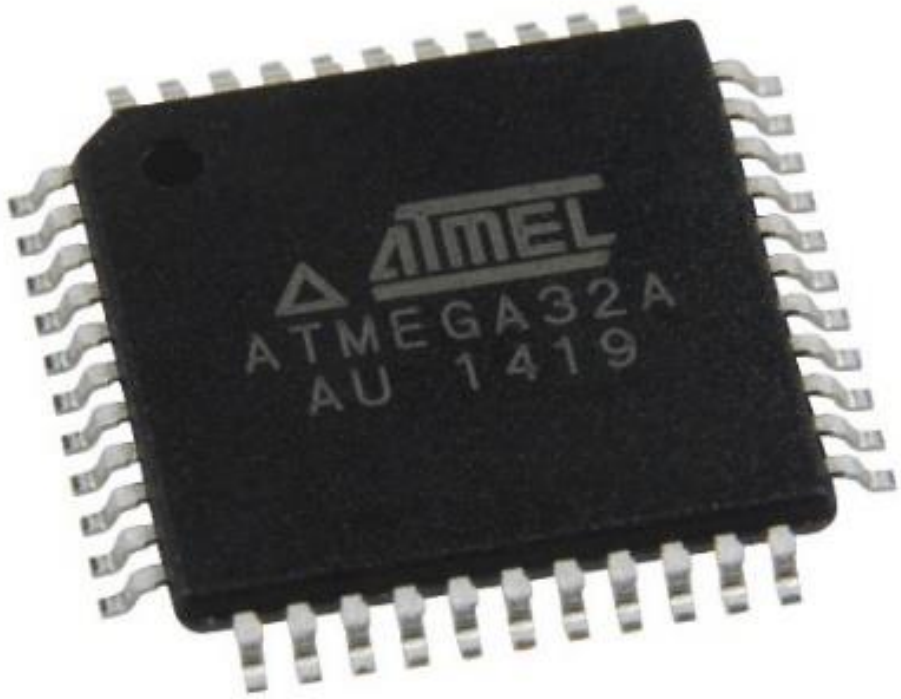
TECHNISCHE UNIVERSITÄT DARMSTADT

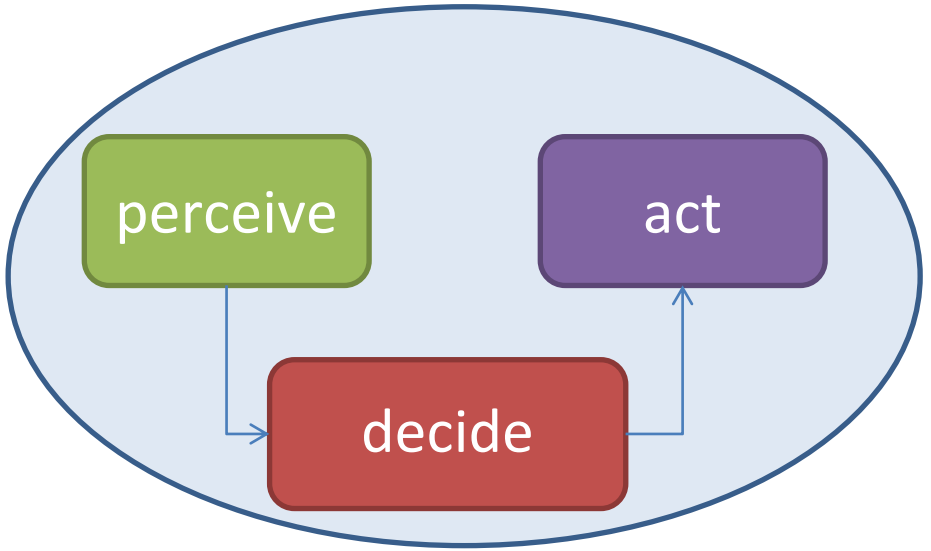
VORLESUNG 3 – MIKROCONTROLLER

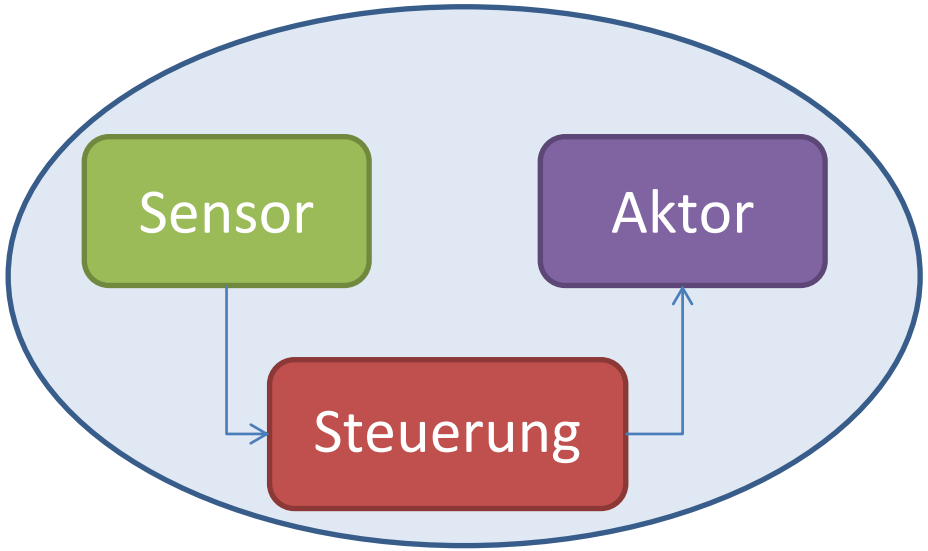
TECHNISCHE UNIVERSITÄT DARMSTADT

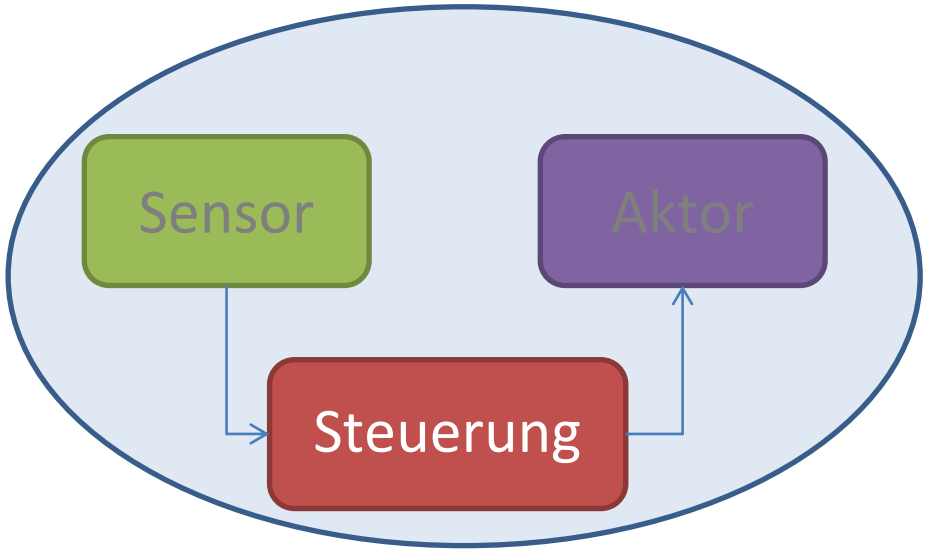
AGENDA

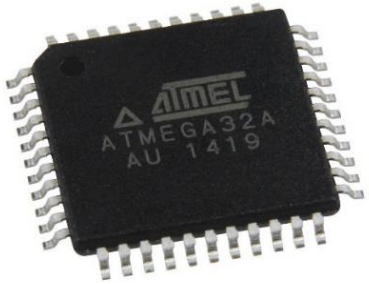
- Steuerungen im AmI-Kontext
- Mikrocontroller-Aufbau
  - Rechen- & Steuerwerk
    - Register
  - Speicherwerk
  - Interrupt-Steuerung
  - Ein-/Ausgabewerk
  - Analog-Digital-Umsetzer




perceive
act
decide


Sensor
Aktor
Steuerung


Sensor
Aktor
Steuerung


ATMEL
ATMEGA32A
AU 1419

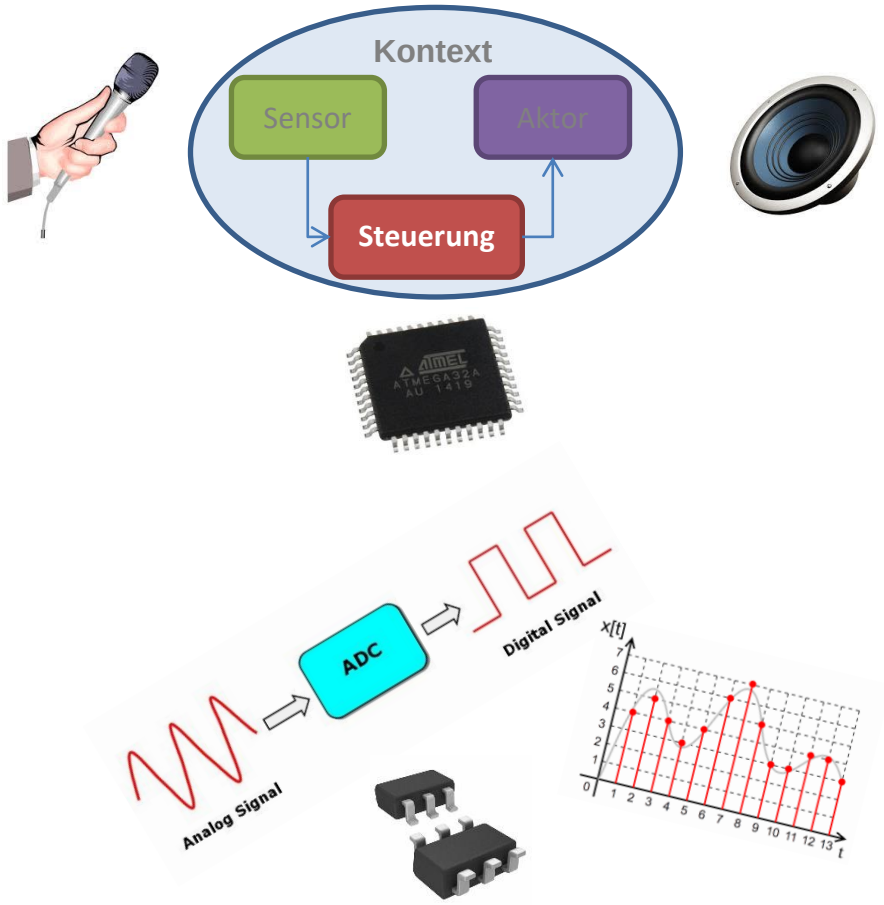
TECHNISCHE UNIVERSITÄT DARMSTADT

MIKROCONTROLLER

TECHNISCHE UNIVERSITÄT DARMSTADT

ANFORDERUNGEN

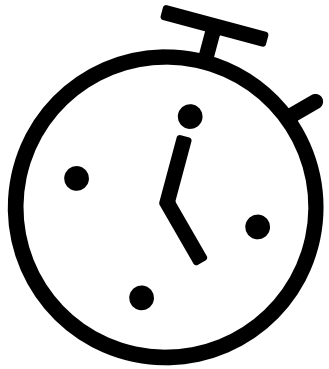
- Eingabe
  - Schnittstellen für Sensorik
  - Aufnahme von Messwerten (=Elektrische Signale von Sensor)
  - Analog-Digital-Wandlung
- Ausgabe
  - Ansteuerung von Aktoren
- Echtzeit-Anwendungen



TECHNISCHE UNIVERSITÄT DARMSTADT

ECHTZEIT

Echtzeit ≠ Schnell



TECHNISCHE UNIVERSITÄT DARMSTADT

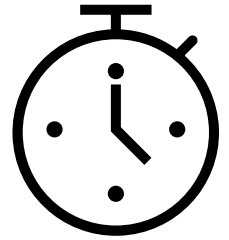
„UNTER ECHTZEIT VERSTEHT MAN …, DASS DIE VERARBEITUNGSERGEBNISSE INNERHALB EINER VORGEGEBENEN ZEITSPANNE VERFÜGBAR SIND.“

DIN 44300, Teil 9

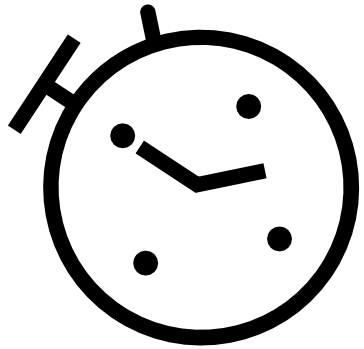
TECHNISCHE UNIVERSITÄT DARMSTADT

WEITERE DEFINITIONEN

- Harte Echtzeit:
  - Definierte Reaktionszeit wird garantiert und niemals überschritten
  - Überschreitung würde zu einem (schweren) Unfall führen!
  - Beispiel: Reaktorsteuerung, Autopilot

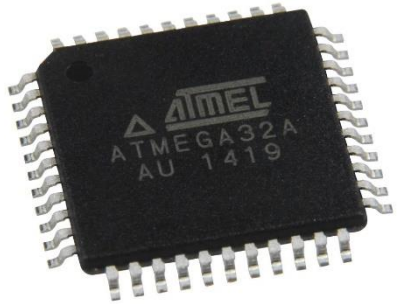


- Weiche Echtzeit:
  - Reaktionszeit wird nur statistisch garantiert
  - Überschreitung führt nicht zu Fehlern!
  - Beispiel: Streaming, Personal Computer (PC)



