Electronics Systems
LM Cyber Security – Fall 2024
Federico Baronti
This slides have been adapted from the lecture given by Prof. Roberto Saletti in the previous years

# Data conversion

**Roberto Saletti**

Sustainable Energy and Mobility Embedded Systems (SEMES) Lab.
Dip. Ingegneria dell'Informazione
University of Pisa
roberto.saletti@unipi.it

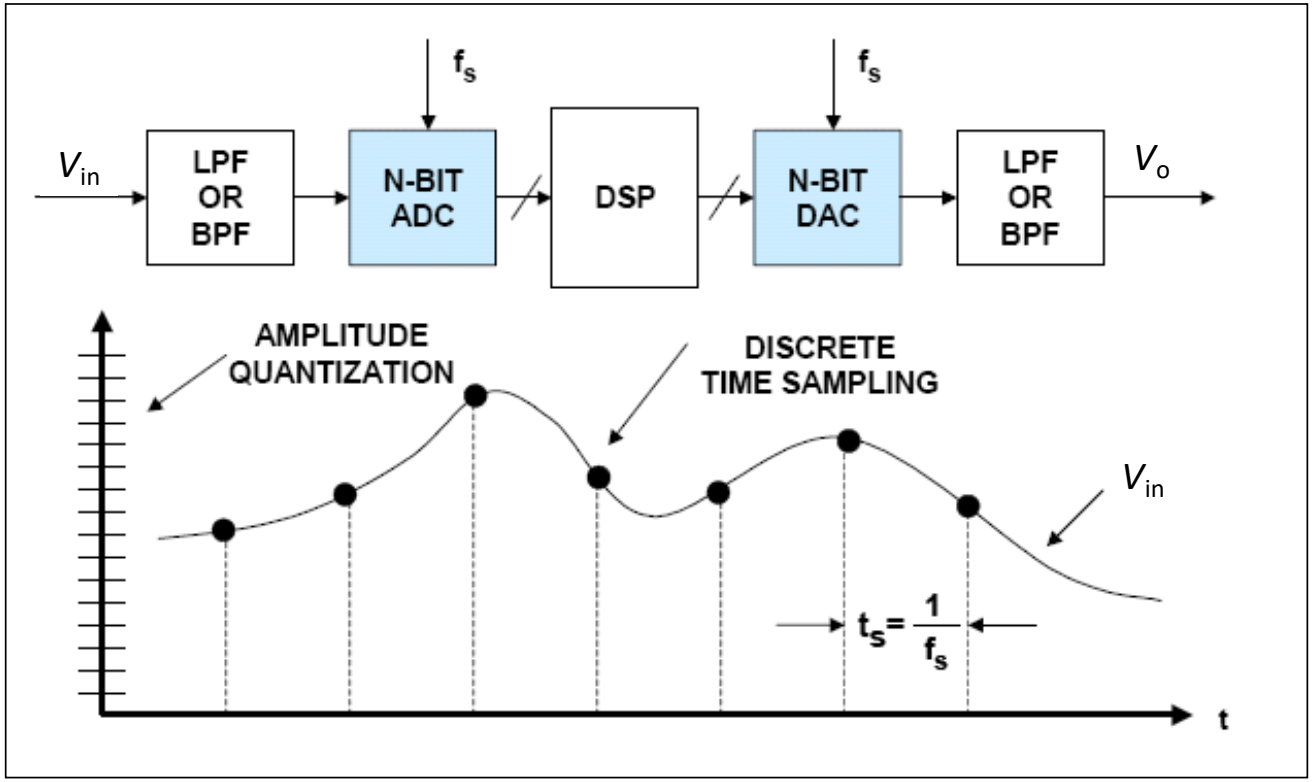

UNIVERSITÀ DI PISA

# Digital representation of signals

- How is it possible to represent analog signals with digital numbers?
  - Sampling Theorem due to Shannon
- Passage from continuous-time to discrete-time
- Analog signals can be represented and processed as sequences of numbers with no losses
- How is it possible to represent analog signals with digital numbers?
  - Quantization of the signal levels
- The numbers representing the signal samples are expressed in binary form with fixed number of digits
- A sequence of binary numbers represents the analog signal with loss

# Basic Sampling Theory

- $f_s$ is the sampling frequency



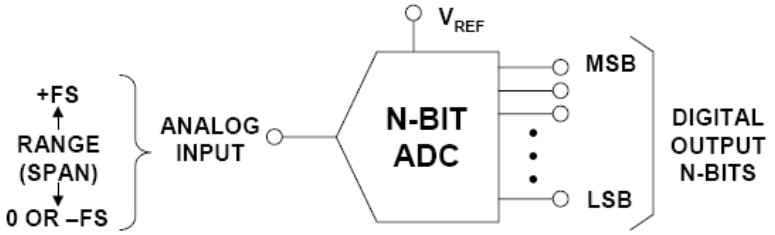
# The Nyquist Criterion

- A continuous analog signal is sampled at discrete intervals $t_s = 1/f_s$ ($f_s$ is the sampling frequency)
- $f_s$ must be carefully chosen for an accurate representation of the original analog signal
- The sampling frequency must be at least twice the highest frequency contained in the signal
- If the Nyquist Criterion is not satisfied information about the signal will be lost
- A phenomenon know as aliasing occurs when the sampling frequency is less than twice the maximum analog signal frequency

# Fundamentals of data conversion

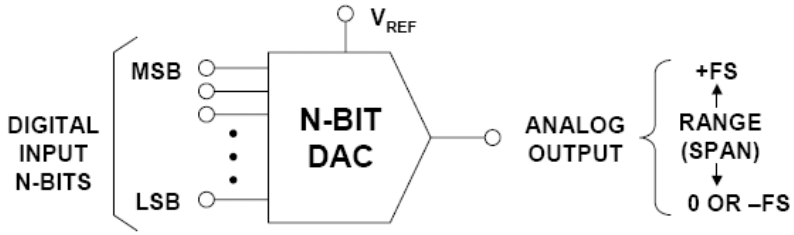
## Analog-to-Digital converters (ADCs)

- Translate “real world” analog quantities to digital language




## Digital-to-Analog converters (DACs)

- Transform the results of digital processing back to the “real world”



# Signal Quantization

- Analog signals can be represented by sequences of numbers
- Possible analog values are continuous and thus infinite
- Digital values are finite
  - 8 bit represents $2^8 = 256$ digital values
  - 16 bit represents $2^{16} = 65536$ digital values
- How we manage with that?
- Quantization (approximation) of the sampled value to the nearest digital value
- It means an error is **always** added by the quantization process

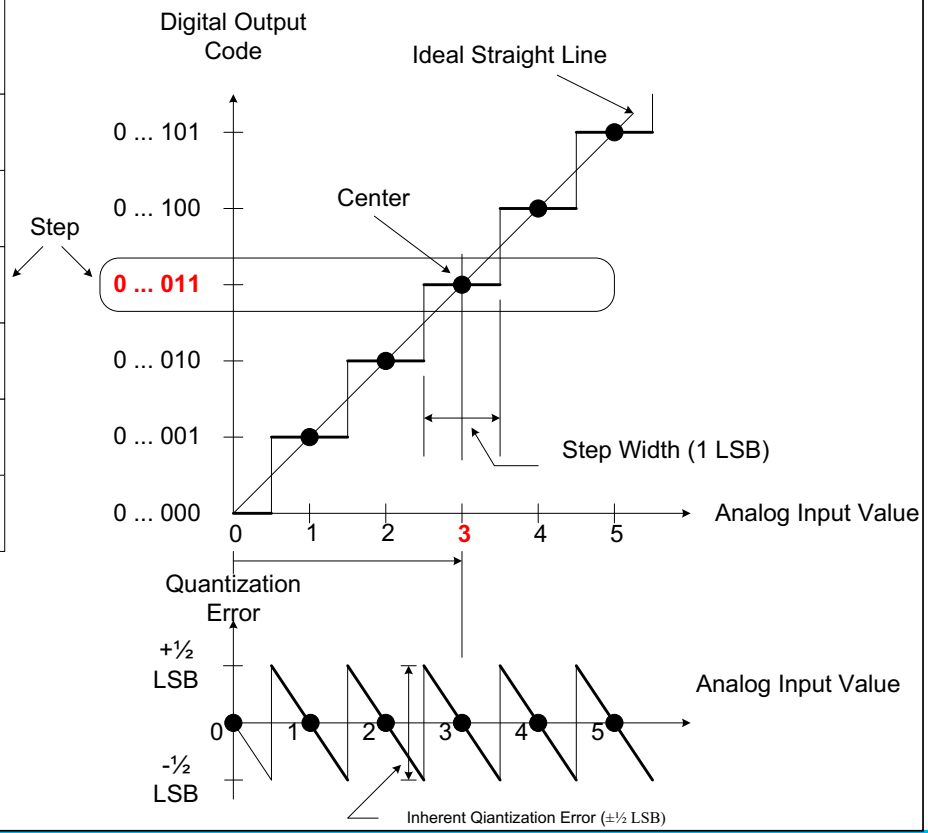
# Analog to Digital Converter (ADC)

