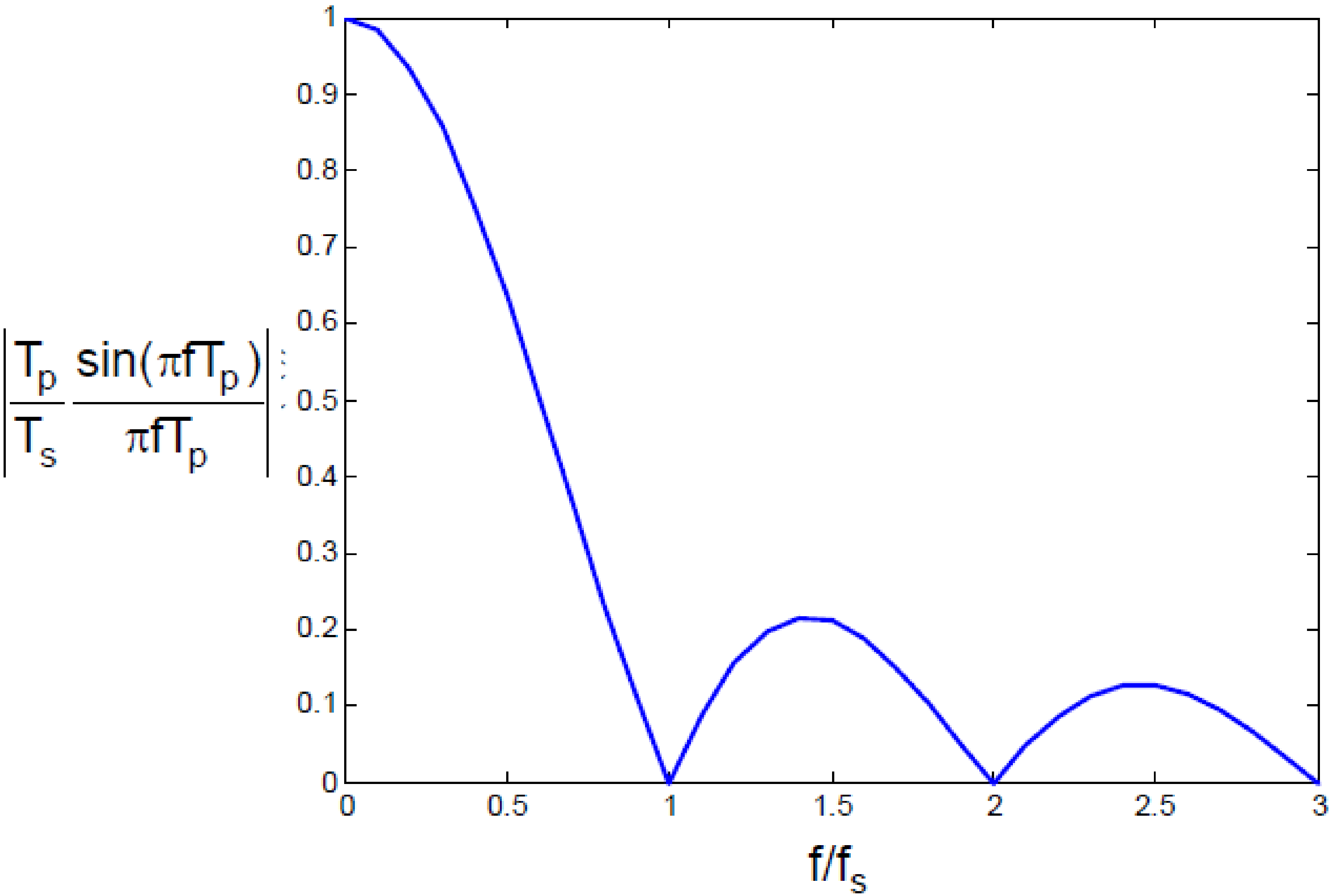
- Very hard to build an analog circuit that does this…

