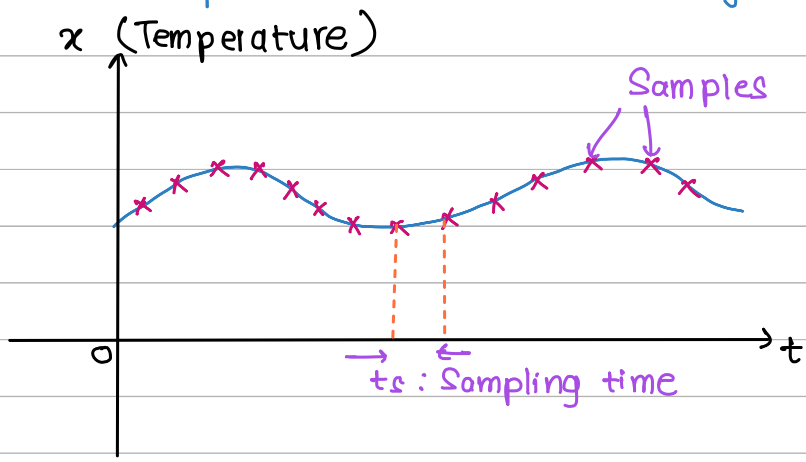
Digital Instrumentation – Sampling 1

→ How to measure and represent changing parameters?
→ How frequently do we have to measure?
→ How to represent the measured parameters without losing essential information?

Consider the temperature variation throughout time



→ Measurements cannot be taken at every time.
→ But that much of time resolution is not needed

Sampling frequency ;

$$f_s = \frac{1}{t_s}$$

Sampling → Uniform sampling
t_s is same throughout
↘ Non - uniform sampling
t_s varies from time to time.

eg: For audio signals we use a uniform sampling frequency of 44100 Hz (or 48000 Hz)
↑ Studio quality.

Variation of temperature.
Sometimes temperature varies rapidly and sometimes

temperature varies slowly. Therefore t_s can be large for slow variations and less for rapid variations
– Non-uniform sampling.

How to determine sampling time?

→ If reduce sampling time
  ↳ Have to take lot of measurements
  ↳ Ends up with large number of data

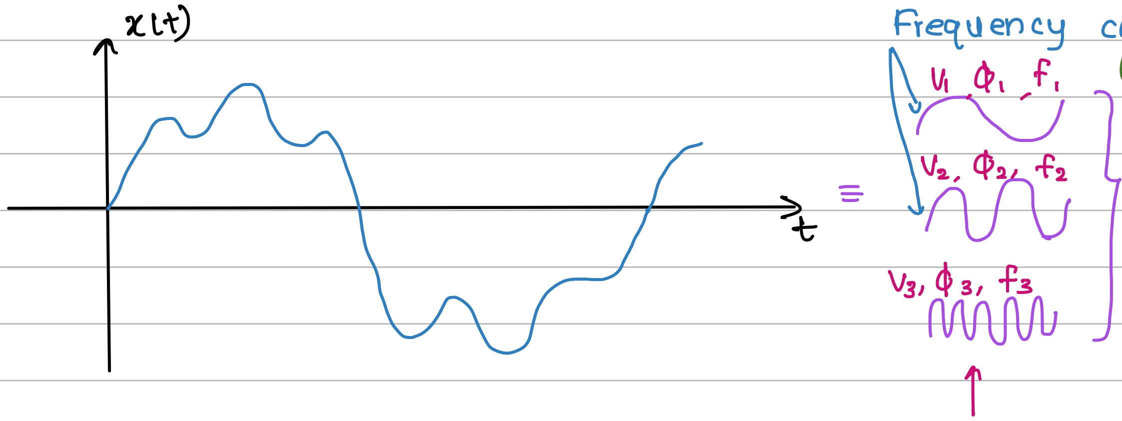
→ If increase sampling time
  ↳ Necessary information can be lost.

Therefore, 1) Reduce the # samples as much as possible
2) Reduce the loss of information.

Quality vs. Quantity

Sampling Theorem : How to Determine the Optimal Sampling Frequency

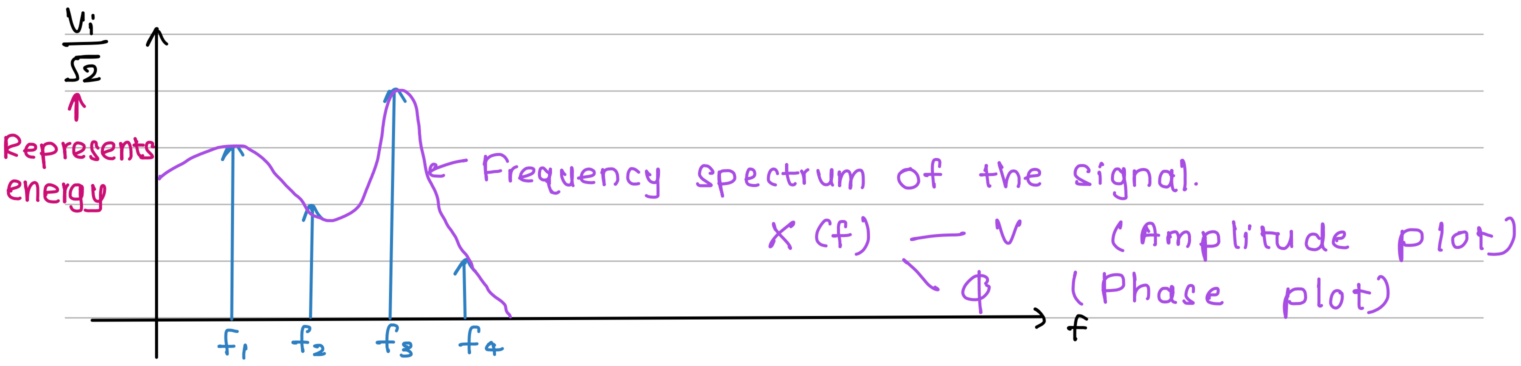
Note: Frequency Domain Representation of a Signal



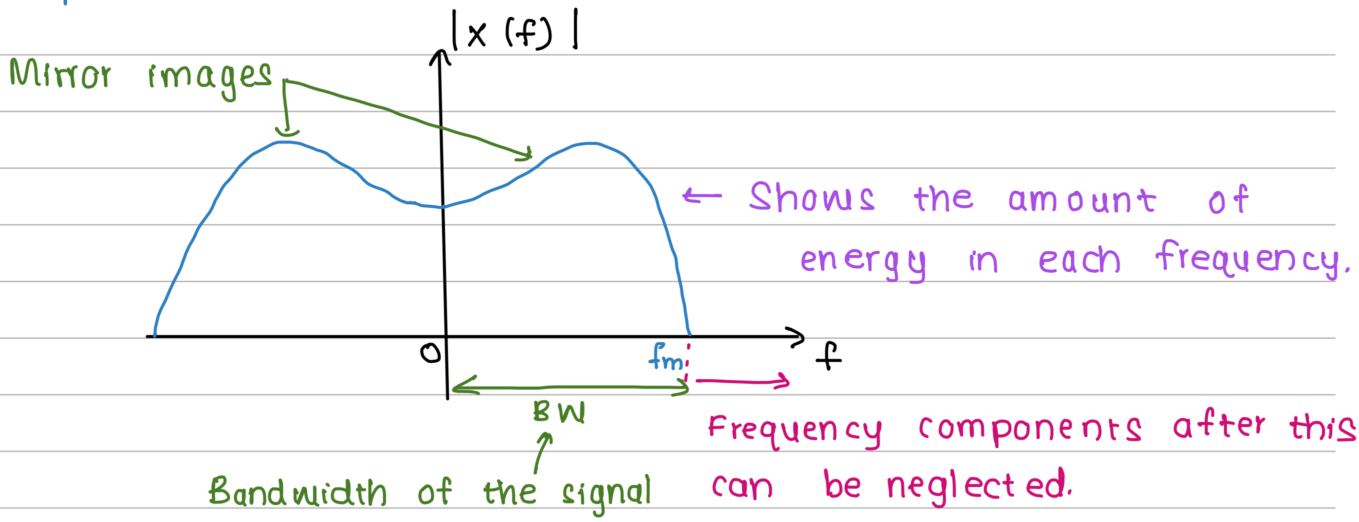
Frequency components.
(Different amplitudes, different frequencies + different ϕ)

Any time domain signal can be represented as a summation of sinusoidal signals
→ In more advanced methods can be represented as an integration (when frequencies become continuous)

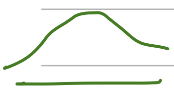
Then, $x(t) = \sum_{i=1}^{N} V_i \cos(2\pi f_i t + \phi_i)$



∴ any signal can be represented using its frequency spectrum.

