EE1320: Measurement Science

Lecture 5: Analog-to-Digital Convertors

Dr. ir. Michiel Pertijs, Electronic Instrumentation Laboratory

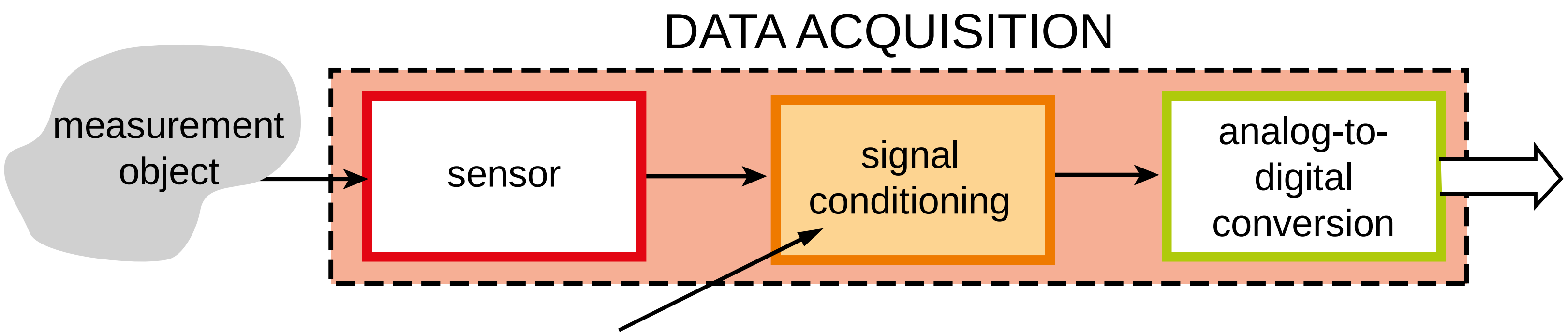
May 21, 2013

Course program 2013

week	date	topic
4.1	Tu 23/4	#1 intro measurements and meas. systems
	Fr 26/4	#2 sensors
4.3	Tu 7/5	#3 sensor readout and signal conditioning
4.4	Tu 14/5	#4 instrumentation amplifiers
	We 15/5	intermediate test
4.5	Tu 21/5	#5 analog-to-digital converters
4.6	We 29/5	#6 measurement instruments I
4.7	Tu 4/6	#7 measurement instruments II
	We 5/6	intermediate test
4.8	Tu 11/6	tutorial
4.11	We 3/7	final exam

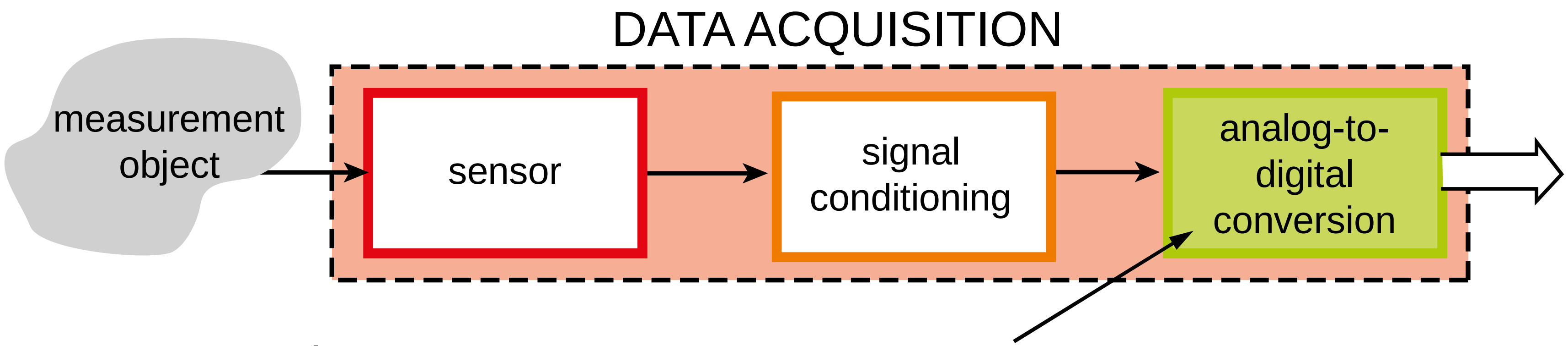
Lecturer: dr. ir. Michiel Pertijs
room HB 15.050, M.A.P.Pertijs@tudelft.nl, 015-2786823

Previous lectures…
signal conditioning



- Signal conditioning needed to adjust output signal of the sensor to the input of the ADC
  - Opamp-based readout circuits, with non-idealities
  - Bridge circuits
  - Differential vs. common-mode signals
  - Difference and instrumentation amplifiers

Today: analog-digital convertors



- Basic principles:
  - Sampling
  - Quantization
- Three important ADC types:
  - Flash ADCs
  - Dual-slope ADCs
  - Successive-approximation ADCs

Overview study material

- Regtien 2.1.1: classification of signals, analog vs. digital
- Regtien 2.2.3: sampling, aliasing
- Regtien 18: D/A en A/D conversion
- Regtien 15.2.2: Sample-and-hold circuits

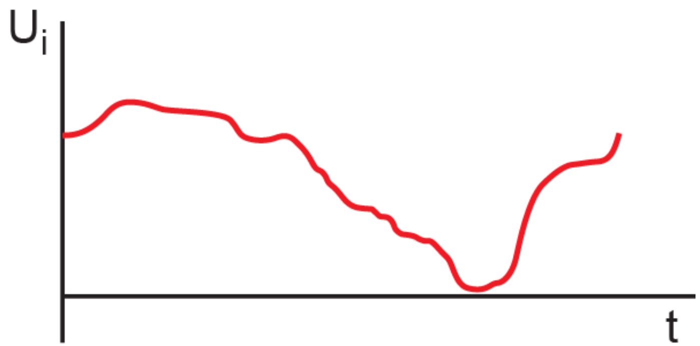
ADC: sampling and quantization

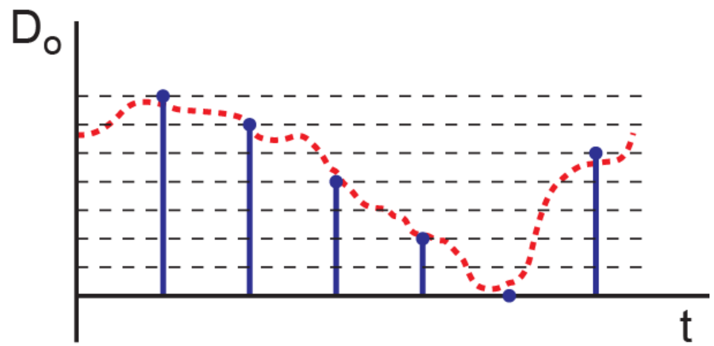
analog input signal

- time continuous
- amplitude continuous

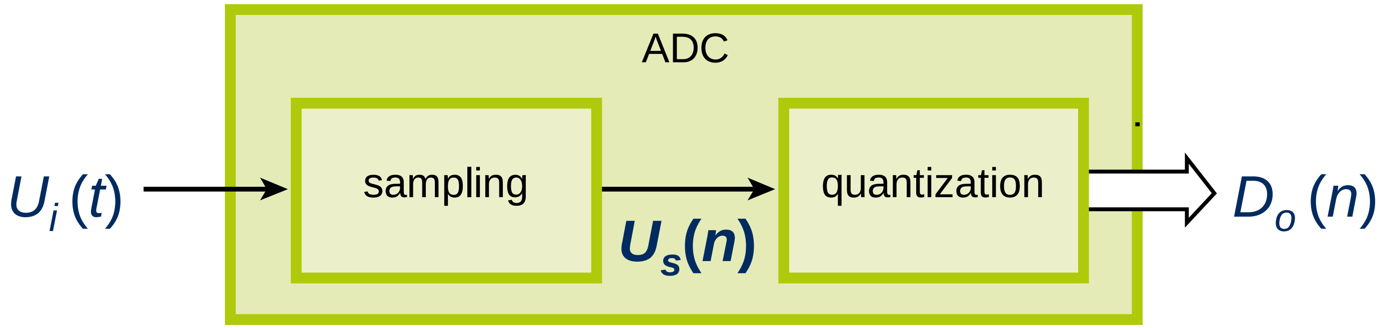
digital output signal

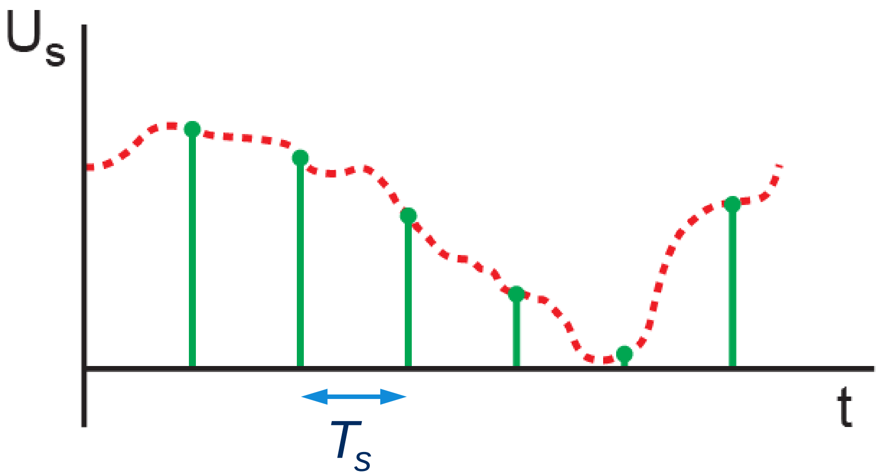
- time discrete: sampled
- amplitude discrete: quantized