# Zero-Order Hold

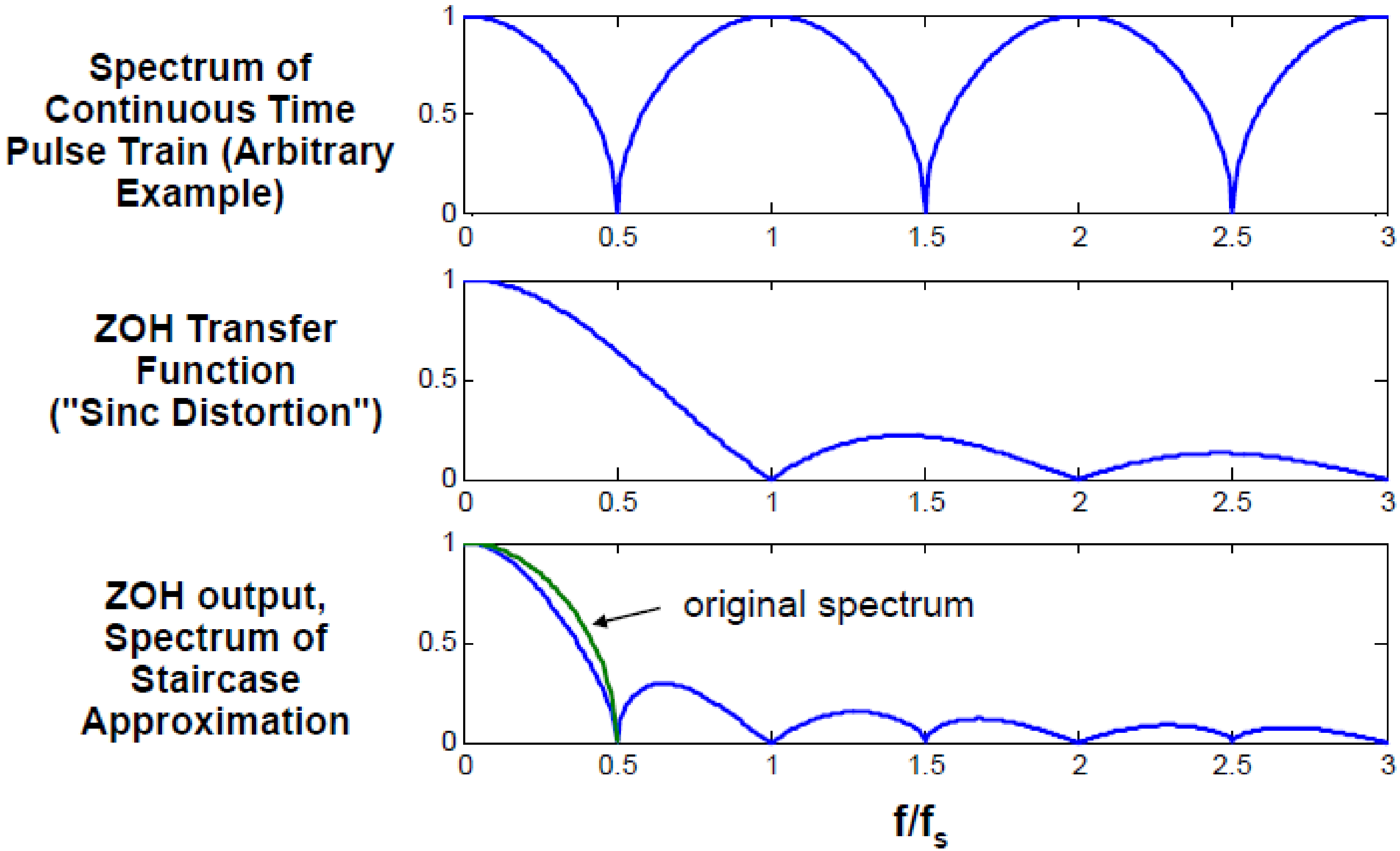



- The most practical way of reconstructing the continuous time signal is to simply "hold" the discrete time values
  - Either for full period $T_s$ or a fraction thereof
  - Other schemes exist, e.g. "partial-order hold"
    - See [Jha, TCAS II, 11/2008]
- What does this do to the signal spectrum?