MIKROCONTROLLER: ABGRENZUNG

- Mikroprozessor: Steuerwerk + Rechenwerk (ALU)
- Mikrorechner: Mikroprozessor + Speicher-Werk + E-/A-Werk
  - Beispiel: PC
- Mikrocontroller (µC): Mikrorechner integriert auf einem Chip
  - Optimiert:
    - Für Steuerungs- oder Kommunikationsaufgabe
    - Auf niedrige Leistungsaufnahme
    - Für harte Echtzeitanforderungen



Rechenwerk

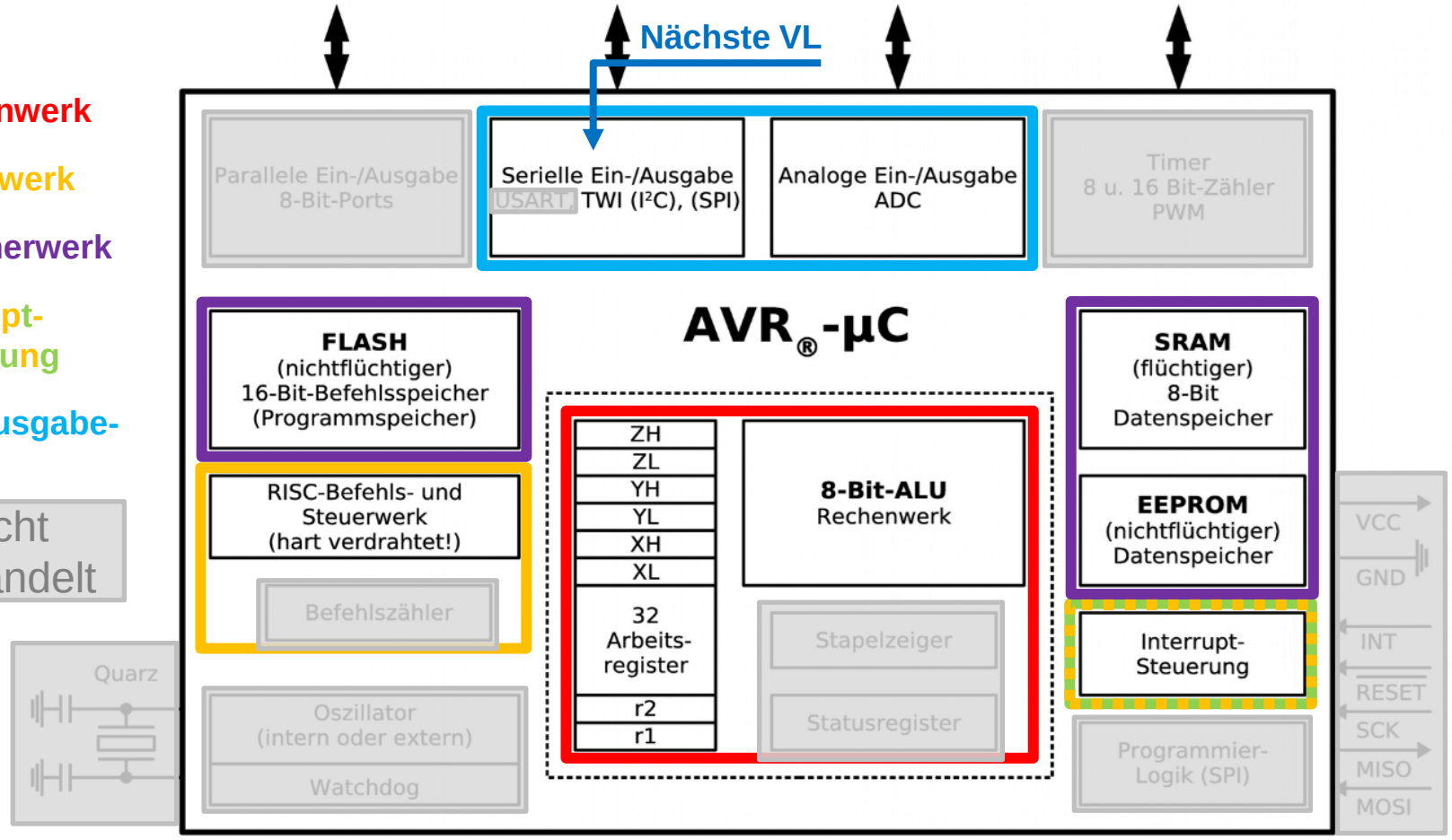
Steuerwerk

Speicherwerk

Interrupt-Steuerung

Ein-/Ausgabe-werk

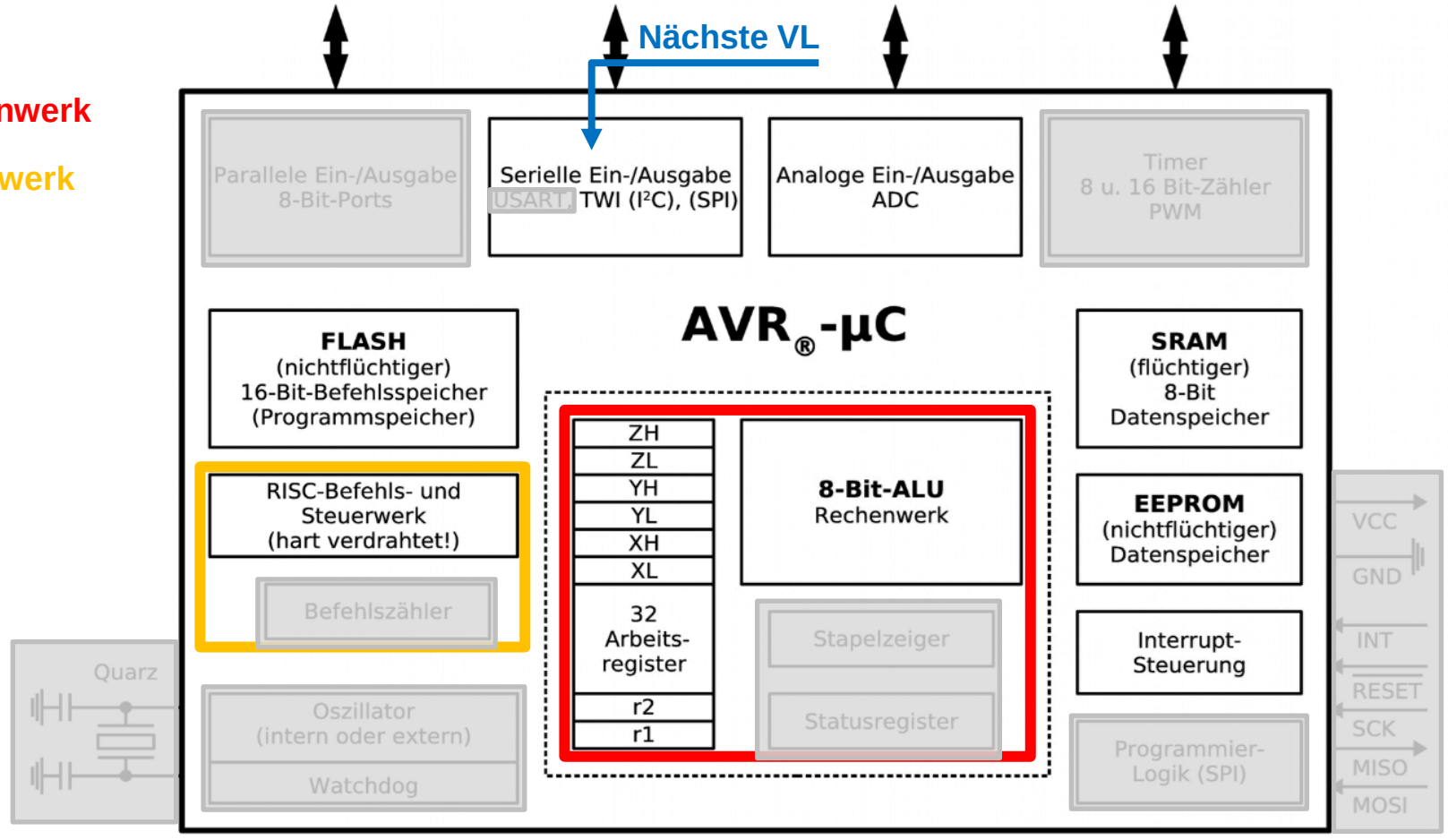
Nicht Behandelt



Blockschaltbild eines AVR-µC, Quelle: Weigu.lu µC-Technik

Rechenwerk

Steuerwerk



Blockschaltbild eines AVR-µC, Quelle: Weigu.lu µC-Technik

TECHNISCHE UNIVERSITÄT DARMSTADT

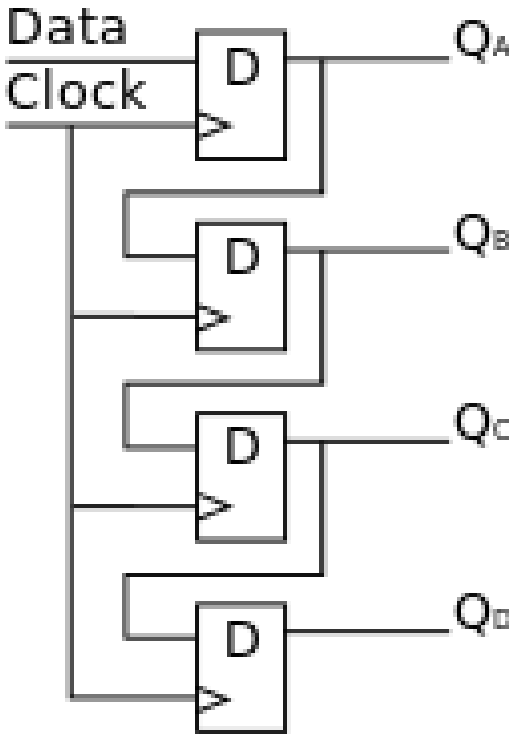
RECHEN- & STEUERWERK

- Arithmetic Logic Unit (ALU) / Rechenwerk:
  - Führt logische und mathematische Operationen aus
  - Statusregister: zeigen Statusinformationen zur vorangegangenen Operation an

- Steuerwerk
  - Steuert den Ablauf der Befehlsverarbeitung:
    - Dekodiert Befehle
    - Versorgt die ALU mit Daten und Befehlen
    - Wertet Statusregister der ALU aus
    - Leitet Ergebnisse der ALU an den Speicher weiter
    - Steuert weitere Funktionseinheiten des µC (z.B. Schnittstellen)

REGISTER

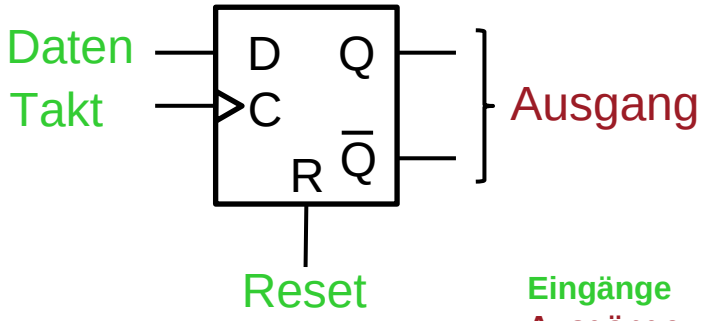
- Speicher (flüchtig!)
- Spitze der Speicherhierarchie: besonders schneller Zugriff
- ALU hat i.d.R. direkten Zugriff
- Input & Output (I/O)
  - -> Pin-Zustände
- Besteht aus FlipFlops



TECHNISCHE UNIVERSITÄT DARMSTADT

BEISPIEL: D-FLIPFLOP (D-FF)

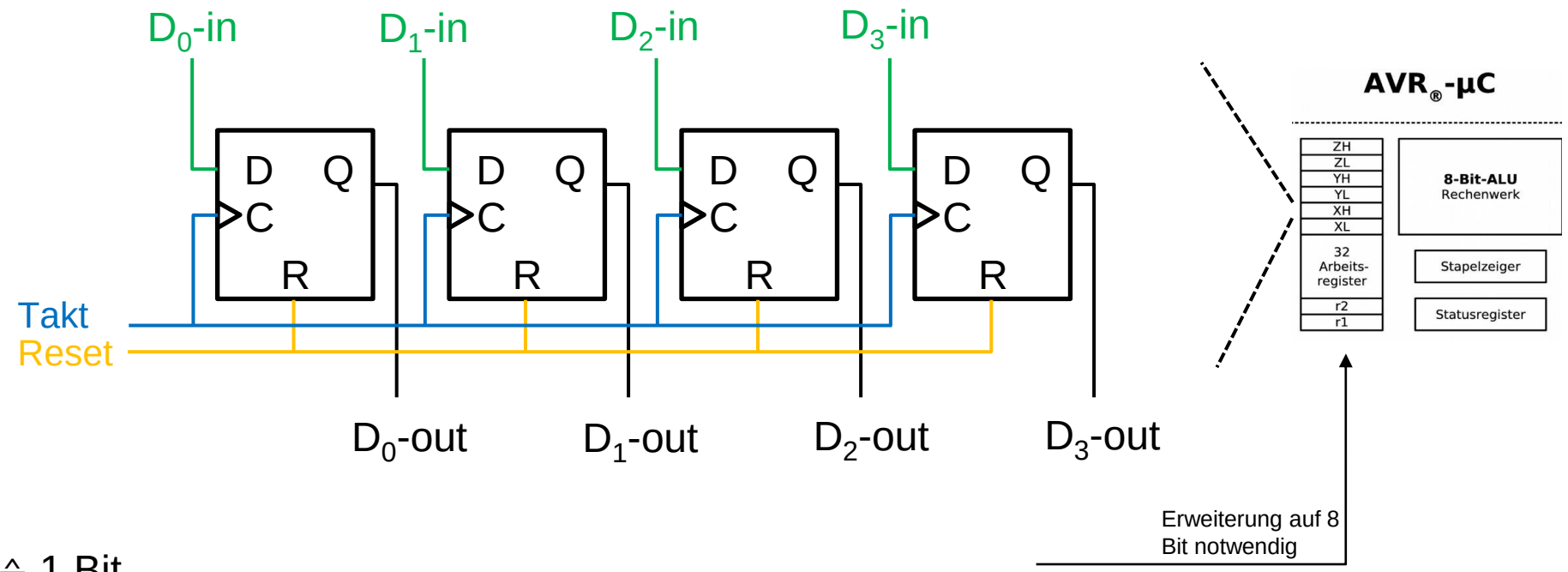
Schaltbild



Wahrheitstablle

D	C	Q	$\bar{Q}$	Funktion
0	0	*	*	Speichern
0	1	0	1	Rücksetzen
1	0	*	*	Speichern
1	1	1	0	Setzen