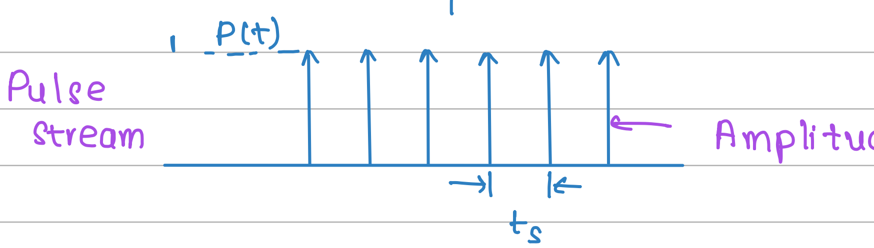


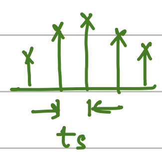
To mathematically model the sampling process,



$x(t) \longrightarrow \otimes \longrightarrow X_s(t)$ Sampled signal

Multiplication

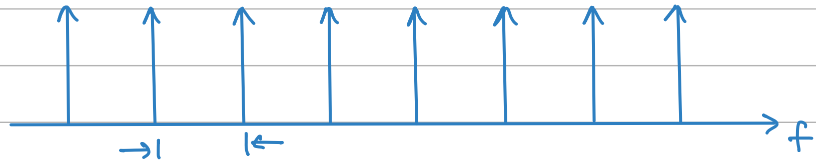




$|X_s(f)| \rightarrow$ Frequency spectrum of the sampled signal.

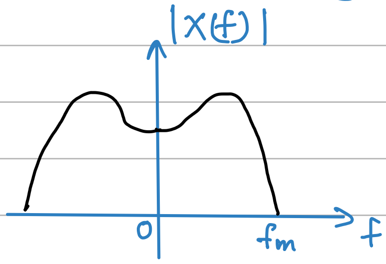
$P(f) \rightarrow$ Pulse stream in the frequency domain.

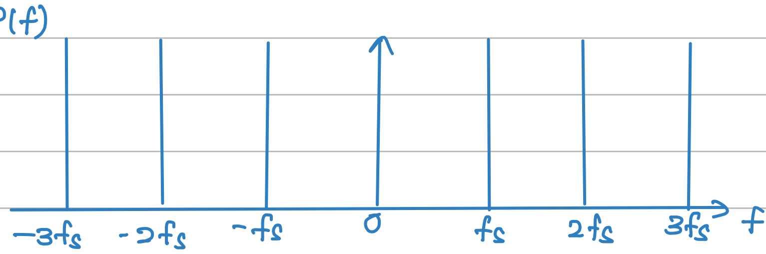
$P(f) \Rightarrow$



$f_s = 1/t_s$

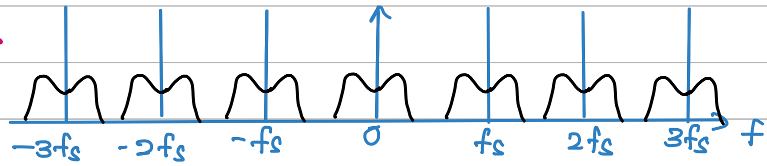
- When 2 signals are multiplied in the time domain, convolution is done in frequency domain.





By convolution of these 2.

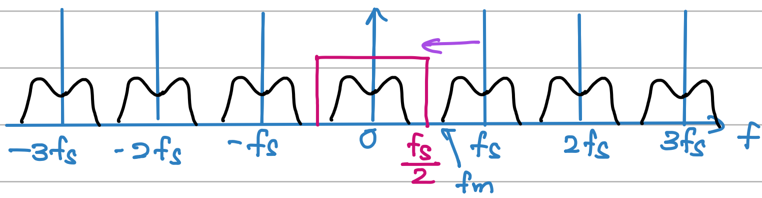
$X_s(f) \longrightarrow$



How to get the original $X(f)$ back?

→ By using a filter

→ Cuts down unwanted frequencies.



Multiply $X_s(f)$ with filter.

What is the minimum value of f_s?

when reducing f_s

$$\frac{f_s}{2} \leq f_s - f_m$$

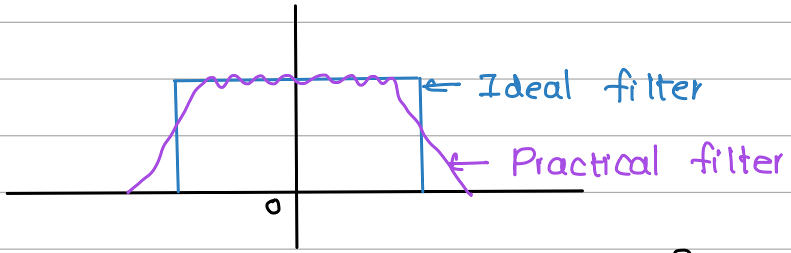
$$2f_m \leq f_s$$

To get the original shape after filtering

* Nyquist sampling theorem,

$f_s \geq 2f_m$ for the signal to be reconstructed.

But practically,



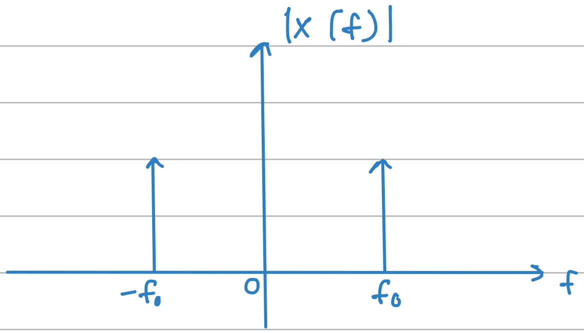
∴ practically $f_s \geq 2.5 f_m$ or $f_s \geq 3 f_m$ } Depends on how good the filter is.

Activity

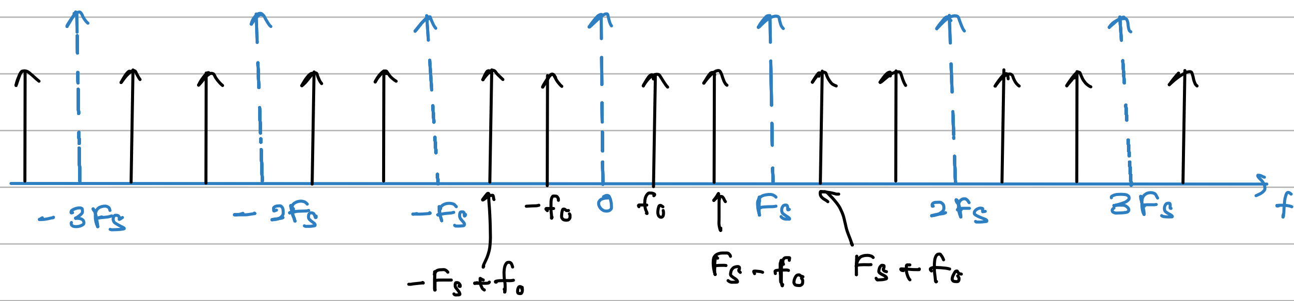
Suppose you sample a pure sinusoidal signal x(t) with frequency f0.
a). Draw the frequency spectrum of the signal x(t)
b). If the signal is sampled at a frequency Fs, where Fs>2f0, draw the frequency spectrum of the sampled signal
c). If Fs<2f0, and if the signal is filtered using an ideal filter (square shape) with width of Fs/2, what will be the frequency of the output signal?