- An electronic component that:
  - Samples the input signal at the sampling frequency
  - Provides a sequence of digital numbers as output
- The sequence of digital numbers consists of the quantized values of the samples of the input signal
- The ADC transfer function describes the relationship between analog input and digital output

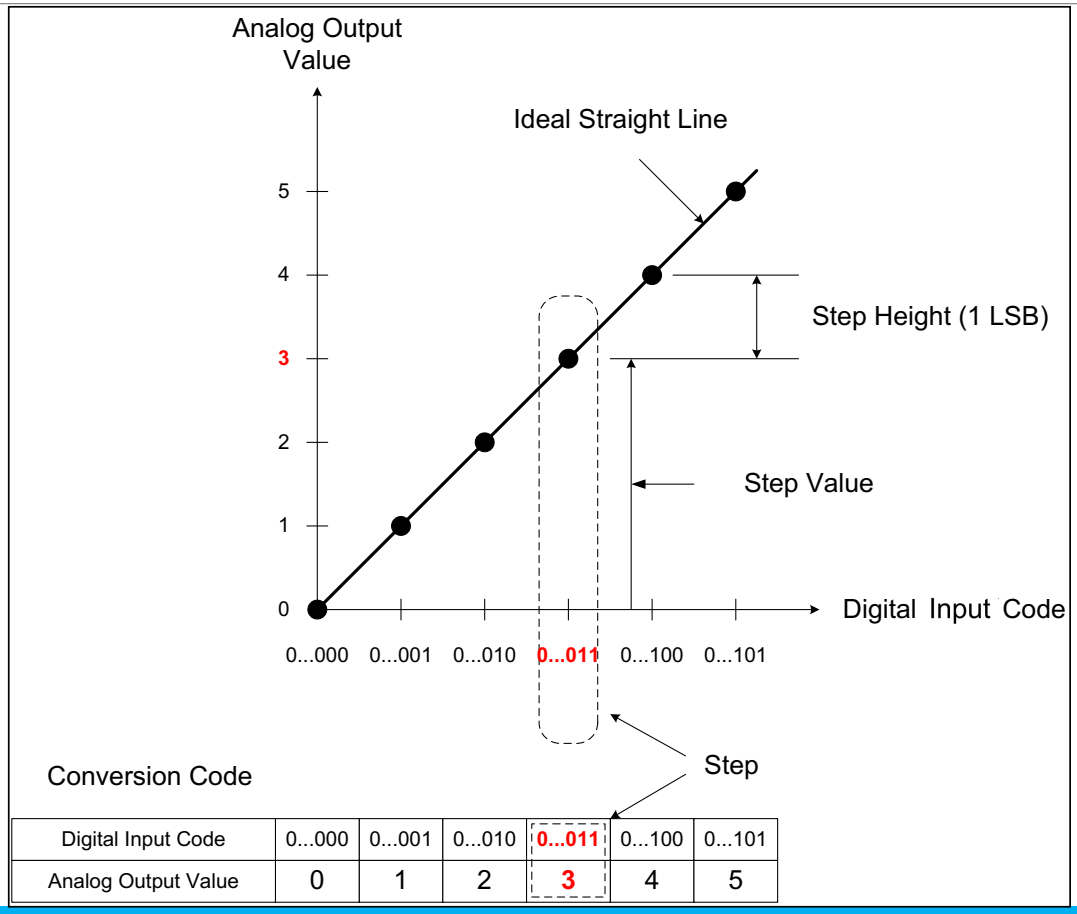
# ADC Ideal Staircase Transfer Function

CONVERSION CODE

| RANGE OF ANALOG INPUT CODE | DIGITAL OUTPUT CODE |
|---|---|
| 4.5 · 5.5 | 0 ... 101 |
| 3.5 · 4.5 | 0 ... 100 |
| **2.5 · 3.5** | **0 ... 011** |
| 1.5 · 2.5 | 0 ... 010 |
| 0.5 · 1.5 | 0 ... 001 |
| 0 · 0.5 | 0 ... 000 |



# DAC Ideal Transfer Function

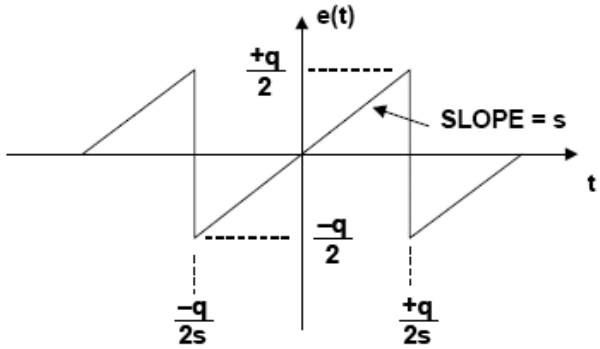



Conversion Code

| Digital Input Code | 0...000 | 0...001 | 0...010 | 0...011 | 0...100 | 0...101 |
|---|---|---|---|---|---|---|
| Analog Output Value | 0 | 1 | 2 | 3 | 4 | 5 |

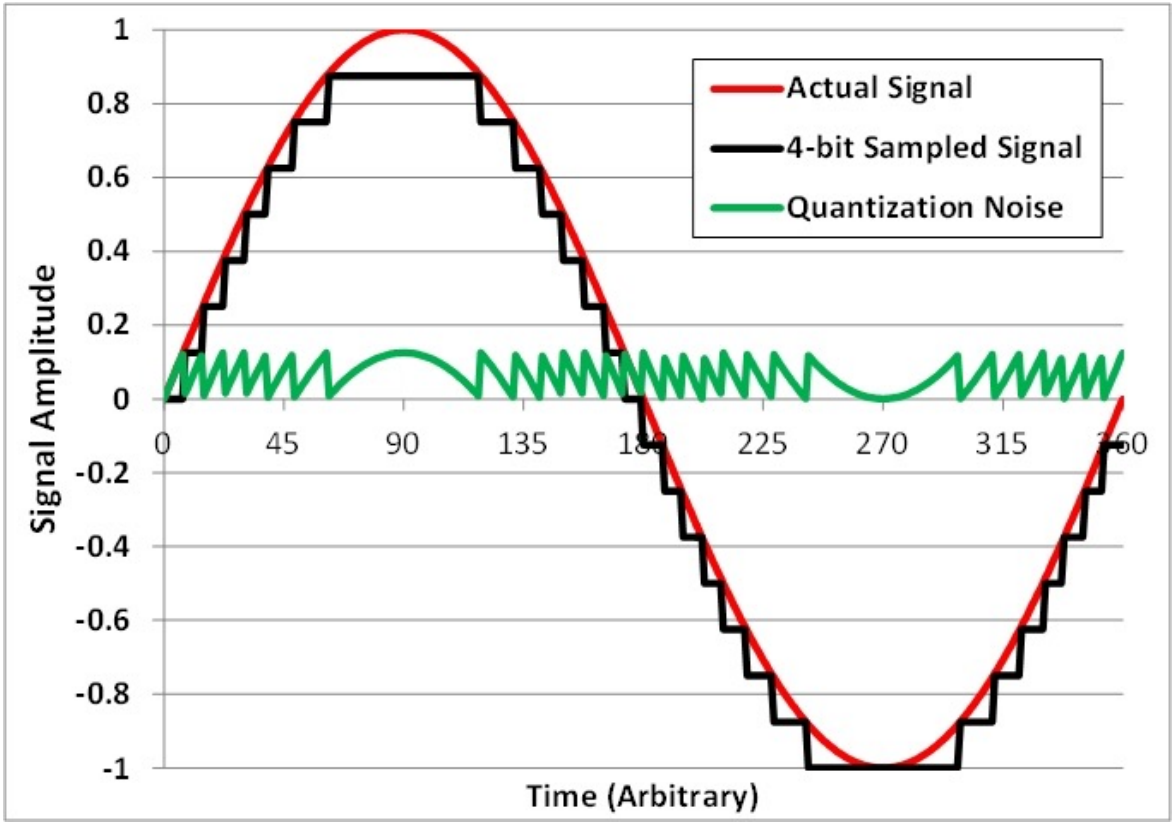
# Quantization Noise as a Function of Time

- An error is always committed in quantization
- Quantization error range is $\pm$LSB/2
- Root Mean Square error is LSB/$\sqrt{12}$
- LSB (Least Significant Bit) amplitude is fundamental
- ADC resolution = *N* (number of bit)