Sampling

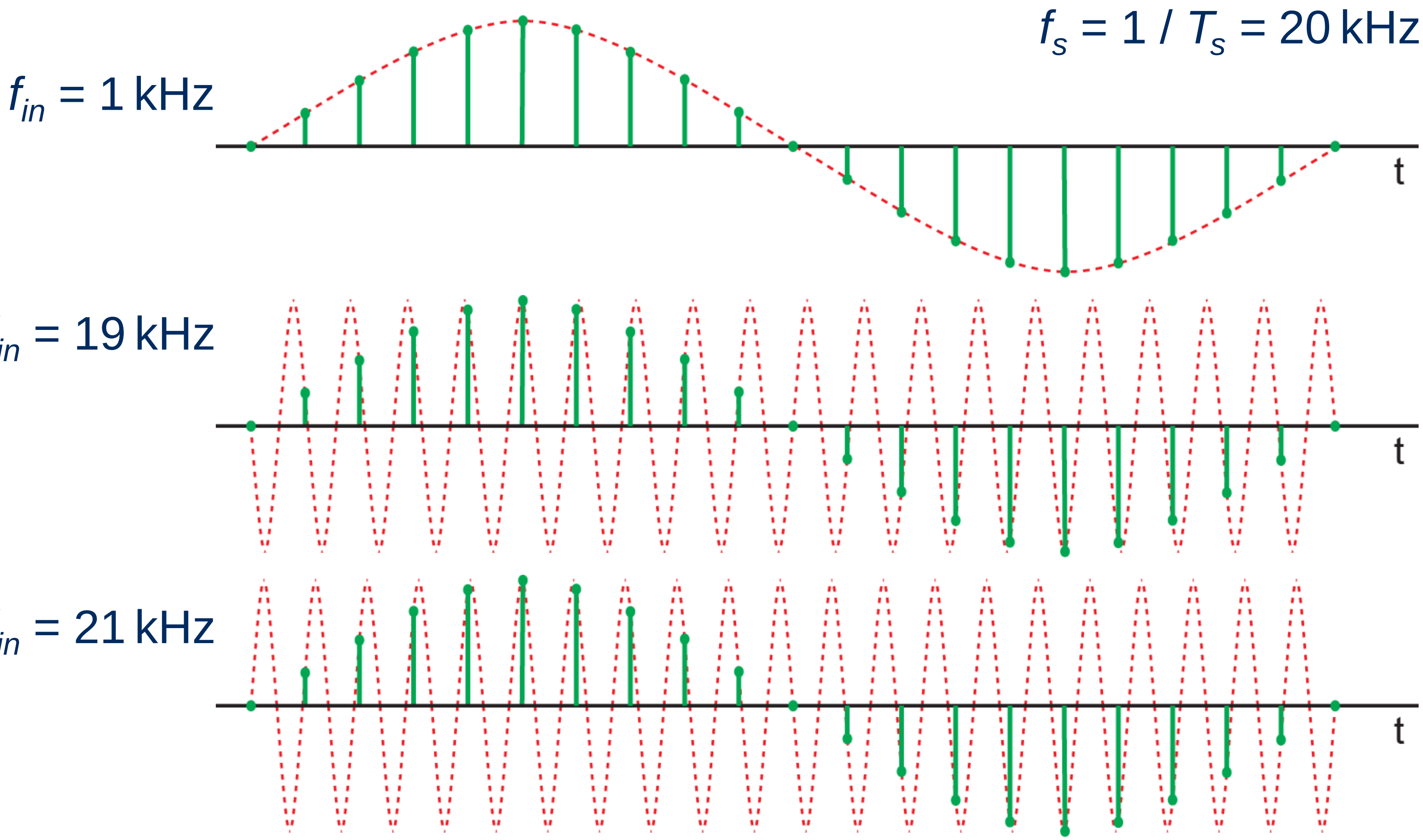




sampled input signal

- amplitude information only at discrete points in time $t = n \cdot T_s$
- sampling frequency $f_s = 1 / T_s$ has to be sufficiently high to be able to track changes in U_i

Sampling

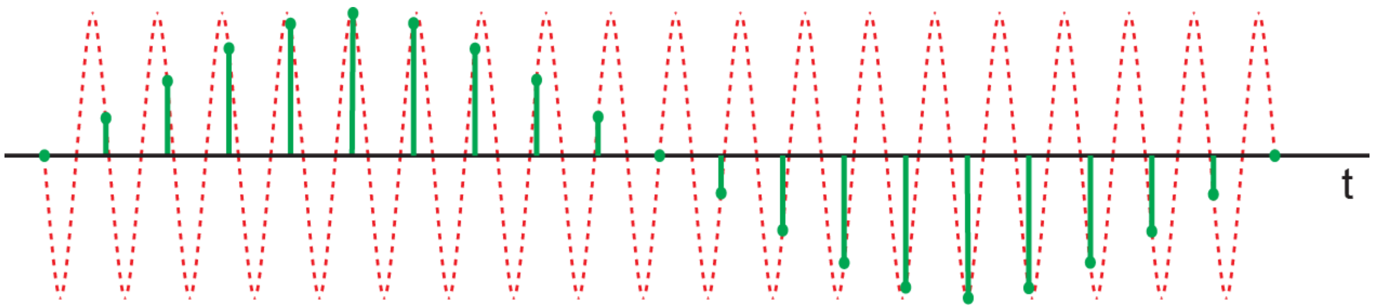


Various input signals yield the **exact same** sampled signal!

Aliasing

f_{in} = 19 kHz

f_s = 20 kHz



- 19 kHz input signal sampled with f_s = 20 kHz
  is indistinguishable from 1 kHz signal
  ⇒ by sampling, the 19 kHz 'folds back' to 1 kHz

- This folding back is called **aliasing**

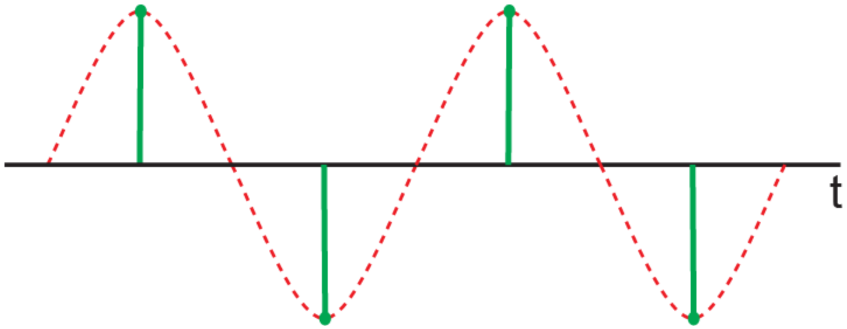
- In order to prevent aliasing,
  the sampling has to be done at a sufficiently high frequency

Shannon-Nyquist Theorem

- A signal that contains no components with a frequency higher than f_{max} can be fully reconstructed based on sampling at $f_s > 2 \cdot f_{max}$ (the Nyquist frequency)

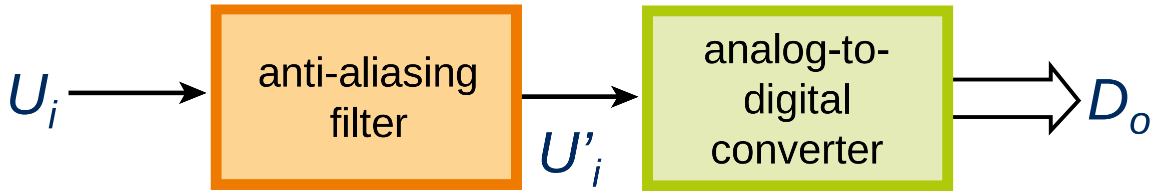
- If $f_s \leq 2 \cdot f_{max}$ aliasing will occur

- So: at least 2 samples per period of the highest frequency component



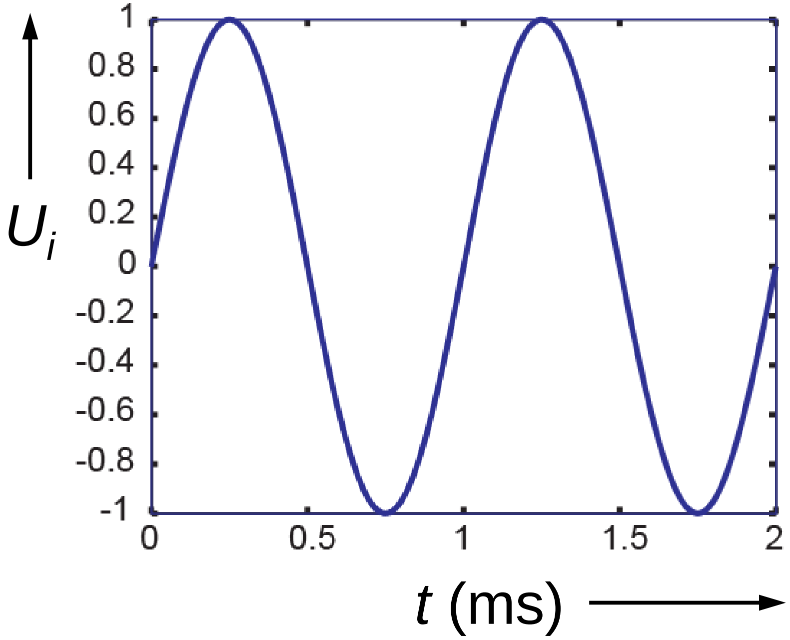
Preventing aliasing

- Two ways:
  - Sample at least at the Nyquist frequency: $f_s > 2 \cdot f_{max}$
  - Limit the bandwidth of the input signal: $f_{max} < f_s / 2$
- **Anti-aliasing filter:** low-pass filter suppressing signal components above $f_s / 2$



Exercise: aliasing

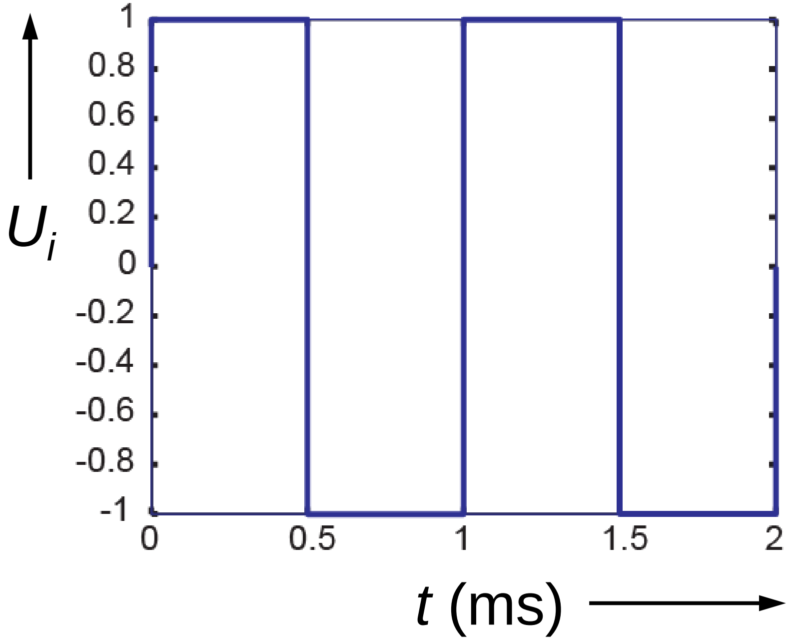
- What is the minimum f_s at which aliasing is prevented?



$$U_i(t) = \sin(2\pi \cdot f \cdot t), \quad f = 1\text{kHz}$$

Oefening: aliasing

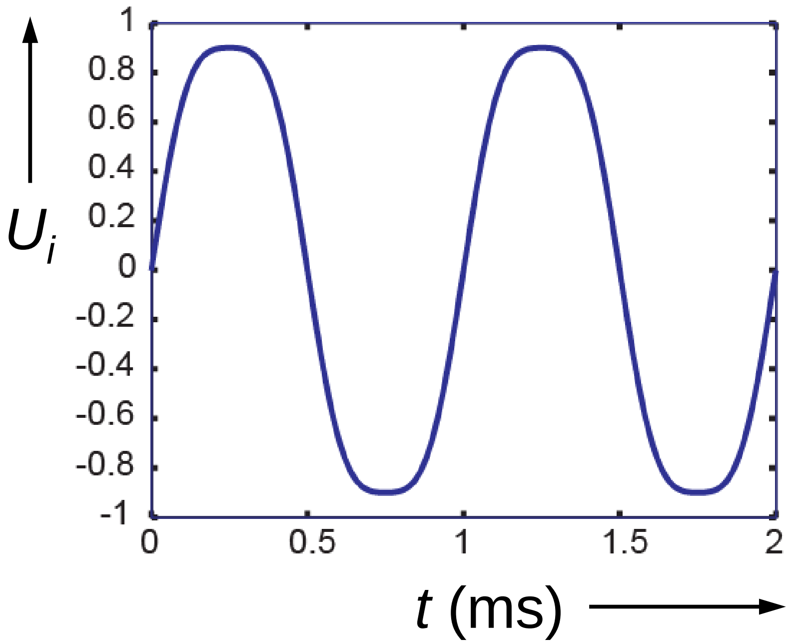
- What is the minimum f_s at which aliasing is prevented?



$$U_i(t) = \text{square wave} \quad f = 1\,\text{kHz}$$

Exercise: aliasing

- What is the minimum f_s at which aliasing is prevented?