$x(t)$ → Pure sinusoidal
→ Frequency $= f_0$

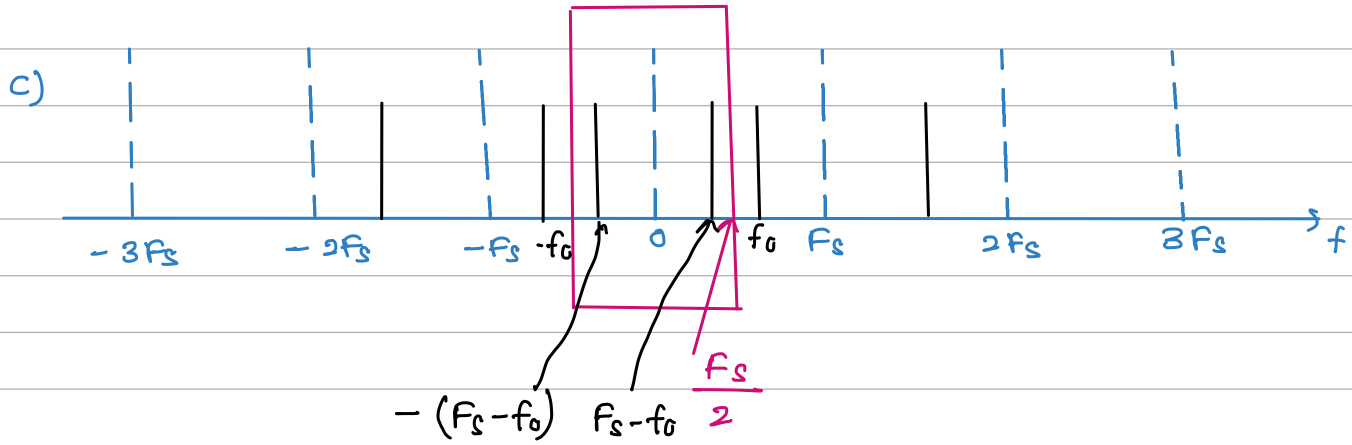
a)



b)

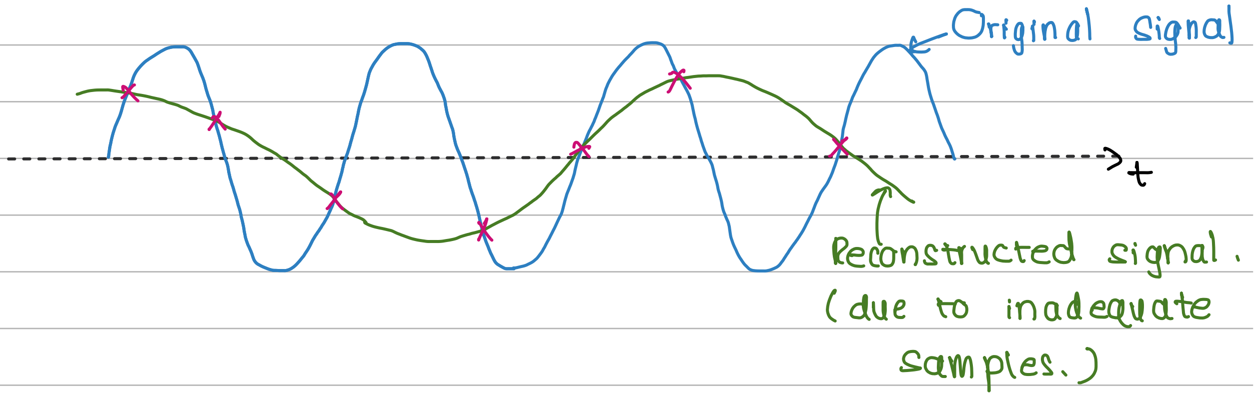


c)



∴ Frequency of the output signal $= F_s - f_o$

This can be explained in time domain also.



Very low sampling rate

Frequency has reduced

This property is called "ALIASING"

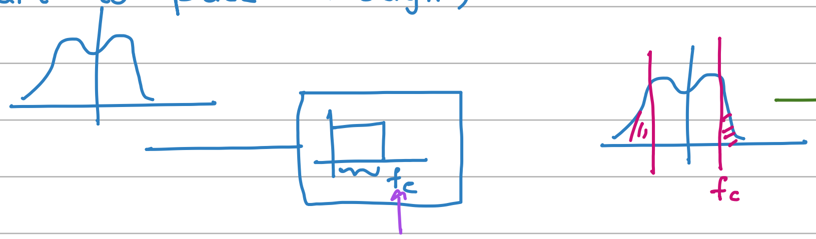
→ Due to aliasing, the signals get distorted.

To avoid aliasing

→ Can select a good sampling frequency (F_s)

→ Make sure that the maximum frequency of the signal to be sampled is $\leq F_s/2$

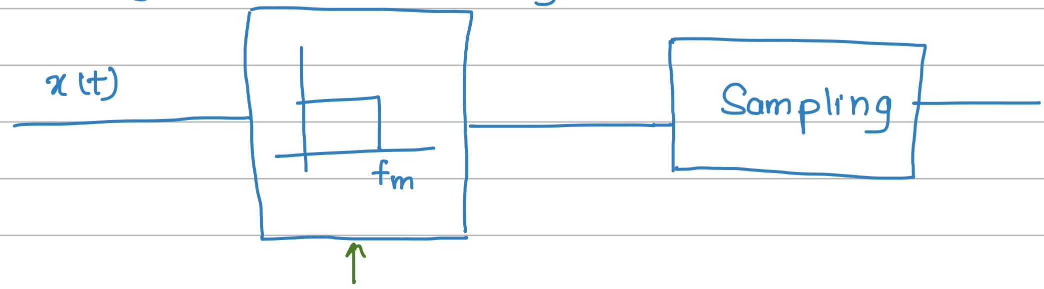
∴ to avoid aliasing we use a low pass filter (an electronic circuit which will allow only the low frequency part to pass through.)



→ Limits the maximum frequency of the spectrum.

Cut off frequency.

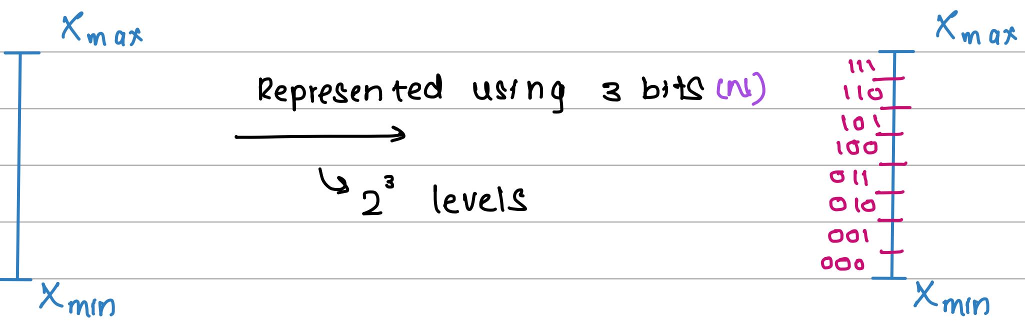
Before sampling an anti-aliasing filter is used of f_m.