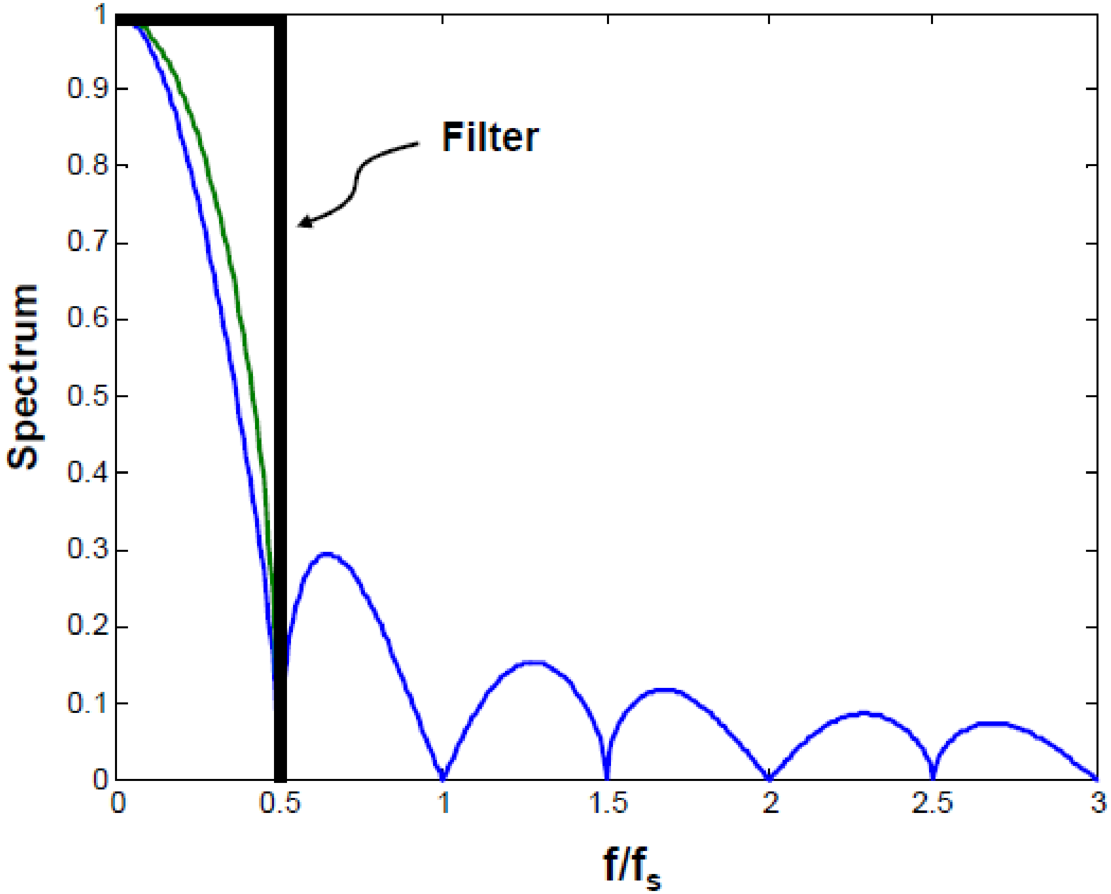
# Envelope with Hold Pulse Tp=Ts



# Signal Spectrum Example



# Reconstruction Filter




- Also called smoothing filter
- Same situation as with anti-alias filter
  - A brick wall filter would be nice
  - Oversampling helps reduce filter order

# Summary of Signal Reconstruction

- Must obey sampling theorem $f_s > 2 \cdot f_{sig,max}$,
  - Usually dictates anti-aliasing filter
- If sampling theorem is met, continuous time signal can be recovered from discrete time sequence without loss of information
- A zero order hold in conjunction with a smoothing filter is the most common way to reconstruct
  - May need to add pre- or post-emphasis to cancel droop due to sinc envelope
- Oversampling helps reduce order of anti-aliasing and reconstruction filters