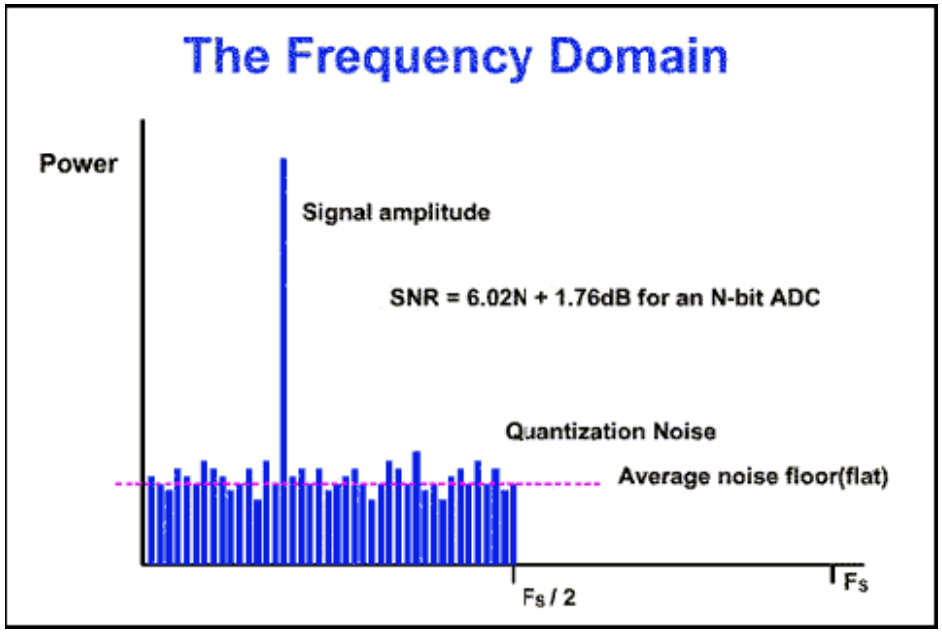

- ERROR = $e(t) = st, \quad \frac{-q}{2s} < t < \frac{+q}{2s}$
- MEAN-SQUARE ERROR = $\overline{e^2(t)} = \frac{s}{q}\int_{-q/2s}^{+q/2s} (st)^2\, dt = \frac{q^2}{12}$
- ROOT-MEAN-SQUARE ERROR = $\sqrt{\overline{e^2(t)}} = \frac{q}{\sqrt{12}}$

# Quantization error of a sine wave



# FFT of an ADC sampled and quantized sine

- This noise is approximately Gaussian
- It spans rather uniformly over the Nyquist bandwidth from 0 to $f_s/2$
- Signal-to-Noise (SNR) ratio is the ratio between the power of signal and the power of quantization noise



# Signal-to-Noise Ratio of a N-Bit Converter

- FS INPUT = $v(t) = \left[\frac{q\,2^N}{2}\right] \sin(2\pi f t)$
- RMS Value of FS Sinewave = $\frac{q\,2^N}{2\sqrt{2}}$
- RMS Value of Quantization Noise = $\frac{q}{\sqrt{12}}$
- SNR = $20 \log_{10} \left[\frac{\text{RMS Value of FS Sinewave}}{\text{RMS Value of Quantization Noise}}\right] = 20 \log_{10} 2^N + 20 \log_{10} \sqrt{\frac{3}{2}}$

$$SNR = 6.02N + 1.76dB$$

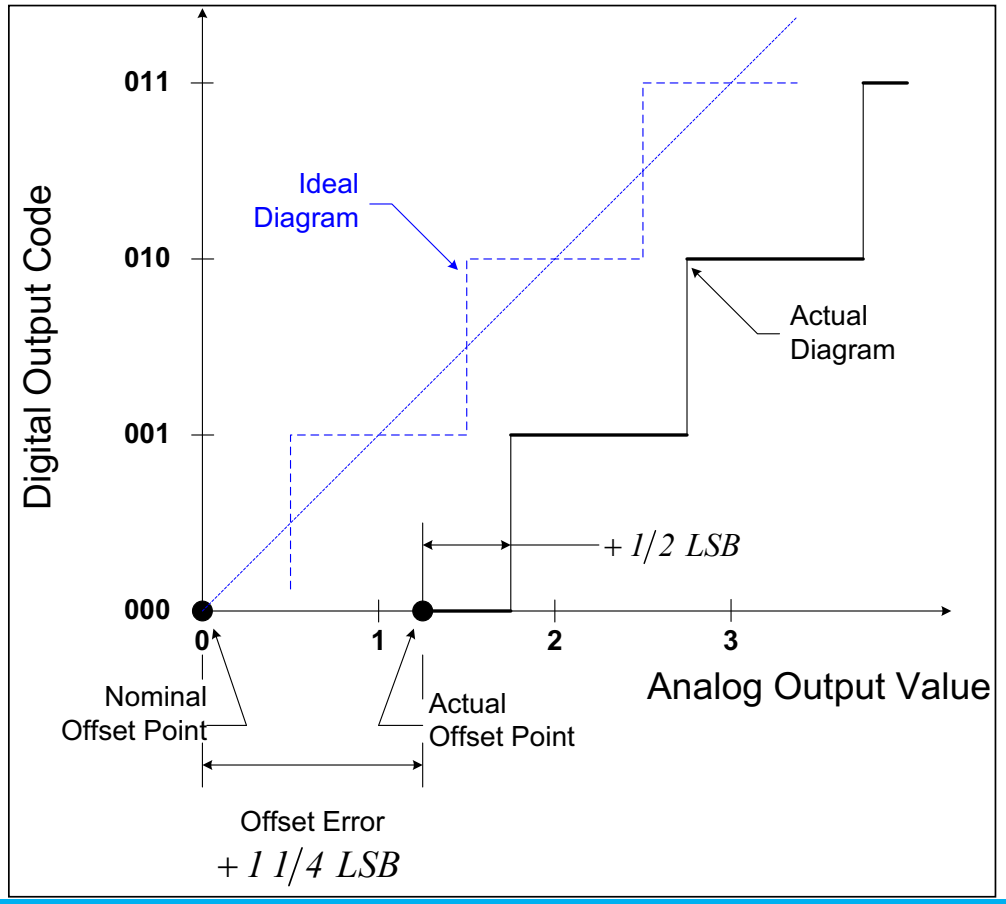
(Measured over the Nyquist Bandwidth : DC to $f_S/2$)

# Other sources of error

- Quantization noise is unavoidable
  - Can be reduced increasing the ADC bit number *N* (ADC resolution)
- Further errors in real converters
  - offset error
  - gain error
  - integral nonlinearity (INL)
  - differential nonlinearity (DNL)
- Usually expressed in LSB units or as a percentage of the FSR (Full Scale Range). E.g.
  - INL = 1.2 LSB
  - INL = 1% FSR

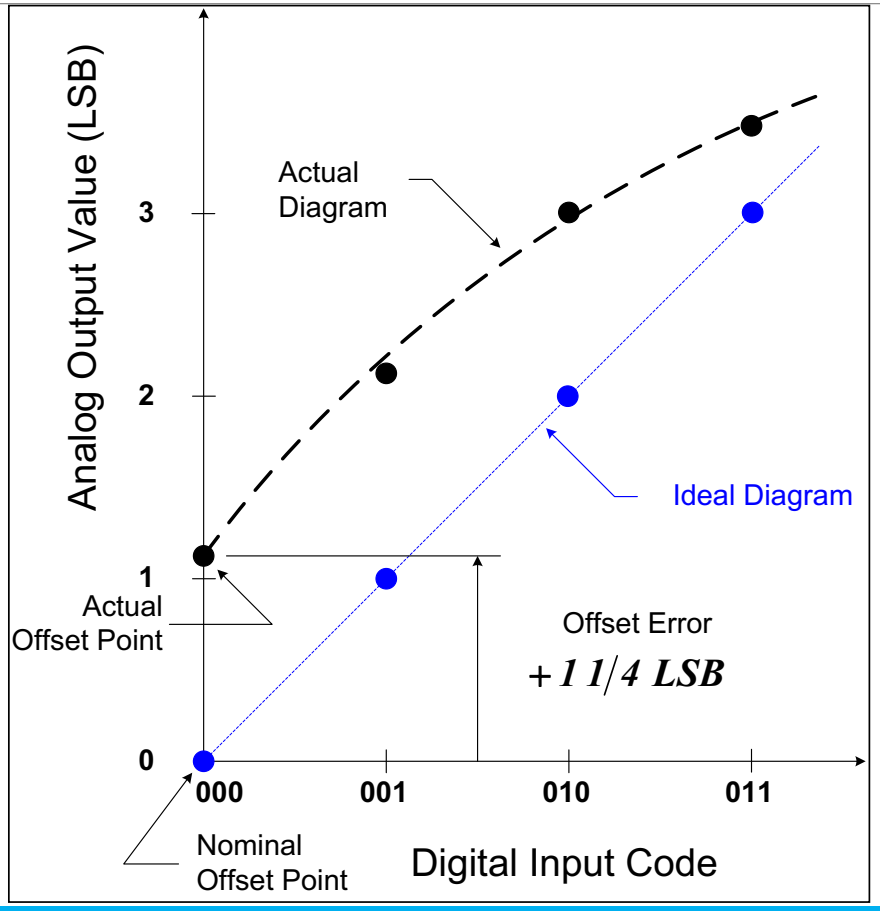
# ADC Offset Error

- The ideal stair is translated
- Difference between the nominal and actual offset points