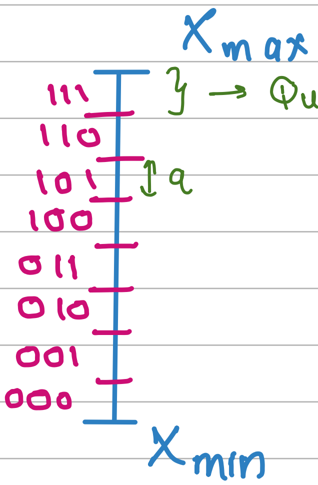


Anti-aliasing filter.

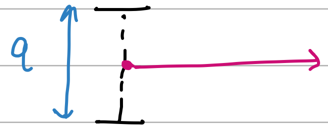
Quantization

- Representing a continuous valued parameter using finite number of values





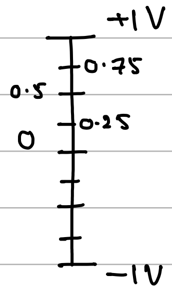
} → Quantization region



Any value in the region will be represented by the middle value when reconstructing the signal

$$q = \frac{X_{max} - X_{min}}{2^N}$$

eg: Sound card :



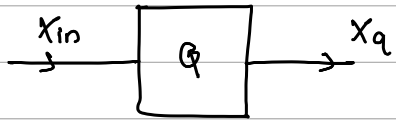
$$q = \frac{1-(-1)}{8} = 0.25\,V$$

For 0.27 V voltage
binary representation → 101
(considering above)
Quantized representation → 0.25 + 0.125
= 0.275 V //

Quantization Error

$-q/2 < e < q/2$ → "Mid riser quantization"

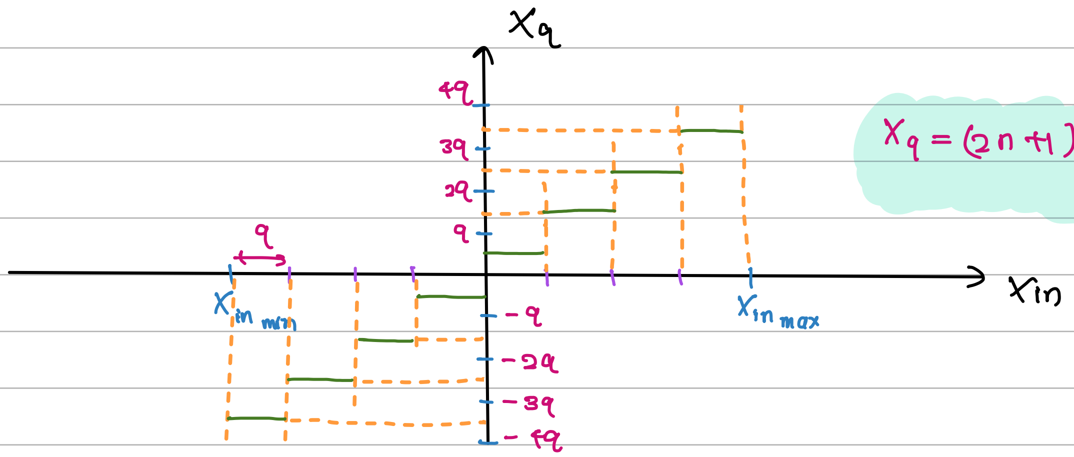
Example



For mid riser quantization,

a) Plot a graph of X_q vs. X_{in} (Take N = 3)

b) Derive an equation for X_q in terms of X_{in} for an N bit system.

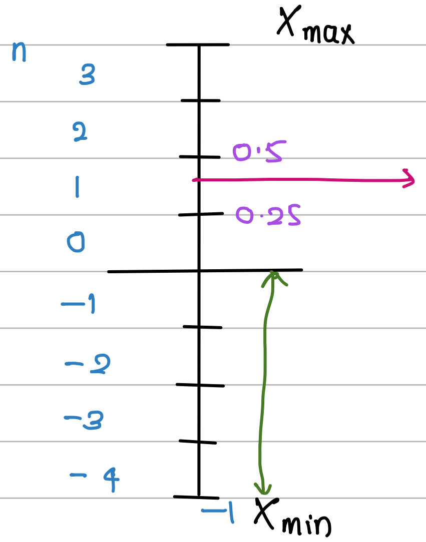


$$X_q = (2n+1)\frac{q}{2}$$

If $X_{in\,min} > 0$ $X_{in\,max} > 0$ → Unipolar

If range of X_{in} $[-V_m, +V_m]$ → Bipolar.

Finding n :



$x = 0.48\,V$

$$n = \left\lfloor \frac{x - X_{min}}{q} \right\rfloor - \left(\frac{L}{2}\right)$$

Floor value

$$X_q = (2n+1)\frac{q}{2}$$

eg : 0.48 V,

$$n = \left\lfloor \frac{0.48 - (-1)}{0.25} \right\rfloor - \frac{8}{2}$$

$$= 1$$

$$X_q = (2 \times 1 + 1)\frac{q}{2}$$

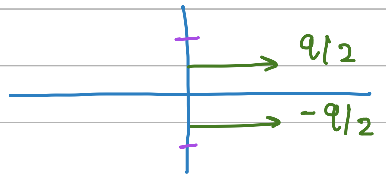
$$= \frac{3q}{2}.$$

$$= 0.375\,V$$

* Quantization error (e) is a random number
* The variance of e increases as number of bits (N) decreases.

Drawbacks of Mid-Riser-Quantization Method.

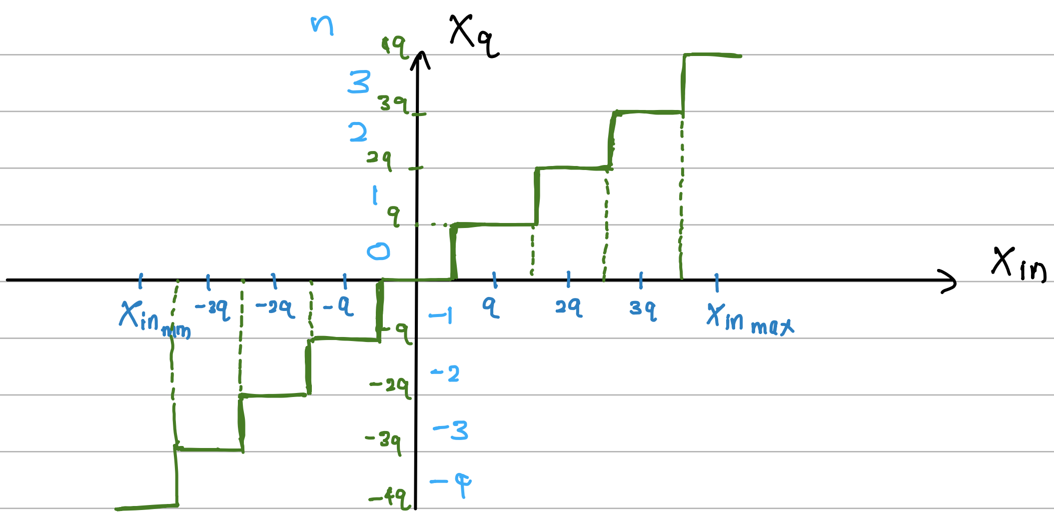
- X_q does not have a zero value



This can be a problem in instrumentation applications.

MID TREAD QUANTIZATION

The mid riser is lowered by q/2.

