EE 435
Lecture 43

Phased Locked Loops and VCOs Over Sampled Data Converters

Final Exam:

Scheduled on Final Exam Schedule:

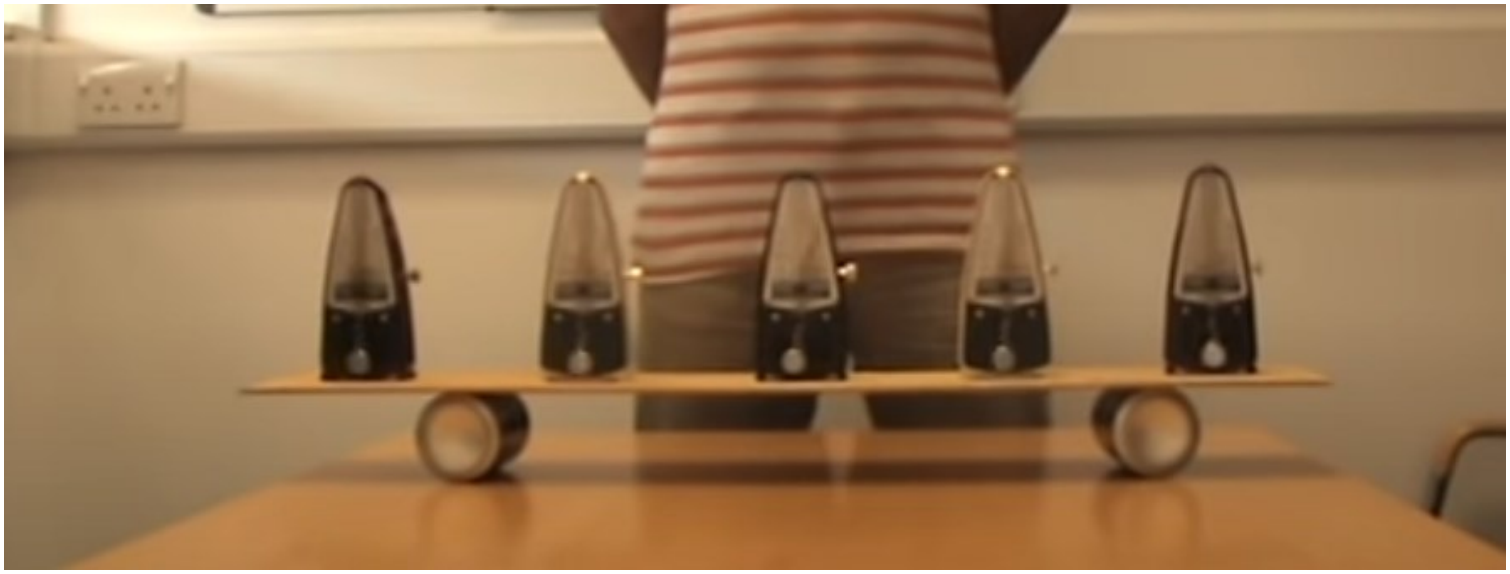
Wednesday May 10 7:30 – 9:30 a.m.

Revised Final Exam:

- Take-home format – open book and open notes
- Will be posted on course WEB site by Tues morning May 9
- Due at 11:59 p.m. on Wednesday May 10: (Hard copy under office door of instructor)

If anyone would prefer an in-class closed-notes exam during the 2-hour scheduled period, arrangements will be made to accommodate those preferences. Such preferences must be conveyed to the instructor by class on Friday May 5

Injection Locking



https://www.youtube.com/watch?v=W1TMZASCR-I

Phase Locked Loops

Special case of injection-locked systems

In 1600's pendulums and organ pipes were occasionally injection locked

In 1919 electronic oscillators were observed that lock or sychronize

For the subsequent several decades, a form of PLLs was used to build radio and television receivers

In 1969 Signetics introduced the first integrated PLLs (NE 565 and NE 567)

In 1970's RCA introduced the 4046 PLL

ON Semiconductor HC4046A

Phase Locked Loops

PLLs widely used throughout industry today

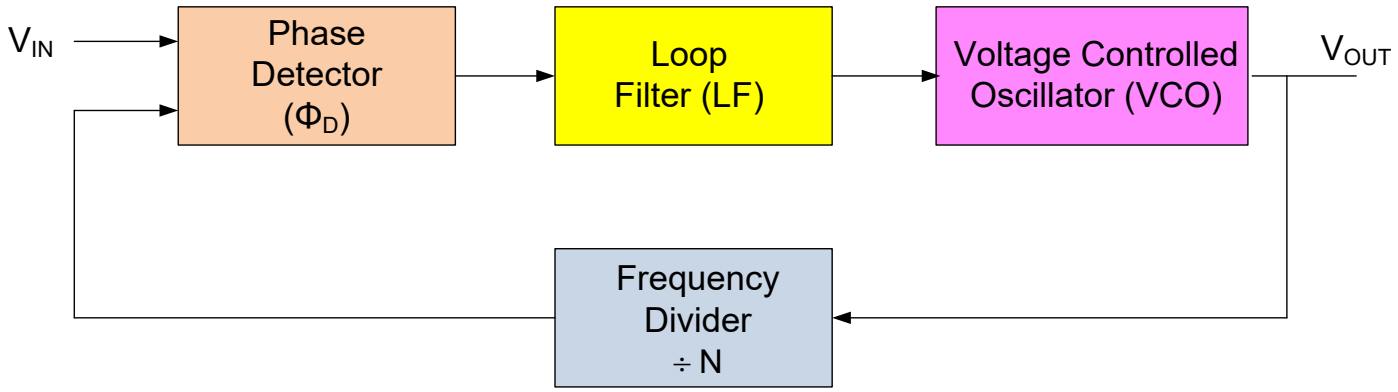
Easy to establish basic operation of a PLL

Highly nonlinear systems with seemingly simple operation but high-end designs can become quite complicated and mathematical rigor is often cumbersome and only approximate

Performance potential of most wireless communication systems and serial data transmission networks strongly dependent upon the performance characteristics of multiple embedded PLLs

Concepts of DLL and PLL are closely related

Basic PLL Architecture

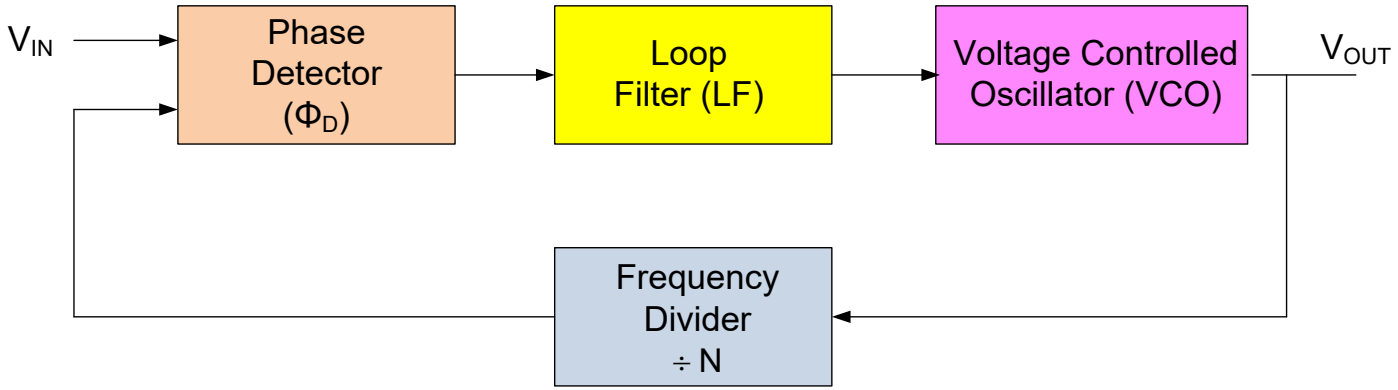


Applications include:

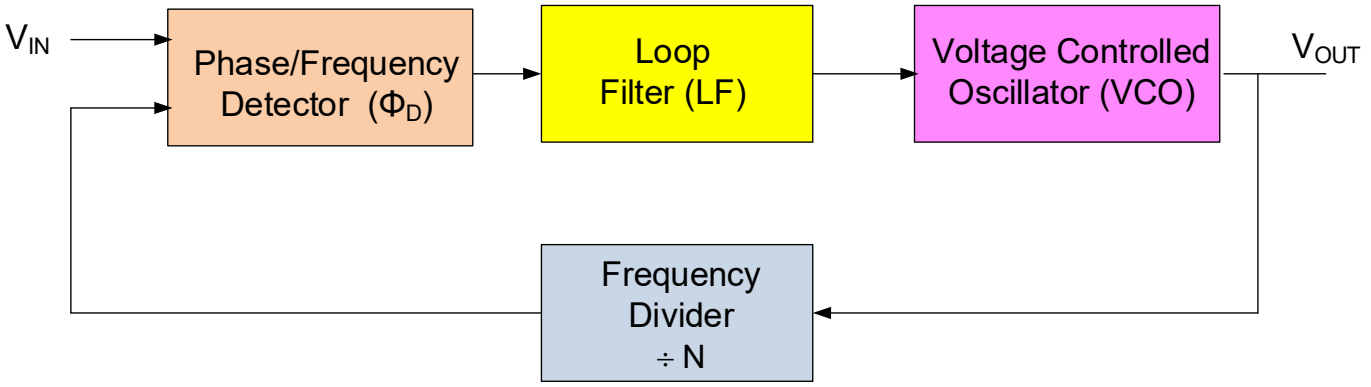
- Frequency Demodulation
- Frequency Synthesis
- Clock Synchronization
- Noise filtering (extreme)
- Tracking and calibrated filters
- …..

- **One of the most widely used analog blocks**
- **Many SoC systems include multiple PLLs**
- **Closely related to Delay Locked Loop (DLL)**

Basic PLL Architecture

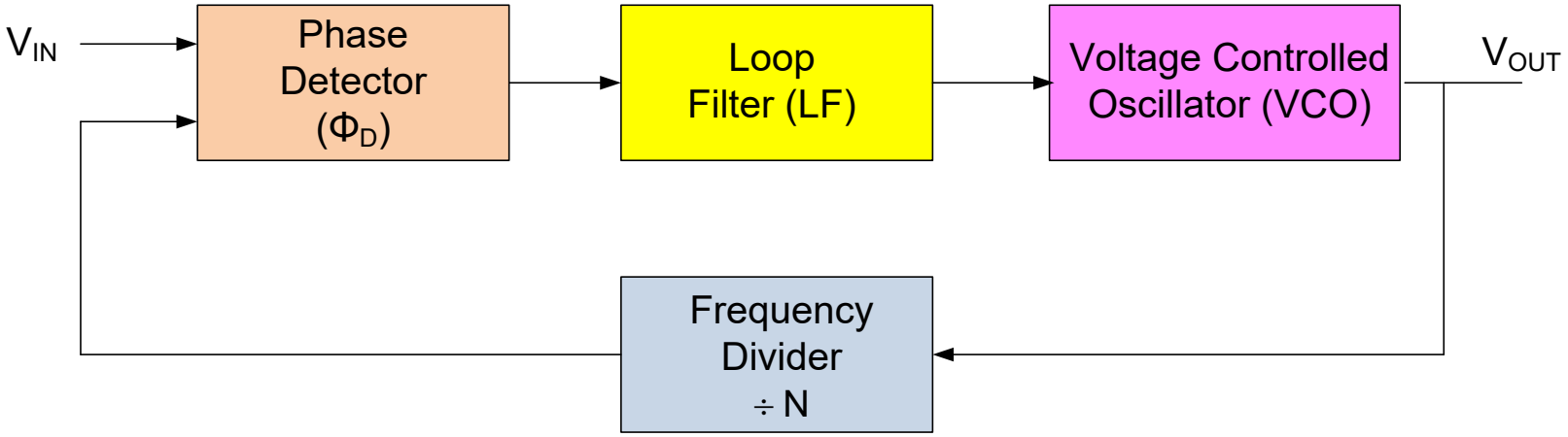


Most basic structure actually uses phase/frequency detector but still termed PLL[1]



[1]Ian Collins, "Phase-Locked Loop (PLL) Fundamentals", Analog Dialogue, July 2018.