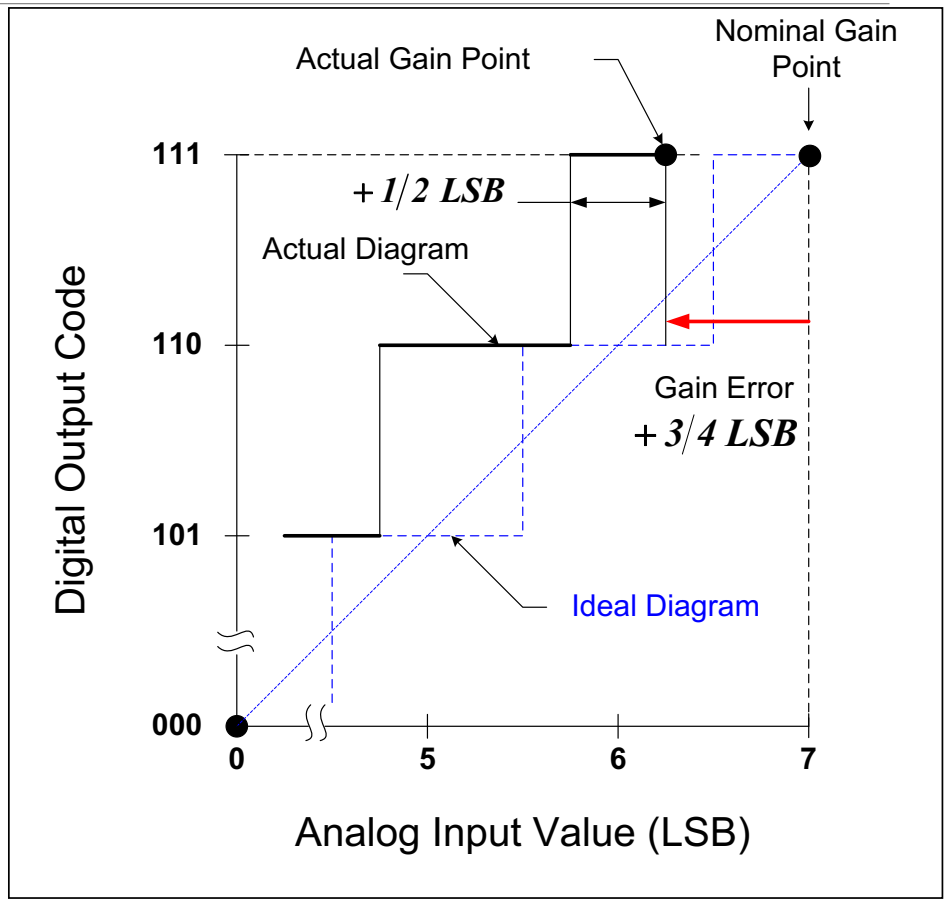
# DAC Offset Error

- The ideal curve is translated
- The offset error is the step value when the digital input is zero



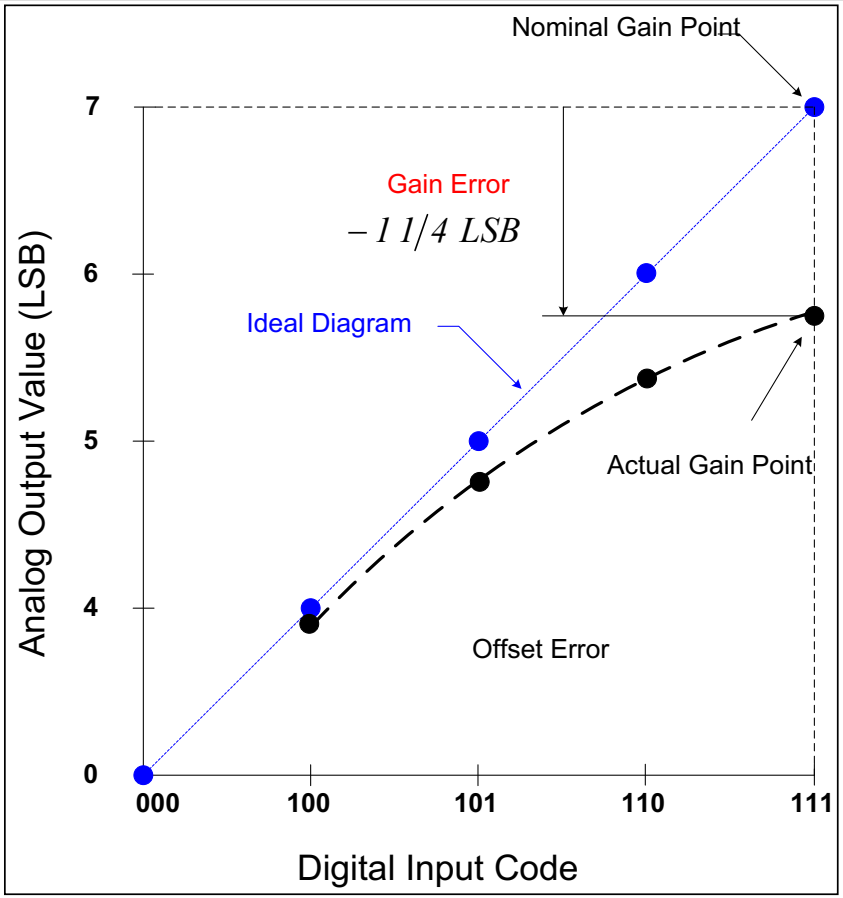
# ADC Gain Error

- Gain error is defined as the difference between the nominal and actual gain points after the offset error has been corrected to zero
- The gain point is the midstep value when the digital output is full scale



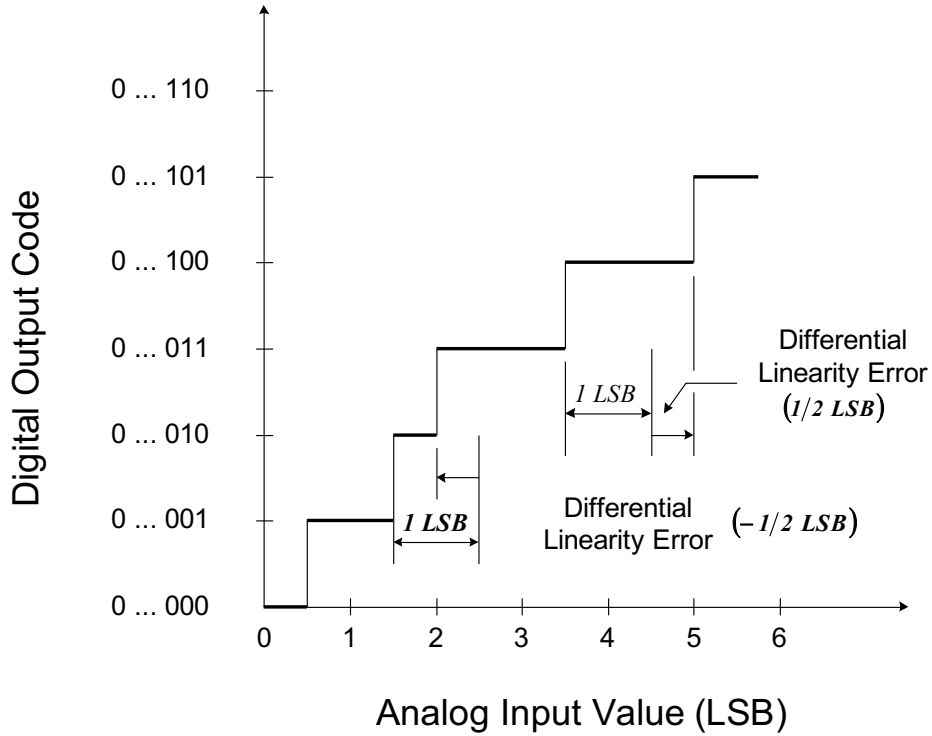
# DAC Gain Error

- DAC Gain Error is the step value when the digital input is full scale
- It represents the difference in the slope of the actual and ideal transfer functions