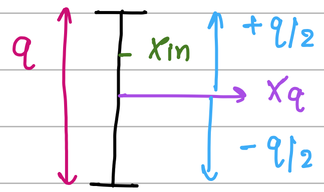


$$n = \text{round}\left(\frac{x - x_{min}}{q}\right) - \frac{L}{2}$$

$$x_q = n\,q$$

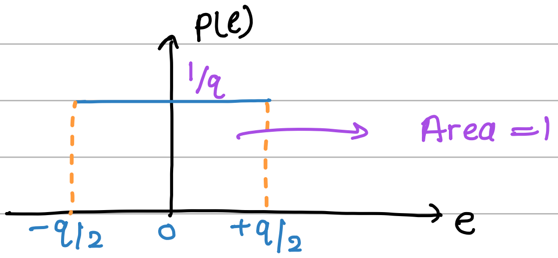
Quantization Error (e)

- A random number



$$e = x_{in} - x_q$$

Probability density function of e $\Rightarrow$



Mean of e $(\bar{e})$ $\Rightarrow$

$$\text{Mean} = \int_{-\infty}^{+\infty} e \cdot p(e)\, de \qquad = 0$$

$$= \int_{-q/2}^{+q/2} e \cdot \frac{1}{q}\, de$$

$$= \left.\frac{e^2}{2q}\right|_{-q/2}^{+q/2}$$

$$= \frac{1}{2q}\left(\frac{q^2}{4} - \frac{q^2}{4}\right)$$

$$= 0$$

$$\bar{e} = 0$$

Variance of $e\ (\sigma_e^2) \Rightarrow$

$$\sigma_e^2 = \int_{-\infty}^{\infty} (e-\bar{e})^2\ p(e)\ de$$

$$= \int_{-q/2}^{q/2} e^2\ (1/q)\ de$$

$$= \left.\frac{e^3}{3q}\right|_{-q/2}^{q/2}$$

$$= \frac{1}{3q}\left(\frac{q^3}{8} + \frac{q^3}{8}\right)$$

$$= \frac{q^2}{12} \longrightarrow \text{Quantization noise power} = \frac{q^2}{12}$$

SNR — Signal to Noise Ratio → Gives an idea about quality of a signal

SQNR — Signal to Quantization Ratio

$$SQNR = 10 \log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right)$$

eg: Consider a sinusoidal signal.

$$\sigma_x = \frac{V}{\sqrt{2}} \rightarrow \text{RMS value}$$

$$\sigma_x^2 = \frac{V_m^2}{2}$$

$$\sigma_e^2 = \frac{q^2}{12}$$

But $q = \dfrac{2V_m}{2^N}$

$$\sigma_e^2 = \frac{\left(\dfrac{V_m}{2^{N-1}}\right)^2}{12}$$

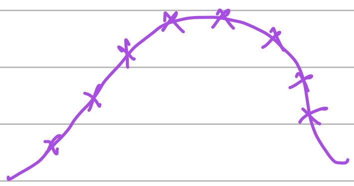
$$\frac{\sigma_x^2}{\sigma_e^2} = \frac{V_m^2/2}{\dfrac{V_m^2}{(2^{N-1})^2 \times 12}}$$

$$= 6 \times 2^{2N-2}$$

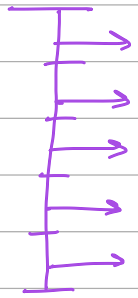
$$SQNR = 10 \log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right)$$
$$= 10 \log_{10}\left(6 \times 2^{2N-2}\right)$$
$$= 10 \log_{10}\left(1.5 \times 2^{2N}\right)$$
$$= 10 \left[\log 2^{2N} + \log 1.5\right]$$
$$= 10 \left[2N \log 2 + \log 1.5\right]$$
$$= 6.02N + 1.76 \longrightarrow$$ SQNR of a sinusoidal signal

* SQNR (even for any signal) linearly increases with the number of bits. (Quality increases)

Sampling

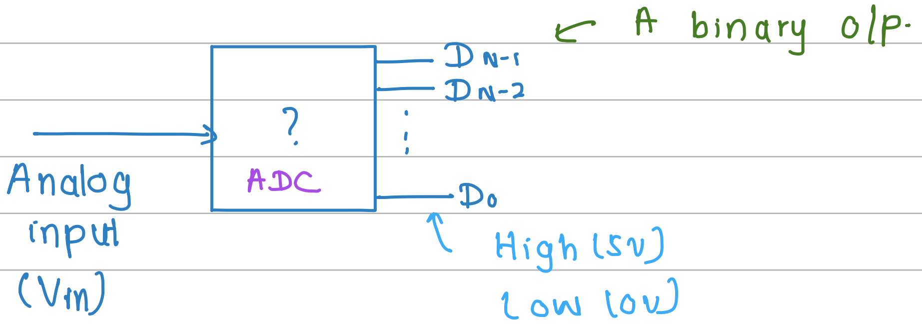


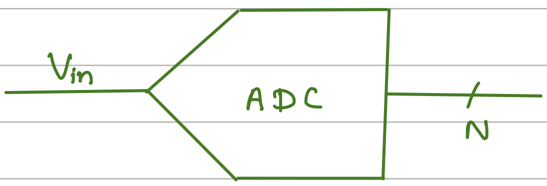
Quantization

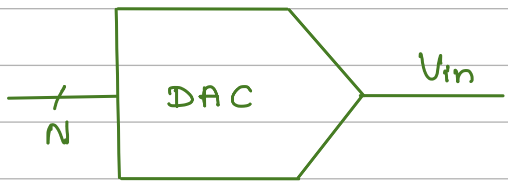


Analog to Digital Conversion (ADC)

- How to do quantization electronically.

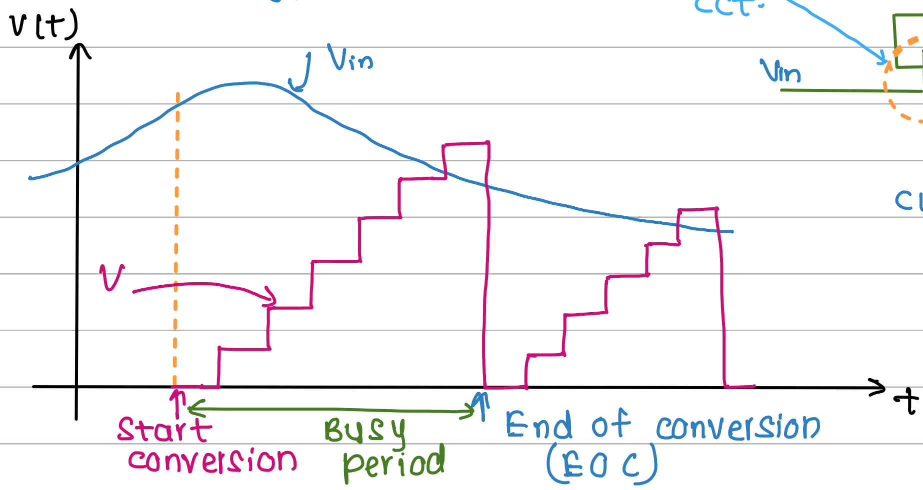


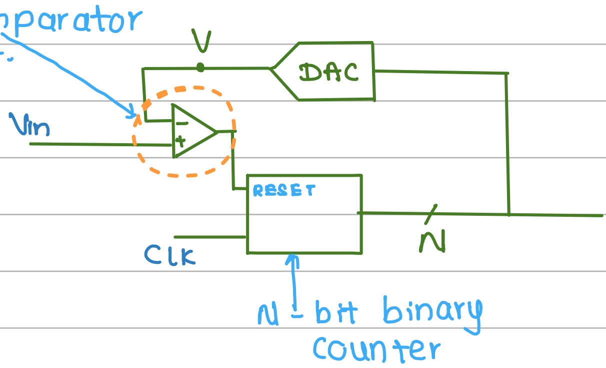




Digital to Analog Conversion.

① Counter Type ADC





- Gives an output binary value propotional to V_{in}.