→ Weiter Informationen: Elektroniktutor - Register



- 1 D-FF ≙ 1 Bit
- Registertiefe = Anzahl paralleler FlipFlops (hier 4 Bit)
- Meist mit zusätzlicher Registerauswahl-Leitung Realisiert
- Weitere Registertypen: Schieberegister, Ringregister,…

→ Weiter Informationen: Elektroniktutor - Register

TECHNISCHE UNIVERSITÄT DARMSTADT

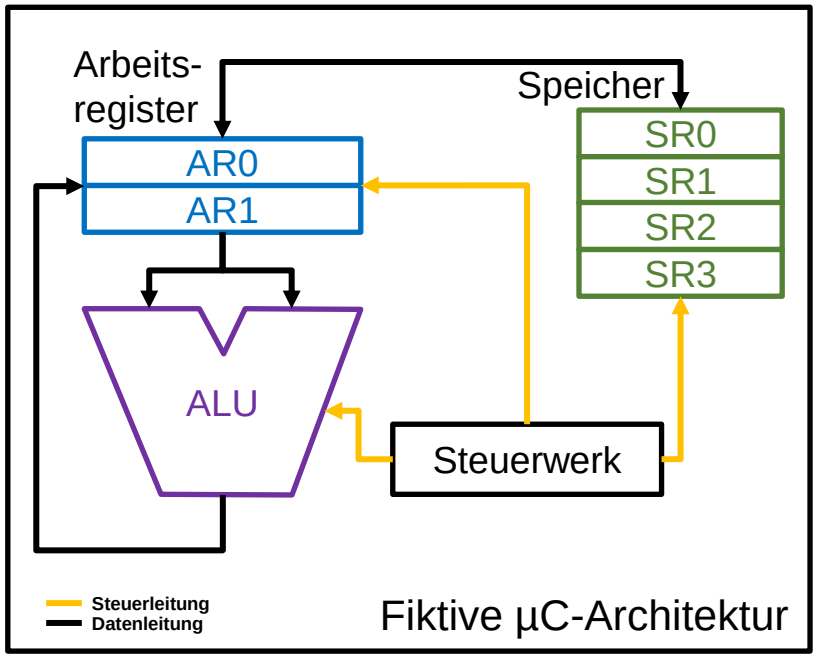
BEFEHLSABARBEITUNG

Beispiel-Aufgabe: SR2 = SR0 + SR3

Mikrocontroller-Daten:

- Arbeitsregister (AR): AR0, AR1
- Speicherregister (SR): SR0 bis SR3

Befehl	Bedeutung
load A B	Register A = Speicher an der Stelle B
add A B	Register A = Register A + Register B
sub A B	Register A = Register A - Register B
move A B	Speicher an der Stelle A = Register B



TECHNISCHE UNIVERSITÄT DARMSTADT

BEFEHLSABARBEITUNG

Beispiel-Aufgabe: SR2 = SR0 + SR3

Pseudo-Code:

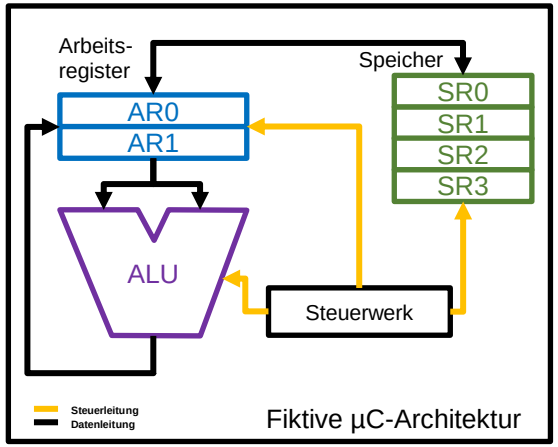
```
(1)     load    AR0     SR0
(2)     load    AR1     SR3
(3)     add     AR0     AR1
(4)     move    SR2     AR0
```

Steuerwerk belädt AR0 mit SR0

Steuerwerk belädt AR1 mit SR3

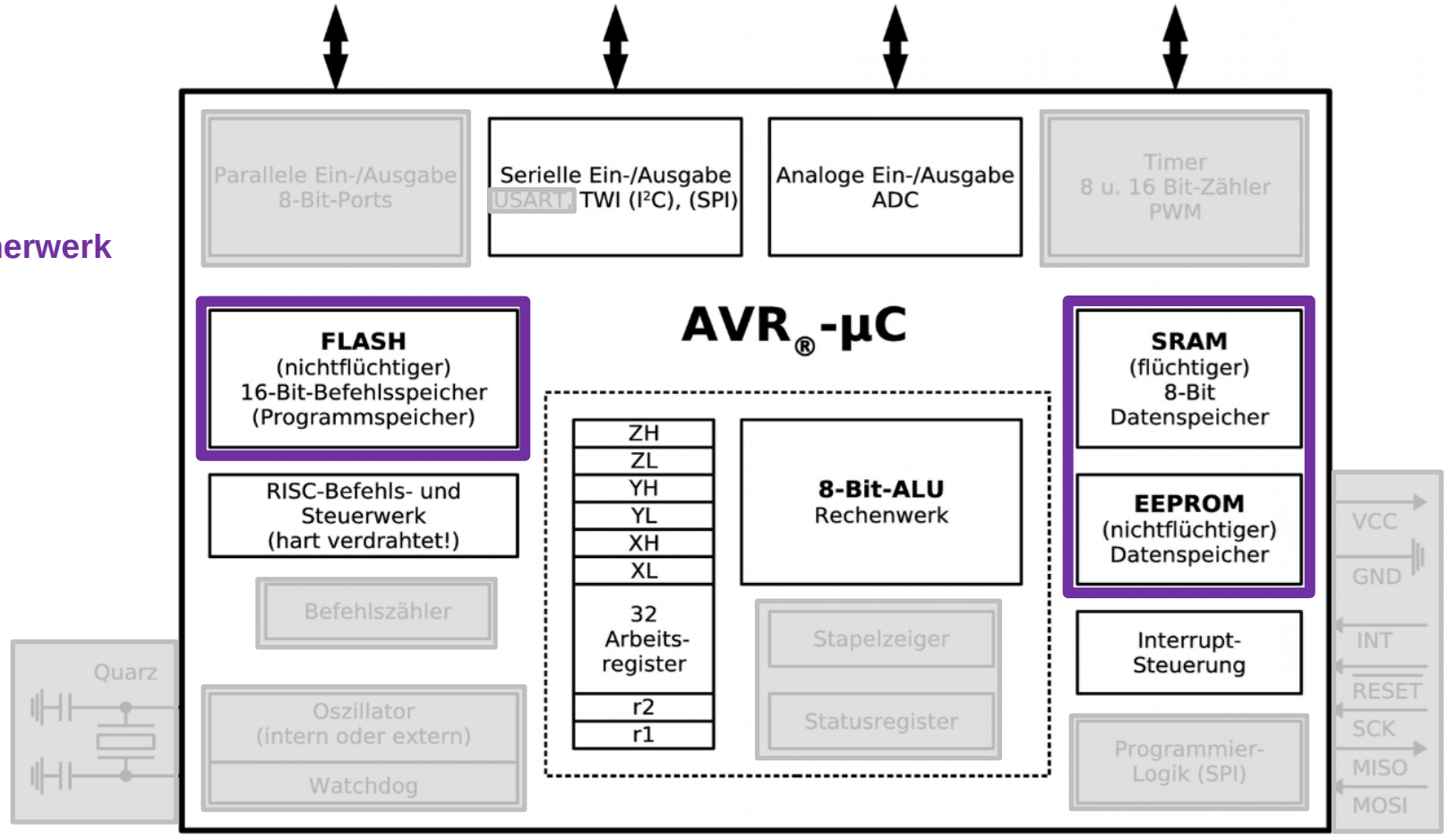
Steuerwerk stellt ALU auf Addition und Zieladresse AR0 ein
ALU führt Addition durch und speichert Ergebnis in AR0

Steuerwerk speichert Wert von AR0 in SR2



Befehl	Bedeutung
load A B	Register A = Speicher an der Stelle B
add A B	Register A = Register A + Register B
sub A B	Register A = Register A - Register B
move A B	Speicher an der Stelle A = Register B

Speicherwerk

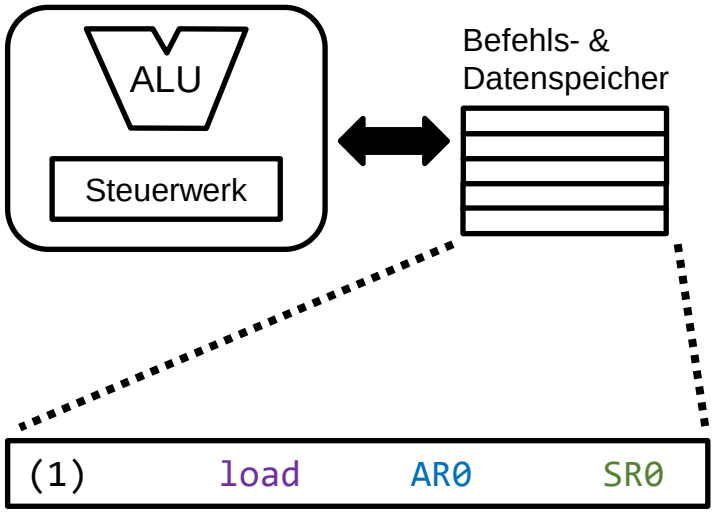


Blockschaltbild eines AVR-µC, Quelle: Weigu.lu µC-Technik

SPEICHERWERK

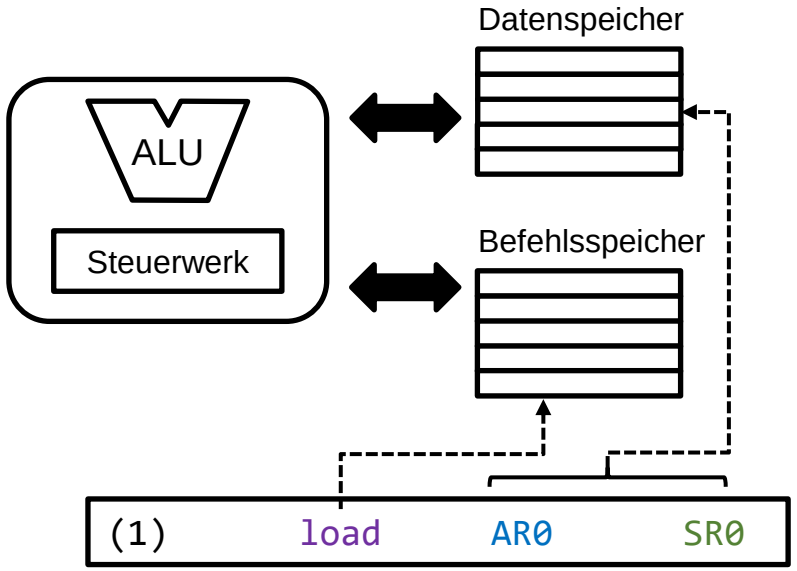
- Speicherung und Bereitstellung von Daten und Befehlen
- Zwei verbreitete Architekturen
  - Von-Neumann
  - Harvard

Von-Neumann-Architektur:



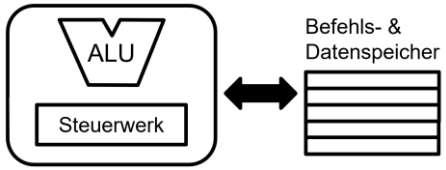
Beispielcode

Harvard-Architektur:



Beispielcode

VON-NEUMANN



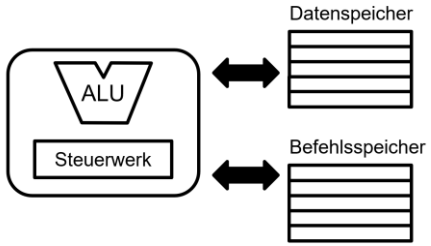
Vorteile:

- Einfach realisierbar
- Hohe Flexibilität:
  - Freier Speicher kann für Daten und Befehle verwendet werden
- Niedrigere Kosten:
  - Geringerer Verdrahtungsaufwand
  - Einfacheres Steuerwerk

Nachteile:

- Verbindung zum Speicher als Flaschenhals
- Langsamer: konkurrierender Speicherzugriff, entweder Daten- oder Befehlscode
- Erzwungener Sequentialismus, keine Parallelität

HARVARD



Vorteile:

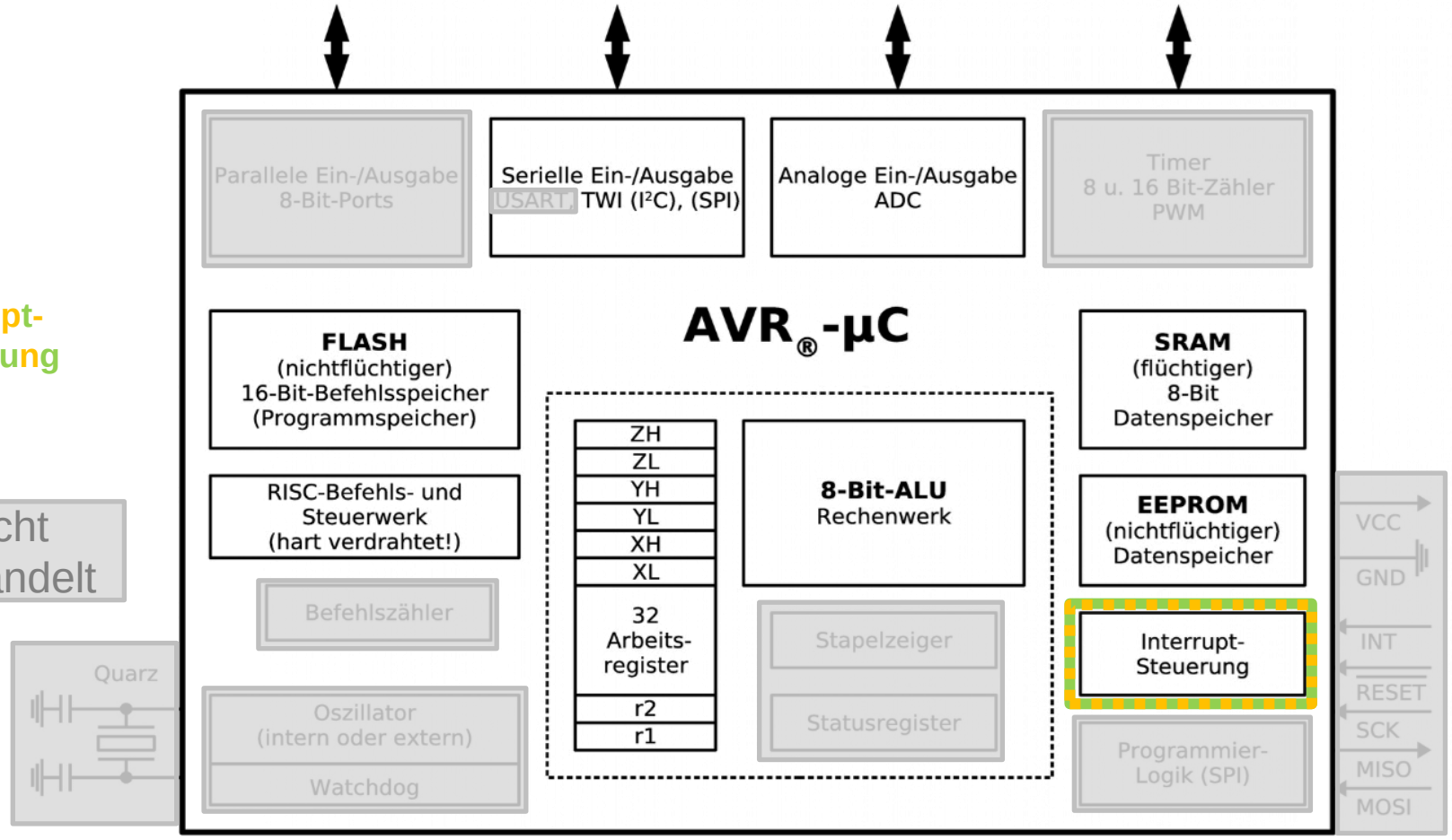
- Sicherheit: Code und Daten sind getrennt
  - Bestimmte Fehler/Angriffe sind unmöglich
  - Selbstmodifizierender Code durch unveränderlichen Speicher vermeidbar
- Buffer-Overflow (Daten überschreiben Code) ist unmöglich
- Schnell, da Daten- und Befehlsspeicher parallel abgefragt werden
- Daten- und Befehlsspeicher können unterschiedlich groß sein (Kostenfaktor)

Nachteile:

- Weniger flexibel: Freier Speicher für Daten / Befehle reserviert
- Teurer:
  Hoher Verdrahtungsaufwand
  Komplexes Steuerwerk

Interrupt-Steuerung

Nicht Behandelt



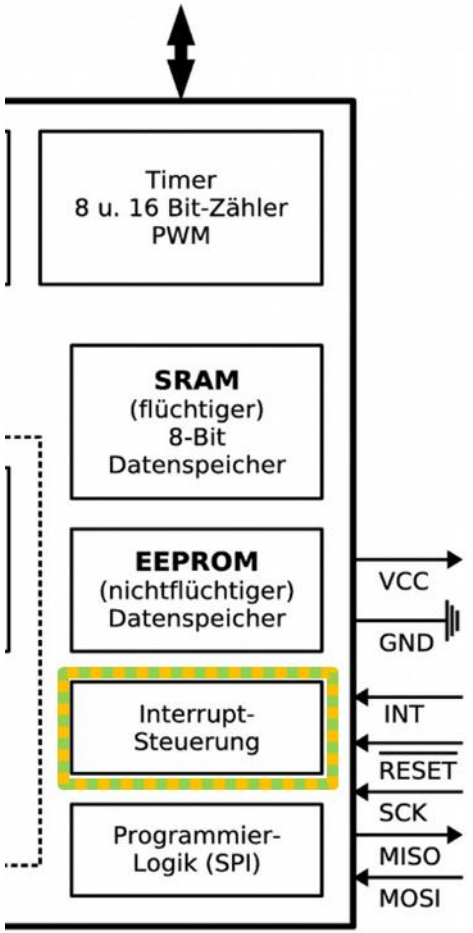
Blockschaltbild eines AVR-µC, Quelle: Weigu.lu µC-Technik

TECHNISCHE UNIVERSITÄT DARMSTADT

INTERRUPT-STEUERUNG

- Spezielle Komponente des Steuerwerks zur Behandlung von besonderen Ereignis
- Schnelle und flexible Reaktion auf Ereignisse

- Interrupt:
  - Asynchrone Unterbrechung des Programmablaufs:
  - Beim Eintreten wird eine vordefinierte Interrupt-Routine ausgeführt
  - Durch interne und externe Ereignisse auslösbar



TECHNISCHE UNIVERSITÄT DARMSTADT

INTERRUPT-ARTEN

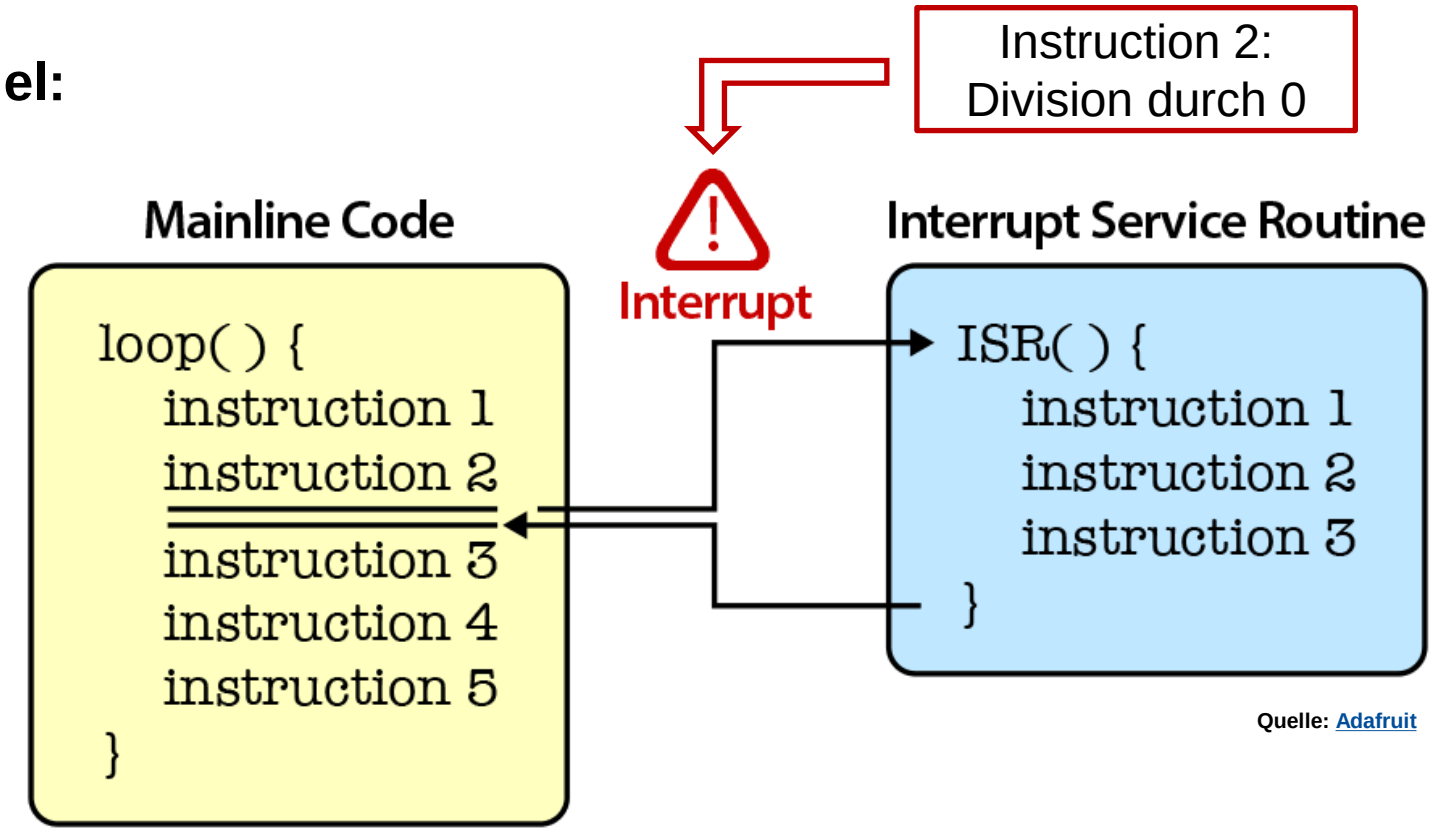
Interne Interrupts

- Durch µC-interne Ereignisse ausgelöst
- Zeitgeber:
  Nach Ablauf einer vorgegeben Zeit
- Schnittstellen:
  Beim Empfang von Daten
- ALU:
  Bei Rechen-Ereignissen (z.B. Division durch 0)

Externe Interrupts

- Durch µC-externe Ereignisse ausgelöst
- Nur an speziellen Eingängen/Schnittstellen möglich
- Meist ausgelöst durch Zustandswechsel
  (0 -> 1 oder 1 -> 0)

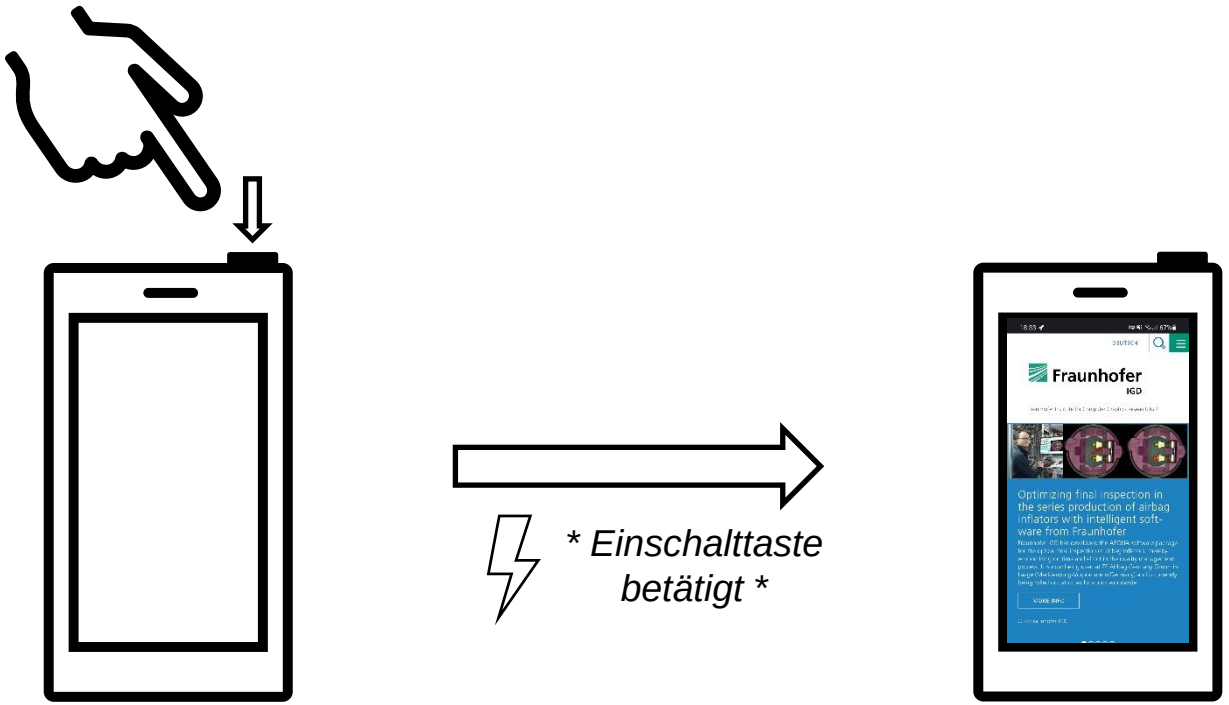
Beispiel:



Quelle: Adafruit

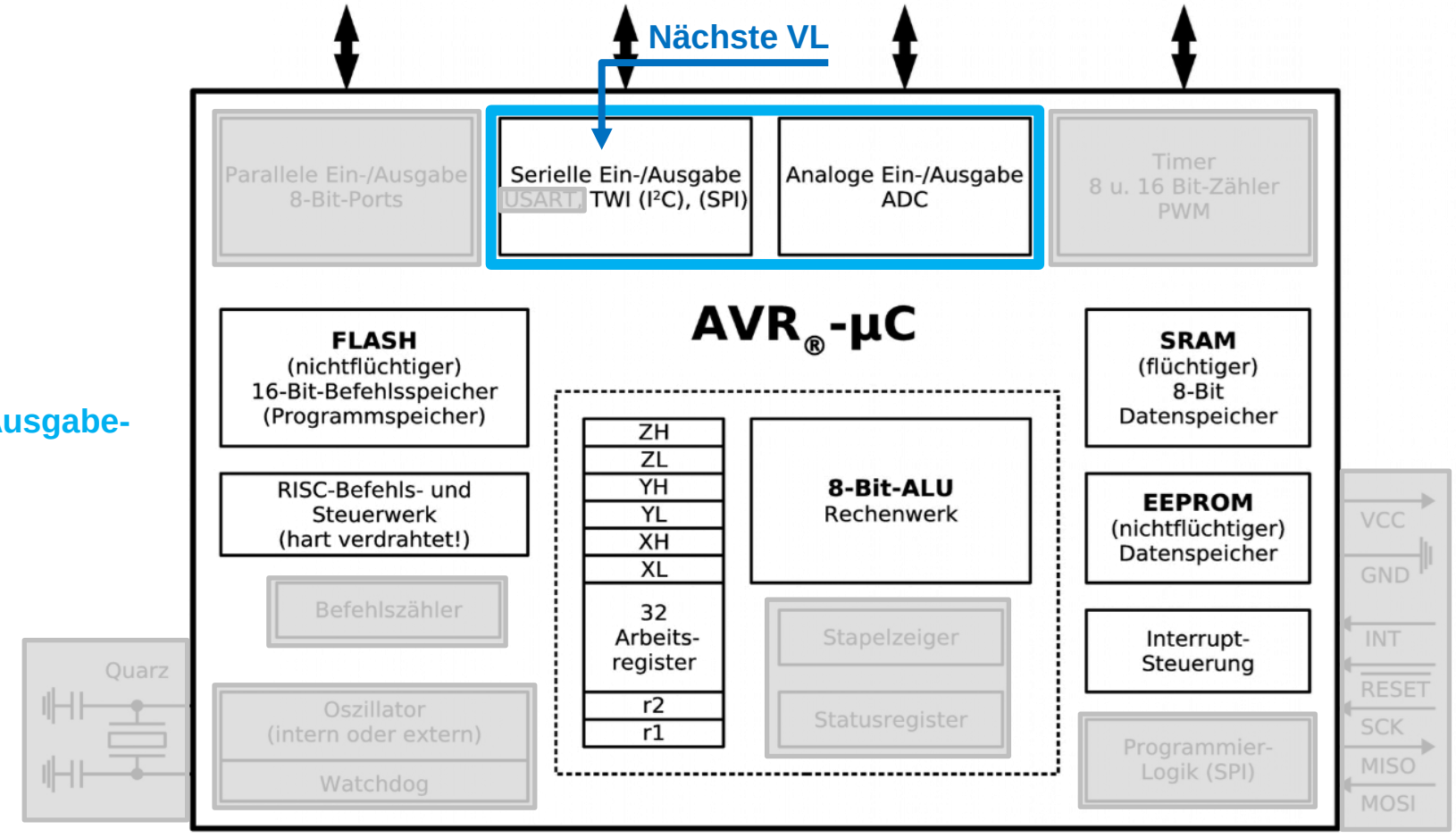
Welche Art von Interrupt wird hier dargestellt? → Internes Interrupt

Beispiel:



Welche Art von Interrupt wird hier dargestellt? → Externes Interrupt

Ein-/Ausgabe-werk



Blockschaltbild eines AVR-µC, Quelle: Weigu.lu µC-Technik

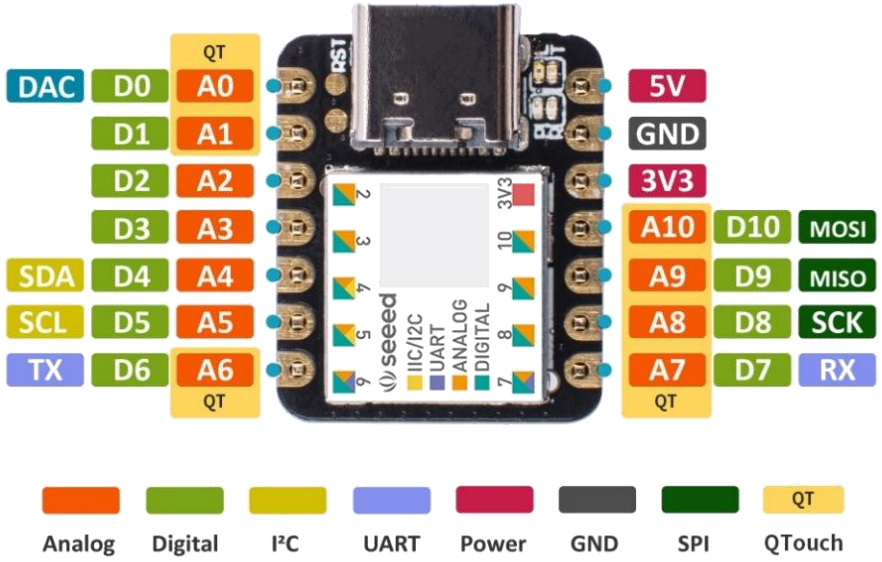
EINGABE-/AUSGABEWERK

- Bindeglied zur Umwelt – Ein- und Ausgabe von Daten:
- Zum Anwender (über Bildschirm u. Eingabegeräte)
- Zu anderen Systemen (über Schnittstellen)

- Schnittstellen: I²C, SPI, UART, …

- Digitale Ein-/Ausgänge:
- Logisches Schalten (0 / 1)
- Bspw. Ein-/Ausschalter, …

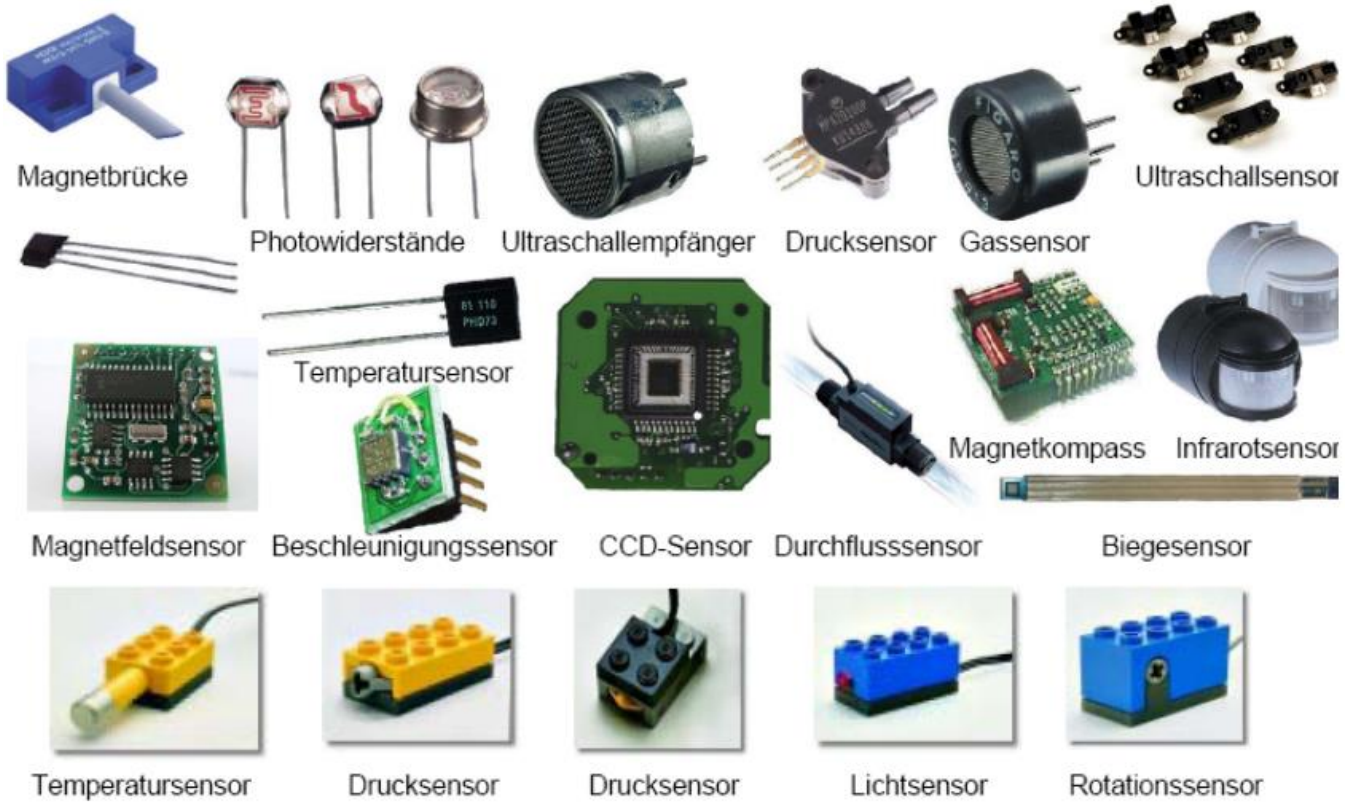
- Analoge Ein-/Ausgänge



µC: SAMD21G18, Quelle: Seeedstudio

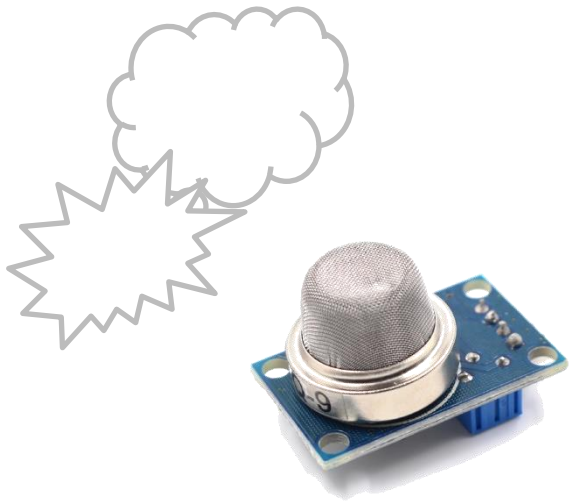
TECHNISCHE UNIVERSITÄT DARMSTADT

ANALOG-DIGITAL-UMSETZER

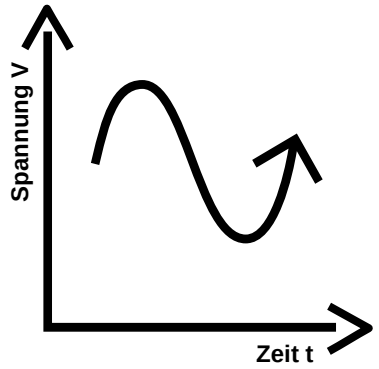


ANALOG-DIGITAL-UMSETZER

nicht-elektrisches, analoges Signal → elektrisches, analoges Signal



Gas-Sensor,
Quelle: Makerfabs



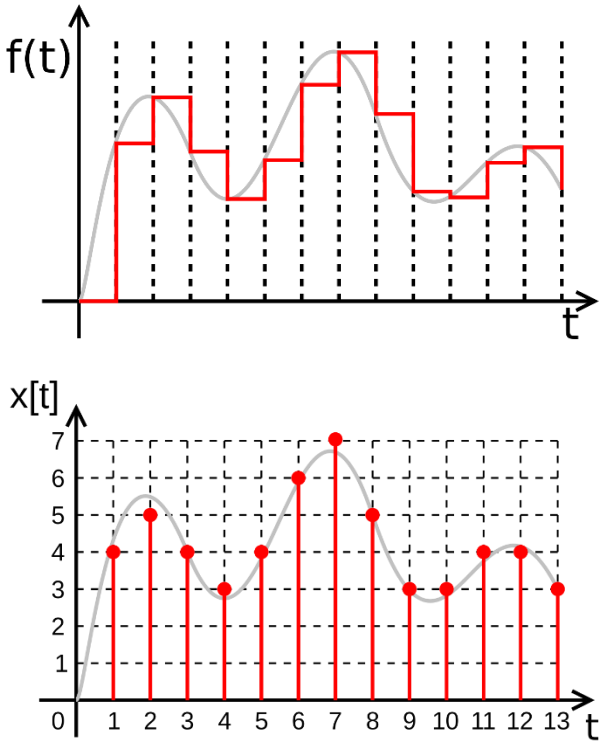
TECHNISCHE UNIVERSITÄT DARMSTADT

ANALOG-DIGITAL-UMSETZER

- Auch A/D-Wandler oder ADC (analog-to-digital converter)
- Elementarer Bestandteil von Mikrocontrollern:
  - Ermöglicht die „Wahrnehmung der Umwelt“