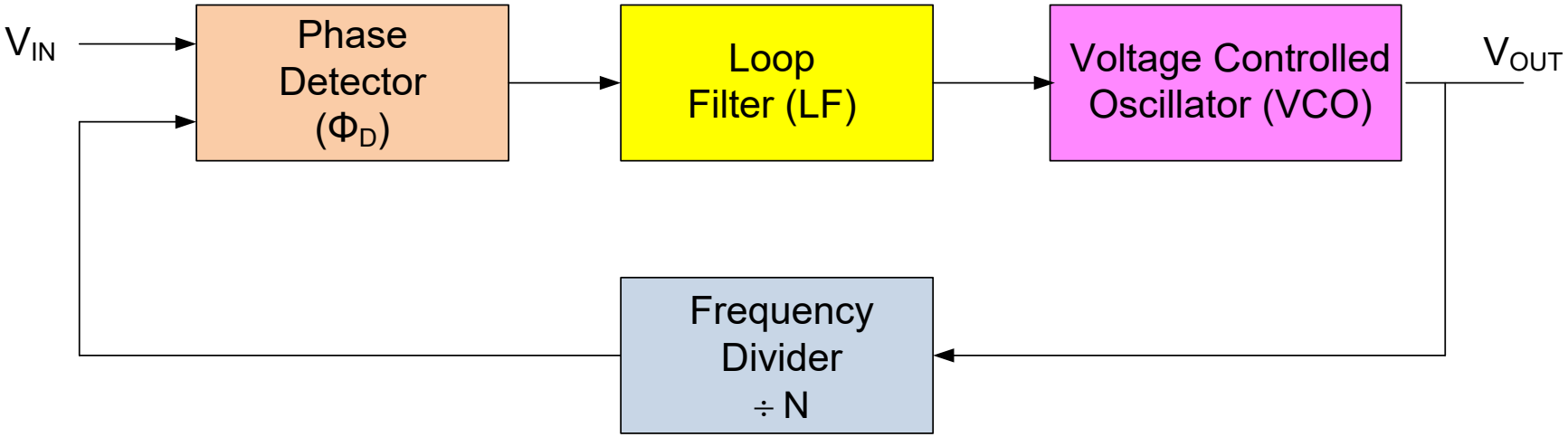
Basis PLL Architecture



Applications by subcategory:

- Clock and Data Recovery
- Recovering signals when SNR <<<1
- Timing generators in digital systems

Desired Operation of PLL



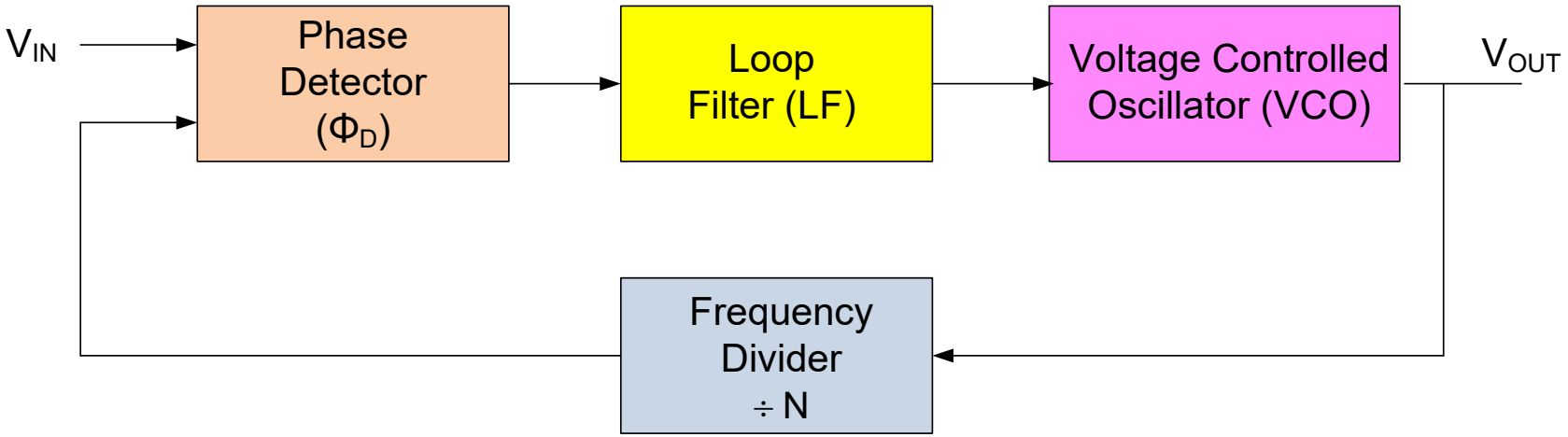
$$V_{IN}=V_M\sin(\omega_{IN}t+\phi_{IN})$$

Desired output when locked:

$$V_{OUT}=V_X\sin(N\omega_{IN}t+\phi_{OUT})$$

- Relationship between V_M and V_X is of little concern
- Frequency relationship is critical
- ϕ_{OUT} is often critical too
- Waveshape of V_{IN} and V_{OUT} is often of little concern
  May be highly distorted or even square waves

Desired Operation of PLL



$$V_{IN}=V_M\sin(\omega_{IN}t+\phi_{IN})$$

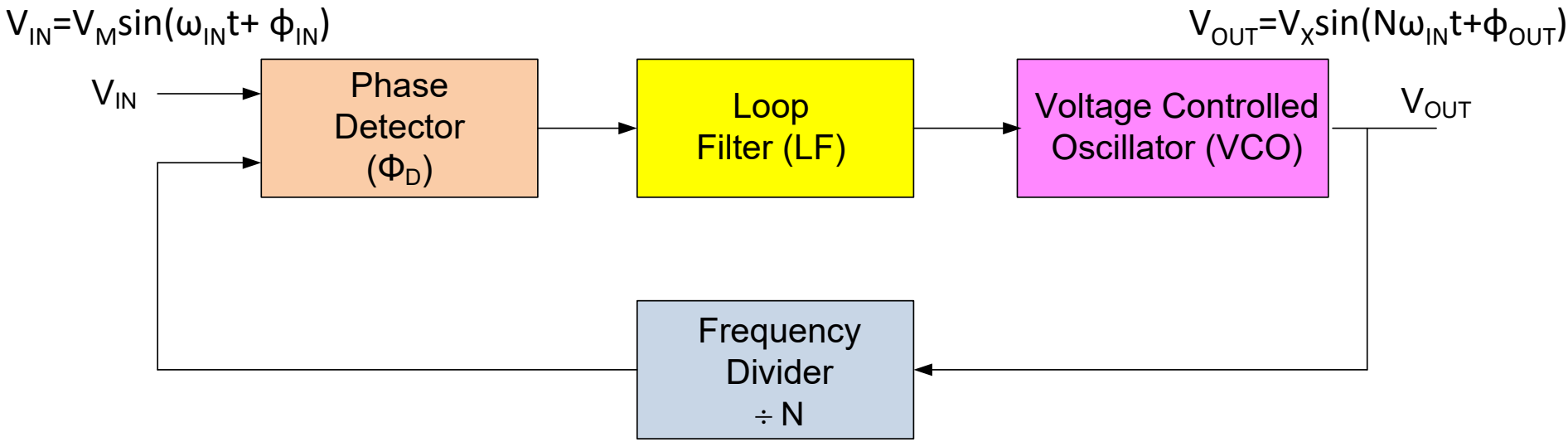
Desired output when locked:

$$V_{OUT}=V_X\sin(N\omega_{IN}t+\phi_{OUT})$$

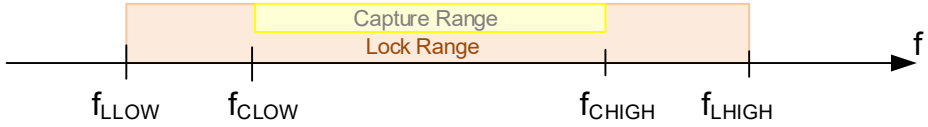
Some Terminology of PLLs

- Locked / Unlocked — Locked when V_{OUT} assumes desired value
- Lock Range — $f_{LLOW} < f_{IN} < f_{LHIGH}$ If locked, will remain locked for f_{IN} in lock range
- Capture Range — $f_{CLOW} < f_{IN} < f_{CHIGH}$ If unlocked, will lock for f_{IN} in capture range
- Free-running frequency — frequency of VCO when not locked
- Harmonic/Subharmonic Lock

Desired Operation of PLL



Capture range always less than lock range $f_{LLOW} < f_{CLOW} < f_{CHIGH} < f_{LHIGH}$

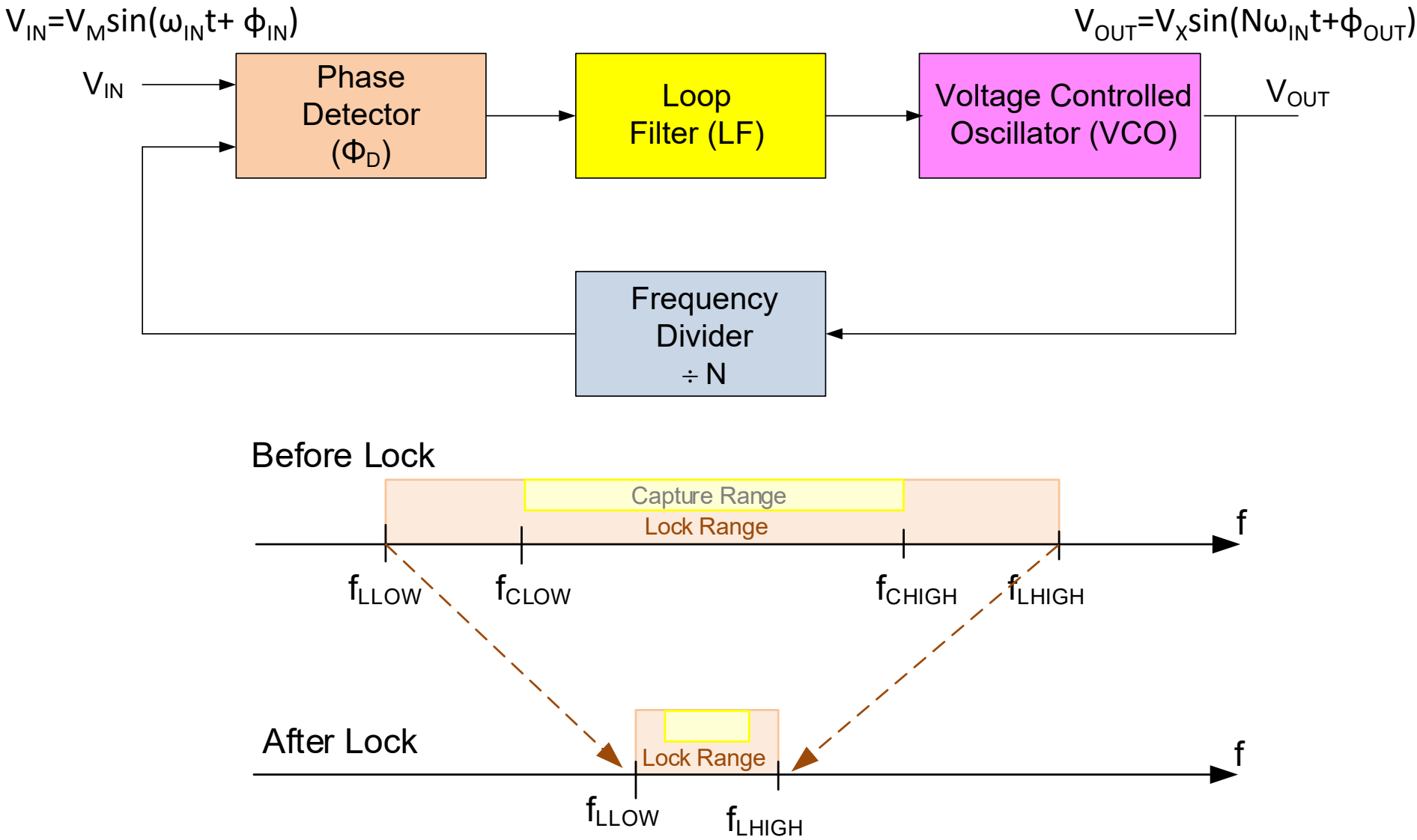


Loop filter controls capture and lock range

Jitter in VCO output strongly dependent upon lock range
(large lock range results in high jitter, low lock range in low jitter)

Loop filter is often dynamic with wide bandwidth prior to lock and narrow BW after lock

Desired Operation of PLL



Loop filter is often dynamic with wide bandwidth prior to lock and narrow BW after lock

Conceptual Operation of PLL

Consider a signal defined for $-\infty < t < \infty$ expressed as

$$V(t) = V_M \sin(\phi(t))$$

If the signal is sinusoidal with frequency ω, the argument ϕ can be expressed as

$$\phi(t)=\omega t+\theta$$

where ϕ is defined to be the phase of the signal and θ is the phase offset on the time axis from the time reference t=0

Taking the time derivative of ϕ(t) , we obtain

$$\frac{d\phi}{dt} = \omega$$

Taking the Laplace Transform, we have

$$\phi_S = \frac{\omega}{s}$$

Is the second statement “If the signal is sinusoidal ….” redundant ?

Is the second statement "If the signal is sinusoidal ...." redundant ?

Consider a signal defined for $-\infty < t < \infty$ expressed as

$$V(t) = V_M \sin(\phi)$$

If the signal is sinusoidal with frequency ω, the argument ϕ can be expressed as

$$\phi = \omega t + \theta$$

Consider any signal f(t) defined for all time (not necessarily periodic but could be)

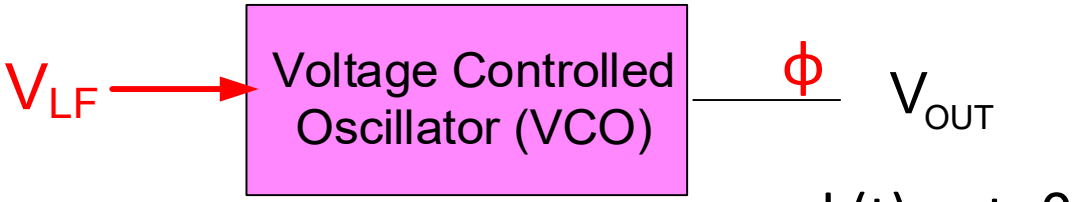
Define ϕ(t) by the expression $\phi(t) = \sin^{-1}\left(\frac{f(t)}{V_M}\right)$

It follows that $V(t) = V_M \sin(\phi(t)) = V_M \sin\left(\sin^{-1}\left(\frac{f(t)}{V_M}\right)\right) = f(t)$

Thus, the first statement gives NO information about the signal V(t)