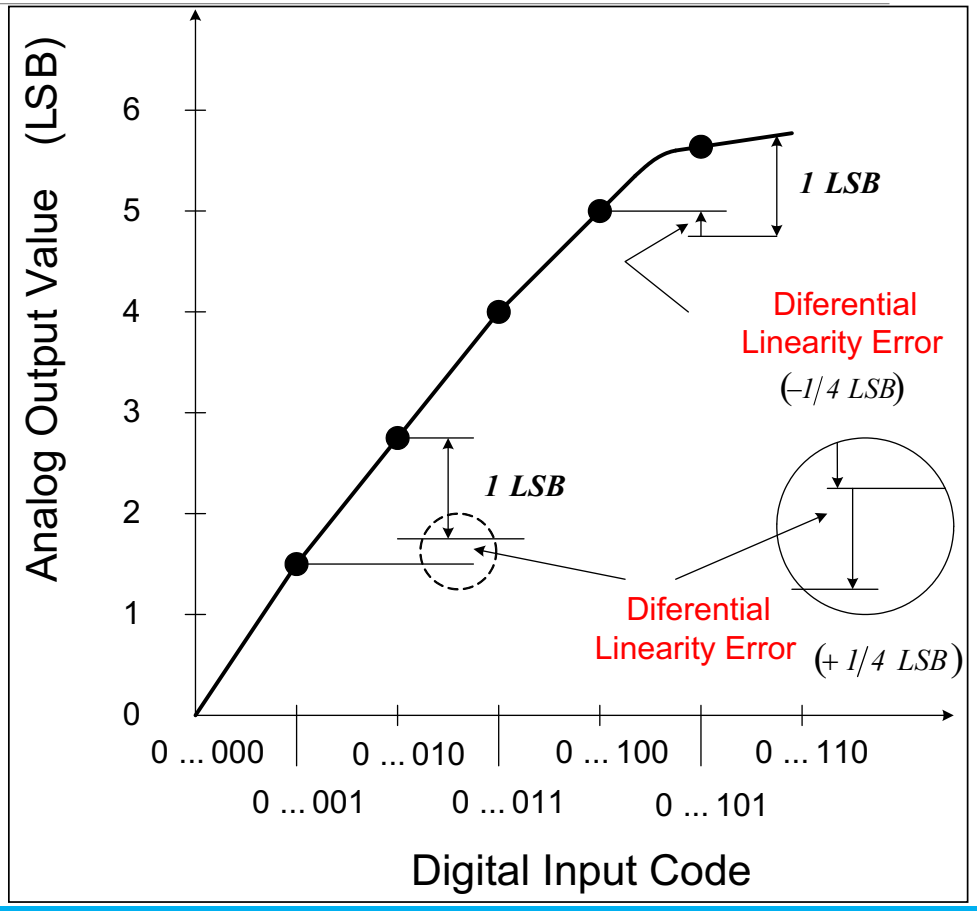
# ADC Differential Nonlinearity (DNL) Error

- DNL is the difference between an actual step width and the ideal value of 1 LSB
- DNL > +1 LSB
  - nonmonotonic (the magnitude of the output gets smaller for an increase in the magnitude of the input)
- DNL < -1 LSB
  - missing codes (one or more of the possible 2N binary codes are never output



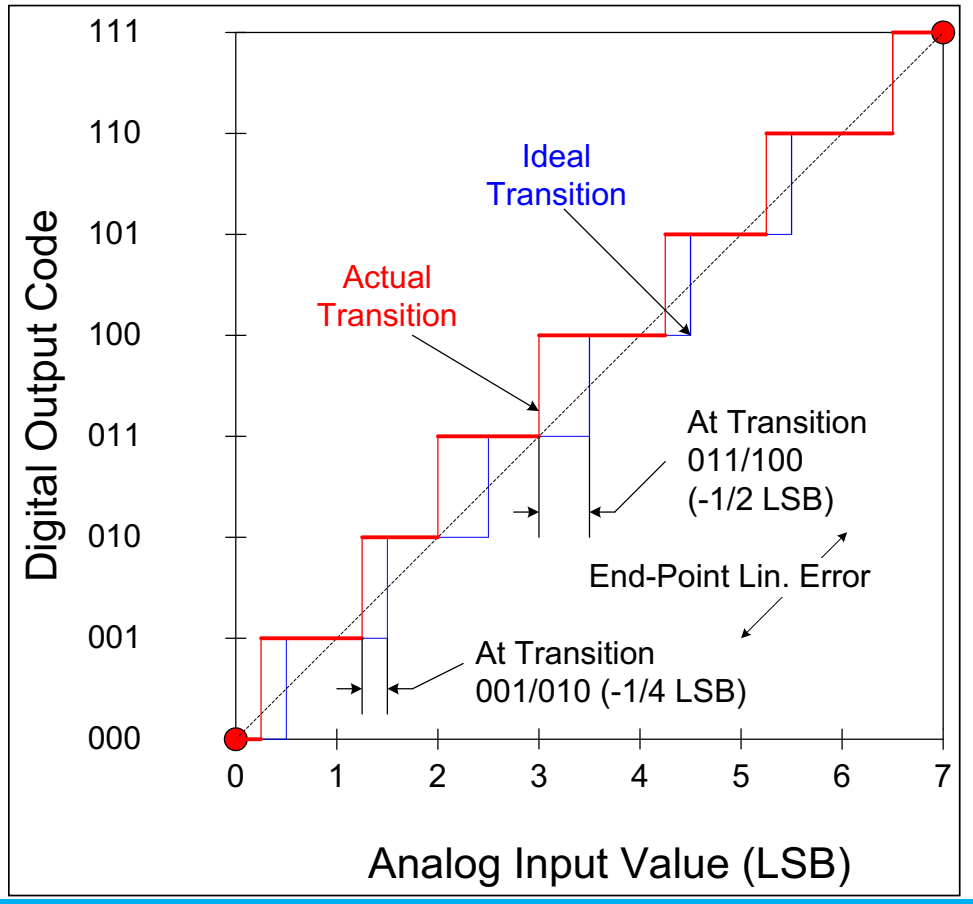
# DAC Differential Nonlinearity (DNL) Error

- DNL error is the difference between an actual step height and the ideal value of 1 LSB
- If the step height is exactly 1 LSB, the DNL error is zero

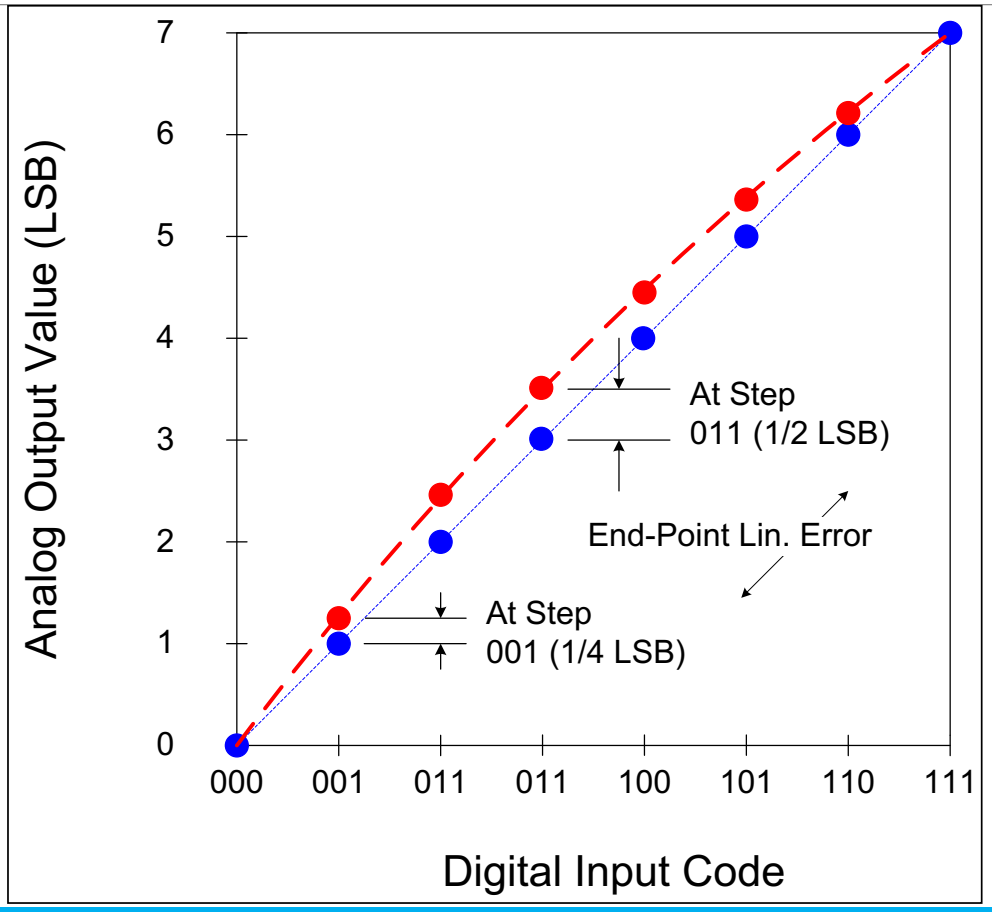


# ADC Integral Nonlinearity (INL) Error

- INL is the deviation of the values on the actual transfer function from a straight line
- This straight line can be
  - best straight line (drawn to minimize these deviations)
  - end points line (first and last points of the transfer function after gain and offset errors are cancelled)