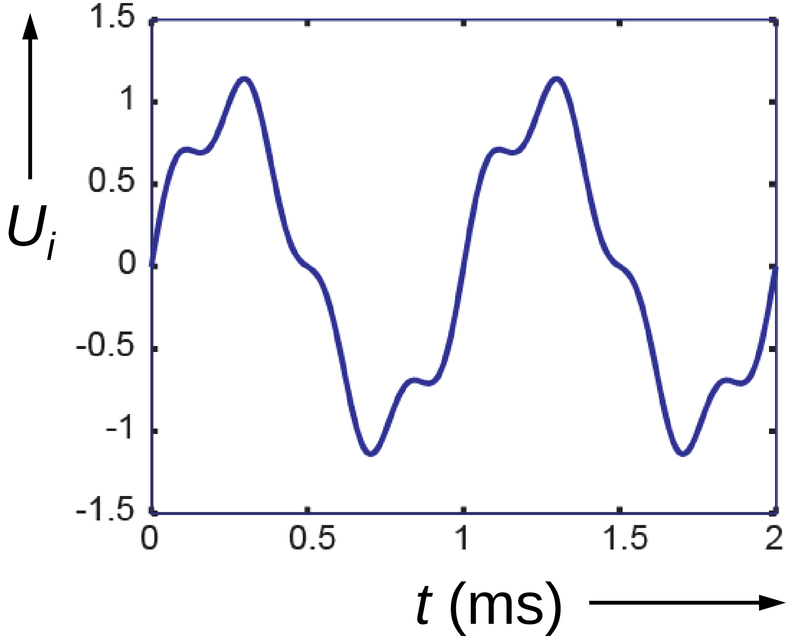


$$U_i(t) = \sin(2\pi \cdot f \cdot t) + 0.1 \cdot \sin(2\pi \cdot 3f \cdot t),$$

$$f = 1\text{kHz}$$

Exercise: aliasing

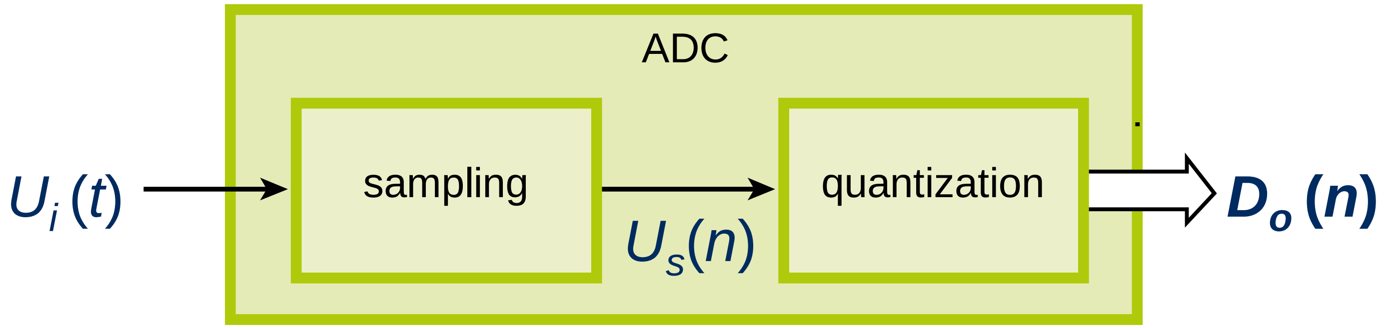
- What is the minimum f_s at which aliasing is prevented?

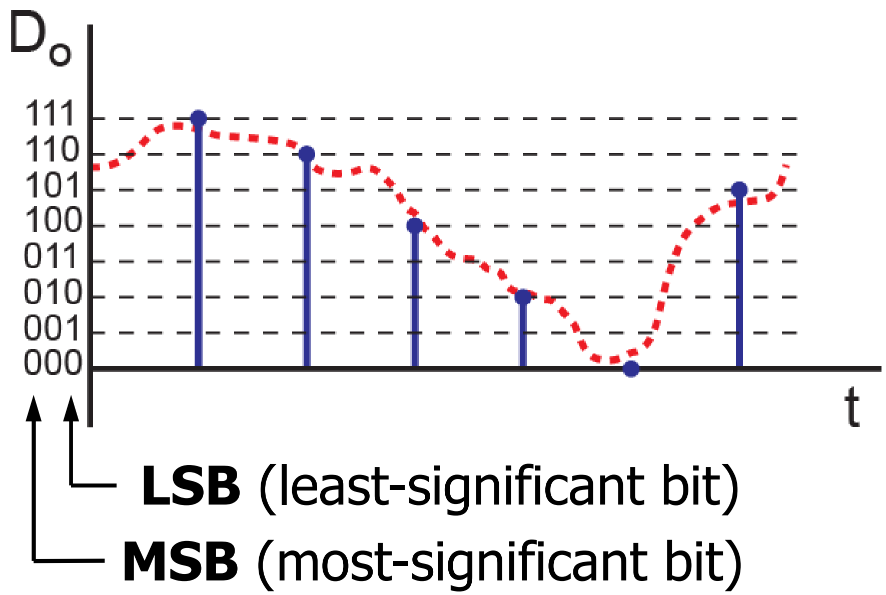


$$U_i(t) = \sin(2\pi \cdot f \cdot t) + 0.2 \cdot \sin(2\pi \cdot 4f \cdot t),$$

$$f = 1\text{kHz}$$

Quantization



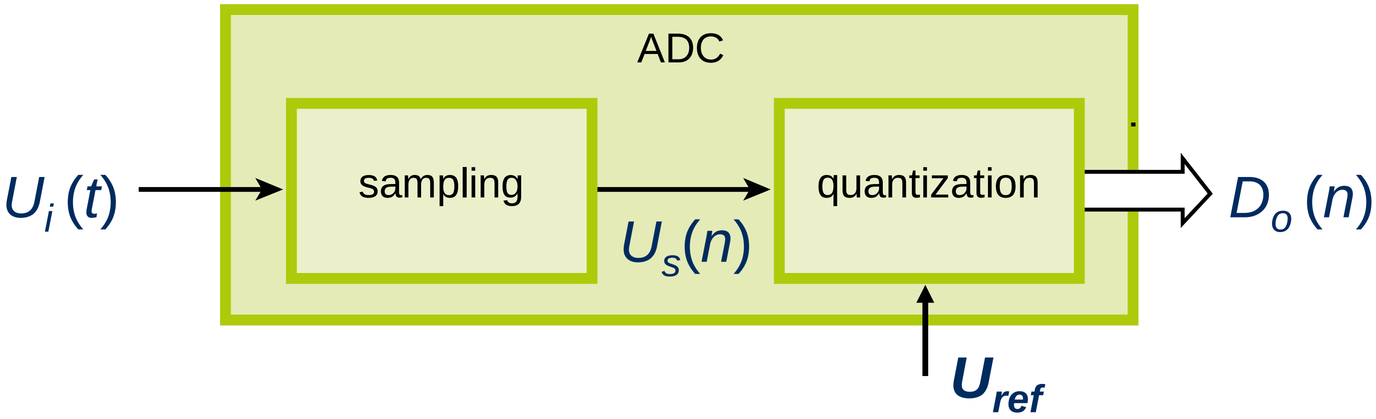


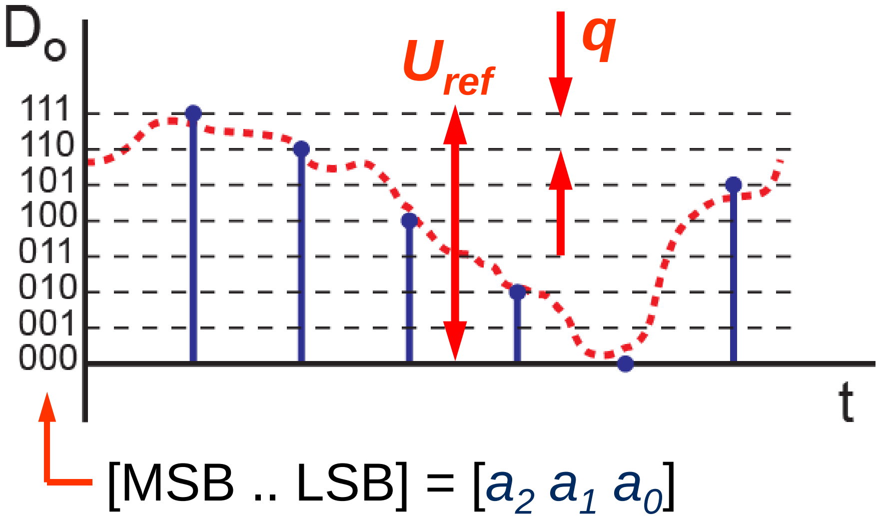
quantized output signal

- sampled values are mapped onto a finite number of quantization levels
- the number of levels determines the **resolution**:

  n-bits ADC ⇔ 2^n levels
- output D_o typically binary encoded using n bits

Quantization





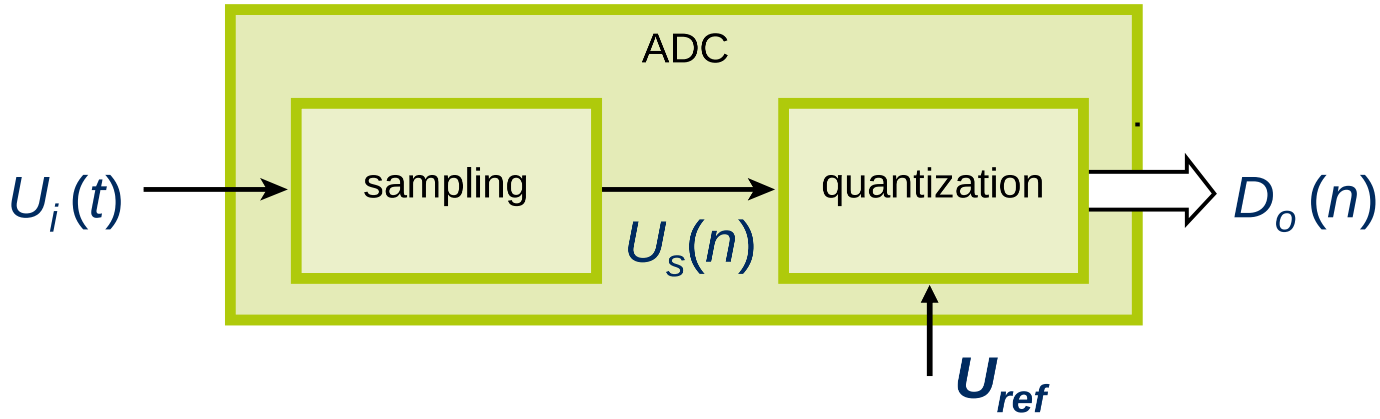
- quantization levels defined w.r.t. **reference voltage U_{ref}**
- quantization interval (LSB step):

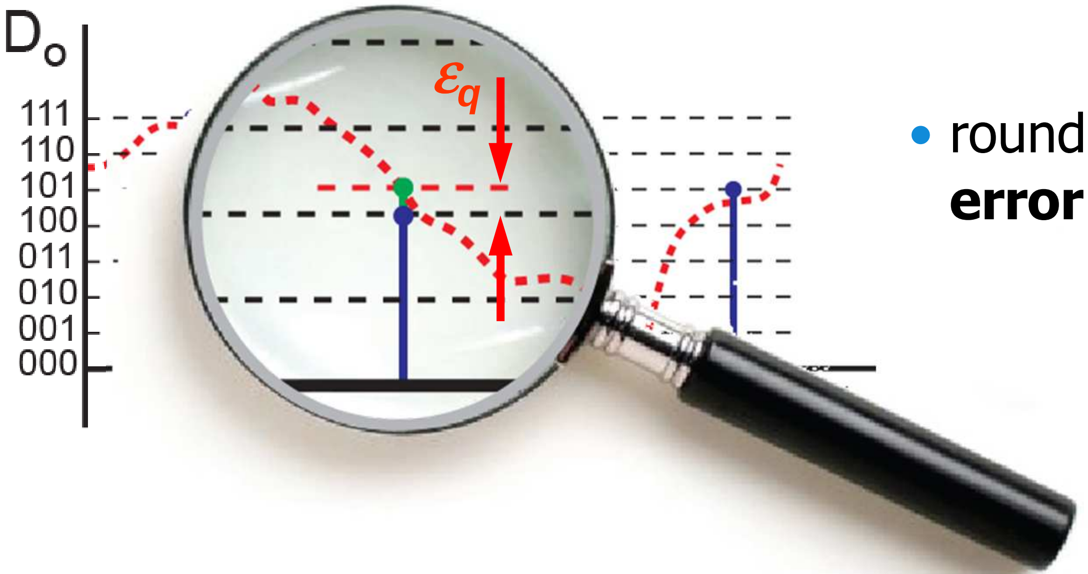
$$q = \frac{U_{ref}}{2^n - 1} \cong \frac{U_{ref}}{2^n} \quad (n >> 1)$$