Consider a VCO where the output is sinusoidal

Assume the output of interest is the phase ϕ



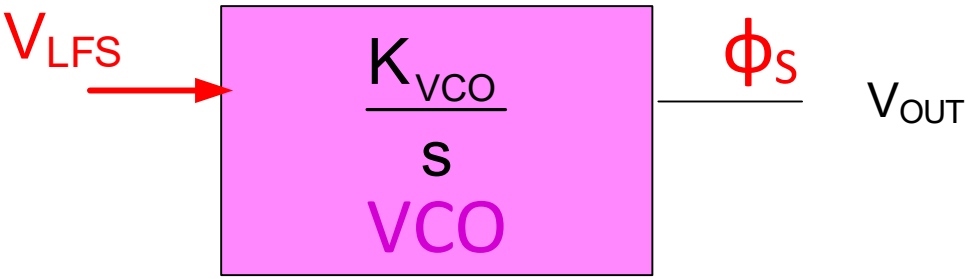
$$\phi(t)=\omega t+\theta$$

$$V_{OUT} = V_m \sin(\omega t + \theta)$$

$$\omega = V_{LF} \bullet K_{VCO}$$

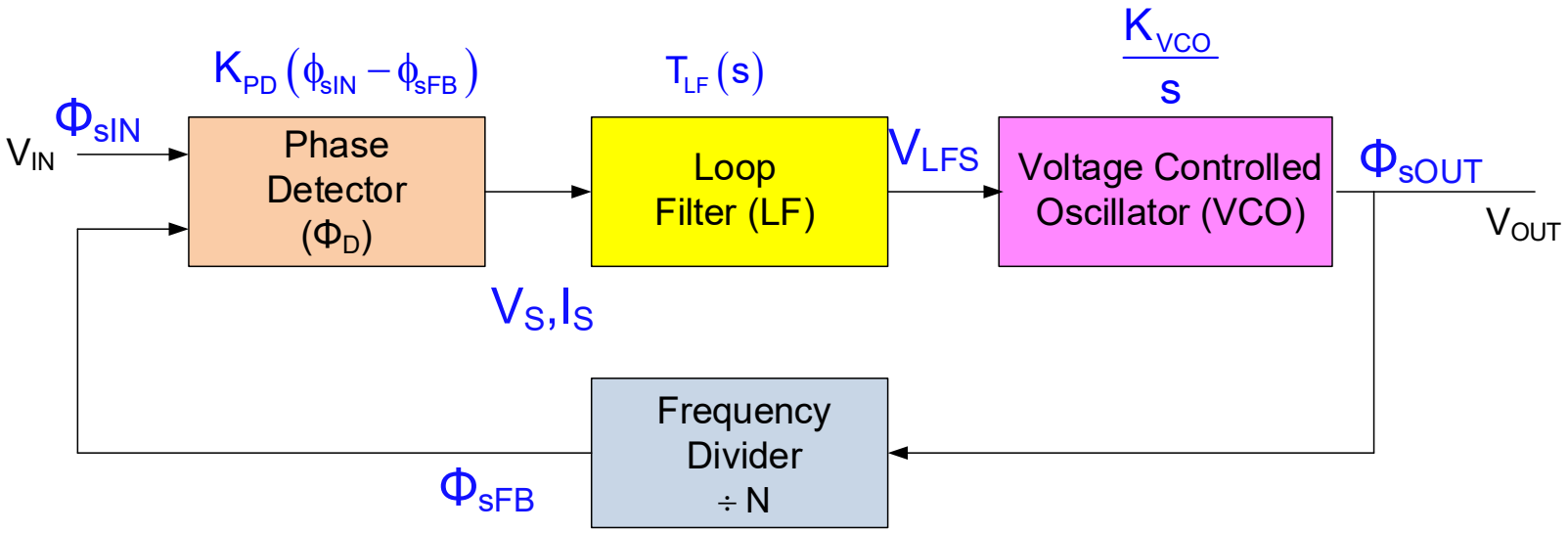
$$\frac{d\phi}{dt} = \omega = V_{LF} K_{VC0}$$

Taking Laplace Transform: $s\phi_S = V_{LFS} K_{VC0}$ ⟹ $\phi_S = V_{LFS} \frac{K_{VC0}}{s}$



Conceptual Operation of PLL

- When locked, PLL can be modeled as a linear system
- Small-signal s-domain analysis when PLL is locked



Note: Dimensions of variables in loop are not the same

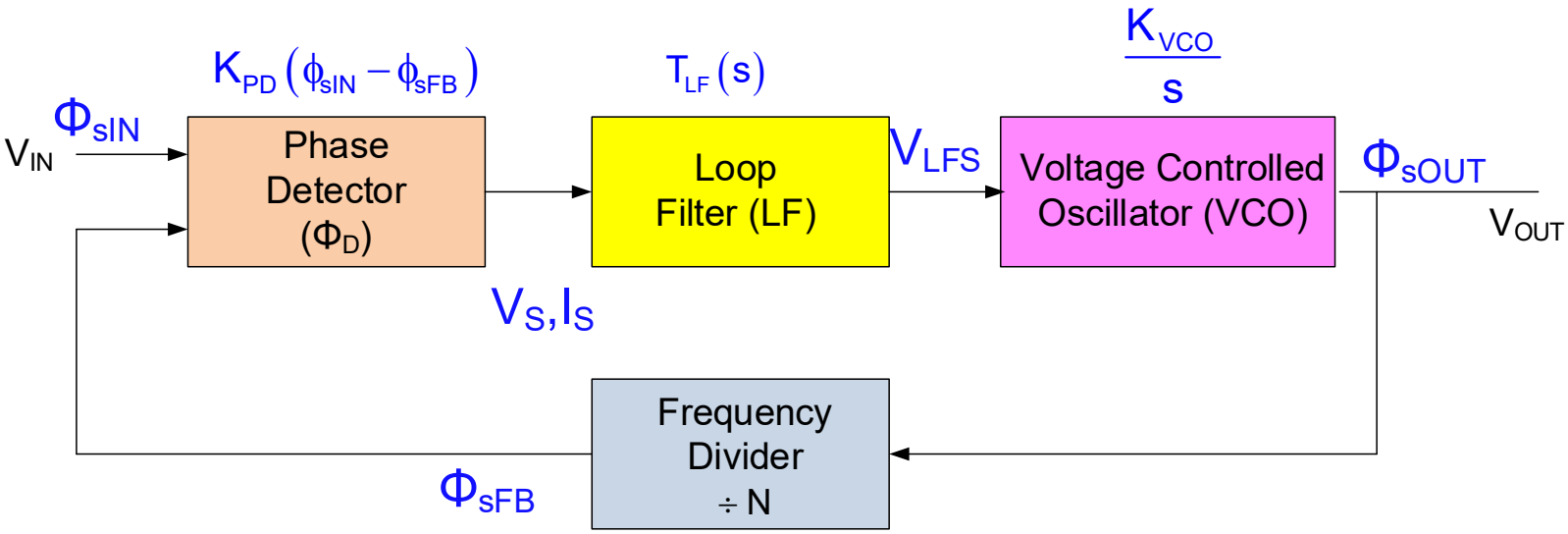
$$V_S = K_{PD}\left(\phi_{sIN}-\phi_{sFB}\right)$$
$$V_{LFS} = T_{LF}(s)V_S$$
$$\phi_{sOUT} = V_{LFS}\frac{K_{VCO}}{s}$$
$$\phi_{sFB} = \frac{\phi_{sOUT}}{N}$$

Solving, we obtain

$$T_{PLL}(s) = \frac{\phi_{sOUT}}{\phi_{sIN}} = \frac{T_{LF}(s)K_{PD}K_{VCO}}{s+T_{LF}(s)\frac{K_{VCO}K_{PD}}{N}}$$

Often the LF is low order

Example: Assume N=1 and $T_{LF}(s) = \dfrac{1}{1+RCs}$

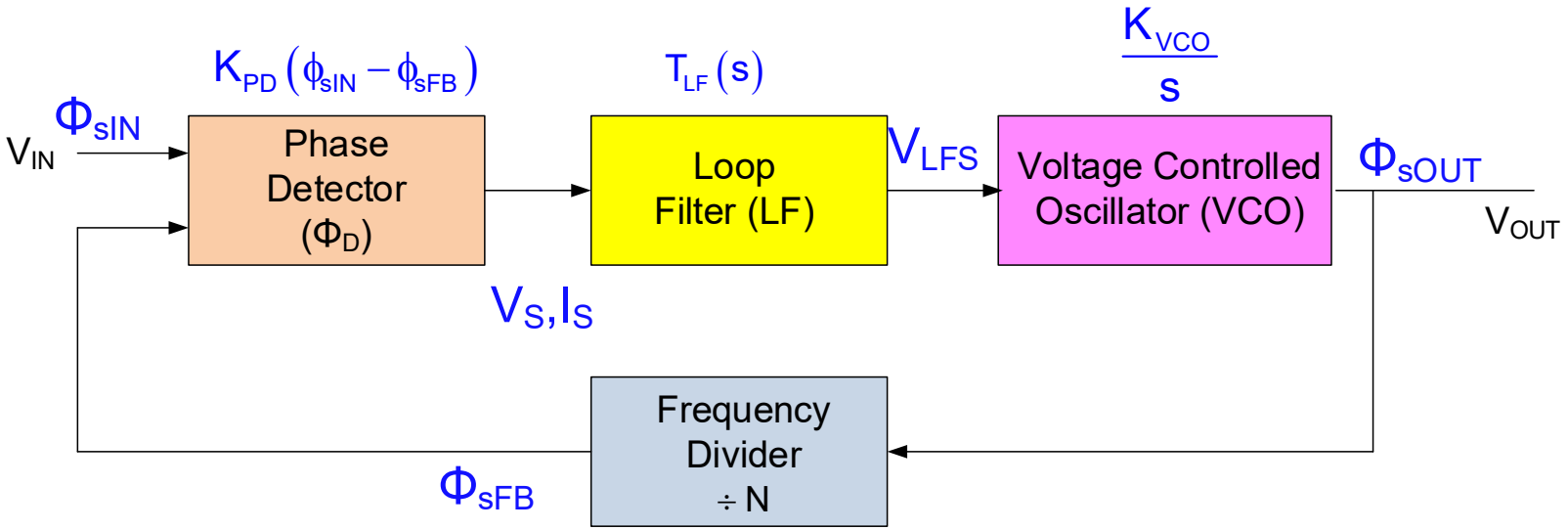


$$T_{PLL}(s) = \frac{T_{LF}(s)K_{PD}K_{VCO}}{s+T_{LF}(s)K_{VCO}K_{PD}} = \frac{K_{PD}K_{VCO}}{s(1+RCs)+K_{VCO}K_{PD}}$$

$$T_{PLL}(s) = \frac{\frac{K_{PD}K_{VCO}}{RC}}{s^2 + s\frac{1}{RC} + \frac{K_{VCO}K_{PD}}{RC}}$$

Often the LF is low order

Example: Assume N=1 and $T_{LF}(s) = \frac{1}{1+RCs}$



$$T_{PLL}(s) = \frac{\frac{K_{PD}K_{VCO}}{RC}}{s^2 + s\frac{1}{RC} + \frac{K_{VCO}K_{PD}}{RC}}$$

$$s^2 + s\frac{1}{RC} + \frac{K_{VCO}K_{PD}}{RC} = s^2 + s\frac{\omega_0}{Q} + \omega_0^2$$

$$\omega_0 = \sqrt{\frac{K_{VCO}K_{PD}}{RC}}$$

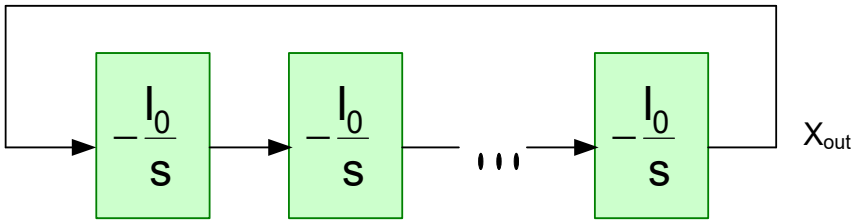
$$Q = \sqrt{RCK_{VCO}K_{PD}}$$

Voltage Controlled Oscillators

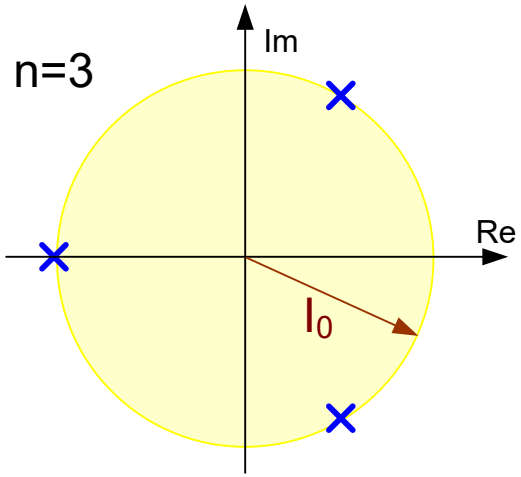
Voltage Controlled Oscillator (VCO)

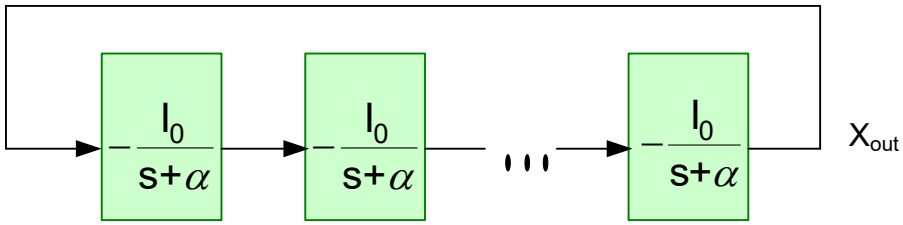
Many different VCOs can be used

Assume I_0 is determined by a controlling voltage

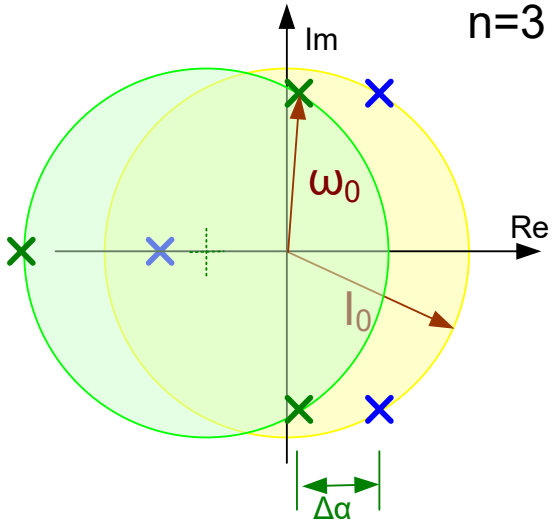


Integrator-Based VCO



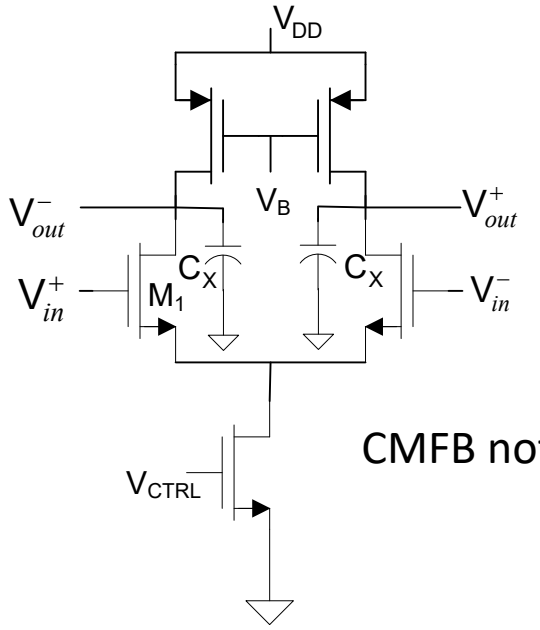


Lossy Integrator-Based VCO



Voltage Controlled Oscillators

Voltage Controlled Oscillator (VCO)