Digitalisierung des analogen Signals:

- **Zeitliche Diskretisierung**: (vertikale Linien)
- Zerlegung des Analogsignals in eine zeitdiskrete Signalfolge
- **Quantisierung**: (horizontale Linien)
- Zerlegung der zeitdiskrete Signalfolge in eine zeit- und wertdiskrete Folge

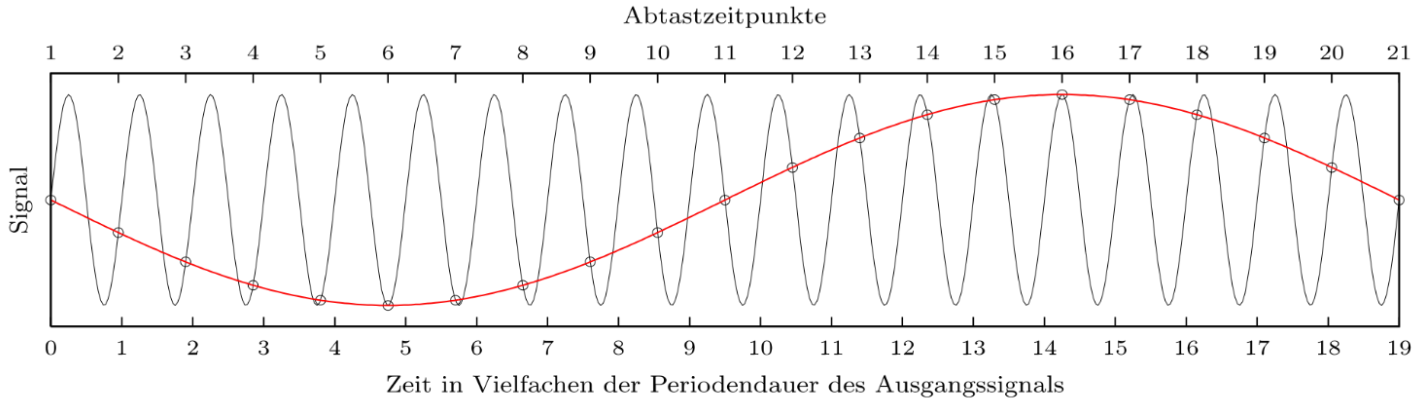


Abbildungsquelle: [Wikipedia - ADC]

TECHNISCHE UNIVERSITÄT DARMSTADT

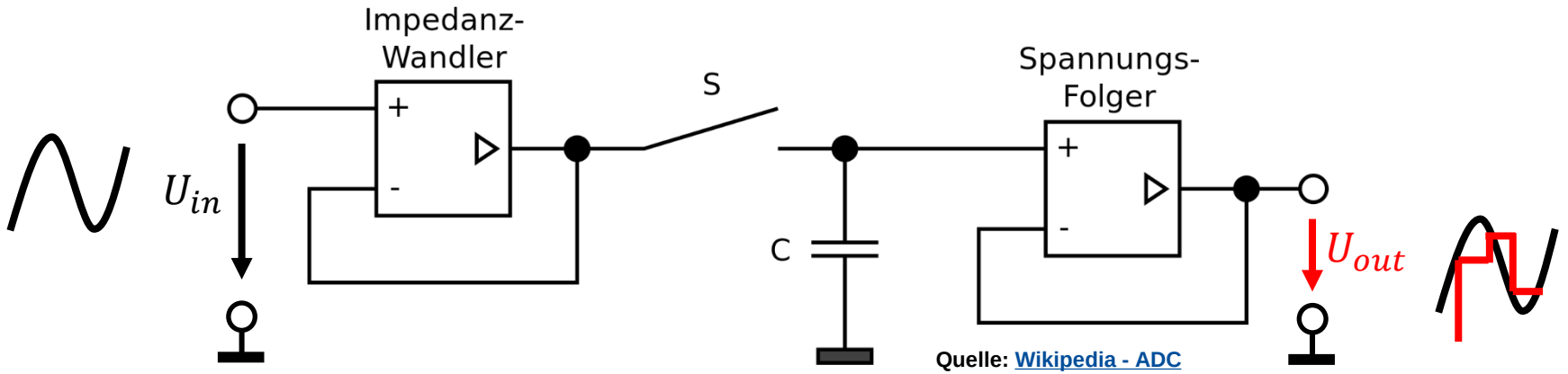
ZEITLICHE DISKRETISIERUNG

- Zerlegung des Analogsignals in eine zeitdiskrete Signalfolge
- Nyquist-Shannon-Abtasttheorem für bandbegrenzte Signale:
  $f_{abtast} \geq 2 \cdot f_{max}$
- Nicht-Beachtung des Abtasttheorems: Unterabtastung (mit ggf. Aliasing-Effekt)



ZEITLICHE DISKRETISIERUNG

- Realisierung mittels Sample-Hold-Glied:
- Kondensator hält in der Haltephase (Schalter S offen) die Eingangsspannung U_in konstant
- Spannungsfolger: verhindert Entladung des Kondensators C
- In der Ladephase (Schalter S geschlossen) wird der Kondensator C geladen
- Impedanzwandler: stromfreie Spannungsmessung am Messort



Quelle: Wikipedia - ADC

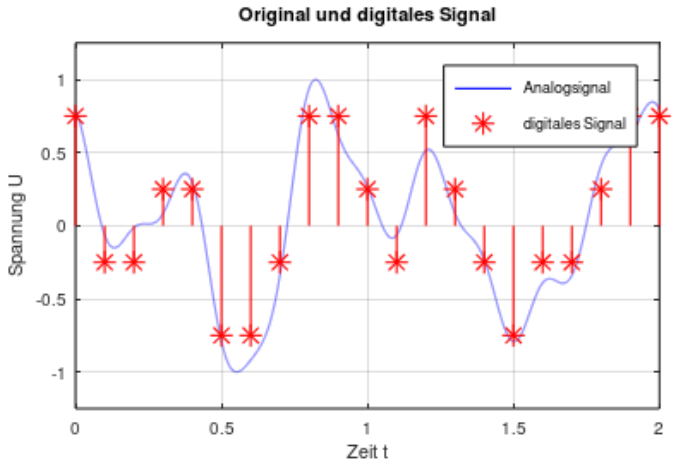
TECHNISCHE UNIVERSITÄT DARMSTADT

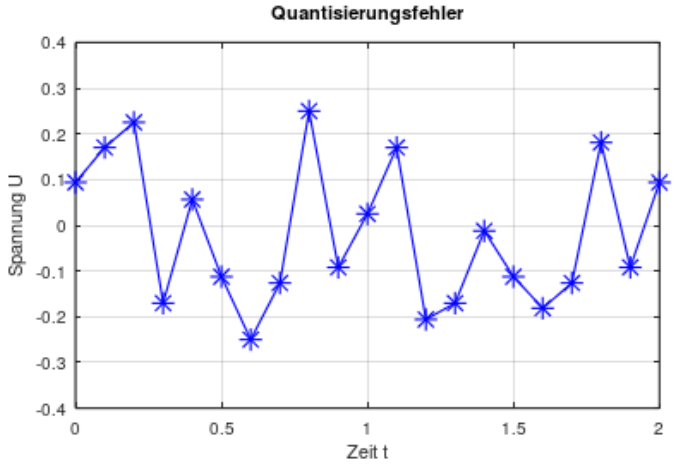
QUANTISIERUNG

- Zerlegung des zeitdiskreten in eine zeit- und wertdiskrete Folge

- Quantisierungsabweichung bzw. Quantisierungsfehler
- Abweichung des digitalen vom analogen Signal
- Abhängig von der Auflösung des ADCs

- ADC-Auflösung = $2^{Bit-Angabe}$
- Beispiel rechts: 2-Bit-ADC (4 diskrete Werte)



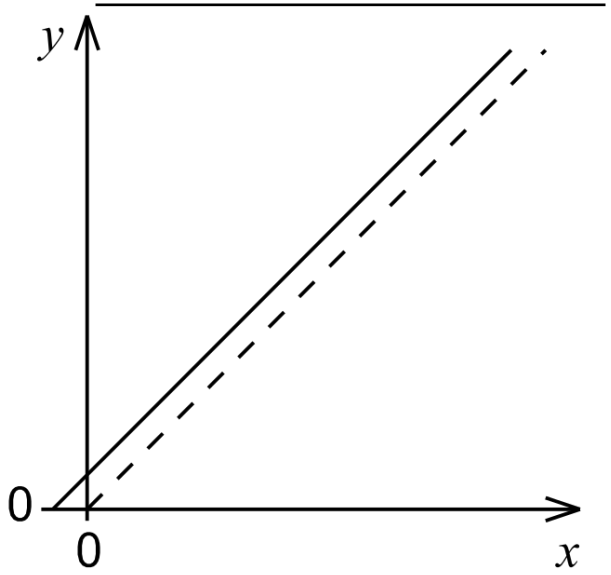


TECHNISCHE UNIVERSITÄT DARMSTADT

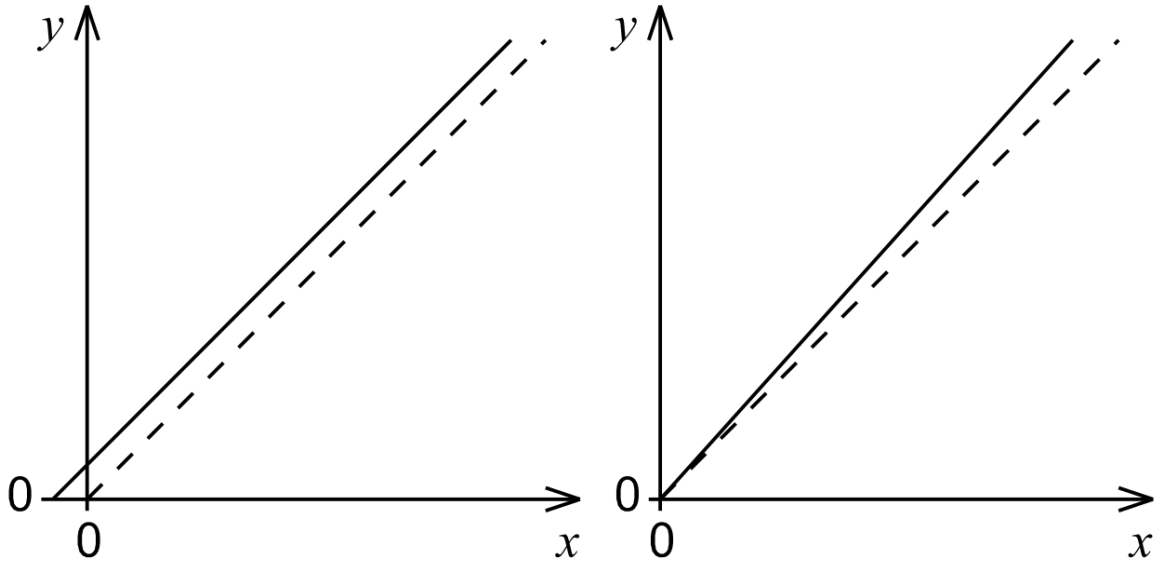
Nullpunktfehler

kontinuierlicher Offset

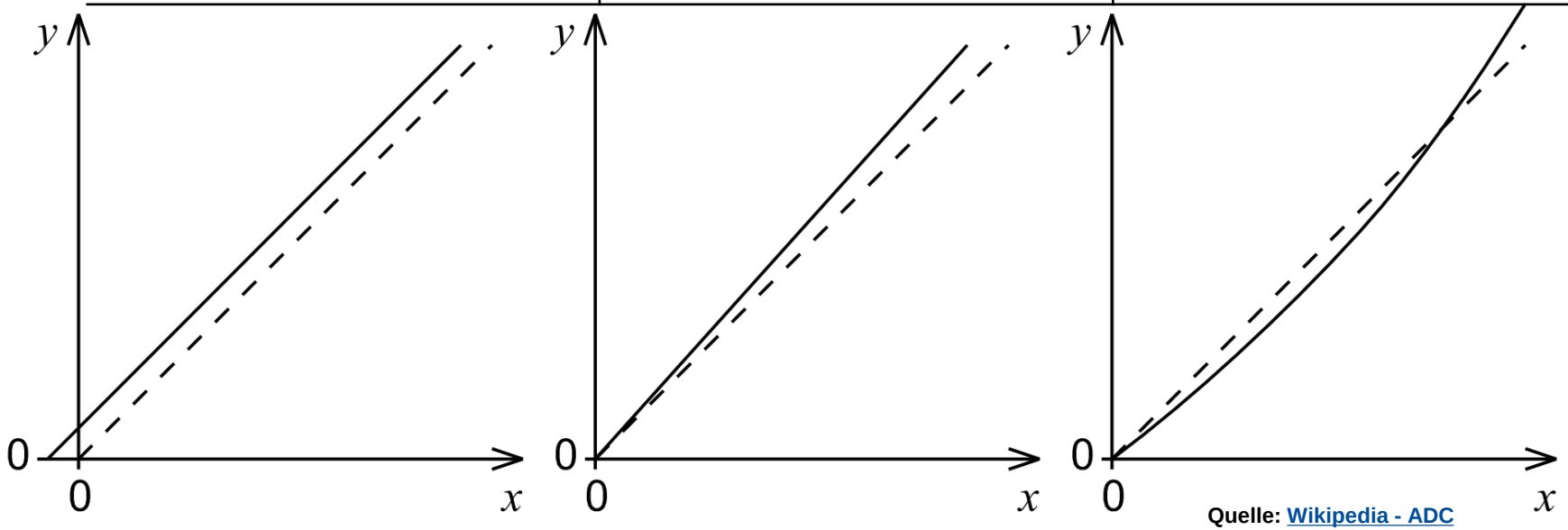
Behebung:
Offset bei Nulleingang bestimmen