- D_o yields U_i as fraction of U_{ref}:

$$U_i = a_{n-1}\frac{U_{ref}}{2} + a_{n-2}\frac{U_{ref}}{4} + \ldots + a_0\frac{U_{ref}}{2^n}$$

Quantization errors



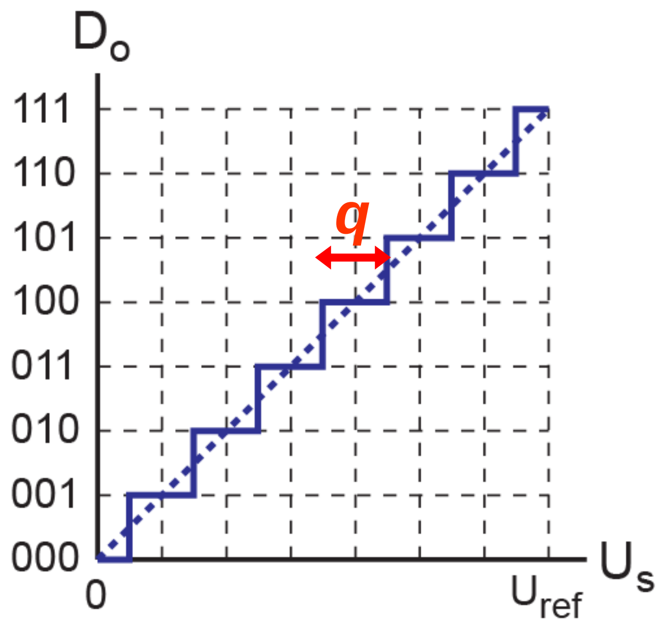


- rounding error = **quantization error** ε_q

how big is the maximum ε_q of an 8-bits ADC with U_{ref} = 1V ?

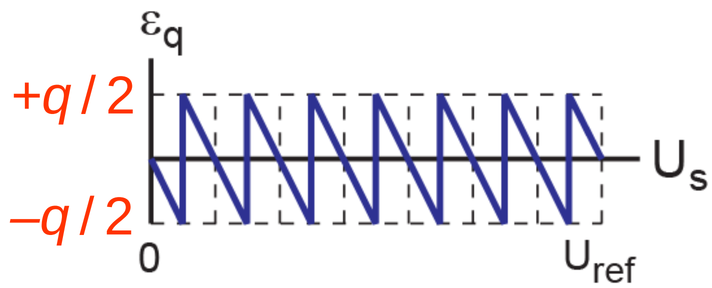
Quantization errors

- ideal transfer ADC:



- maximal quantization error ε_q

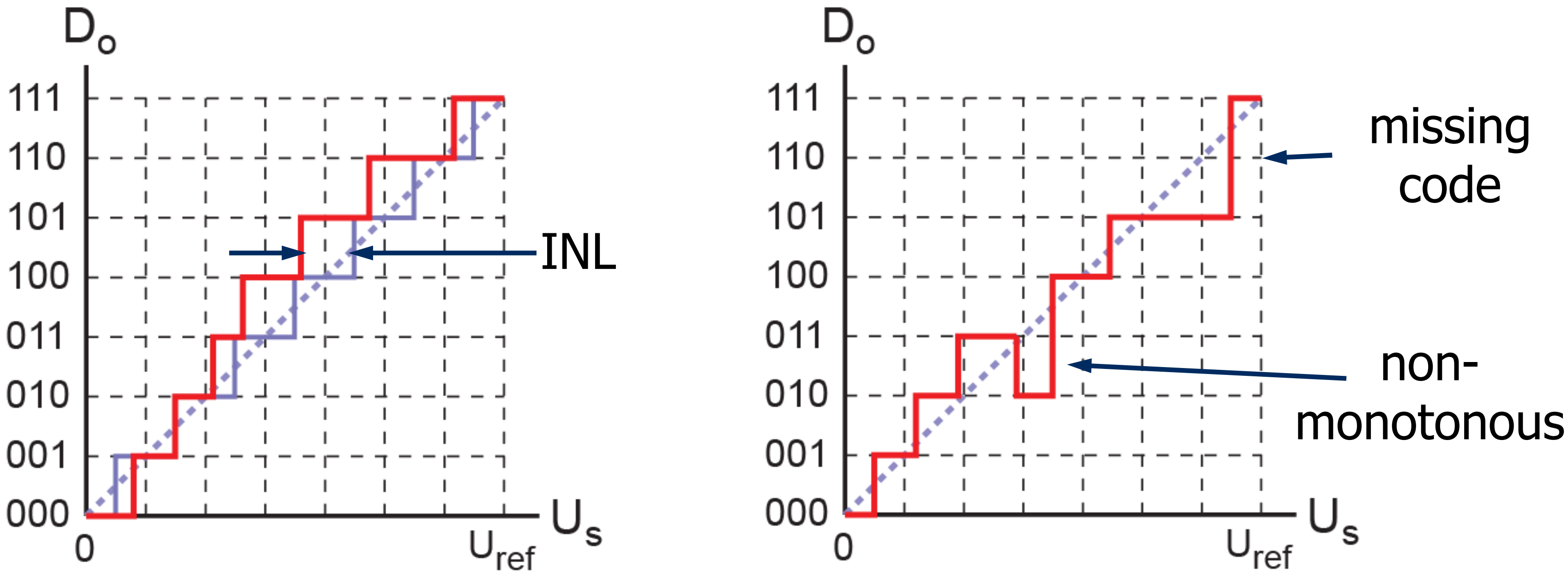
$$\left|\varepsilon_q\right| \leq \frac{q}{2} = \frac{U_{ref}}{2\left(2^n - 1\right)} \cong \frac{U_{ref}}{2^{n+1}} \quad (n >> 1)$$



- corresponds to ±0.5 LSB

Quantization errors

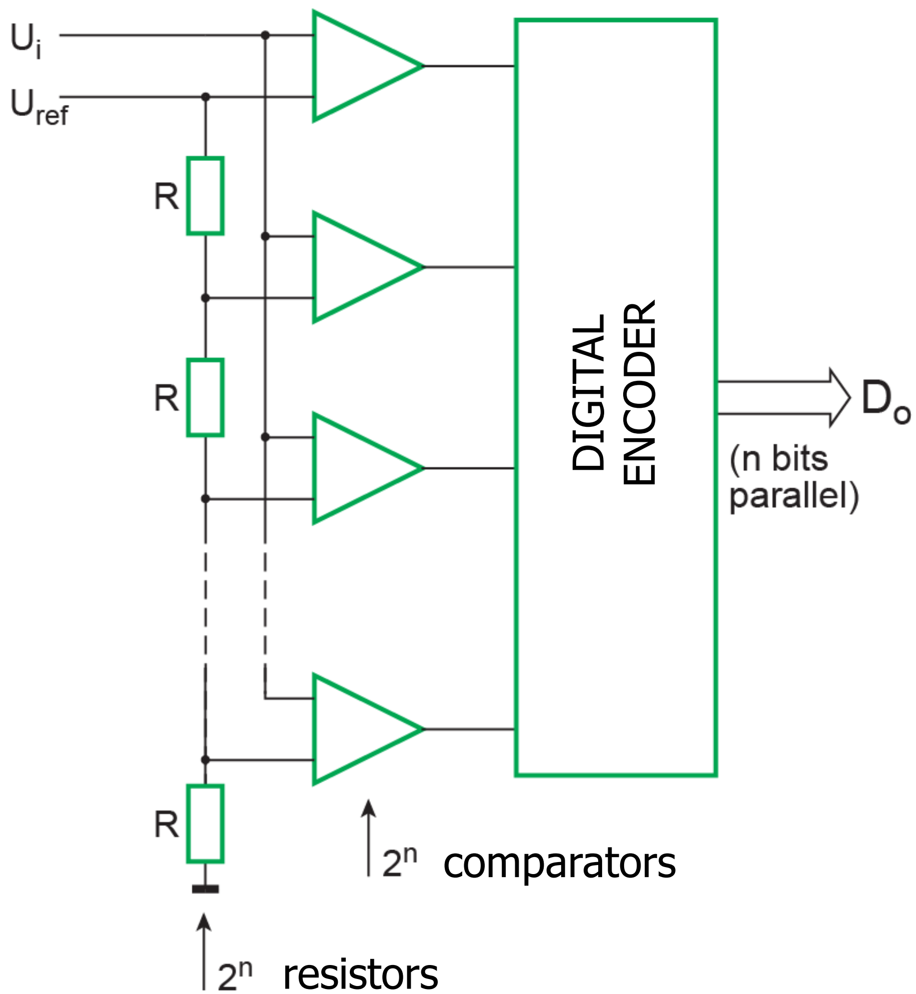
- In addition to (ideal) quantization errors, practical ADCs have more limitations:
  - offset and gain errors
  - non linearity: integral (INL), differential (DNL)
  - and sometimes: non-monotonous transfer (ambiguity) or missing codes



Types of analog-to-digital convertors

- **Direct** ADCs
  - conversion takes place in one step
  - example: flash ADC
- **Integrating** ADCs
  - conversion trough an intermediate step: time period, frequency
  - example: dual-slope ADC
- **Compensating** ADCs
  - input signal is compensated (stepwise) by output signal of digital-to-analog convertor (DAC)
  - example: successive-approximation ADC (SAR ADC)

Flash ADC

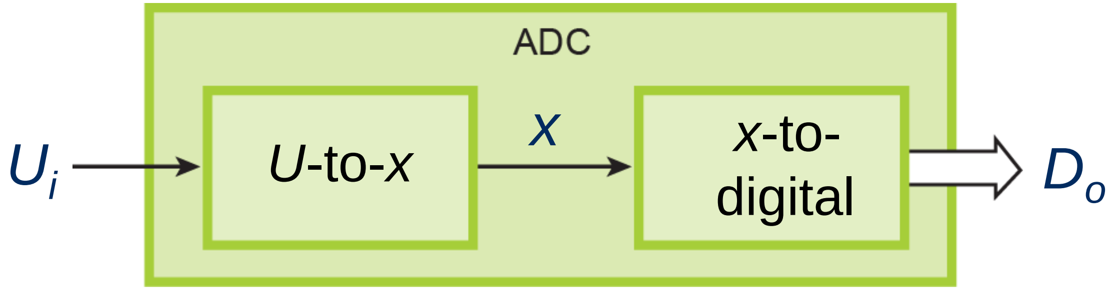


- Resistive divider with 2^n resistors produces quantization levels
- Comparators compare U_i to quantization levels
- Encoder produces *n*-bit binary code

- Fast (>1 GHz possible), but many components!
- Can be found in digital oscilloscopes for example