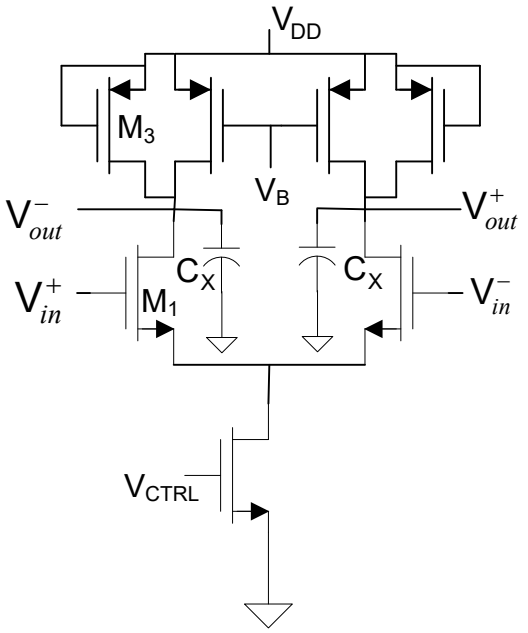


CMFB not shown

$$I_{0d}(s) = \frac{g_{m1}}{sC_X}$$

$$I_0 = \frac{g_{m1}}{C_X}$$

Integrator for: Integrator-based VCO



$$I_{0d}(s) = \frac{g_{m1}}{sC_X + g_{m3}}$$

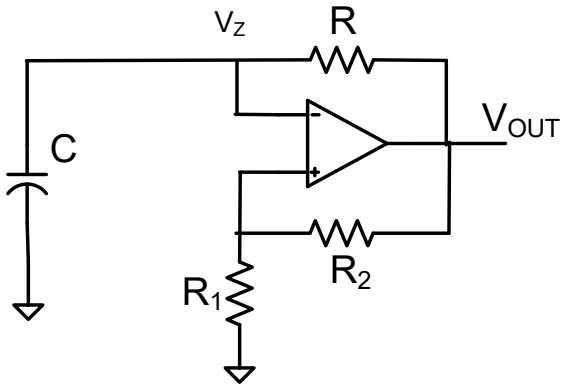
$$I_0 = \frac{g_{m1}}{C_X} \qquad \alpha = \frac{g_{m3}}{C_X}$$

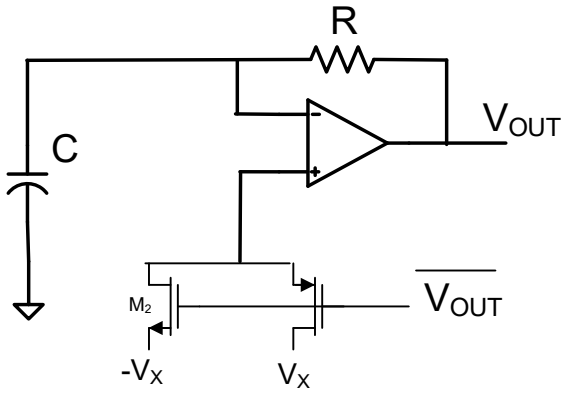
Integrator for: Lossy Integrator-based VCO

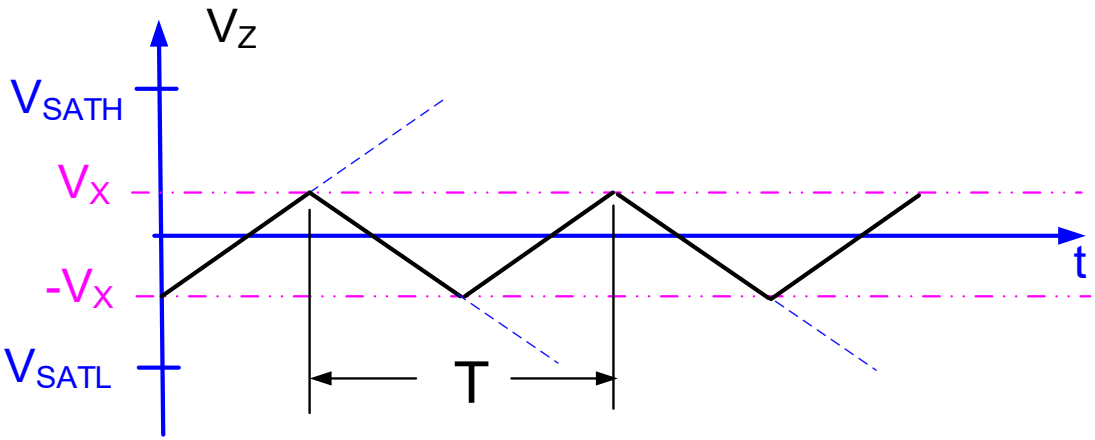
Voltage Controlled Oscillators

Voltage Controlled Oscillator (VCO)

Relaxation Oscillator Derived VCO (comparator not Op Amp)







V_Z is approximately triangle wave, V_{OUT} is square wave

Can have either triangle wave or square wave ouputs

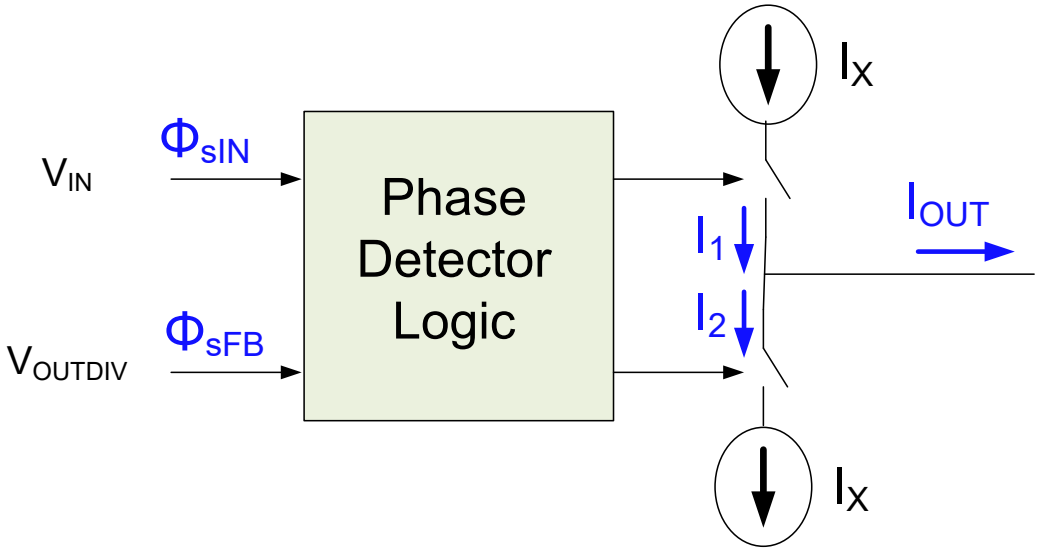
Phase Detectors

Phase Detector (Φ_D)

Many different Phase Detectors can be used

Some Popular Phase Detector Circuits

- Analog Multiplier
- Exclusive OR Gate
- Sample and Hold
- Charge Pump



Charge-pump based Phase Detector

Average I_{OUT} is the average phase

Phase Detectors

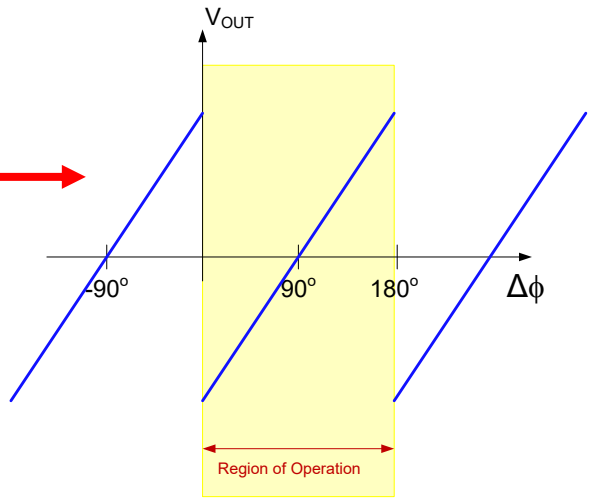
Many different Phase Detectors can be used

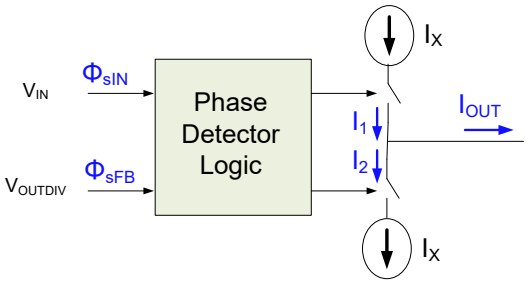
Phase Detector (Φ_D)