# DAC Integral Nonlinearity (INL) Error



# Dynamic Range of ADCs

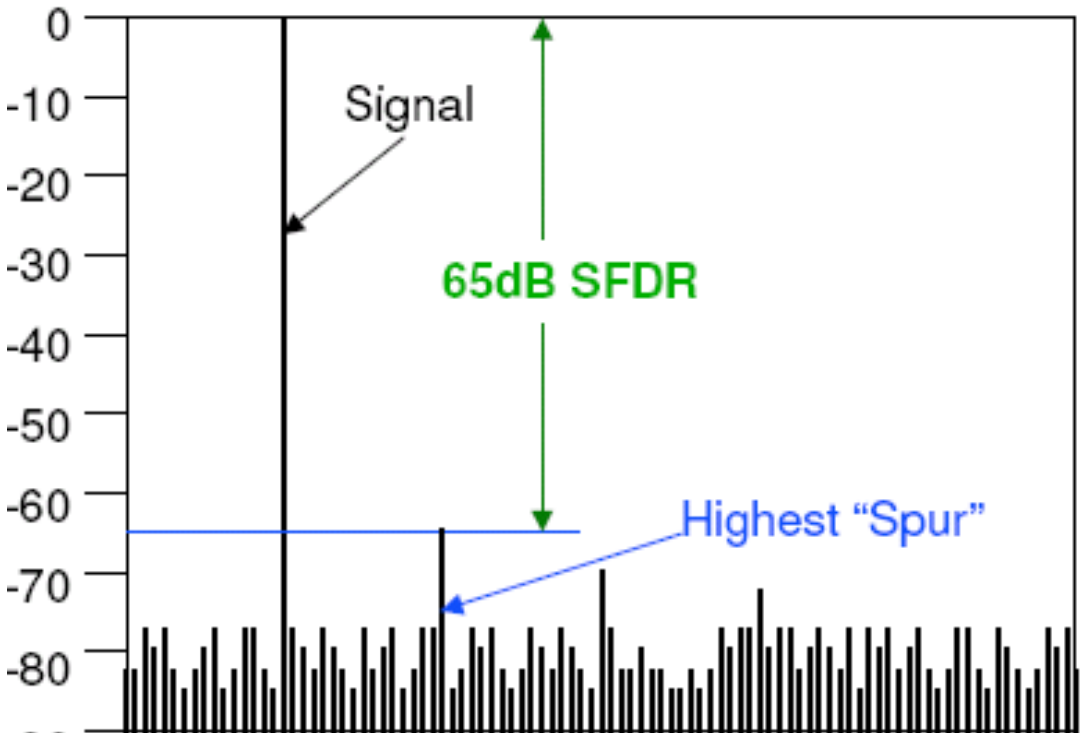
Dynamic Range is the ratio of the largest to the smallest possible signals that can be resolved. DO NOT confuse with Spurious Free Dynamic Range (SFDR).

| Resolution (Bits) | Dynamic Range (dB) |
|---|---|
| 6 | 36.0 |
| 8 | 48.1 |
| 10 | 60.2 |
| 12 | 72.2 |
| 14 | 84.3 |
| 16 | 96.3 |
| 18 | 108.4 |
| 20 | 120.4 |

$$\text{Dynamic Range} = 20 * \text{Log}(2^n - 1)$$

# Spurious Free Dynamic Range (SFDR)

- SFDR of an ADC is the ratio of the rms signal amplitude to the rms value of the peak spurious spectral content measured over the bandwidth of interest



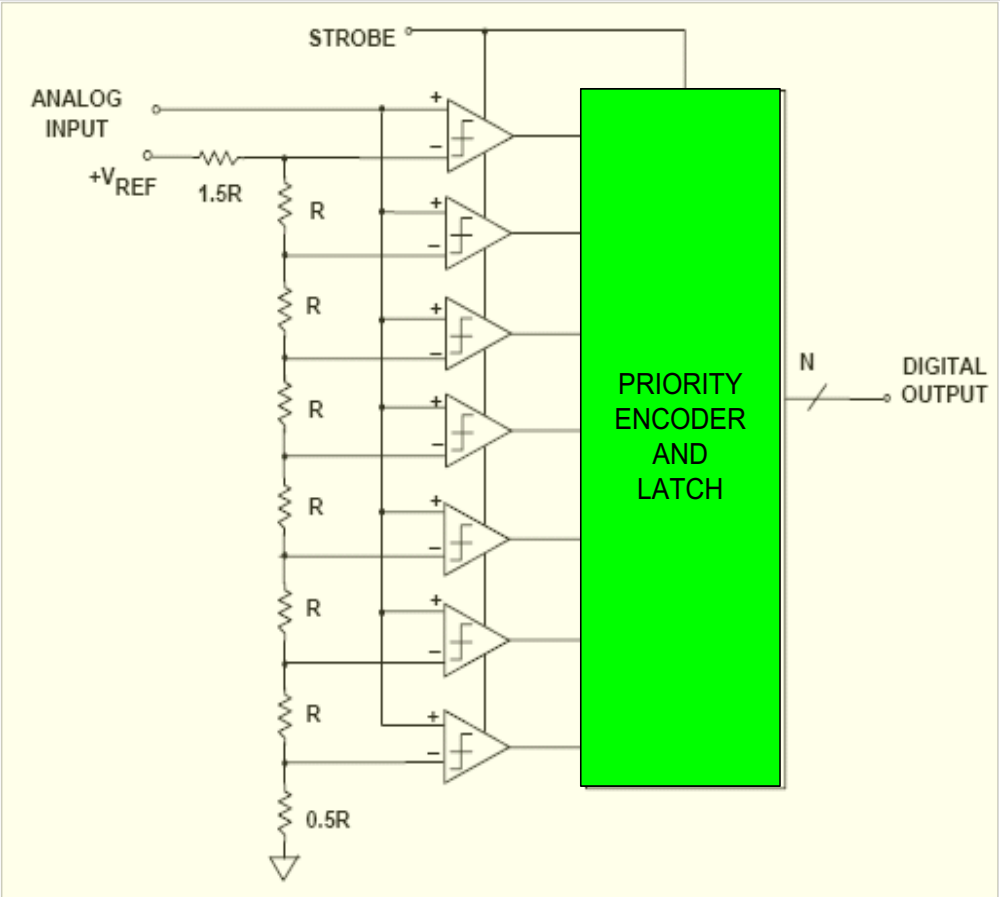
# ADC Architectures

- Flash Converters
- Successive Approximation ADCs
- Pipelined ADCs
- Integrating ADC
- Sigma-Delta ADC

# Flash Converter Features

- Also known as parallel ADCs
- The fastest way to convert an analog signal to a digital signal
- Features
  - very large bandwidths
  - high power consumption
  - relatively low resolution
  - quite expensive
- An $N$-bit flash ADC consists of $2^N$ resistors and $2^N-1$ comparators