Indirect ADCs

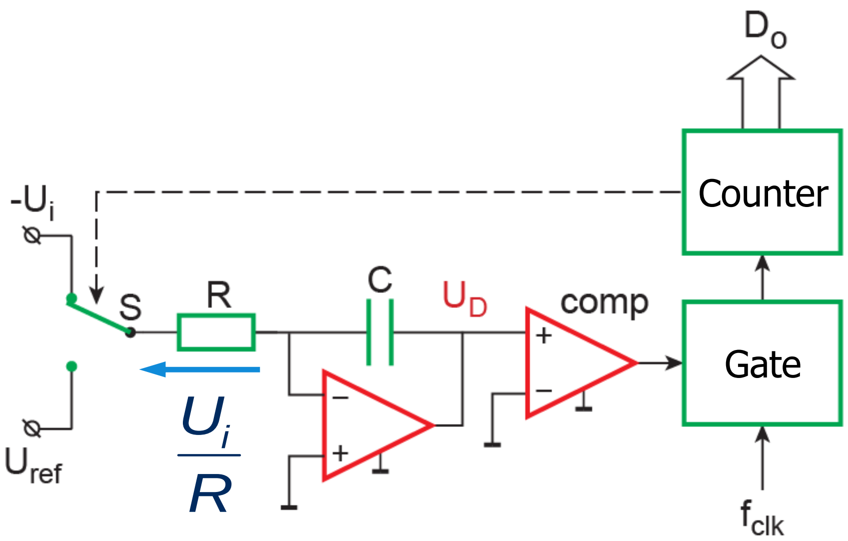
- Use an intermediate step

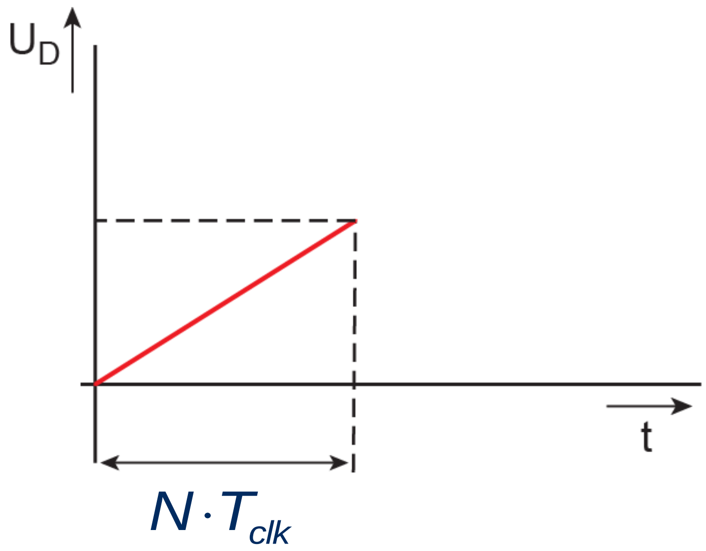


- Intermediate signal x: time step, frequency, bitstream…

- Relatively slow, but a high resolution is possible
- Often used in multimeters

Dual-slope ADC

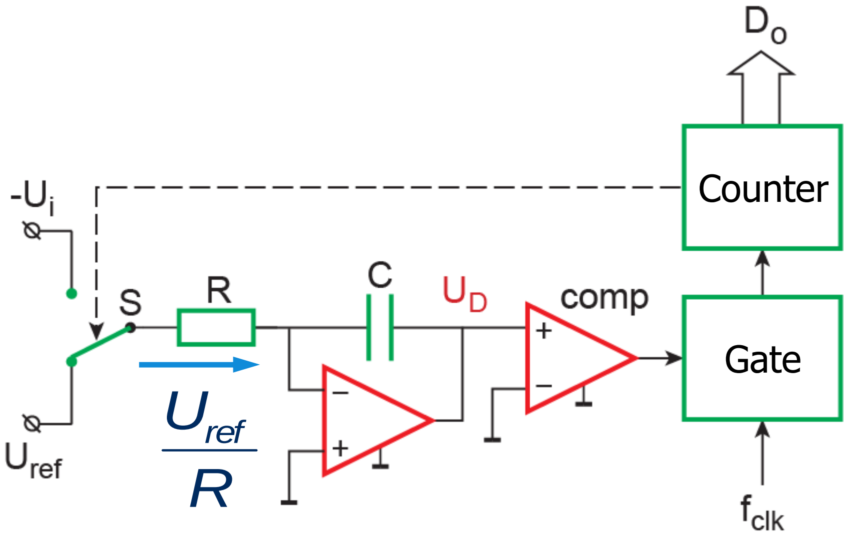


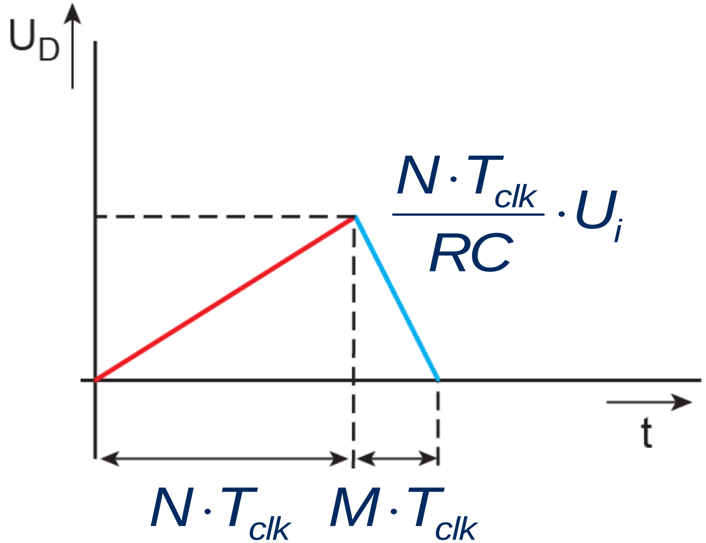


- Input signal is integrated over a fixed period $N \cdot T_{clk} = \frac{N}{f_{clk}}$

$$U_D = \frac{1}{RC} \int_0^{NT_{clk}} U_i(t) \cdot dt = \frac{N \cdot T_{clk}}{RC} \cdot U_i$$

Dual-slope ADC

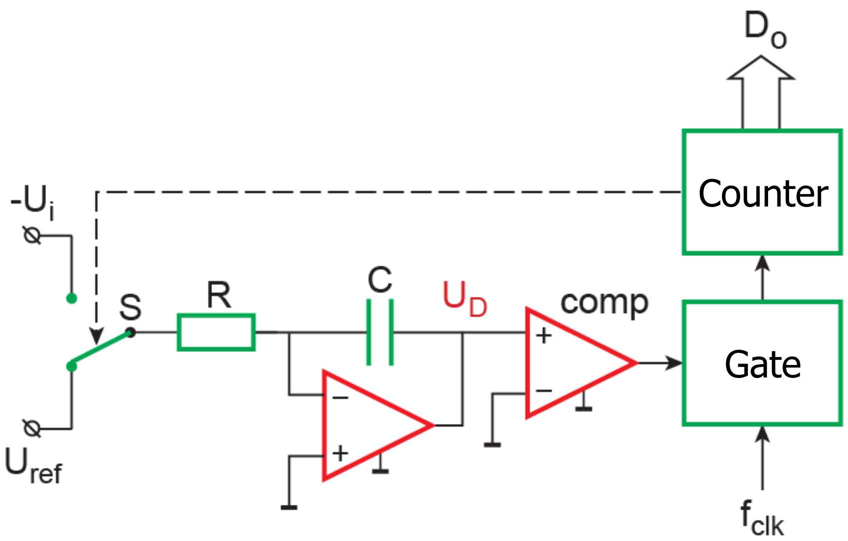


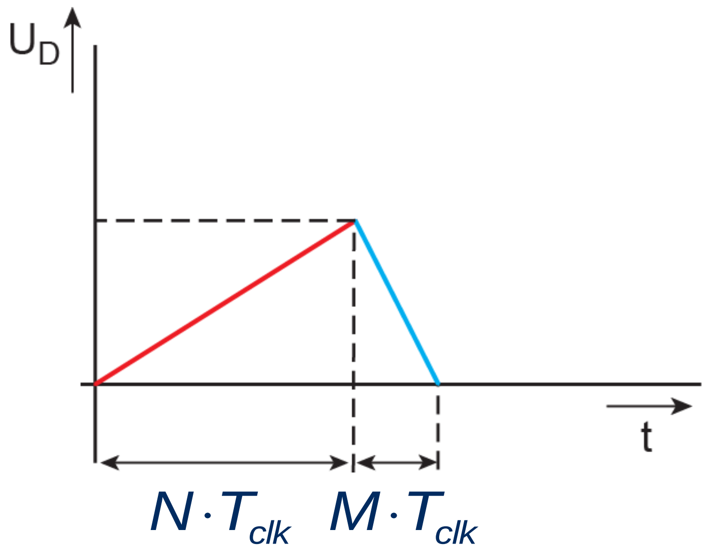


- Input signal is integrated over a fixed period $N \cdot T_{clk} = \frac{N}{f_{clk}}$
- Reference is integrated until U_D is back to 0

$$\frac{N \cdot T_{clk}}{RC} \cdot U_i = \frac{M \cdot T_{clk}}{RC} \cdot U_{ref} \Rightarrow M = \frac{U_i}{U_{ref}} N$$

Dual-slope ADC

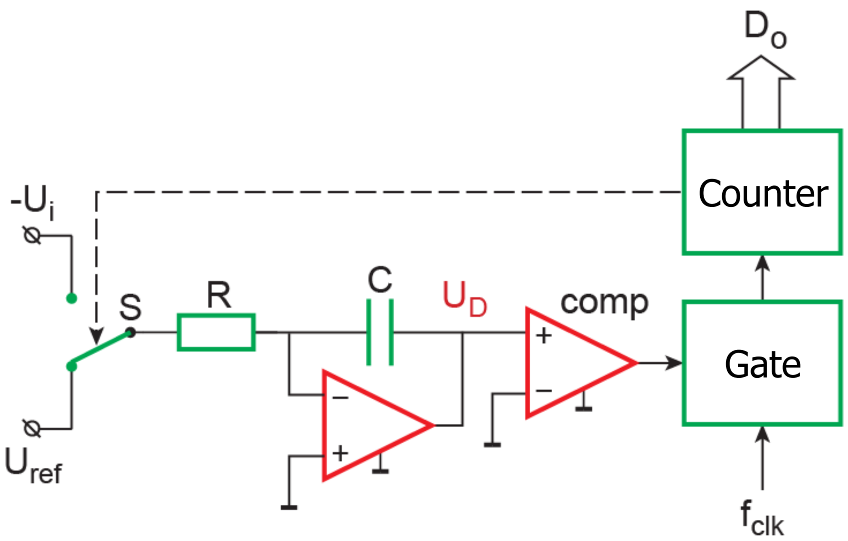


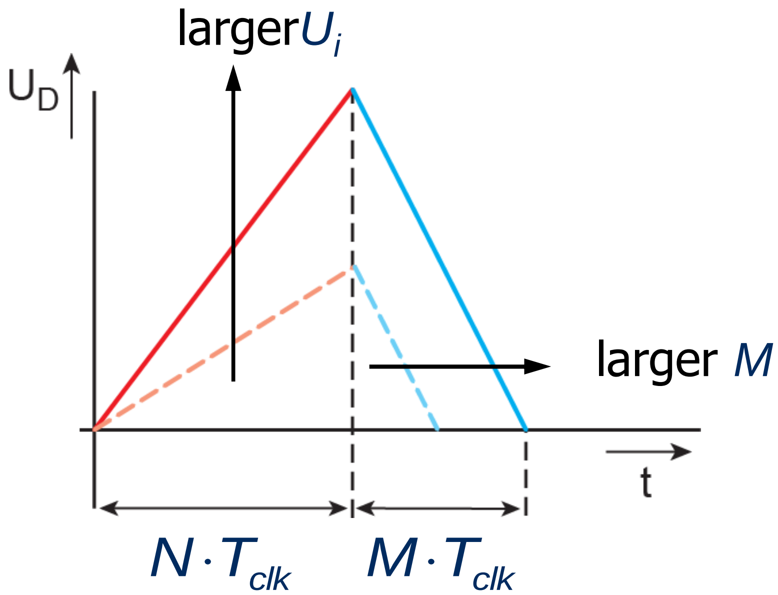


- Number of clock periods M is counted until U_D returns to 0 and is a direct measure for U_i / U_{ref}:

$$D_o = M = \frac{U_i}{U_{ref}} N$$

Dual-slope ADC

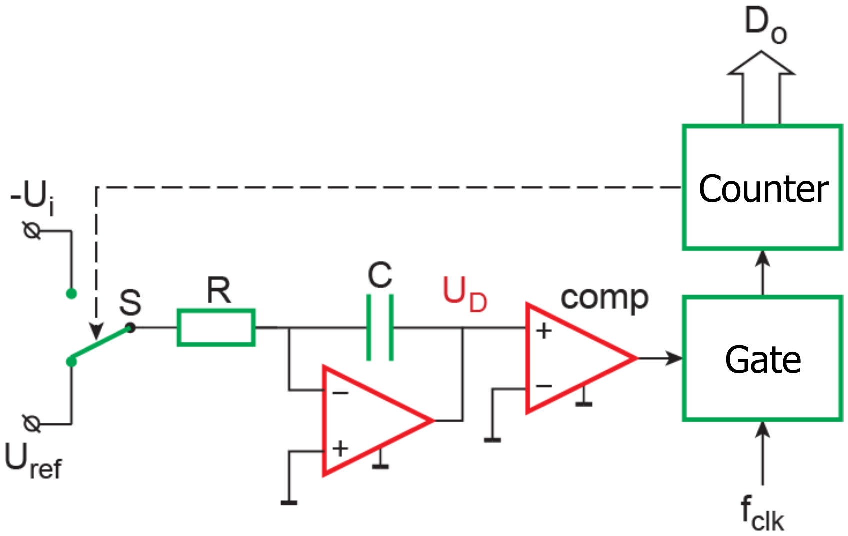


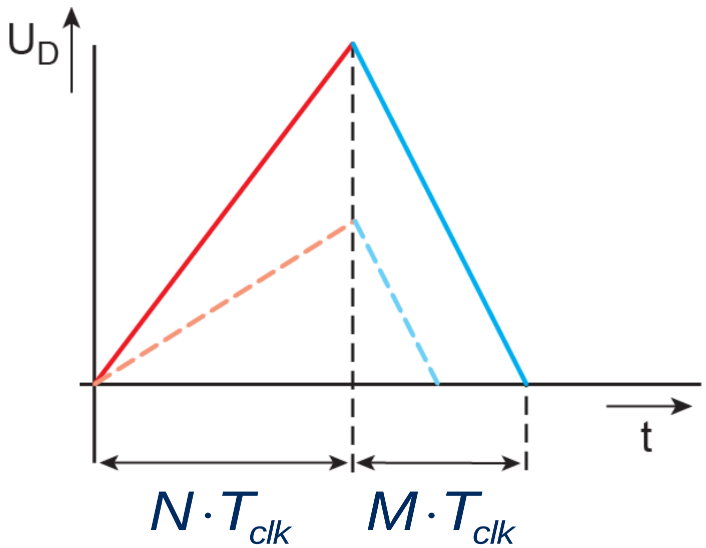


- Number of clock periods M is counted until U_D returns to 0 and is a direct measure for U_i / U_{ref}:

$$D_o = M = \frac{U_i}{U_{ref}} N$$

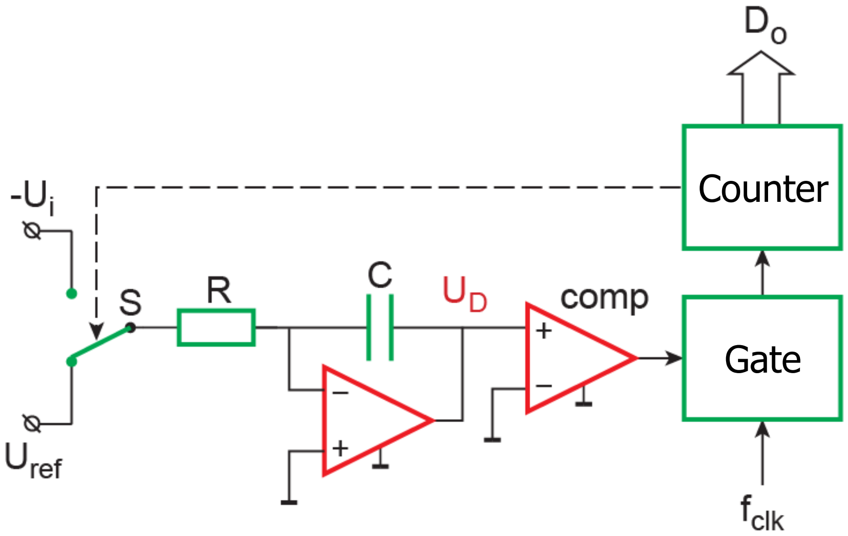
Dual-slope ADC

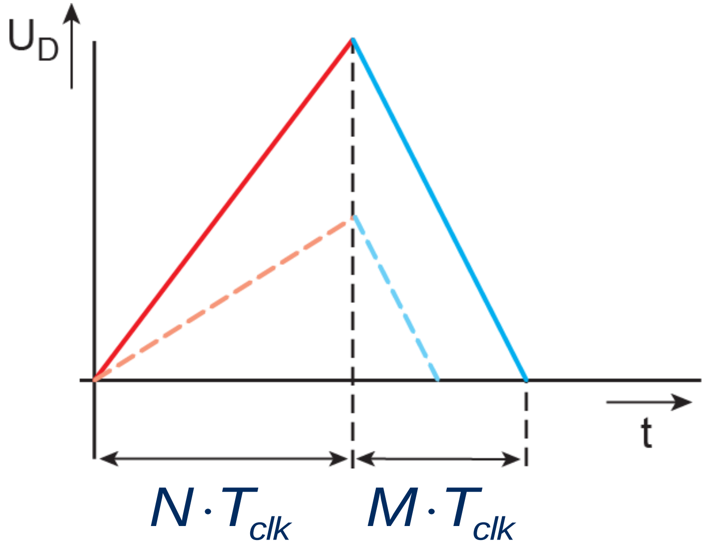




- Note:
  - transfer independent of exact values of *R*, *C*
  - U_i is integrated ⇒ noise / distortion are averaged out
  - integration time $N \cdot T_{clk}$ is often a multiple of 20ms – why?

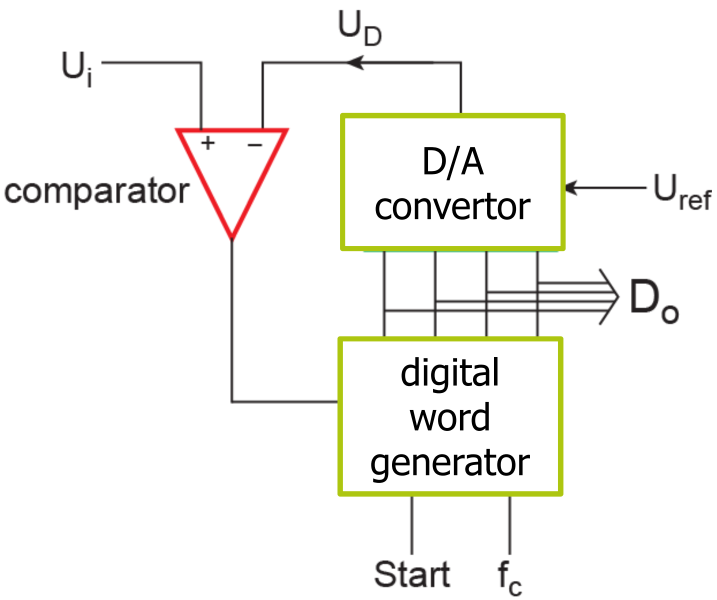
Exercise: dual-slope ADC





- Given: $U_{ref} = 1\,\text{V}$, $N = 16$
  - If the input $U_i = 0.25\,\text{V}$, what is the value of M ?
  - Which U_i result in a counter reading $M = 8$?
  - What is the resolution of this ADC?

Compensating ADCs

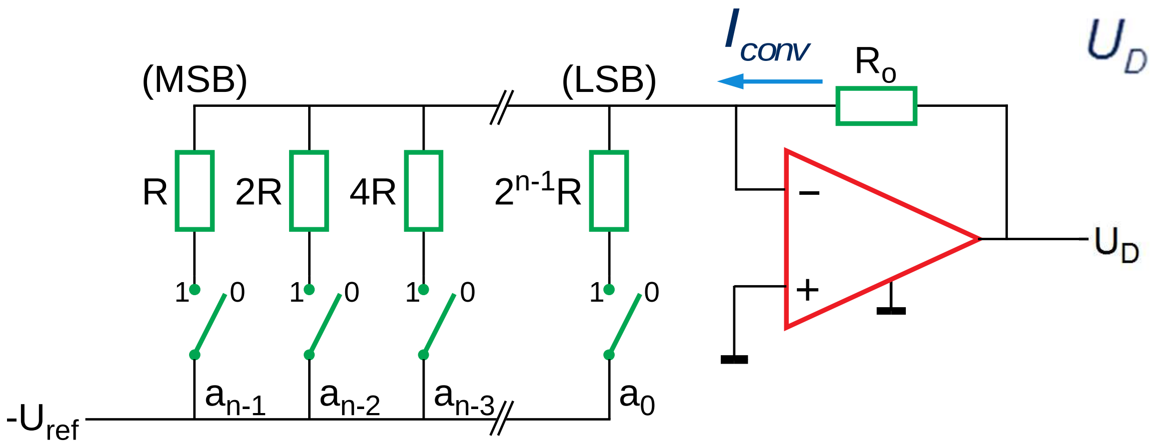


- Comparator compares U_i with output of a digital-to-analog convertor (DAC)
- Feedback: result is used to adjust input DAC (through an algorithm)
- Consequence: U_D tracks U_i
  $\Rightarrow D_o$ tracks U_i / U_{ref}
- Compensating ADCs are often found in data-acquisition systems