Advantages

- Simple circuit

Disadvantages

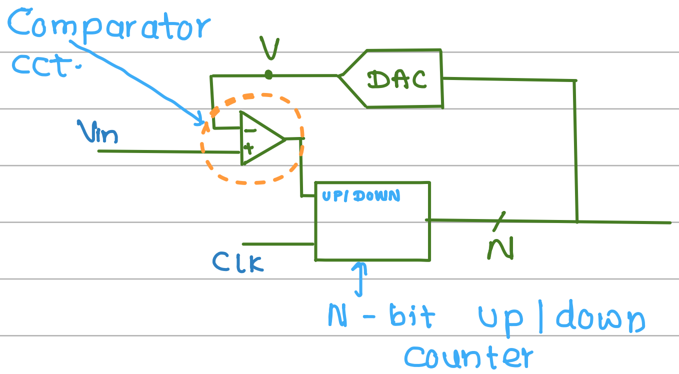
- Conversion time is high.

Due to this value of V_{in} recorded at the end of conversion is different from value of V_{in} at start of conversion.

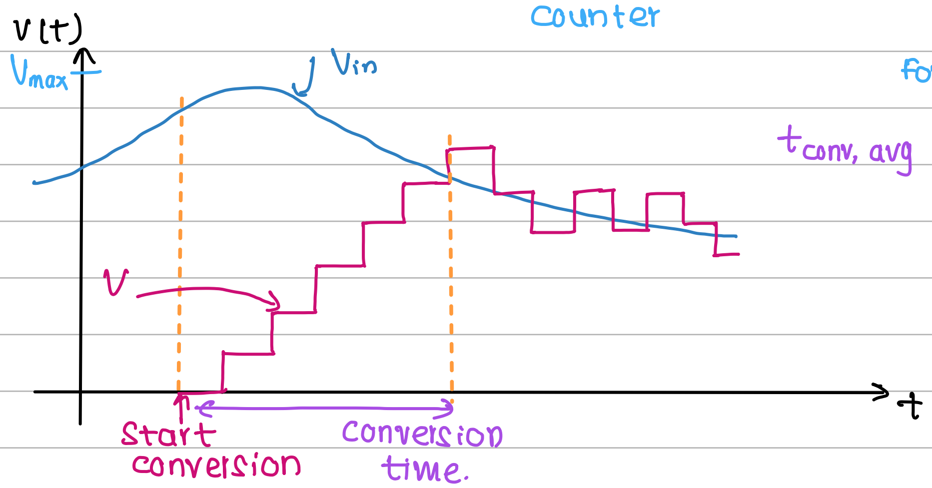
- Not tracking V_{in}
  Goes to zero at end of conversion, instead of tracking the value of V_{in}.

Tracking Type ADC

Modify this circuit, so that the ADC output tracks the input V_{in}.



Yippie!!!



$$t_{conv,\,avg} = \frac{\overset{\text{for } V = V_{in}}{0} + \overset{\text{for } V_{in} = V_{max}}{2^N} \cdot t_{clk}}{2}$$

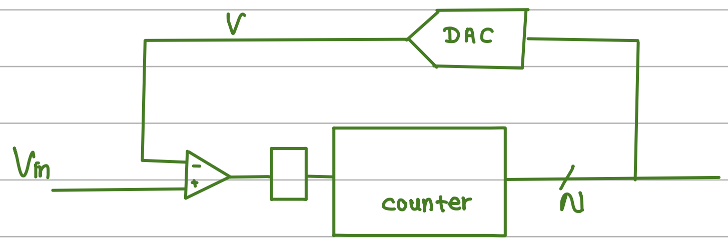
$$= 2^{N-1} \cdot t_{clk}$$

* This also has a relatively higher conversion time.

Propose a method to reduce the conversion time.

My ans: Using a counter with lower number of bits.
Decreasing clock pulse time.

③ Successive Approximation ADC



N-1	N-2	N-3		0	1
0	0	0	. . .	0	0

→ Initially

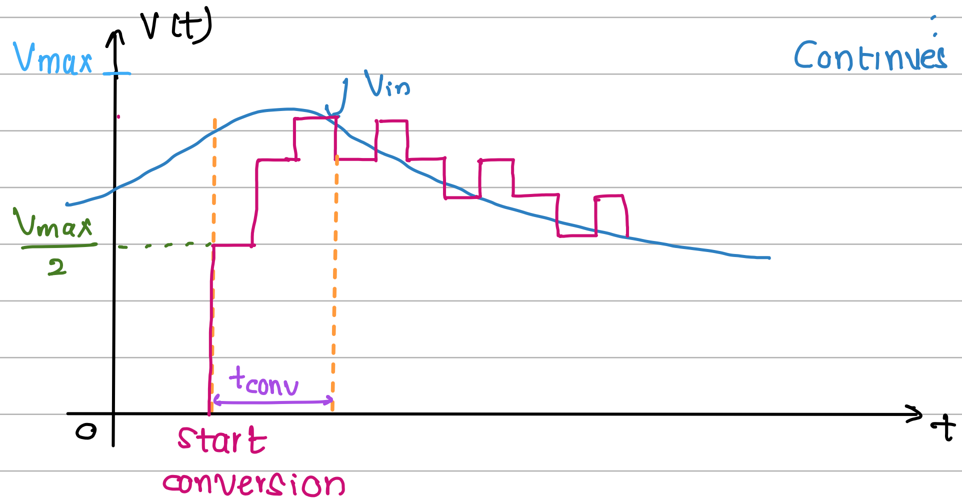
↓

N-1	N-2	N-3		0	1
1	0	0	. . .	0	0

→ First step
↓
Compare V and Vin

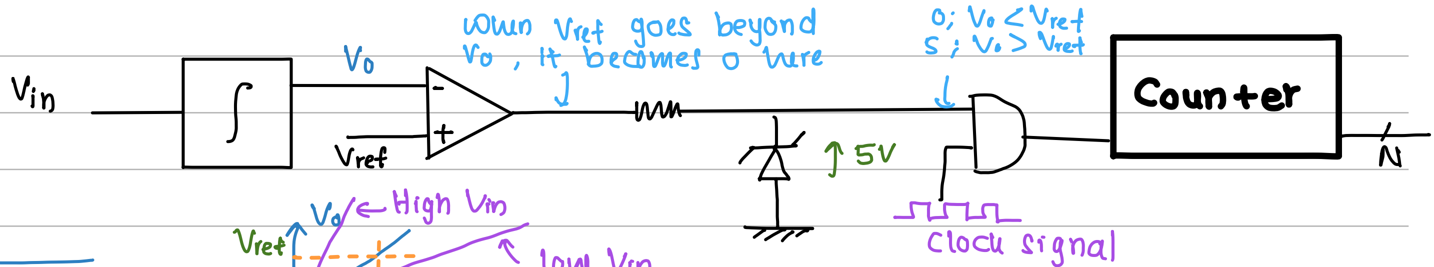
N-1	N-2	N-3		0	1
1	1	0	. . .	0	0

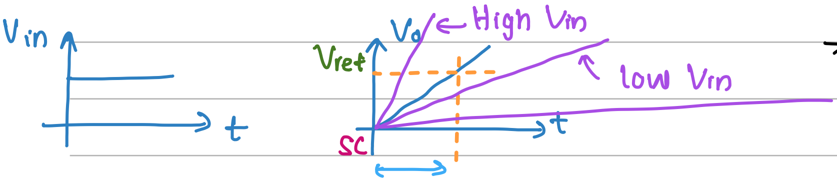
→ Secondly
↓
Again compare V and Vin
⋮
Continues until V > Vin



$t_{conv} = N \cdot t_{clk}$ → Bcz we only have to switch on N bits to get to Vmax

④ Single Slope Integrator Type ADC.