# Flash Converter architecture

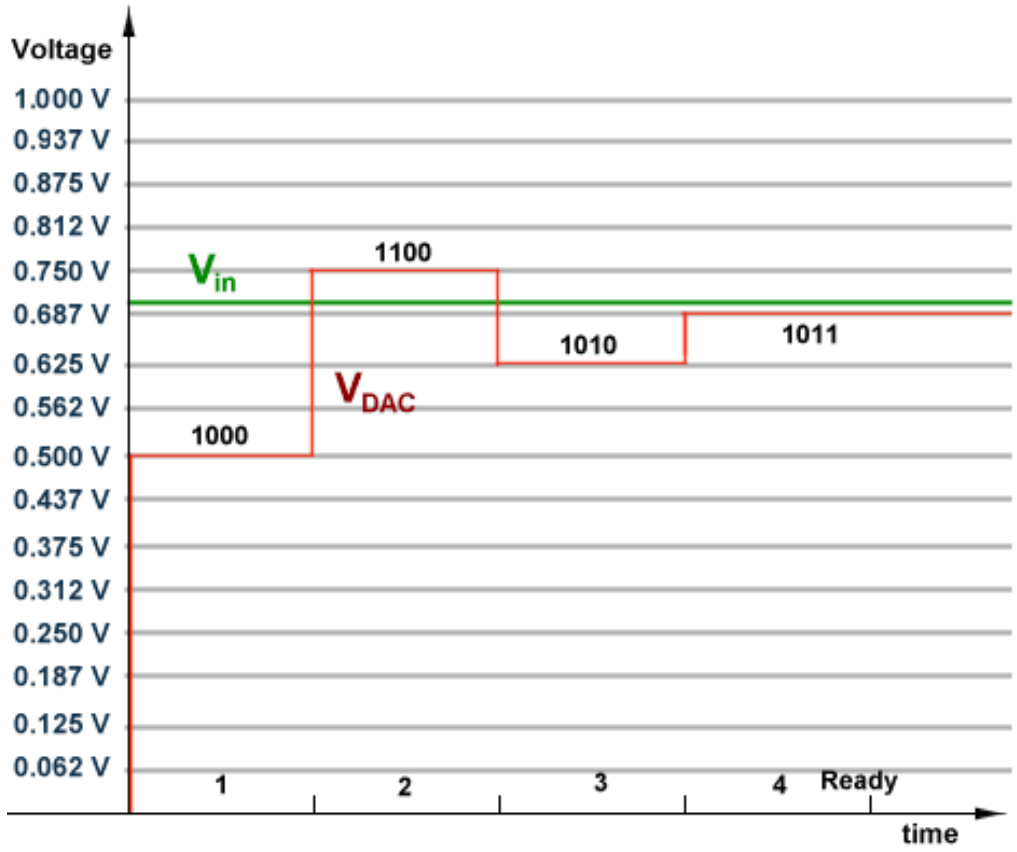


# Successive-Approximation ADCs (SAR)

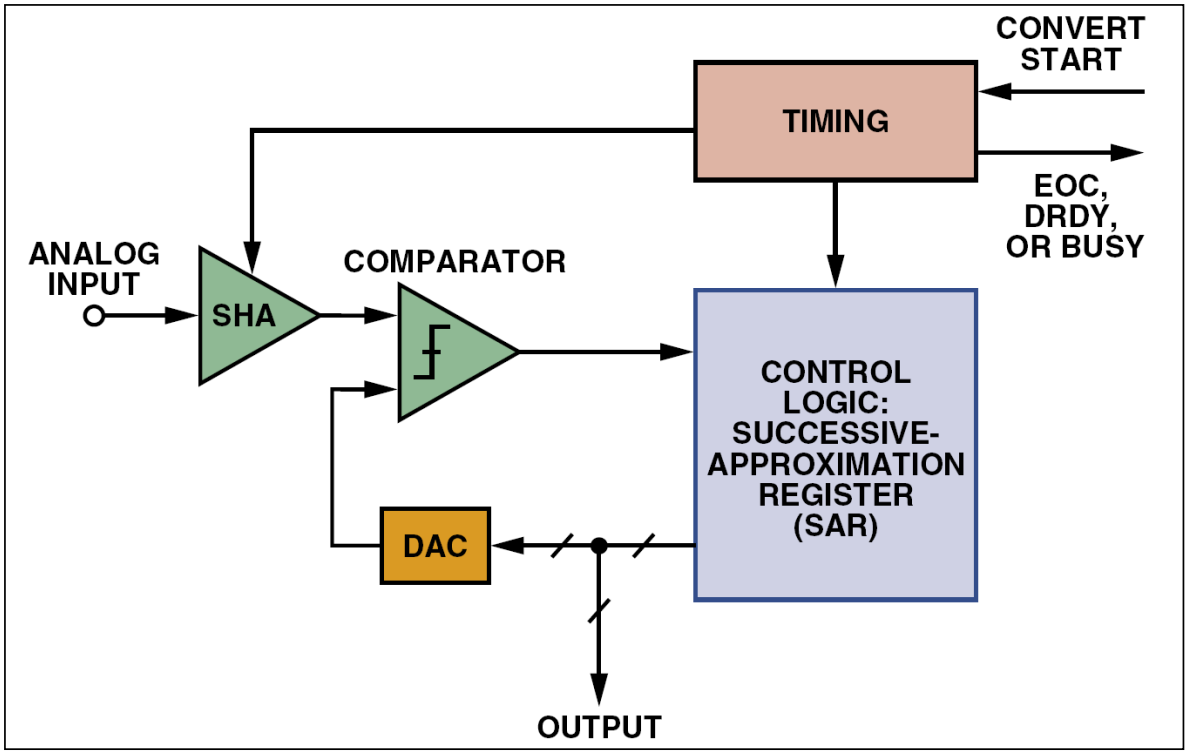
- The most popular architecture for data-acquisition applications
- Commercial SAR ADCs feature resolutions from 8 bits to 18 bits, with sampling rates up to several MHz
- Output data are generally provided via a standard serial interface ($I^2C$ or SPI), but some devices are available with parallel outputs
- SAR ADCs are typically provided with a sample-and-hold (SHA) to keep the input signal constant during the conversion cycle

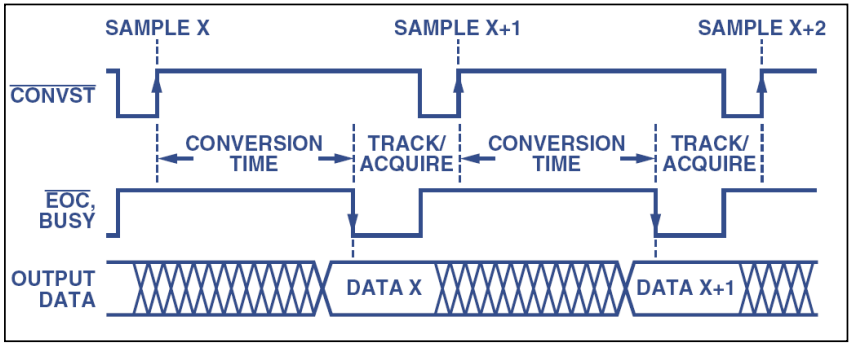
# Basic idea behind SAR ADC

- Input value is approached with approximation steps in which one bit per step is decided
- Conversion is final after *N* steps where *N* is the ADC bit resolution



# SAR Architecture



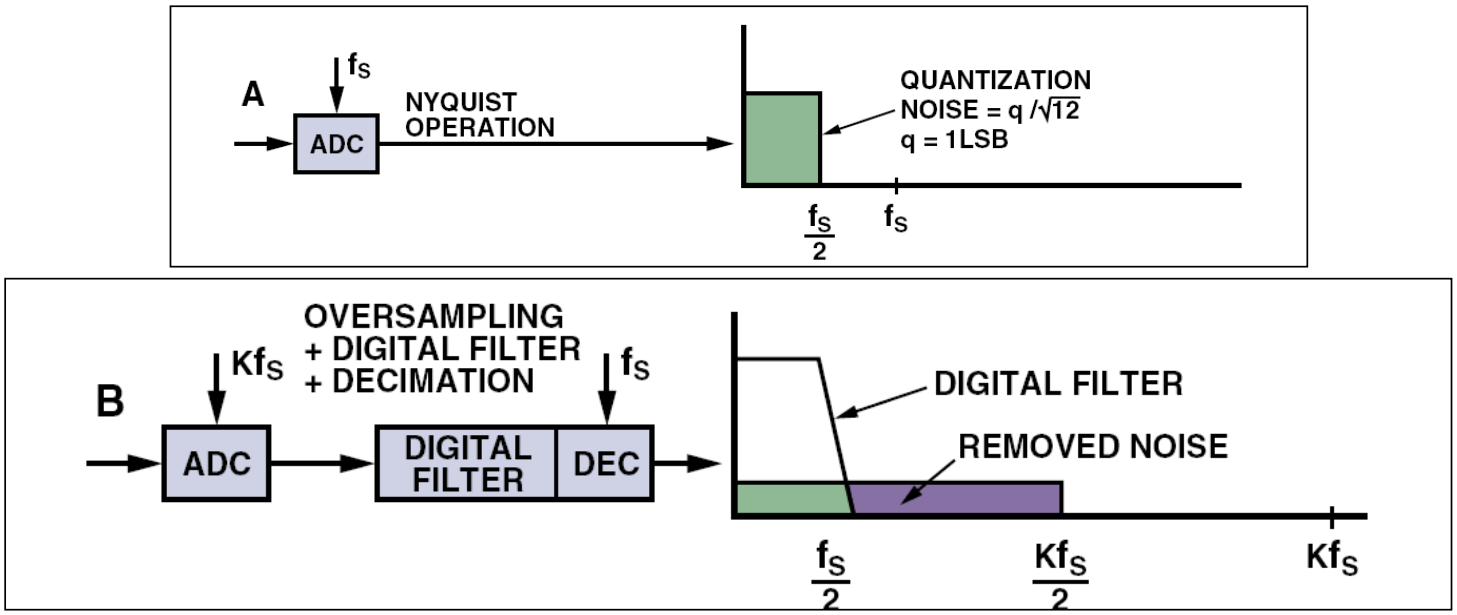

The overall accuracy and linearity of the SAR ADC are determined primarily by the internal DAC's characteristics

# SAR Operation Algorithm

- Binary-tree-research alike conversion algorithm
- The internal D/A converter (DAC) is set to midscale
- The comparator determines whether the input is greater or less than the DAC output, and the result (the most-significant bit (MSB) of the conversion) is stored in the successive-approximation register (SAR) as a 1 or a 0
- The DAC is then set either to 1⁄4 scale or 3⁄4 scale (depending on the value of the MSB), and the comparator makes the decision for the second bit of the conversion
- The result (1 or 0) is stored in the register, and the process continues until all of the bit values have been determined