Digital-to-analog convertor

- Compensating ADCs need a DAC
  - determines resolution, linearity

- Example: DAC with binary-weighed resistors

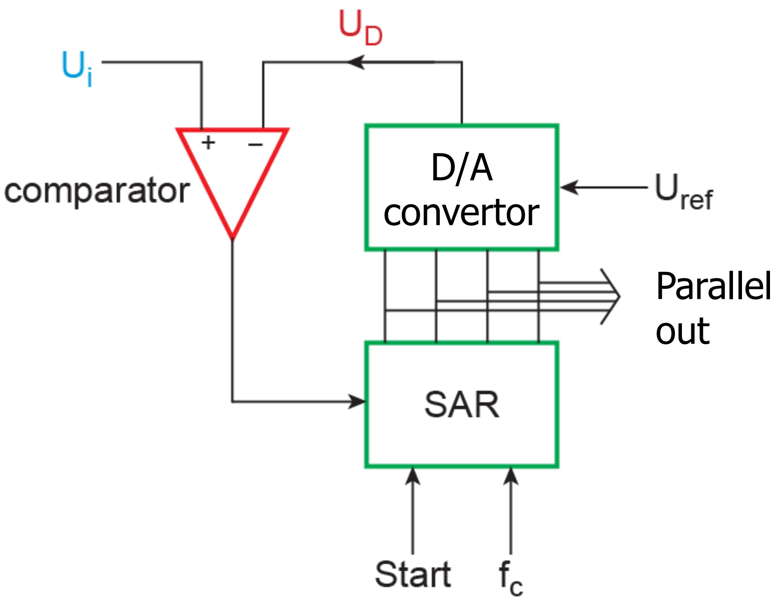


$$U_D = I_{conv} R_o$$

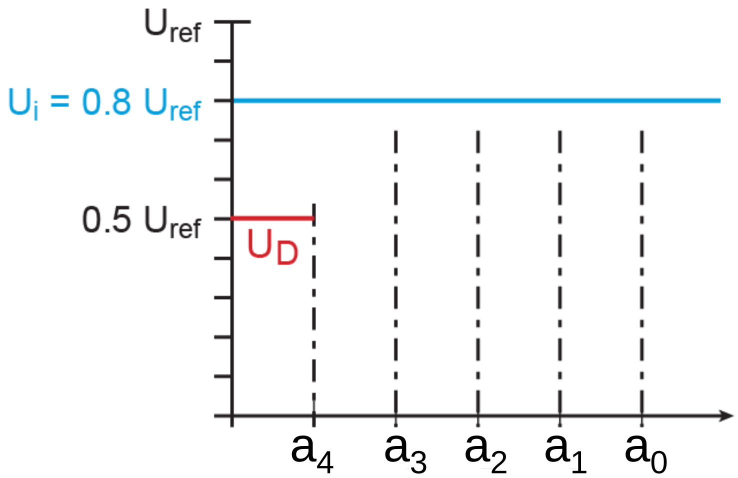
$$= U_{ref} \left(\frac{a_{n-1}}{R} + \frac{a_{n-2}}{2R} + \cdots \right) I_{conv} R_o$$

$$= U_{ref} \frac{R_o}{R} \sum_{i=0}^{n-1} a_i 2^{-n+i+1}$$

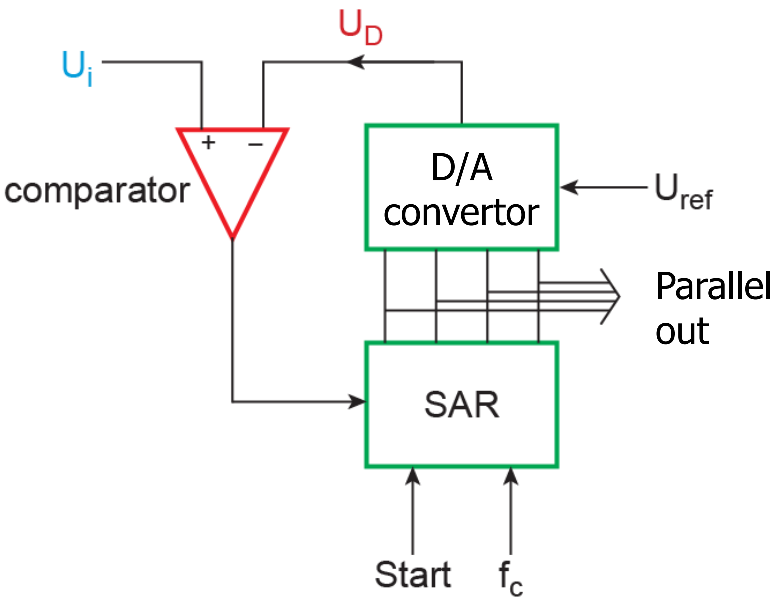
Successive-approximation ADC



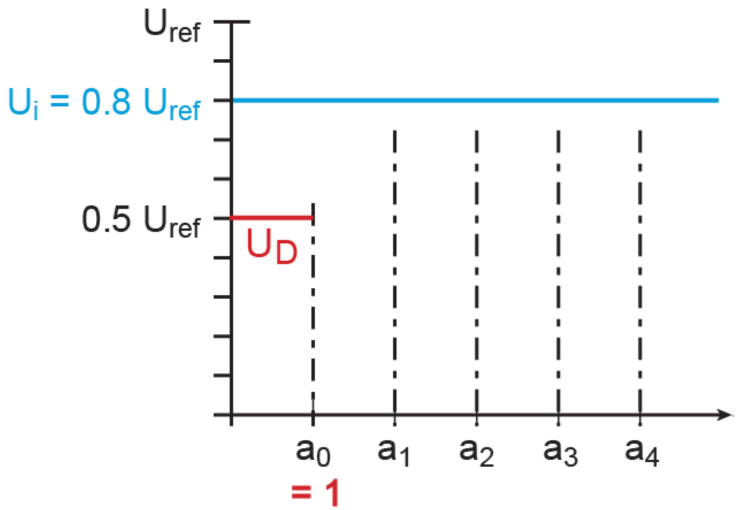
- U_D tracks U_i through a successive-approximation search algorithm



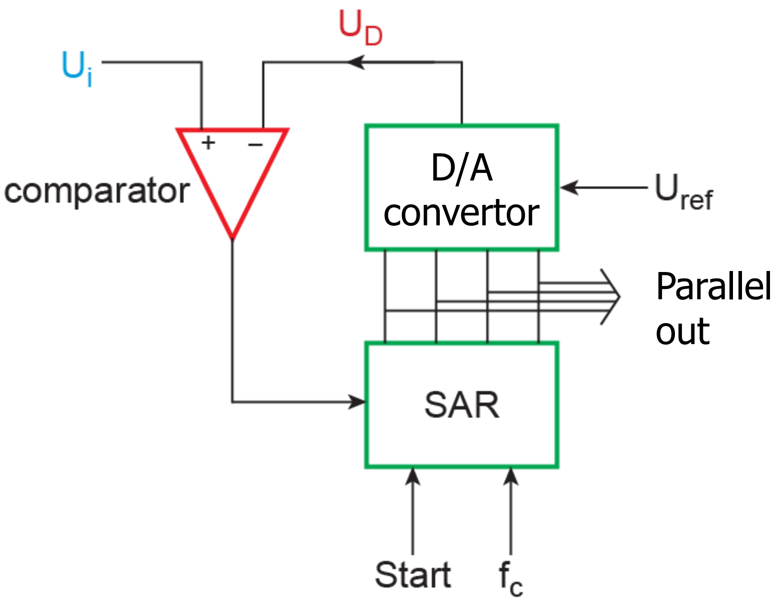
Successive-approximation ADC



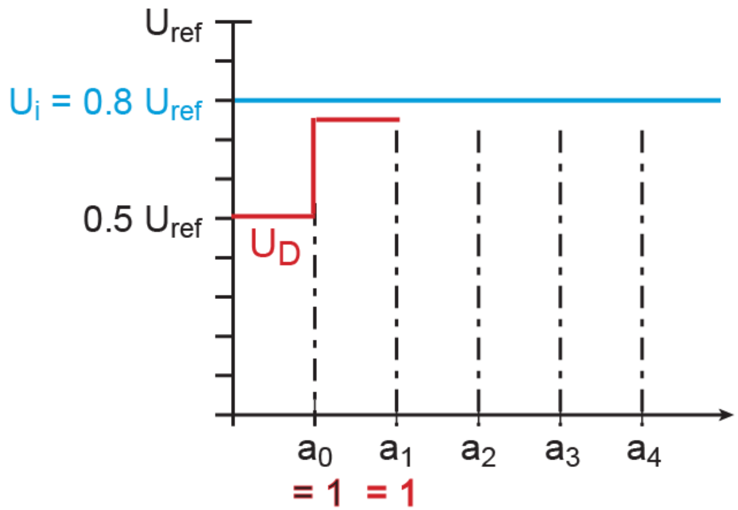
- U_D tracks U_i through a successive-approximation search algorithm



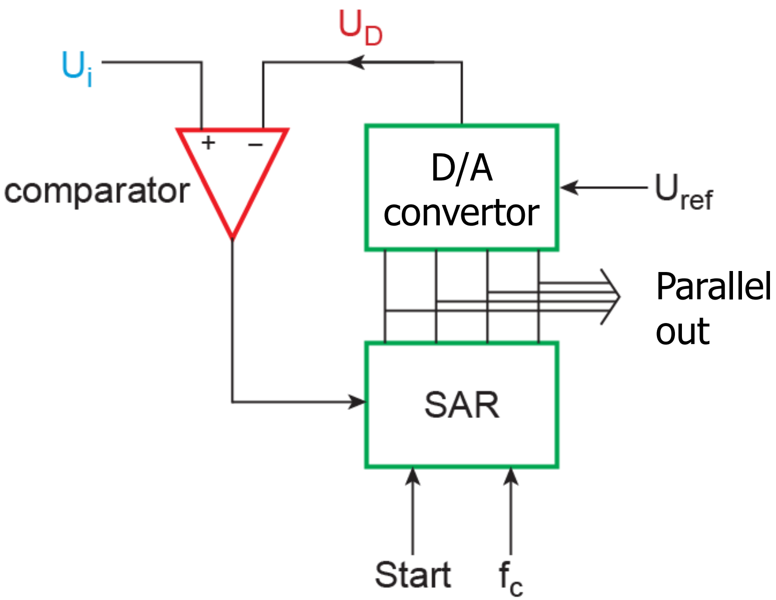
Successive-approximation ADC



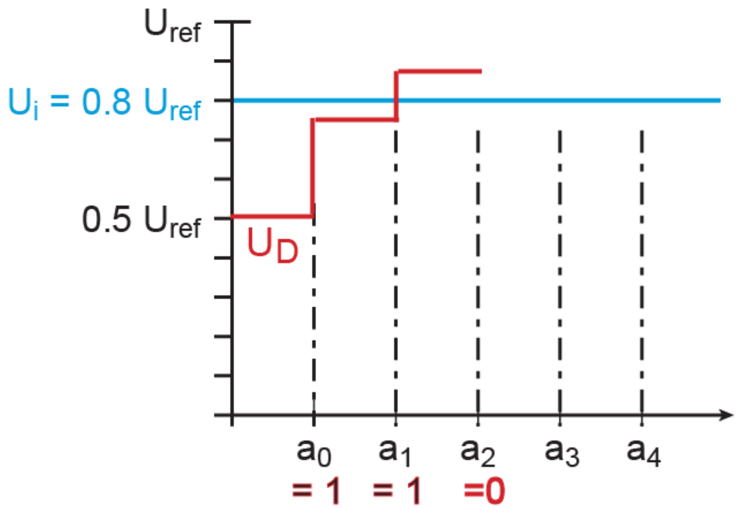
- U_D tracks U_i through a successive-approximation search algorithm