Nullpunktfehler	**Verstärkungsfehler**
kontinuierlicher Offset	multiplikativer Fehler
Behebung: Offset bei Nulleingang bestimmen	Behebung: Vergleich von Null mit Vollausschlag



Nullpunktfehler	Verstärkungsfehler	Nichtlinearitätsfehler
kontinuierlicher Offset	multiplikativer Fehler	Fehler als Funktion f(x)
Behebung: Offset bei Nulleingang bestimmen	Behebung: Vergleich von Null mit Vollausschlag	Behebung: Herstellerkalibration anwenden, linearen Bereich nutzen



Quelle: Wikipedia - ADC

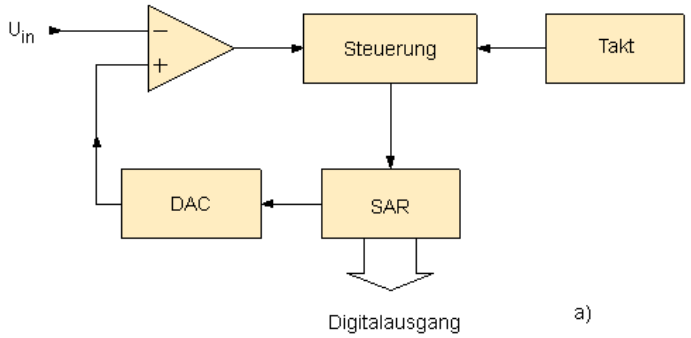
SUKZESSIVE APPROXIMATION

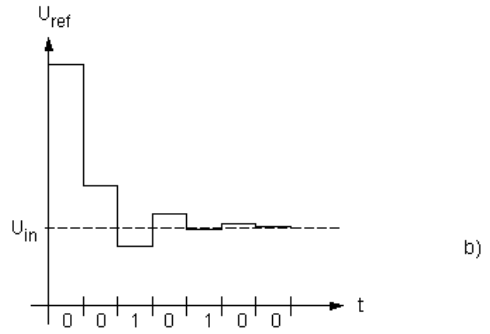
→ U_{in} wird in k Schritten approximiert:

- Über den DAC wird eine Spannung U_{ref} erzeugt
- $U_{ref}^{(n)}$ wird im Komparator mit U_{in} verglichen:
  - $\boldsymbol{U_{ref}^{(n)} < U_{in}}$:
    $U_{ref}^{(n+1)}$ in nächster Iteration um $\frac{U_{ref}^{(n)}}{2}$ hochsetzen
  - $\boldsymbol{U_{ref}^{(n)} \geq U_{in}}$:
    $U_{ref}^{(n+1)}$ in nächster Iteration um $\frac{U_{ref}^{(n)}}{2}$ runtersetzen

$(n+1) < k$?

- Anzahl k der Iterationen entspricht der Auflösung des DAC (typischerweise 12 – 14 Bit)

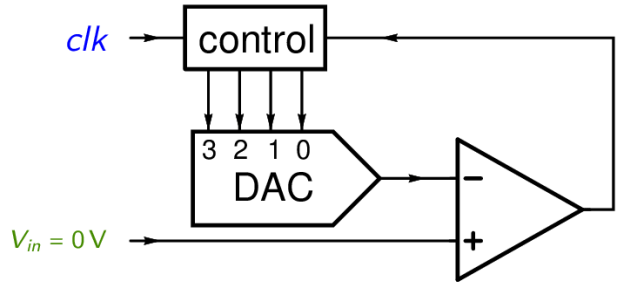


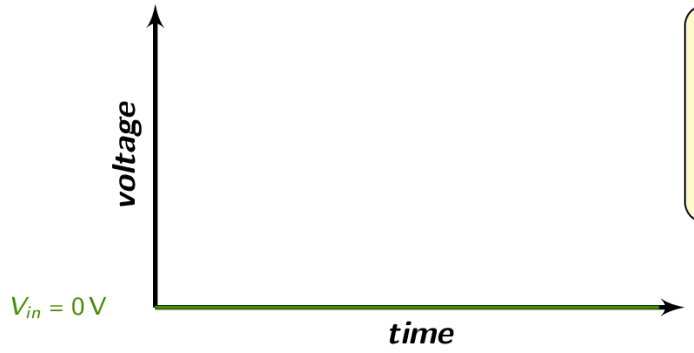


TECHNISCHE UNIVERSITÄT DARMSTADT

SUKZESSIVE APPROXIMATION

Successive Approximation – example of a 4-bit ADC





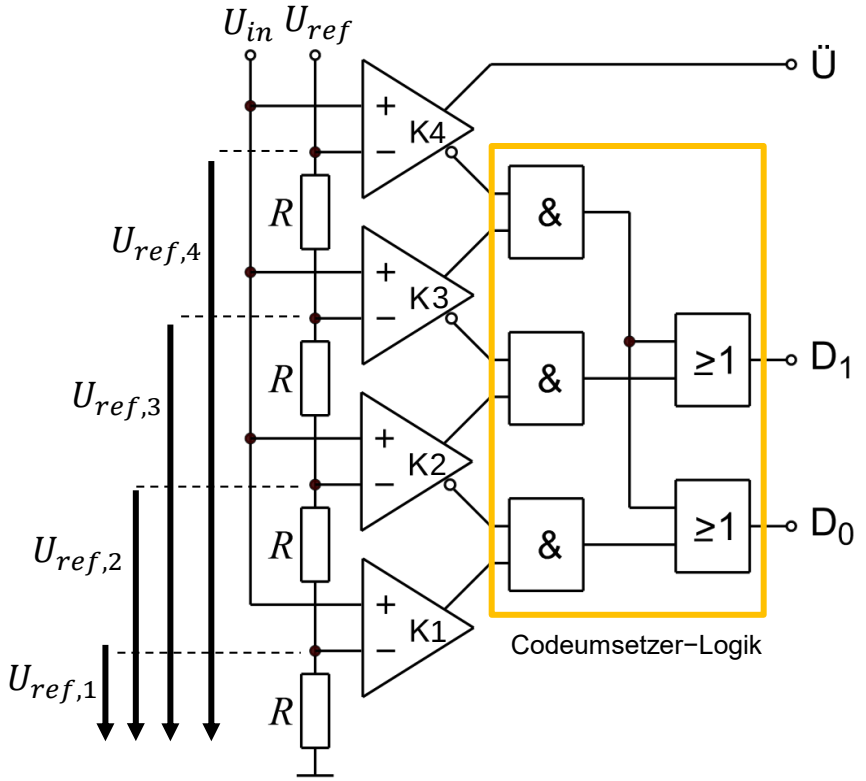
Resolution:	
5 V × 1/2	2.5000 V
5 V × 1/4	1.2500 V
5 V × 1/8	0.6250 V
5 V × 1/16	0.3125 V
...	
5 V × 1/1024	0.0049 V

Quelle: [Wikipedia - SAR-ADC]

FLASH-/PARALLEL-UMSETZER

- Direkte Messung der Spannung mittels Komparator-Logik-Schaltung
  - Spannung U_{ref} wird an den Vorwiderständen R in Teilspannung $U_{ref,n}$ zerkleinert
    - Teilspannung $U_{ref,n} = U_{ref} \cdot \frac{n}{n_{ges}}$;

      $n \coloneqq Nummer\ d.Komparatorstufe$
  - U_{in} wird über die Komparatoren K mit den Teilspannungen $U_{ref,n}$ verglichen
  - Codeumsetzer-Logik erzeugt digitales Signal



2-Bit-Flash-ADC,
Quelle: Wikipedia - ADC

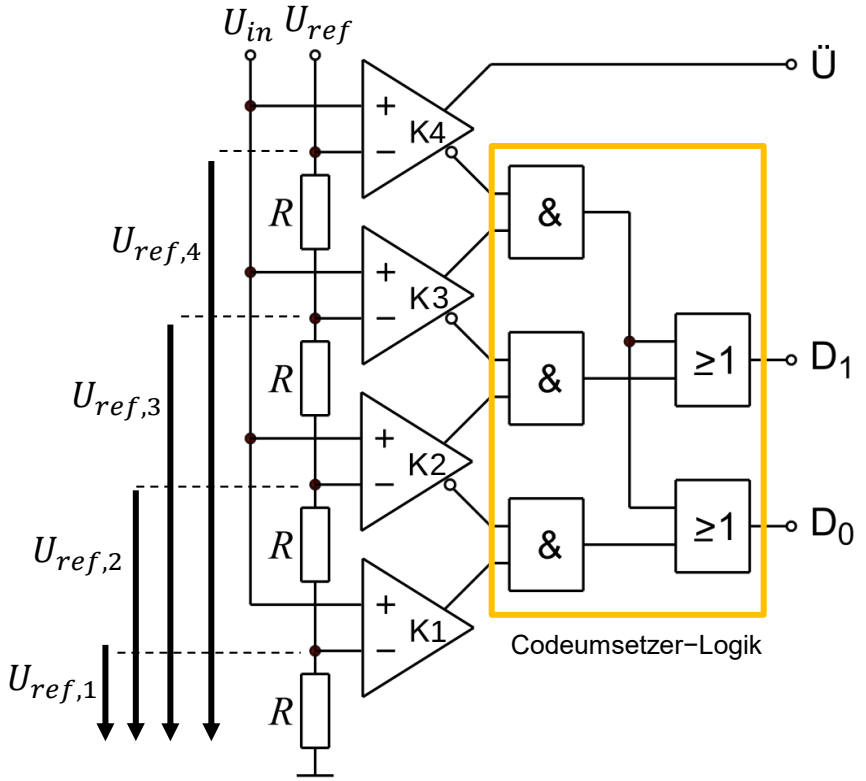
FLASH-/PARALLEL-UMSETZER

$U_{ref} = 4V; U_{in} = 2V;\ R = 10k\Omega;\ U_{ref,n} = U_{ref} \cdot \frac{n}{n_{ges}}$

Komparator (K)		
Bedingung	**Aus**	**Aus ○**
+ ≥ -	1	0
+ < -	0	1

ODER-Glied (≥1)		
In 1	**In 2**	**Aus**
0	0	0
0	1	1
1	0	1
1	1	1

UND-Glied (&)		
In 1	**In 2**	**Aus**
0	0	0
0	1	0
1	0	0
1	1	1

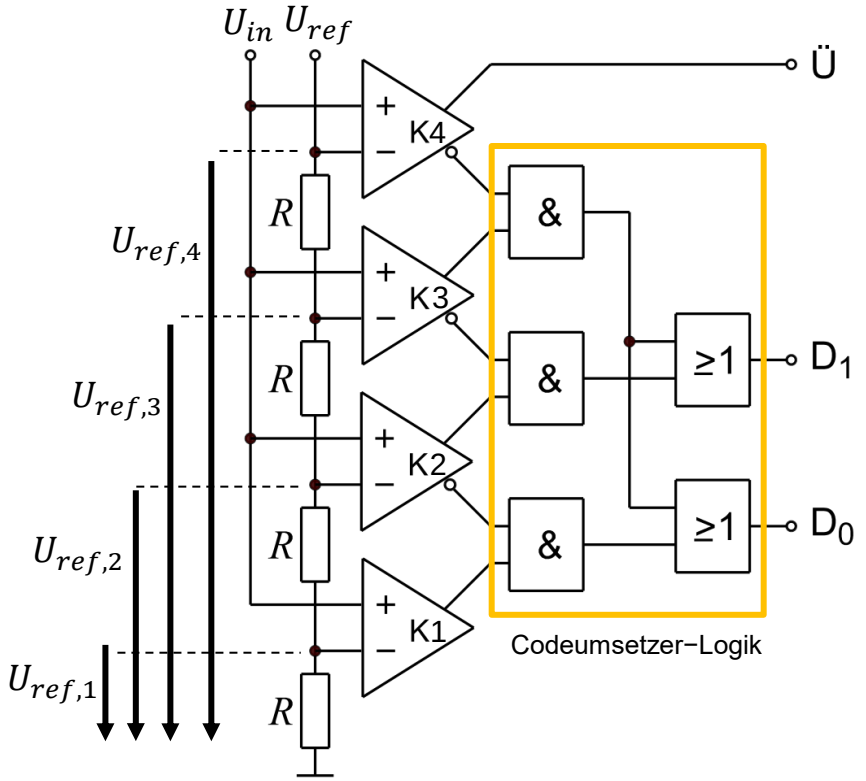


2-Bit-Flash-ADC,
Quelle: Wikipedia - ADC

FLASH-/PARALLEL-UMSETZER

$U_{ref} = 4V;\ \ U_{in} = 2V;\ \ R = 10k\Omega;\ \ U_{ref,n} = U_{ref} \cdot \frac{n}{n_{ges}}$:

- $U_{ref,4} = 4V \cdot \frac{4}{4} = 4V \rightarrow U_{ref,4} > U_{in}$
- $U_{ref,3} = 4V \cdot \frac{3}{4} = 3V \rightarrow U_{ref,3} > U_{in}$
- $U_{ref,2} = 4V \cdot \frac{2}{4} = 2V \rightarrow U_{ref,2} = U_{in}$
- $U_{ref,1} = 4V \cdot \frac{1}{4} = 1V \rightarrow U_{ref,1} < U_{in}$