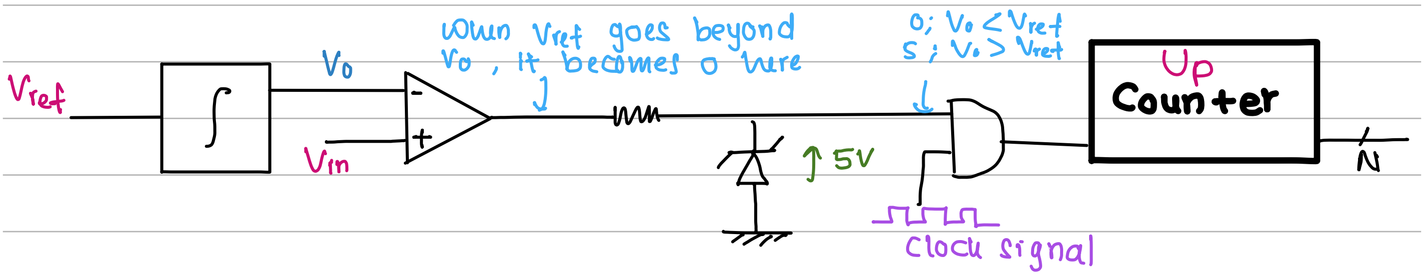


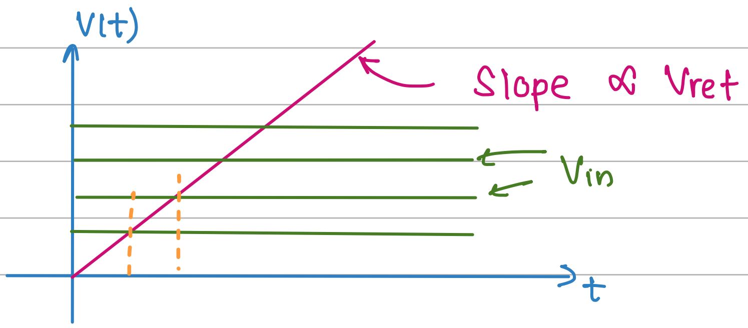


A binary down counter is used here,
→ Starts counting with the maximum value.
* Then output of counter is propotional to Vin

For very low Vin, it takes a very long time to give the output → Not practical.

To solve this,
interchange Vin and Vref



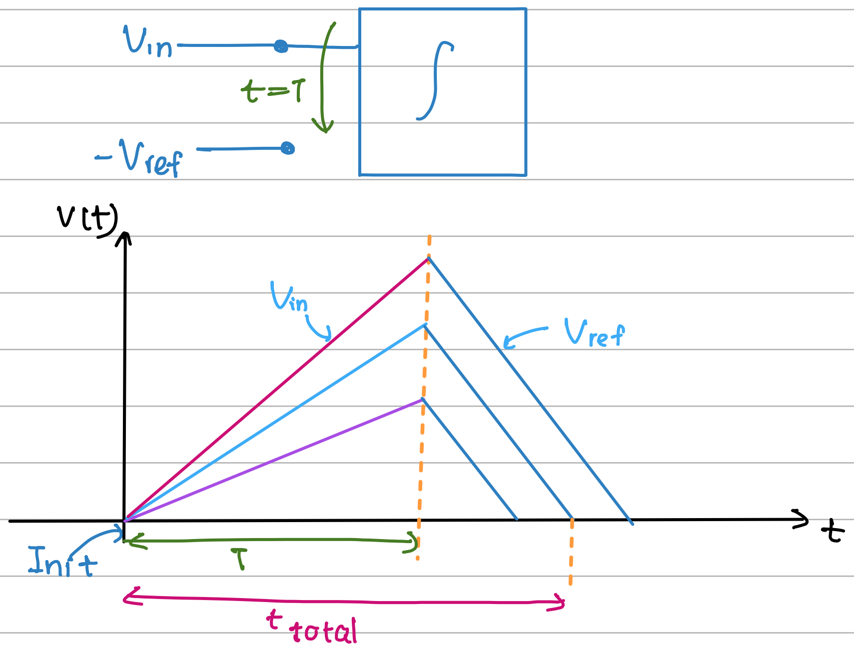


When Vin ↓ time taken to stop counting is less.
∴ an upcounter can be used directly.

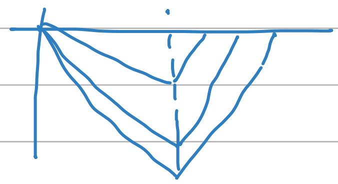
When Vin is used for the integrator cct, the integrator cct filters out the noise in Vin signal → An advantage.

⑤ Dual Slope Integrator Type ADC

- Here both noise cancellation and ability respond quickly for small Vin s are combined together



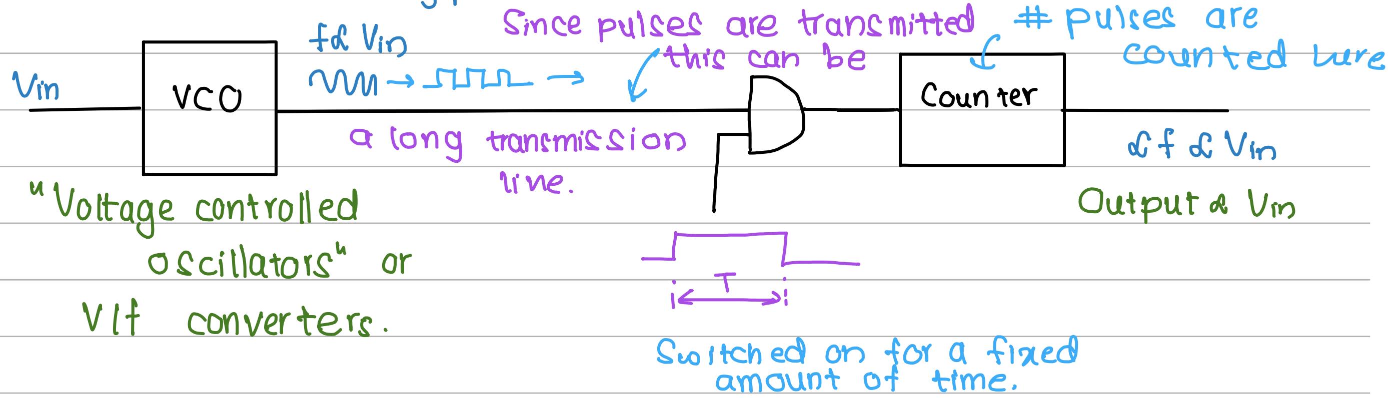
In real op-amp cct



In the negative side.

$t_{total} \propto V_{in}$

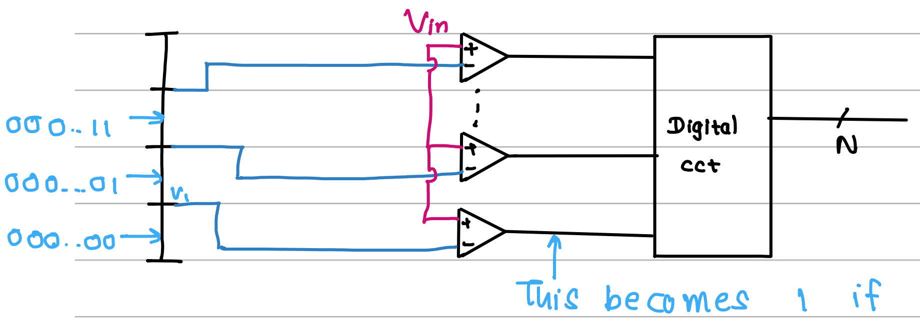
⑥ V/f Converter Type ADC



"Voltage controlled oscillators" or V/f converters.

* Pulses can be recovered much more easier than direct voltage.