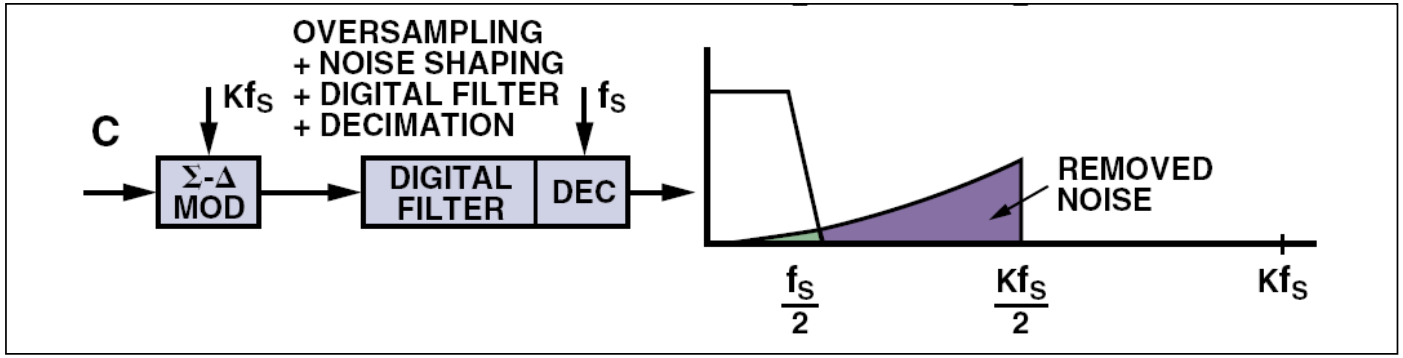
# Principle of oversampling

- Oversampling, digital filtering and decimation
- Quantization noise effect reduction
  - SNR between 0 and $f_s/2$ increases by 3 dB for each doubling of $K$



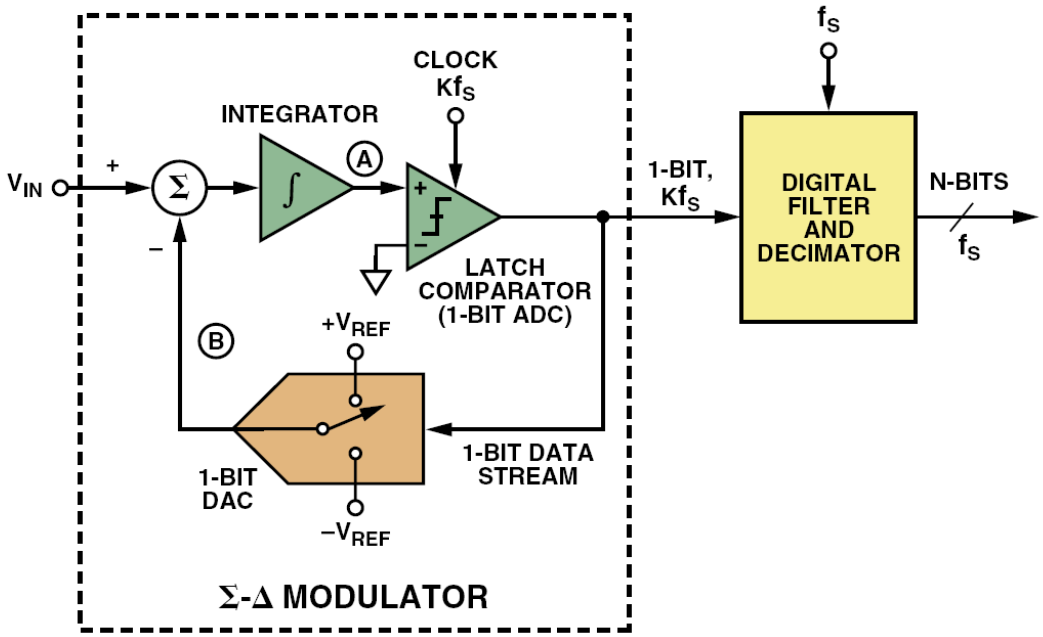
# Σ-Δ ADC basic principles

- The traditional ADC is replaced by an oversampled Σ-Δ modulator
- The effect on noise is a shaping at high frequency
- Digital filtering further improves the SNR in the useful bandwidth



# Σ-Δ ADC Architecture

- Σ-Δ modulator consists of a 1-bit ADC and a 1-bit DAC closed in feedback
- The modulator output is a 1-bit stream of data
- Quantization noise is shaped towards high frequency



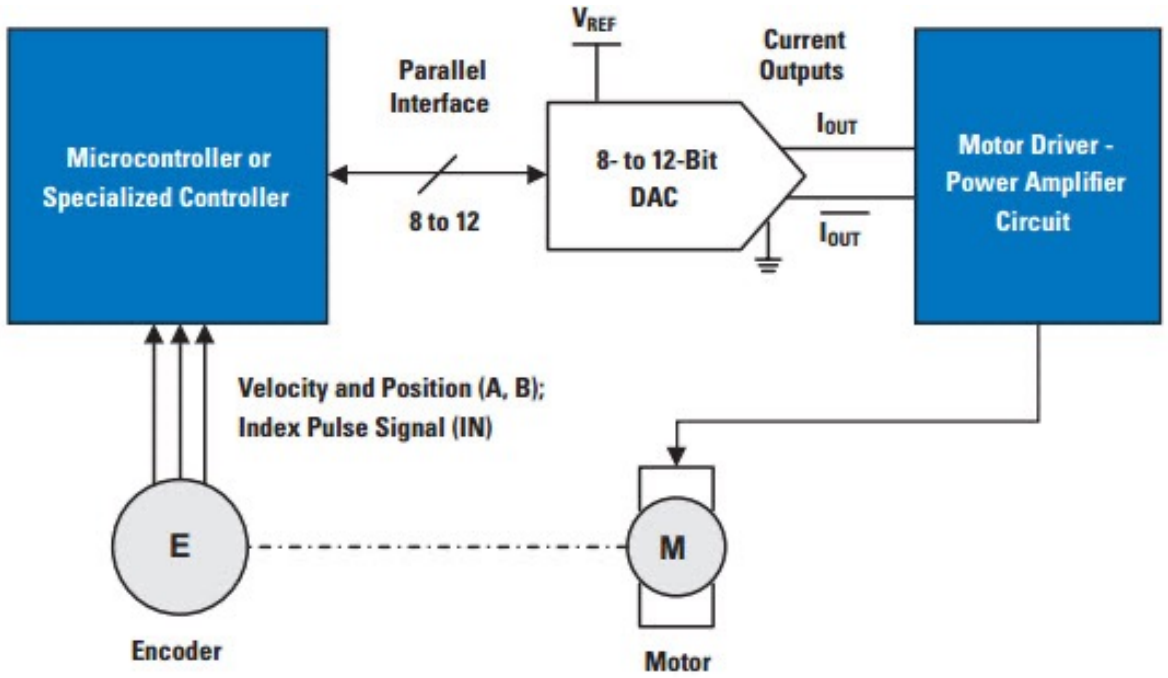
# Σ-Δ ADC Pros and Cons

- Noise shaping effect reduces noise in the signal bandwidth
- Very simple 1-bit architecture
- Most of computation is made on digital

- Oversampling requires higher clock frequency
- Filtering requires several clock cycles and a latency between input and output shows up

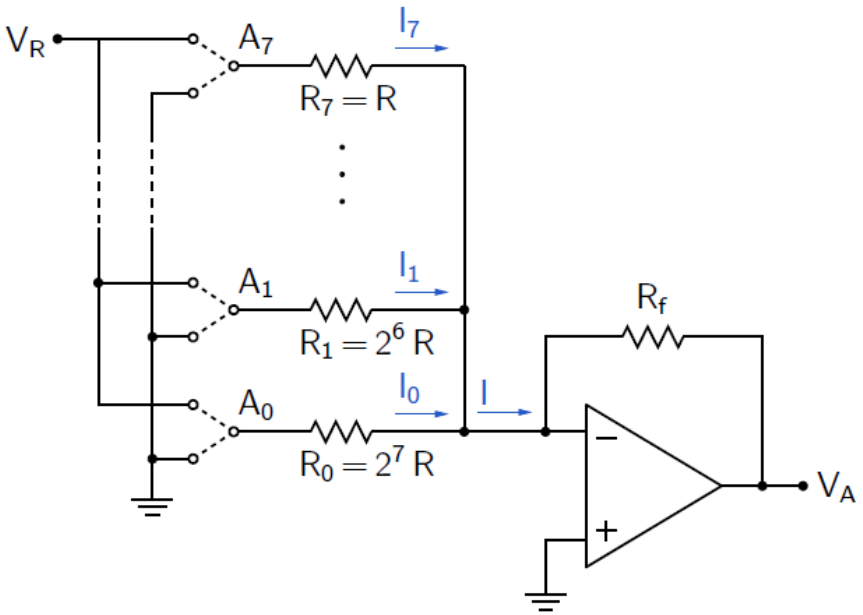
# Digital to Analog Converters (DACs)

- Typical DAC application in motor control



# DACs Architectures

**Binary weighted resistors**




**R-2R ladder resistors**