Desired phase difference often 0° or 90°

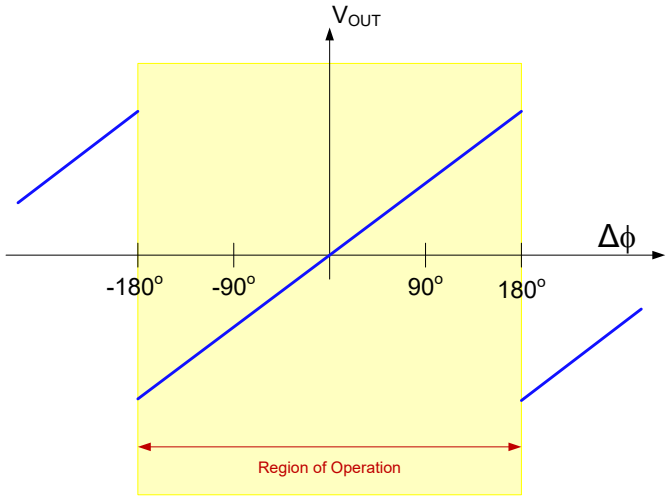
Could be simply

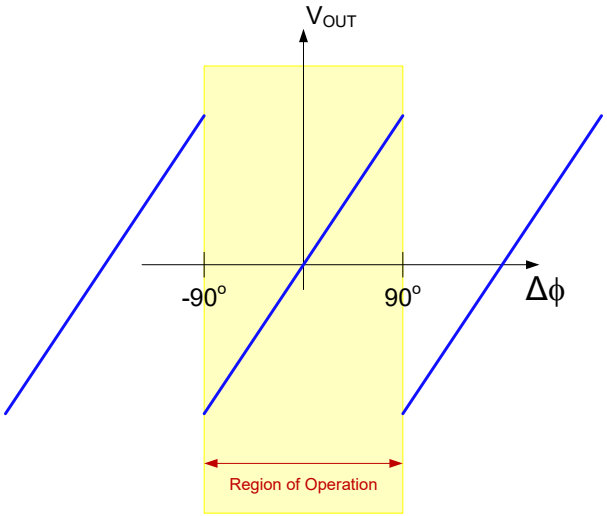
90° nominal phase diff



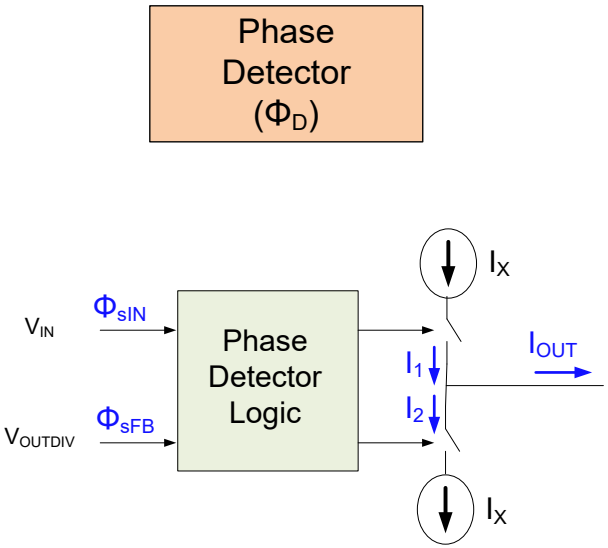


Average I_{OUT} is the average phase

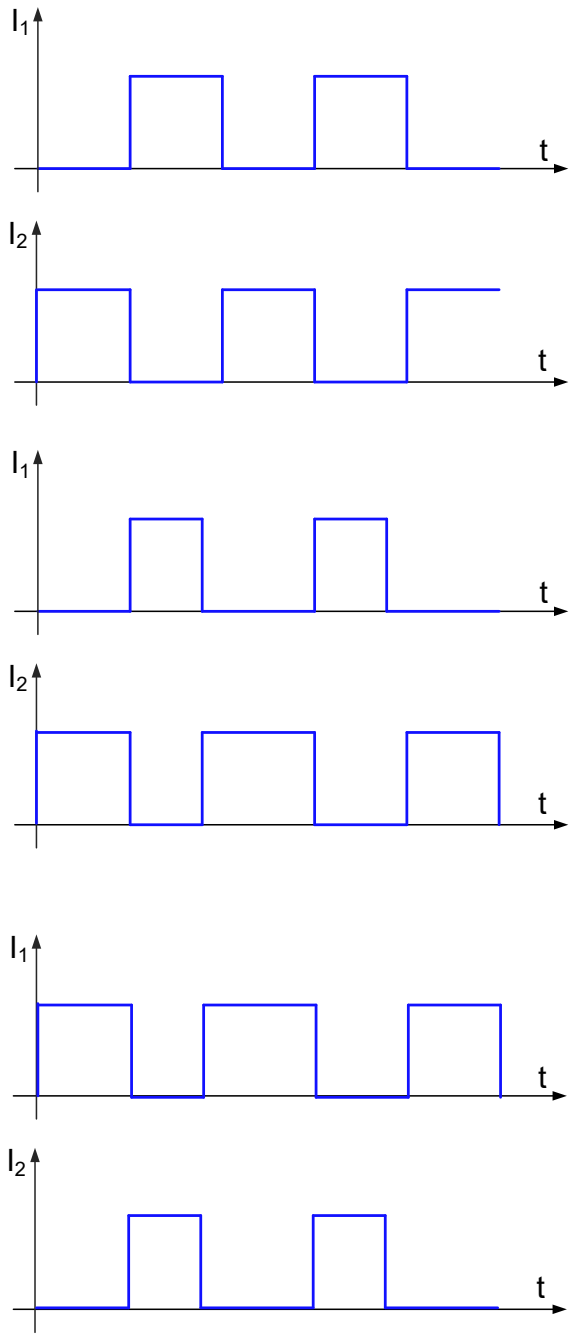




Phase Detectors

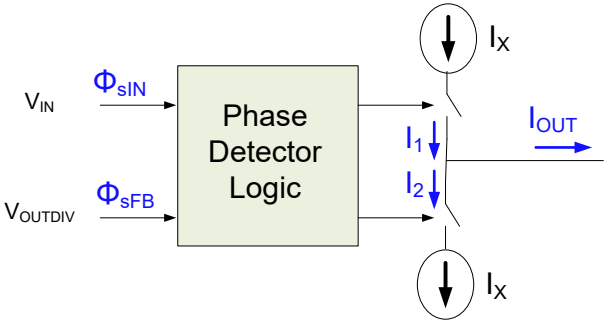


Average I_{OUT} is the average phase

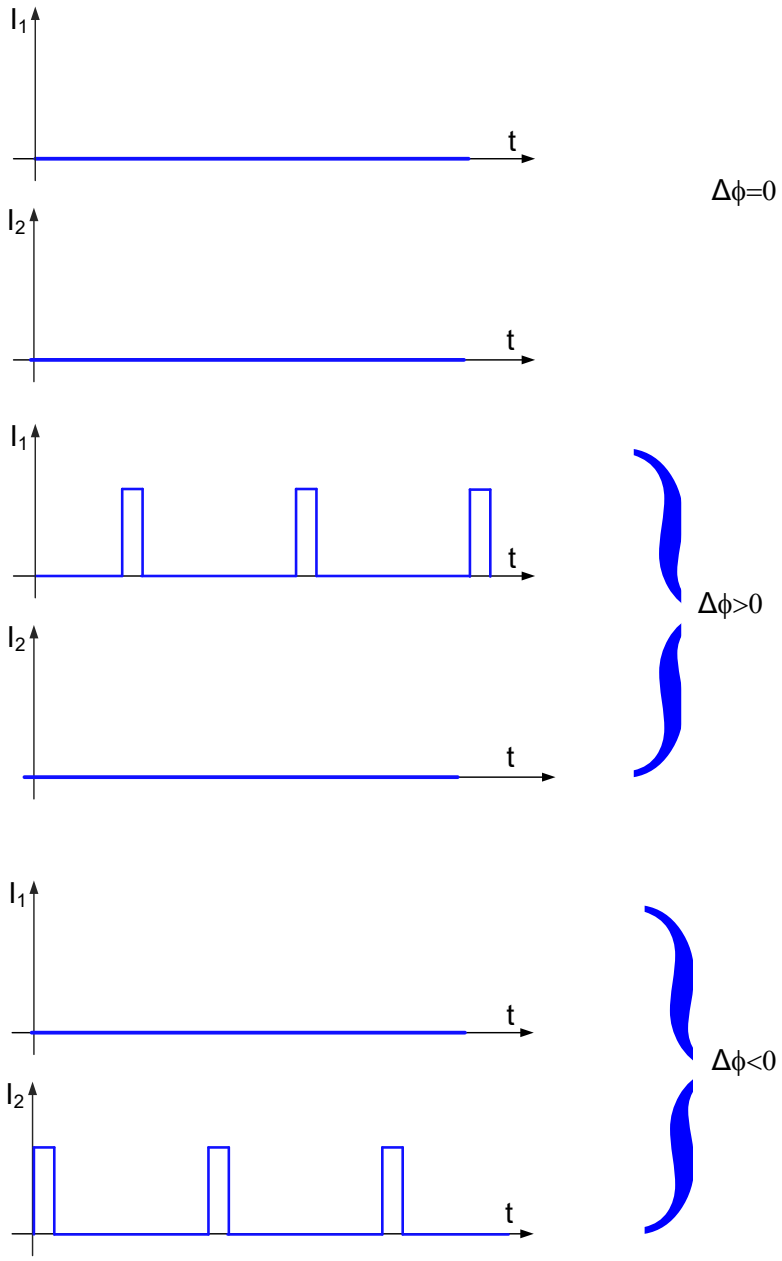


Phase Detectors

Phase Detector (Φ_D)



Average I_{OUT} is the average phase



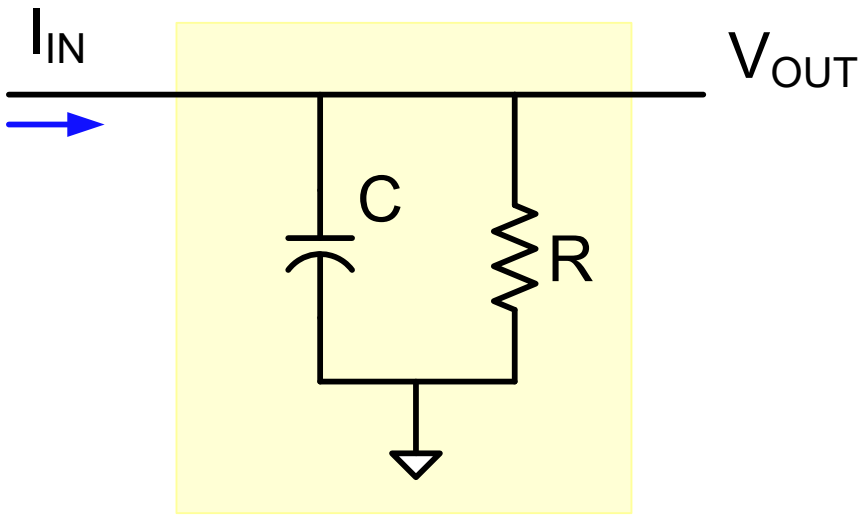
Vulnerable to dead zone problem

Loop Filters

Many different Phase Detectors can be used
Often the loop filter is first or second order
Usually the loop filter circuit is very simple

$T_{LF}(s)$

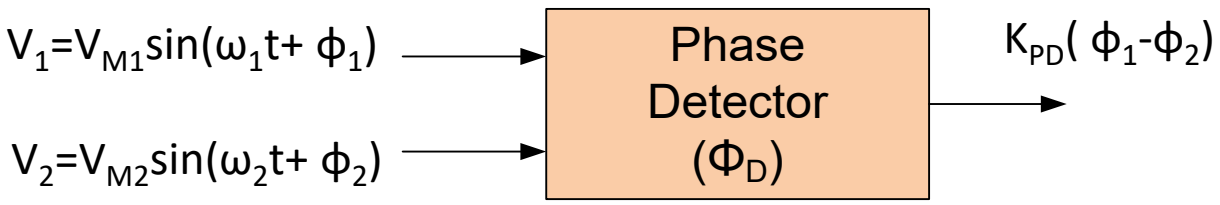
Loop Filter (LF)



$$\frac{V_{OUT}}{I_{INAVG}} = T_{LF}(s) = \frac{R}{1+RCs}$$

Basic first-order LF with average current difference as input

What is the phase of a signal?



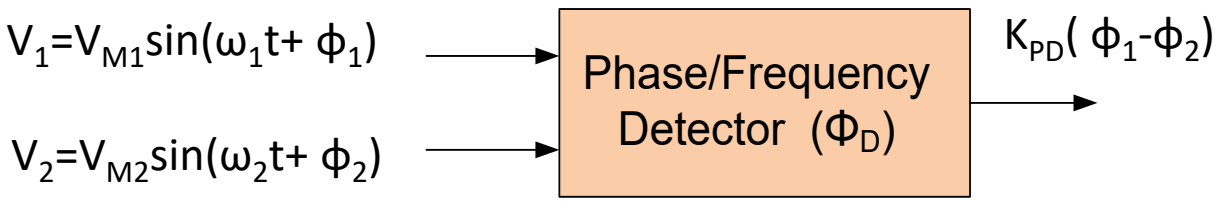
Assume $\omega_1 = \omega_2=\omega$

If V_I can be expressed as $V_1=V_{M1}\sin(\omega t+\phi_1)$ the phase is ϕ_1

But what is the phase if ω is time varying? Or what is the "phase" if this functional form does not really characterize V(t) ? Or what if $\omega_1 \neq \omega_2$?

What does a phase detector do if the two inputs are not at the same frequency?

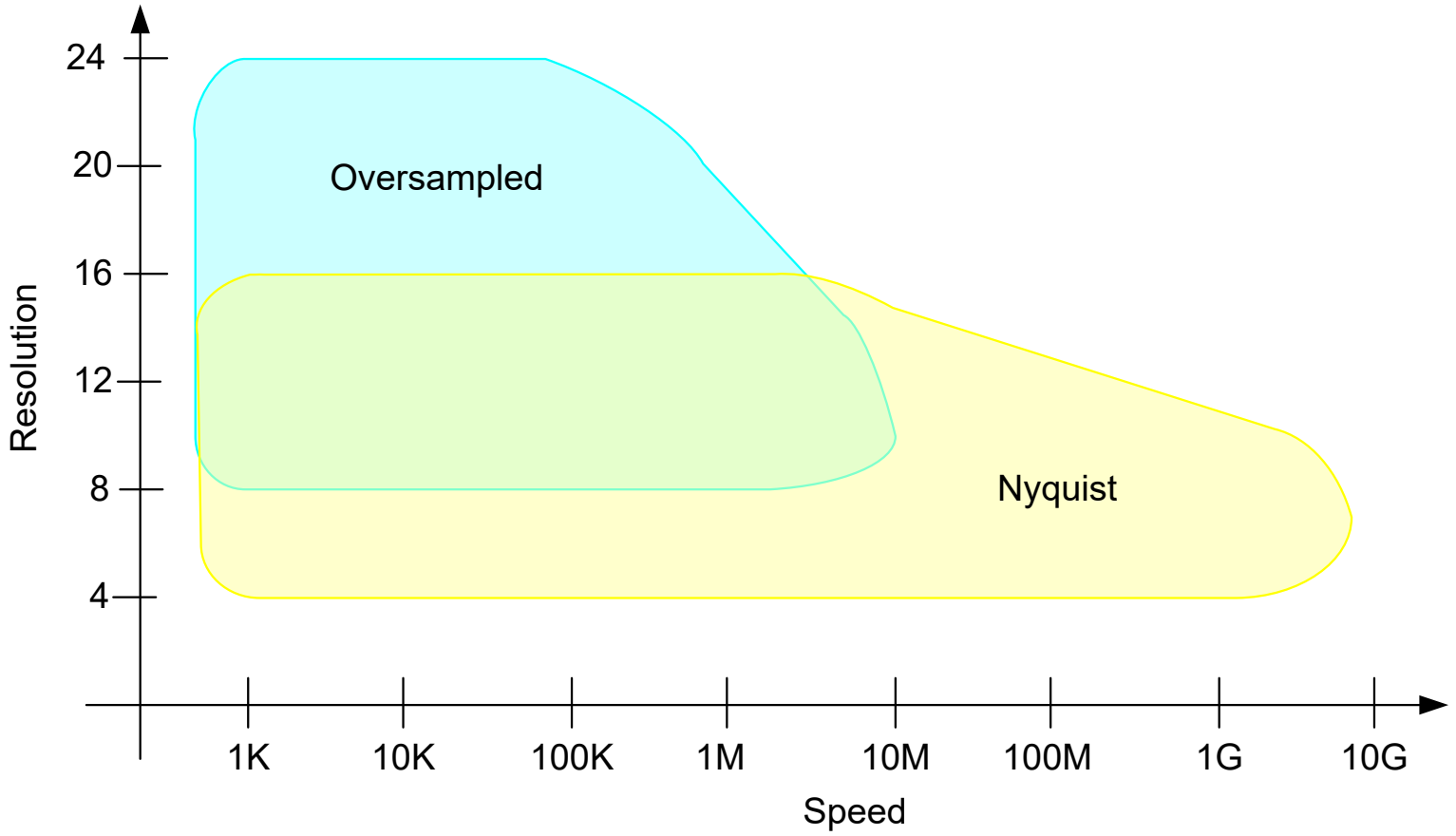
What is the phase of a signal?