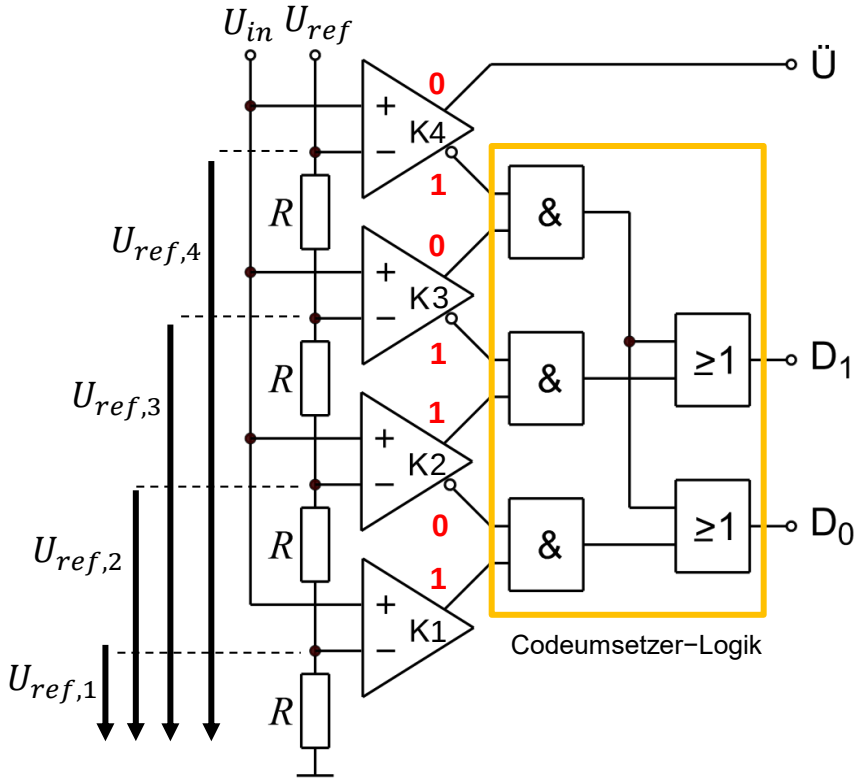
2-Bit-Flash-ADC,
Quelle: Wikipedia - ADC

FLASH-/PARALLEL-UMSETZER

$U_{ref} = 4V;\ \ U_{in} = 2V;\ \ R = 10k\Omega;\ \ U_{ref,n} = U_{ref} \cdot \frac{n}{n_{ges}}$:

- $U_{ref,4} > U_{in}$
- $U_{ref,3} > U_{in}$
- $U_{ref,2} = U_{in}$
- $U_{ref,1} < U_{in}$

Komparator (K)		
Bedingung	**Aus**	**Aus ○**
+ ≥ -	1	0
+ < -	0	1



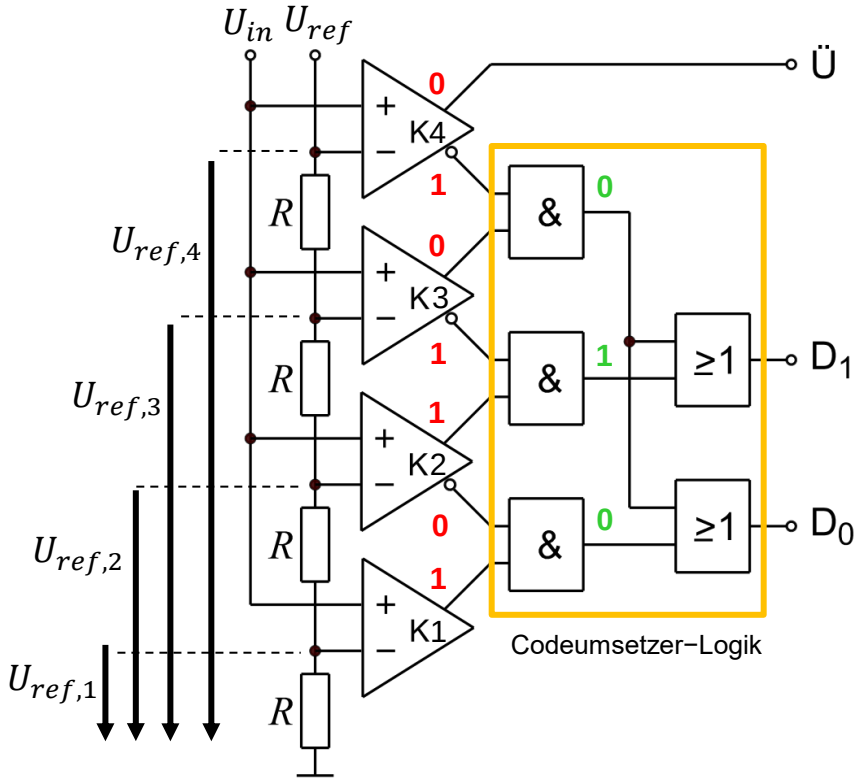
2-Bit-Flash-ADC,
Quelle: [Wikipedia - ADC]

FLASH-/PARALLEL-UMSETZER

$U_{ref} = 4V;\ \ U_{in} = 2V;\ \ R = 10k\Omega;\ \ U_{ref,n} = U_{ref} \cdot \frac{n}{n_{ges}}$:

- $U_{ref,4} > U_{in}$
- $U_{ref,3} > U_{in}$
- $U_{ref,2} = U_{in}$
- $U_{ref,1} < U_{in}$

UND-Glied (&)		
In 1	**In 2**	**Aus**
0	0	0
0	1	0
1	0	0
1	1	1



2-Bit-Flash-ADC,
Quelle: Wikipedia - ADC

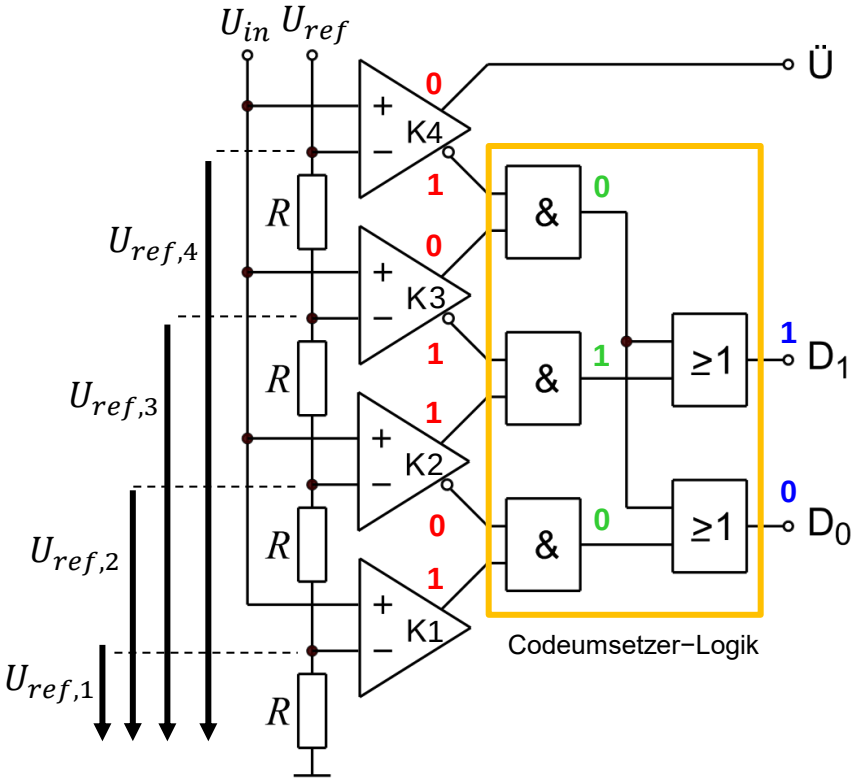
FLASH-/PARALLEL-UMSETZER

$U_{ref} = 4V;\ \ U_{in} = 2V;\ \ R = 10k\Omega;\ \ U_{ref,n} = U_{ref} \cdot \frac{n}{n_{ges}}$:

- $U_{ref,4} > U_{in}$
- $U_{ref,3} > U_{in}$
- $U_{ref,2} = U_{in}$
- $U_{ref,1} < U_{in}$

- $D = \{1\ 0\} \approx 2V = U_{in}$

ODER-Glied (≥1)		
In 1	**In 2**	**Aus**
0	0	0
0	1	1
1	0	1
1	1	1



2-Bit-Flash-ADC,
Quelle: Wikipedia - ADC

TECHNISCHE UNIVERSITÄT DARMSTADT

LERNZIELE

Sie …

- können Mikrocontroller zu anderen Rechnersystemen abgrenzen und die Hauptkomponenten erklären (Steuerwerk, Rechenwerk, Speicherwerk, Eingabe-/Ausgabewerk).
- können die Funktionsweise des Rechen- und Steuerwerks anhand von Pseudo-Code beschreiben.
- kennen die Eigenschaften der Von-Neumann- und Harvard-Architektur.
- wissen, was Interrupts sind, und können diese einordnen.
- können die Funktionsweise verschiedener ADCs und typische Wandlungsfehler an Beispielen beschreiben.

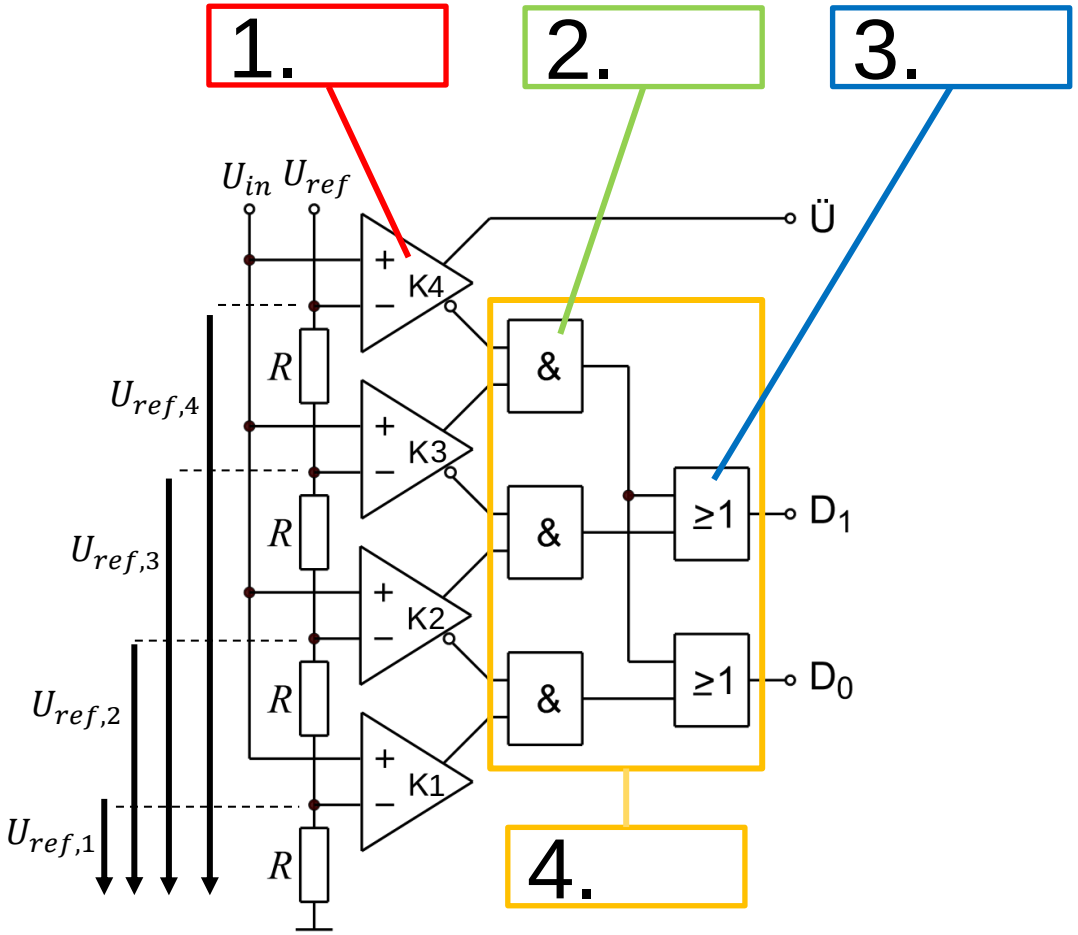
TECHNISCHE UNIVERSITÄT DARMSTADT

NICHT-LERNZIELE

Sie …

- … können einen Mikrocontroller elektronisch korrekt skizzieren.
- … kennen sämtliche Registerschaltungen und FlipFlop-Varianten.
- … 30 verschiedene Anwendungsgebiete für ADCs aufzählen.
- … kennen sich perfekt mit Schaltnetzen und allen Logikgattern aus.

Kein Bulimie-Lernen!


1.
2.
3.
4.
U_{in} U_{ref}
Ü
K4
K3
K2
K1
R
$U_{ref,4}$
$U_{ref,3}$
$U_{ref,2}$
$U_{ref,1}$
&
≥1
D_1
D_0

TECHNISCHE UNIVERSITÄT DARMSTADT

LITERATUR

(OPTIONAL – Nicht klausurrelevant)

- Uwe Brinkschulte & Theo Ungerer - Mikrocontroller und Mikroprozessoren
  - SpringerLink: Mikrocontroller und Mikroprozessoren | SpringerLink
- Helmut Bähring - Mikrorechner-Technik
  - SpringerLink: Mikrorechner-Technik | SpringerLink
- Herbert Bernstein – Mikrocontroller
  - SpringerLink: Mikrocontroller | SpringerLink
- Klaus Wüst – Mikroprozessortechnik
  - SpringerLink: Mikroprozessortechnik | SpringerLink