⑦ Flash (Parallel) ADC



000...00
000...01
000...11
000...111

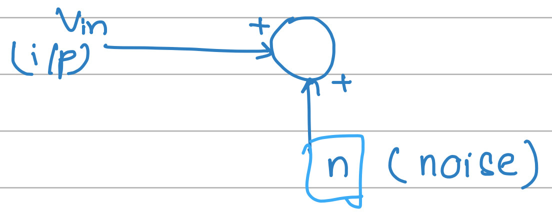
Advantage → Fast (Only the delay in the circuit components

Disadvantage → The # circuit components increase exponentially as the # bits increase.

Comparators = # levels − 1

8. Sigma - Delta ADC

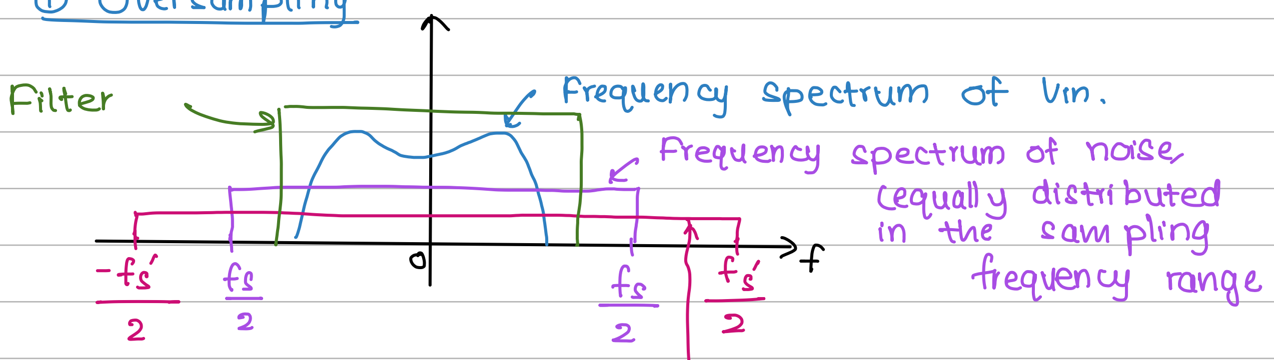
- Quantization can be modeled as noise addition to the input signal (since quantization resuls in quantization error getting added to the output)



- If we reduce n, then SQNR (Signal to Quantization Noise Ratio) increases

There are 2 techniques for this,

1. Oversampling

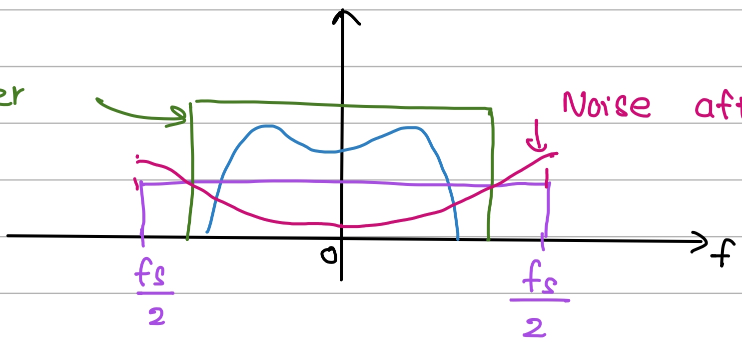


Total error distibuted equally → Frequency spectrum when fs is increased, The height of the graph decreases to keep the area constant.

Area ⇒ Power in noise

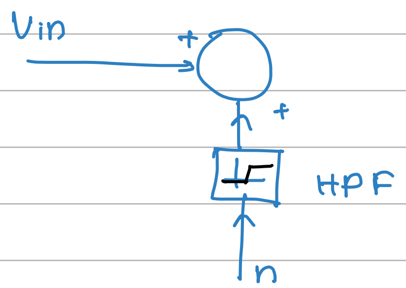
* The amount of noise power coming inside the filter has reduced when fs increased

② Noise Shaping



- SQNR improves

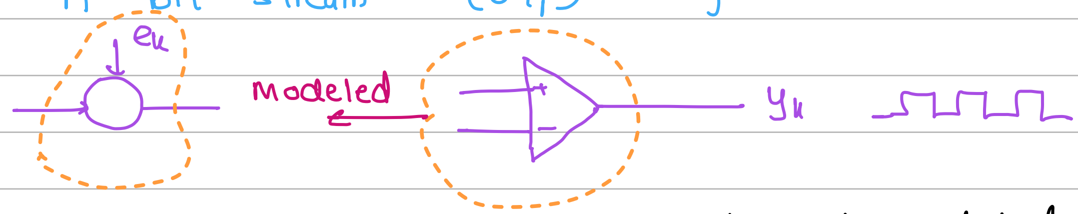
- This can be done by appying a high pass filter to noise

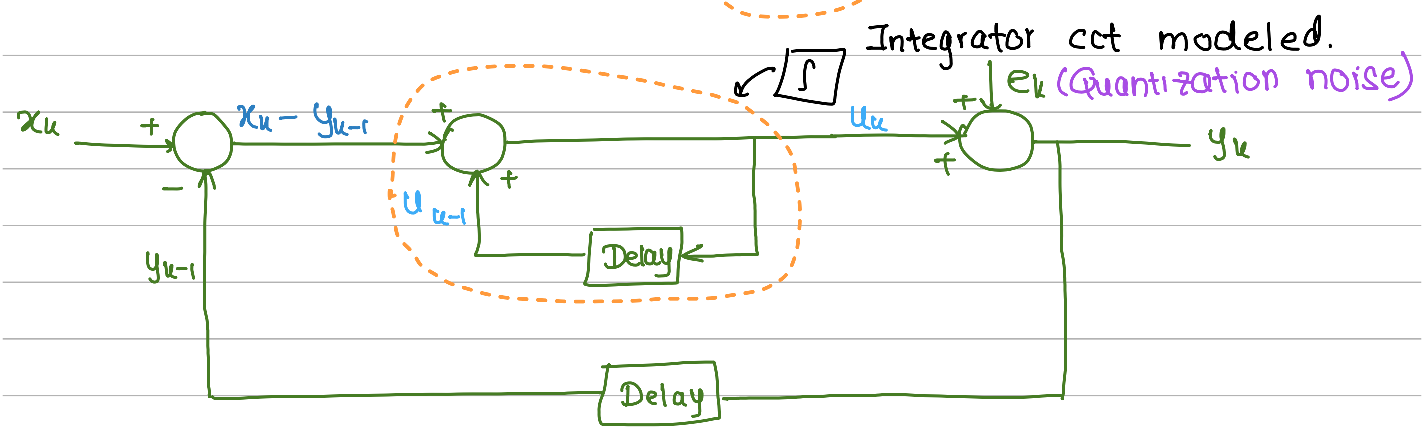


Energy in noise remains same

- Even though we model the quantization process as adding noise, actually there is no such noise source

$x_k \rightarrow$ Sequence of samples (i/p) } of ADC
$y_k \rightarrow$ A bit stream (o/p)





$$u_k = x_k - y_{k-1} + u_{k-1}$$

$$y_k = u_k + e_k$$

$$y_{k-1} = u_{k-1} + e_{k-1}$$

$$u_k = x_k - y_{k-1} + (y_{k-1} - e_{k-1})$$

$$y_k = x_k - \cancel{y_{k-1}} + \cancel{y_{k-1}} - e_{k-1} + e_k$$

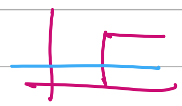
$$= x_k - e_{k-1} + e_k$$

Yippieee!!!

$$y_k = x_k + e_k - e_{k-1}$$

$$y_k = x_k + \underbrace{(e_k - e_{k-1})}_{}$$

A differentiator $\longrightarrow$ Works as a high pass filter.



Find information about different types of ACDs used in different practical applications (eg: Audio and video recorders, digital voltmeters, data acquisition cards, etc.)