Most Phase Detectors are actually Phase/Frequency Detectors

- Large output when frequency difference exists
- Also provides output when phase difference exists after frequencies are matched

Data Converter Type Chart

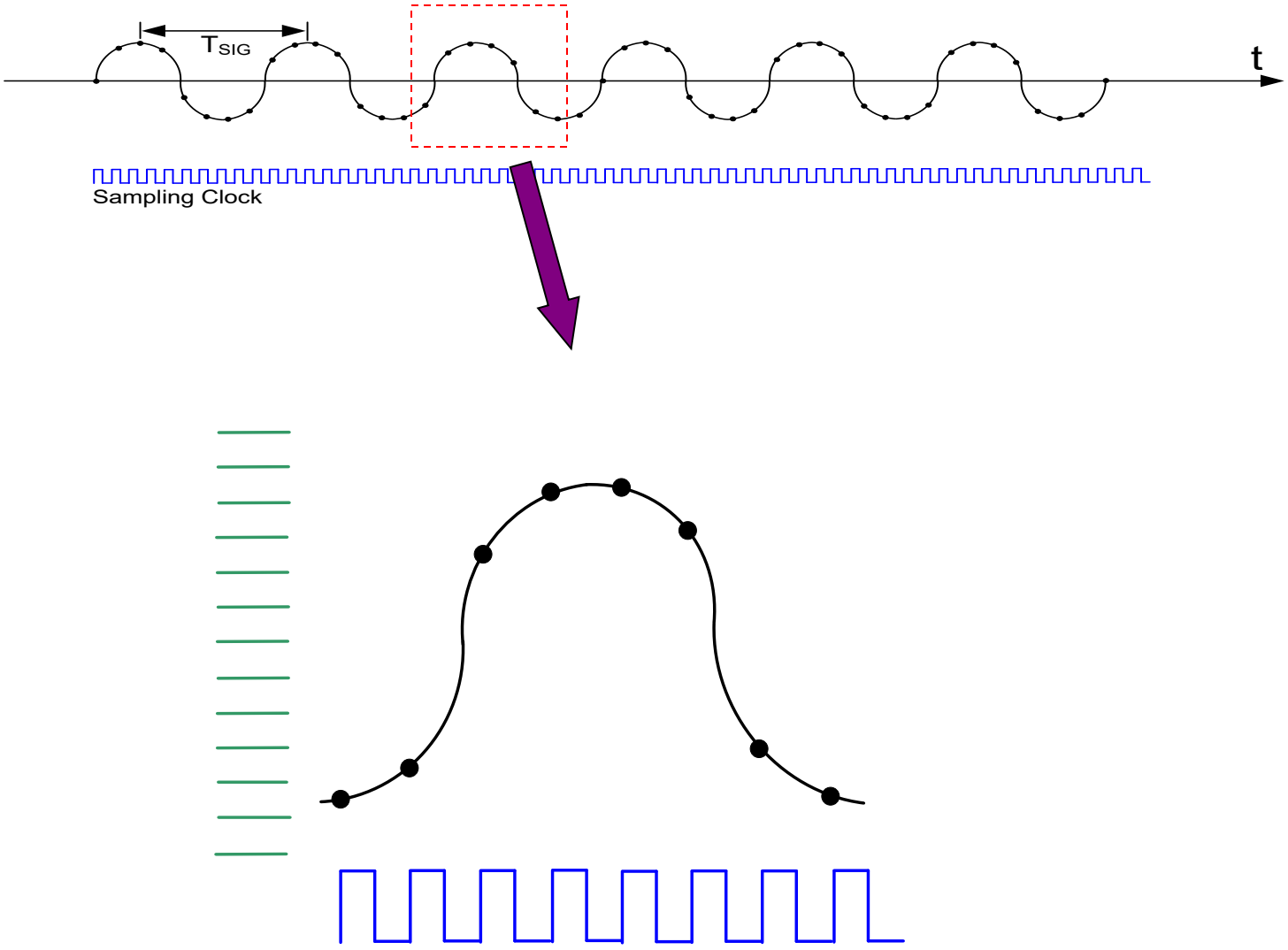


Over-Sampled Data Converters

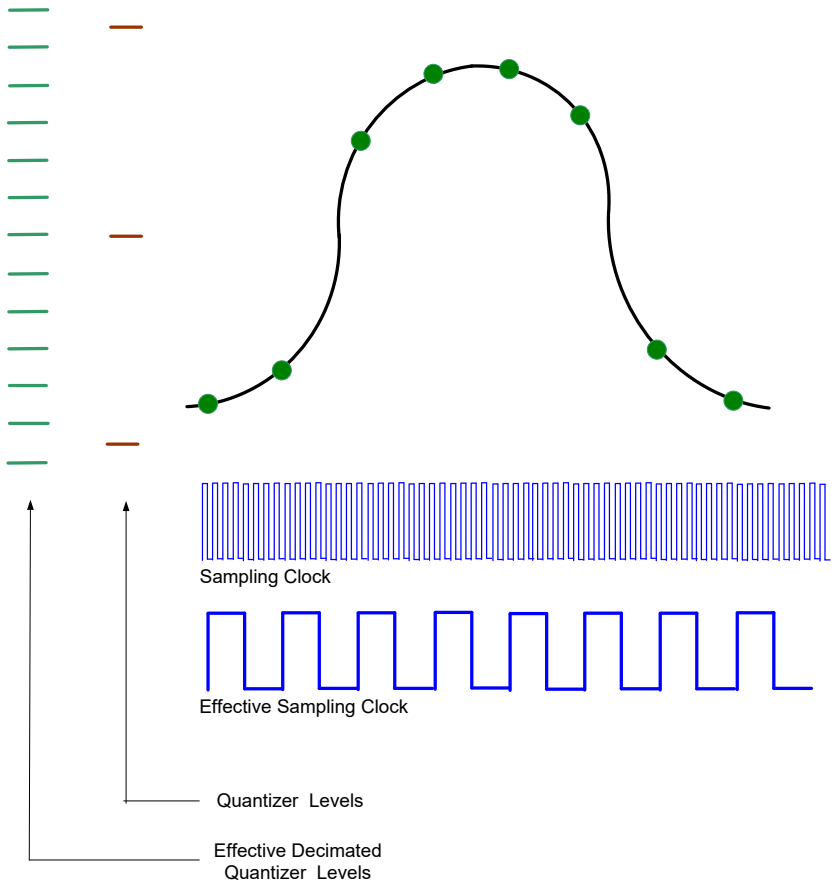
General Classes

- Single-bit
- Multi-bit
- First-order
- Higher-order
- Continuous-time

Nyquist Rate



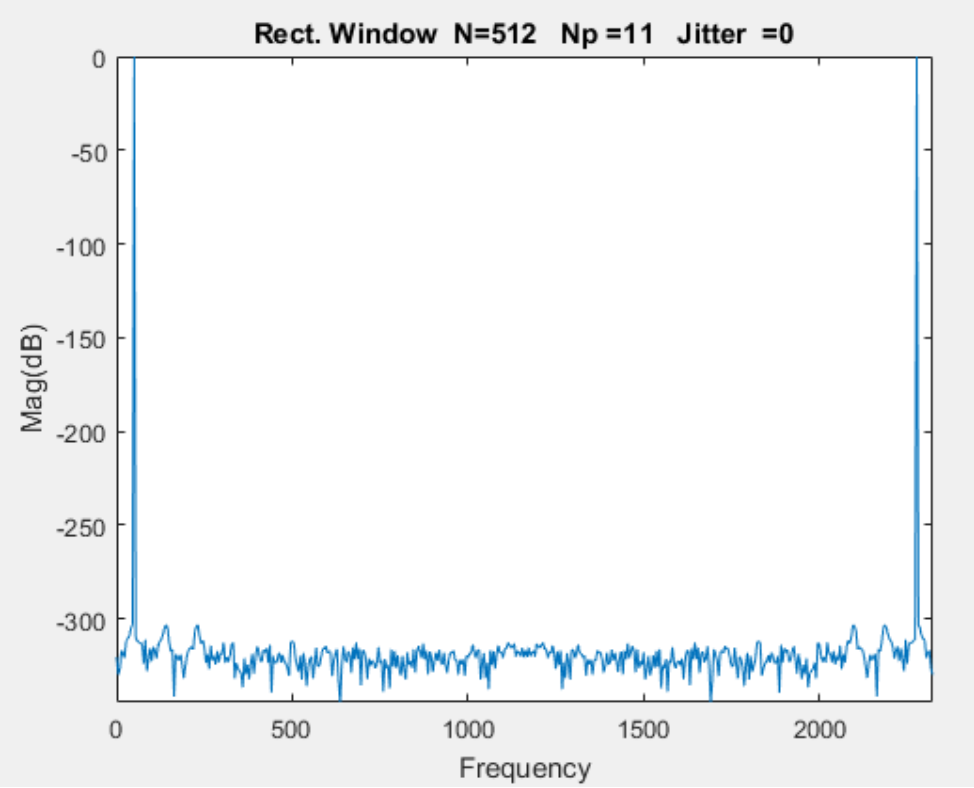
Over-Sampled



Over-sampling ratios of 128:1 or 64:1 are common
Dramatic reduction in quantization noise effects
Limited to relatively low frequencies

Recall:

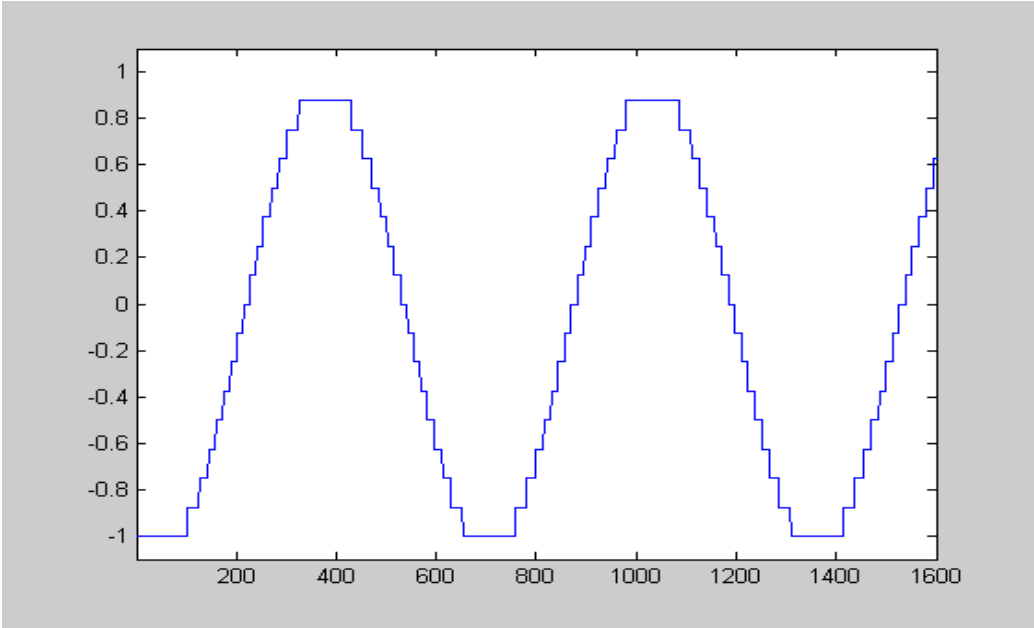
f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=2.3KHz
Oversampled: 23:1



MatLab Results

Recall:

Quantization Effects



Simulation environment:

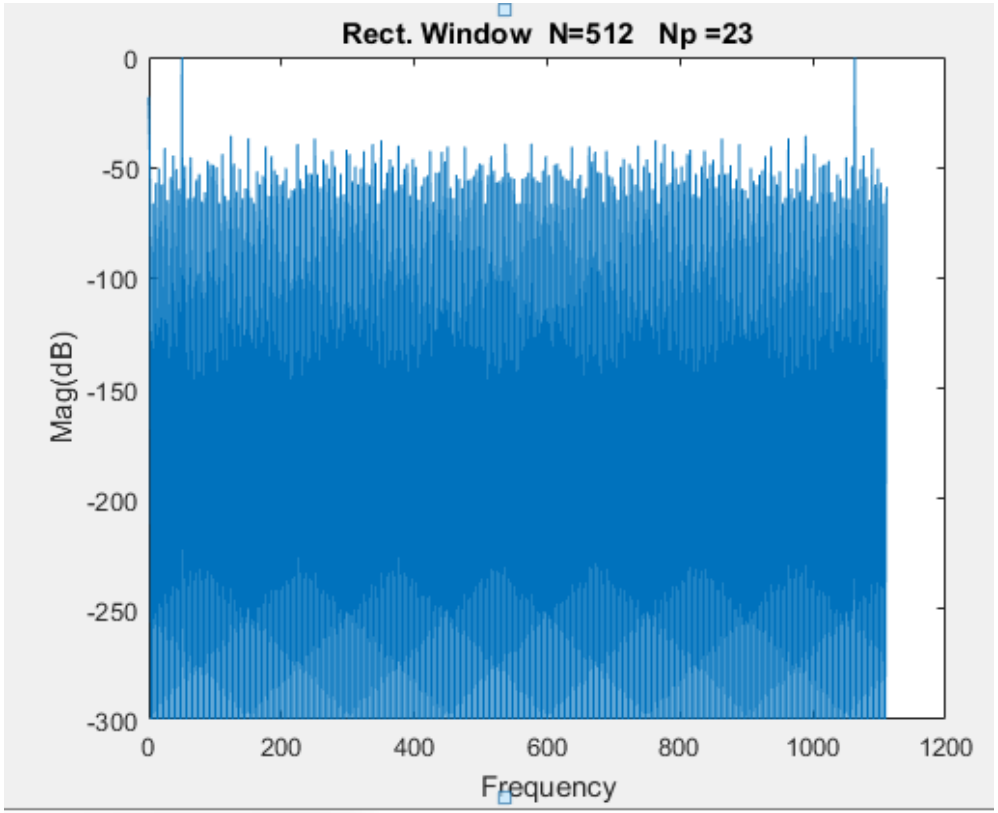
$N_P=23$

$f_{SIG}=50Hz$

Recall:

Quantization Effects

Res = 4 bits



f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=1113KHz
Oversampled: 11:1

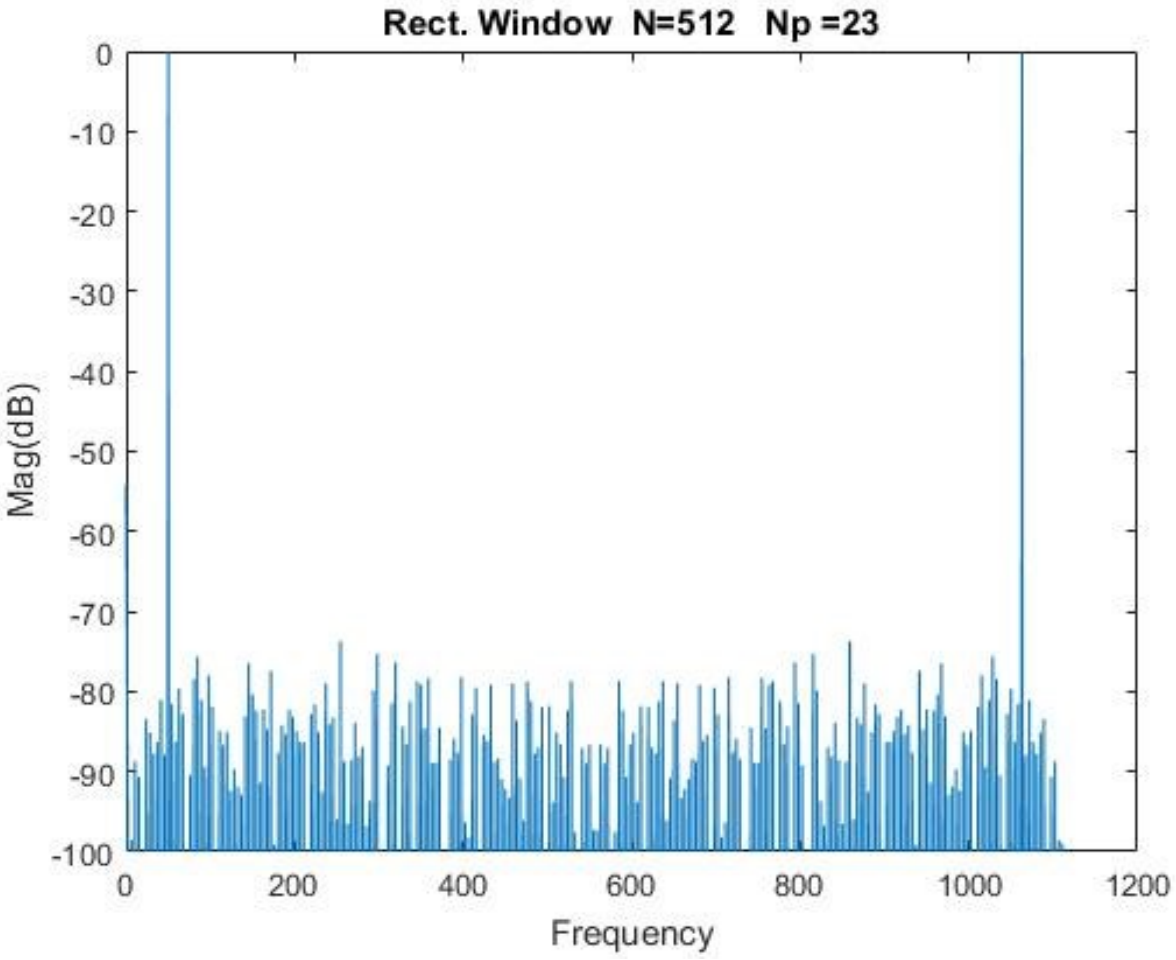
RMS Quantization Noise: $E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$

Lets now increase resolution

Recall:

Quantization Effects

Res = 10 bits



f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=1113KHz
Oversampled: 11:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

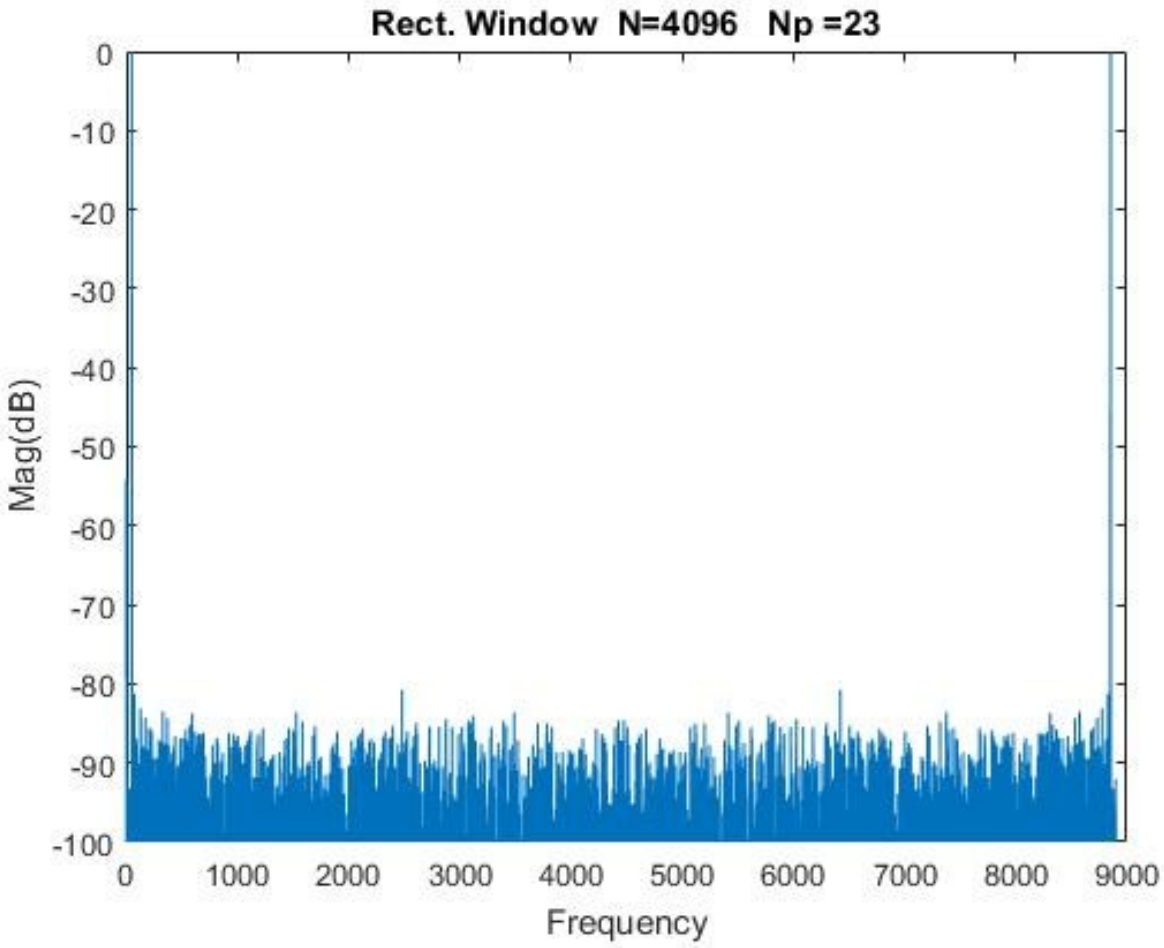
Quantization noise is much lower but still significant

Lets now increase oversampling ratio (i.e. number of samples)

Recall:

Quantization Effects

Res = 10 bits



f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{\mathcal{X}_{LSB}}{\sqrt{12}}$$

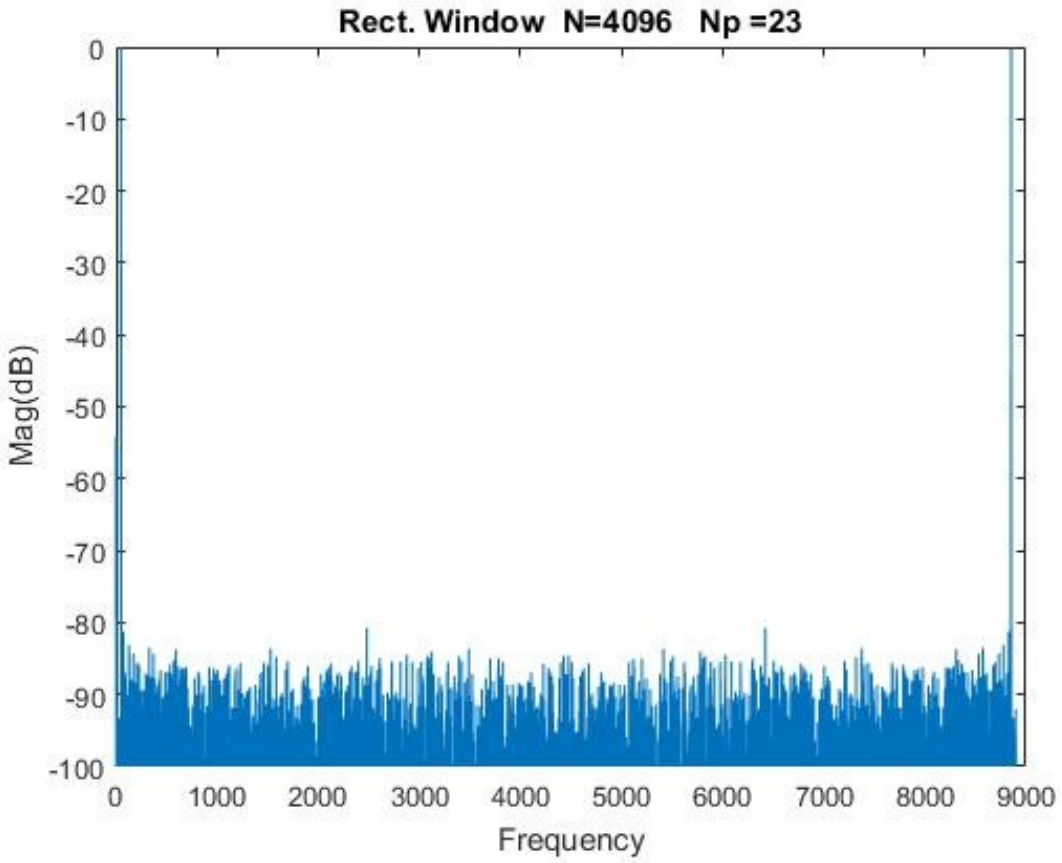
Compared to the previous slide, it appears that the quantization noise has gone down

But has it ? Magnitude of quantization DFT terms decreased but E_{RMS} unchanged

Recall:

Quantization Effects

Res = 10 bits

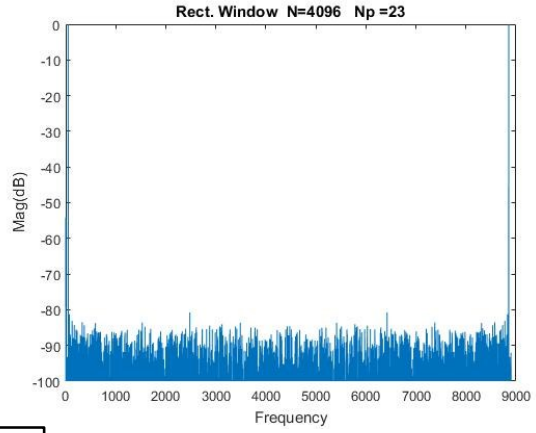


f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{\mathcal{X}_{LSB}}{\sqrt{12}}$$

Can any additional useful information about the input be obtained since we have many more samples than are needed?

Over-Sampling

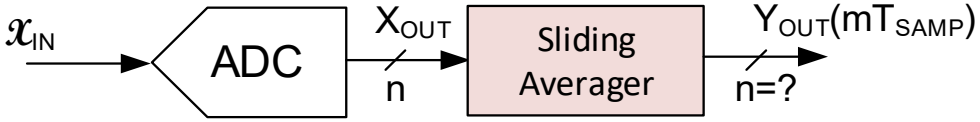


Res = 10 bits

f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

What would happen if we break the 4096 samples into groups of 20 samples and form?

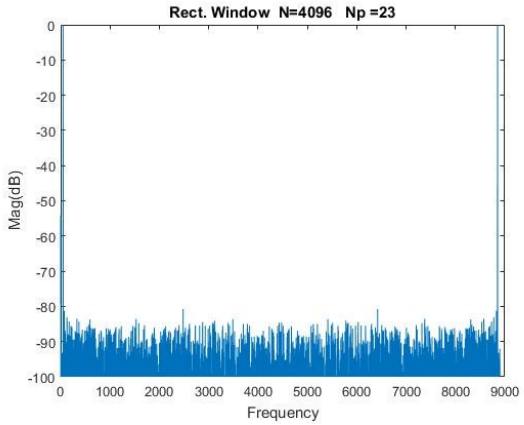


$$E_{RMS} = ?$$

$$\hat{X}_{OUT}(k \bullet 20T_{SAMP}) = \frac{1}{20}\sum_{j=1}^{20} x_{OUT}\left(jT_{SAMP} + 20kT_{SAMP}\right)$$

- Though the individual samples have been quantized to 10 bits, the arithmetic operations will have many more bits
- The effective sampling rate has been reduced by a factor of 20 but is still over 4 times the Nyquist rate
- Has the quantization noise been reduced (or equivalently has the resolution of the ADC been improved?
- Is there more information available about the signal?

Over-Sampling

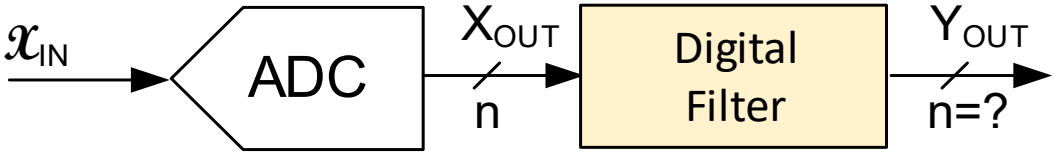


Res = 10 bits

f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

Since the quantization noise is at high frequencies, what would happen if filtered the Boolean output signal?

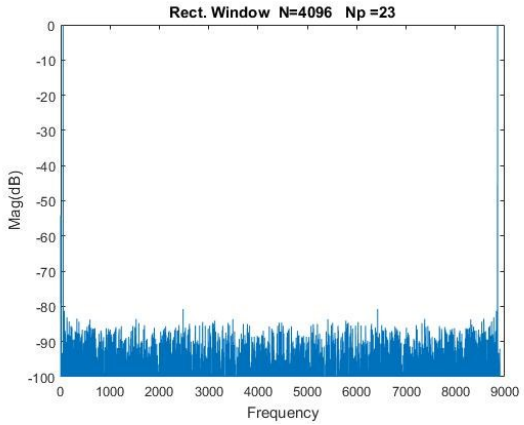


$$E_{RMS} = ?$$

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}(k - jT_{SAMP})$$

Or $$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}(k - jT_{SAMP}) + \sum_{j=1}^{h} b_j Y_{OUT}(k - jT_{SAMP})$$

Over-Sampling



Res = 10 bits

f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

What does this difference equation represent?