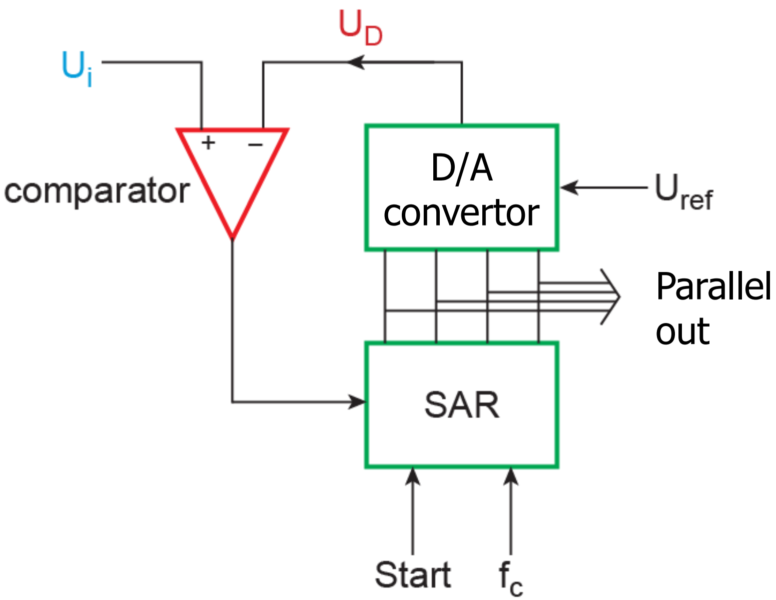
Successive-approximation ADC



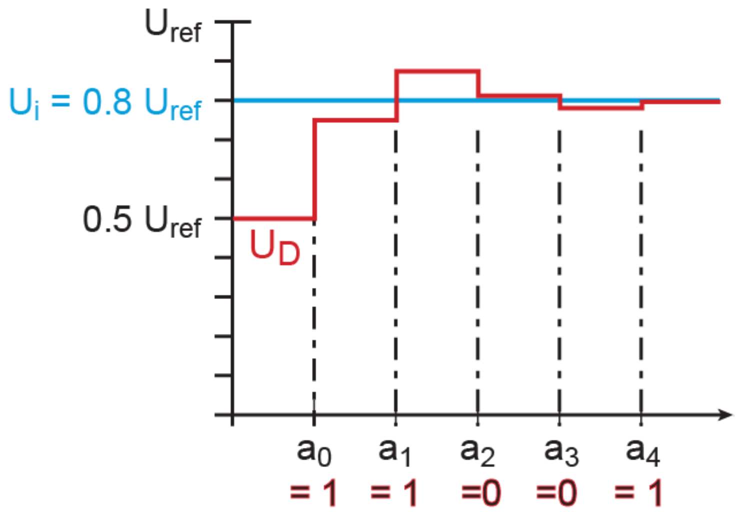
- U_D tracks U_i through a successive-approximation search algorithm



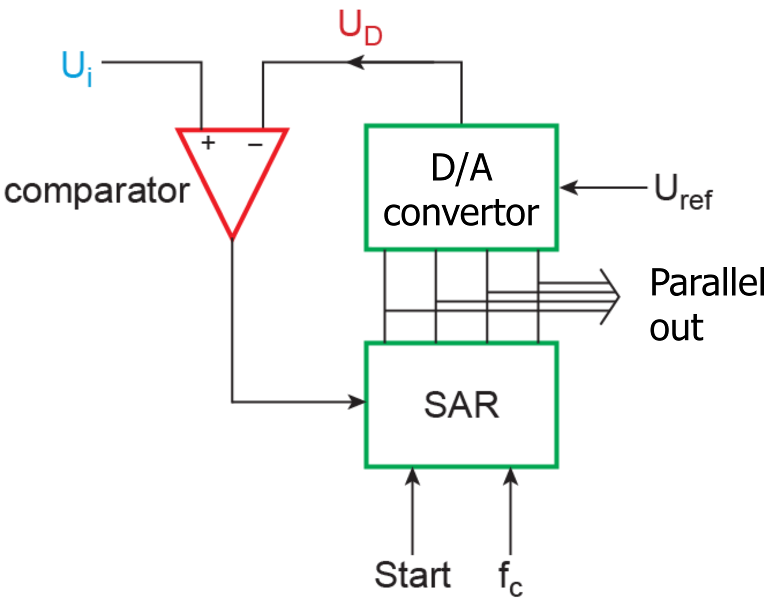
Successive-approximation ADC



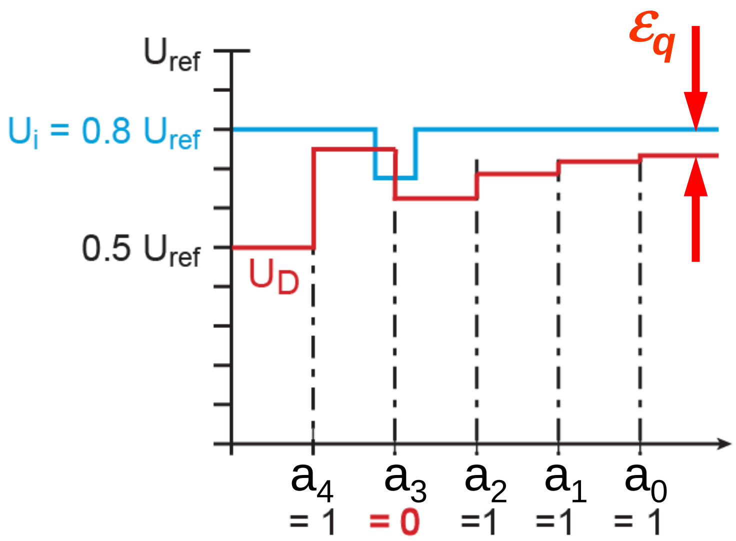
- U_D tracks U_i through a successive-approximation search algorithm



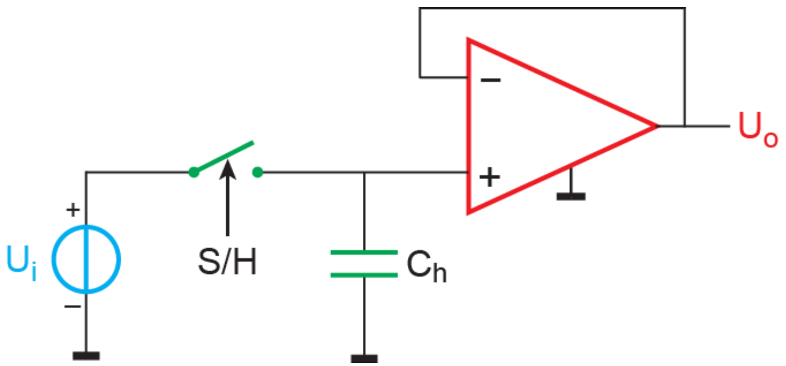
Importance of a stable input signal

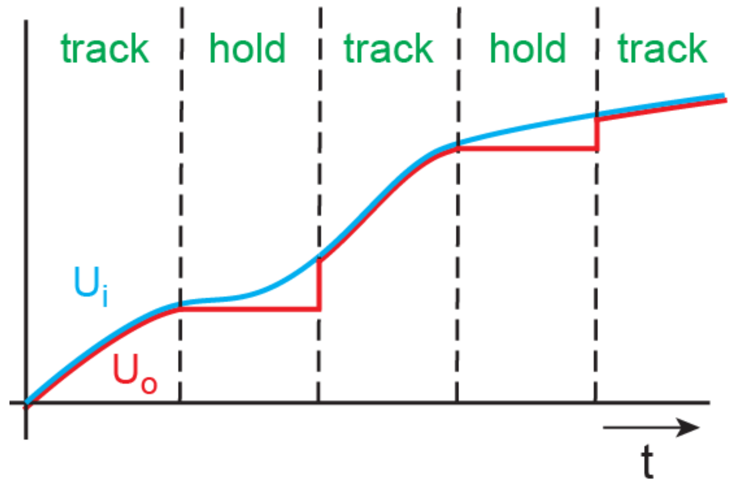


- When U_i changes during the conversion, large errors can occur



Sample-and-hold circuit

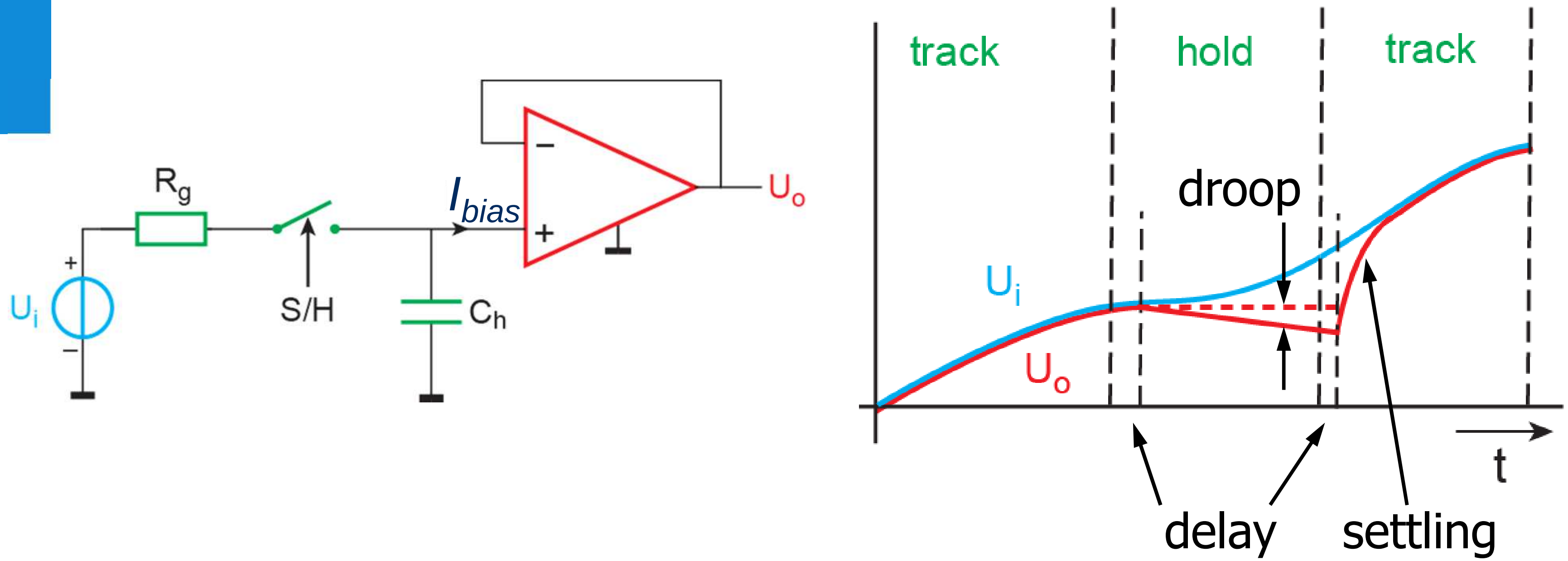




- track mode: output tracks input
- hold mode:
  - input voltage is stored in C_h
  - opamp manages buffering
- AD conversion during hold mode
  $\Rightarrow$ variations in U_i do not disturb conversion

what are possible sources of error here??

Sample-and-hold sources of error



- **Delay**: at transitions between track and hold
- **Droop**: charge leaking from C_h due to bias current (I_{bias})
- **Settling**: limited tracking speed due to R_g and switch resistance R_{on}: $\tau = (R_g + R_{on}) \cdot C_h$

Summary

- ADCs convert a time- and amplitude-continuous input signal into a sampled and quantized output signal
- During sampling, the input signal is only passed at discrete time instances
  - Here, **aliasing** can occur
  - To prevent this, sampling has to be done at least at the **Nyquist** frequency: 2 x the highest frequency component
- Quantization maps the amplitude onto a digital code
  - Here, **quantization errors** occur, due to finite **resolution**

Summary

- Three frequently used ADC types:
- **Direct** ADCs:
  - single-step conversion
  - example: flash ADC in a digital oscilloscope
- **Integrating** ADCs:
  - conversion with an intermediate step (e.g. in the time domain)
  - example: dual-slope ADC in a digital multi-meter
- **Compensating** ADCs:
  - input signal is compensated by the output of a DAC
  - example: SAR ADC in a sensor interface

What's next?

- Study:
  - Regtien sections 2.1.1, 2.2.3, 18, 15.2.2

- Practice:
  - See the practice exercises on Blackboard!

- Questions, things unclear? Let me know!
  M.A.P.Pertijs@tudelft.nl

Next time:
Measurement instruments