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}(k - jT_{SAMP})$$

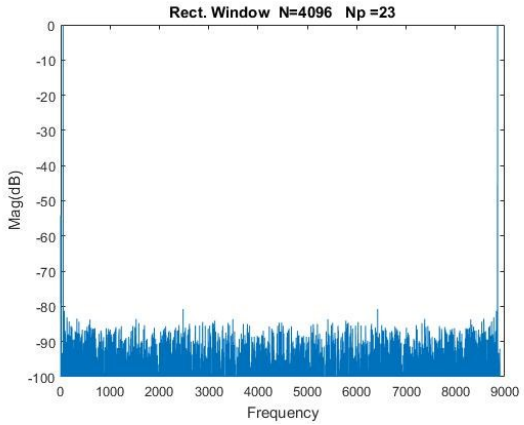
- Moving Average (MA) Digital Filter
- Filter shape (e.g. low-pass, band-pass, high-pass, … dependent upon <a_i> coefficients)

What does this difference equation represent?

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}(k - jT_{SAMP}) + \sum_{j=1}^{h} b_j Y_{OUT}(k - jT_{SAMP})$$

- Auto Regressive Moving Average (ARMA) Digital Filter
- Filter shape (e.g. low-pass, band-pass, high-pass, … dependent upon <a_i> and <b_j> coefficients)

Over-Sampling



Res = 10 bits

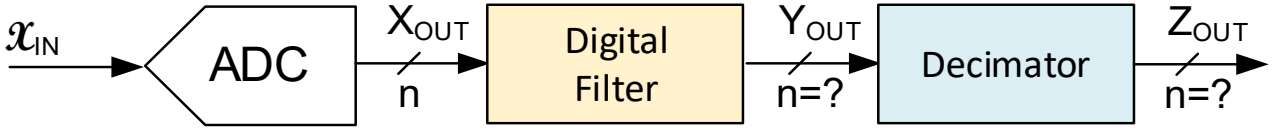
f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

Since the quantization noise is at high frequencies, what would happen if filtered and decimated the Boolean output signal?

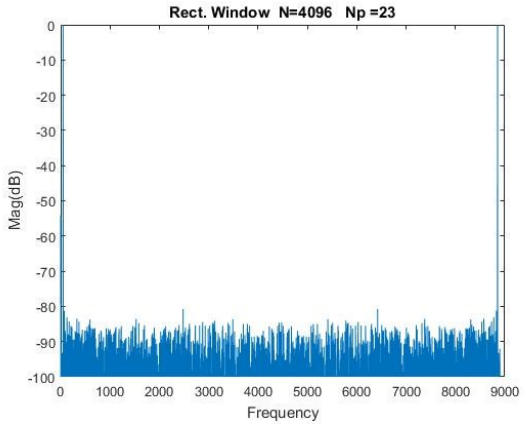
$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}(k - jT_{SAMP})$$

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}(k - jT_{SAMP}) + \sum_{j=1}^{h} b_j Y_{OUT}(k - jT_{SAMP})$$

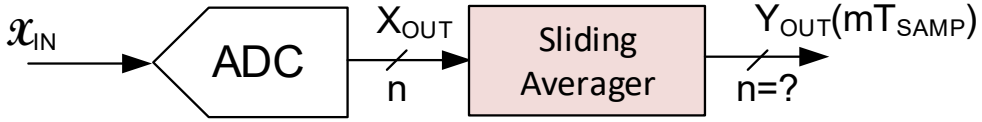


$$E_{RMS} = ?$$

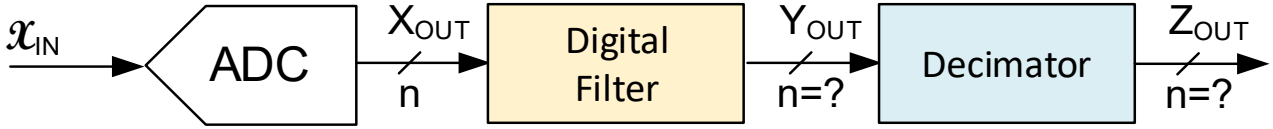
Over-Sampling



$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$



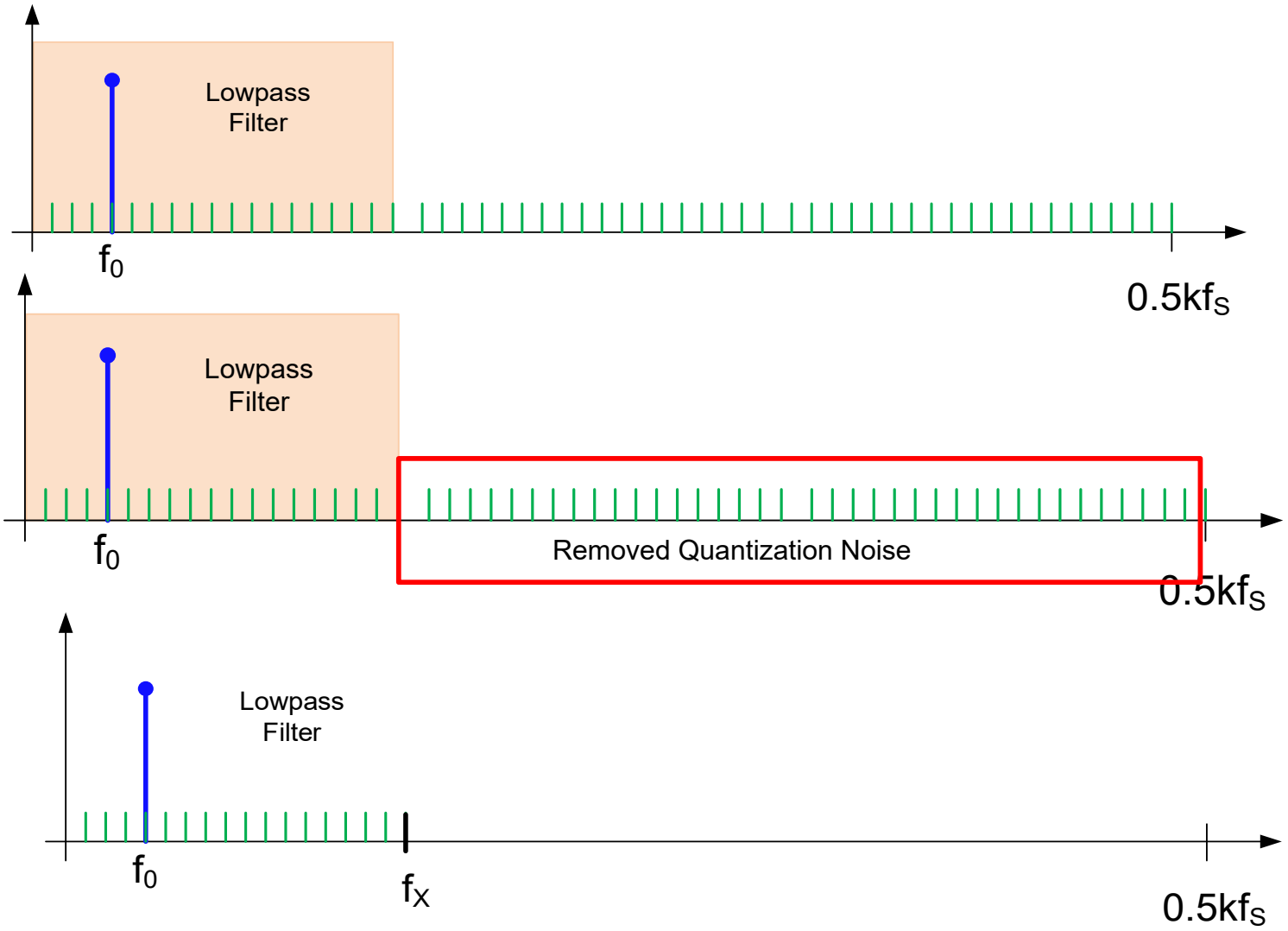
$$E_{RMS} = ?$$



$$E_{RMS} = ?$$

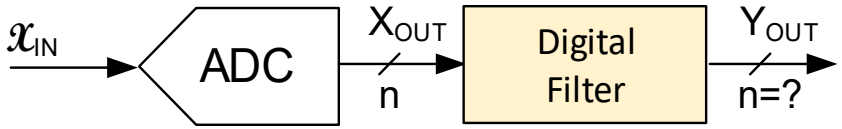
- What is the overhead?
- What is the performance potential?
- How can these or related over-sampling approaches be designed?
- Though this approach may help quantization noise, will not improve ADC linearity

Over-Sampled Spectrum showing quantization noise with digital lowpass filter



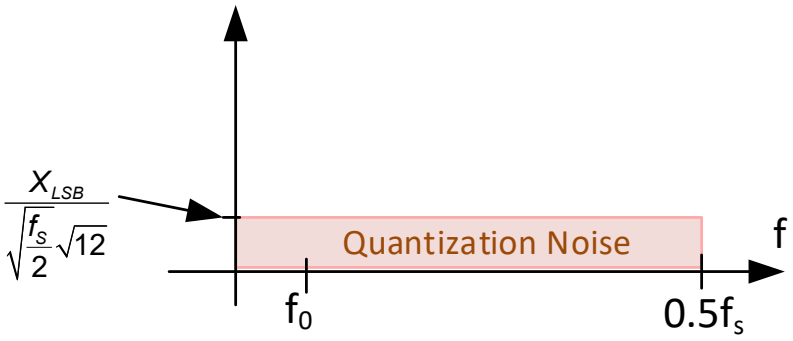
Residual Quantization Noise if Filter Band-edge at f_X

Over-Sampling



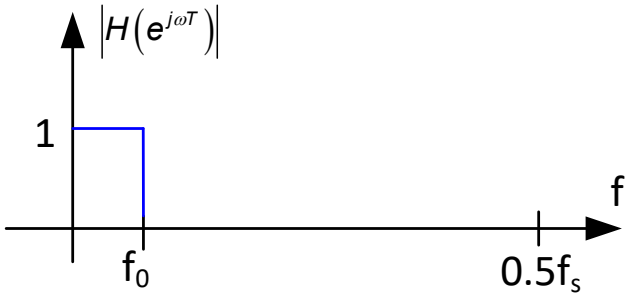
$f_S = f_{SAMP}$

$$OSR = \frac{f_S}{2}\frac{1}{f_0}$$



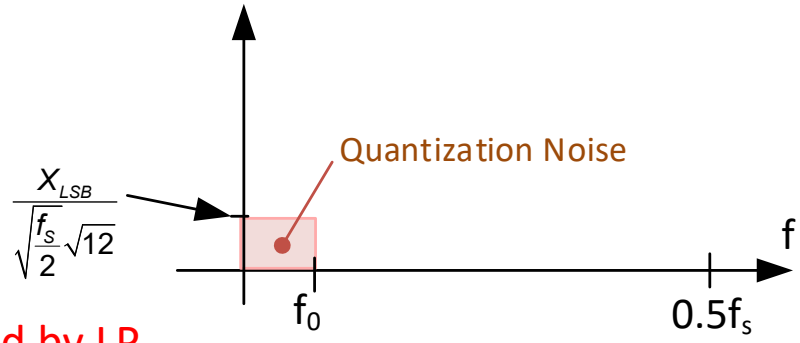
With ideal lowpass filter with band-edge at f_0

$$V_{Qrms} = \frac{V_{LSB}}{\sqrt{12}}\frac{1}{\sqrt{OSR}}$$



For sinusoidal input with p-p value V_{REF}

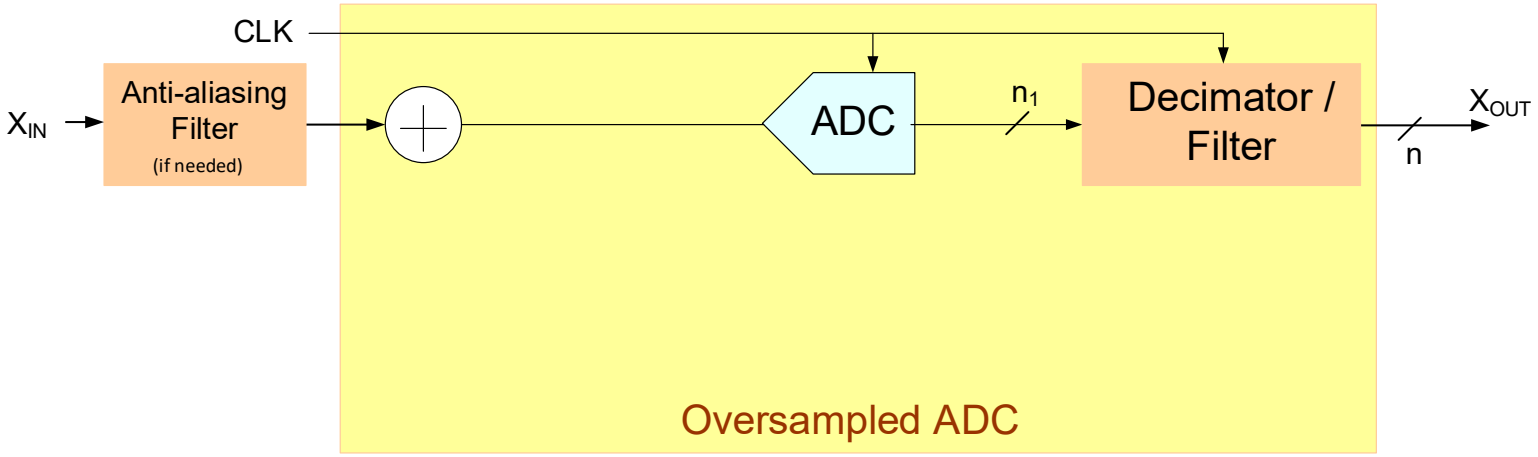
$$SNR = 6.02n + 1.76 + 10\log(OSR)$$



Improvement of 3dB/octave or 0.5bits/octave

Oversampling increases resolution and if followed by LP filter Reduces Quantization Noise!

Over-sampled ADC



- Anti-aliasing filter at the input (if needed) to limit bandwidth of input signal
- ADC is often simply a comparator
- CLK is much higher in frequency than effective sampling rate (maybe 128:1 though lower OSR also widely used)
- Can obtain very high resolution but effective sampling rate is small
- **With clever design, this approach can reduce quantization effects and improve linearity**

Stay Safe and Stay Healthy !

End of Lecture 43