

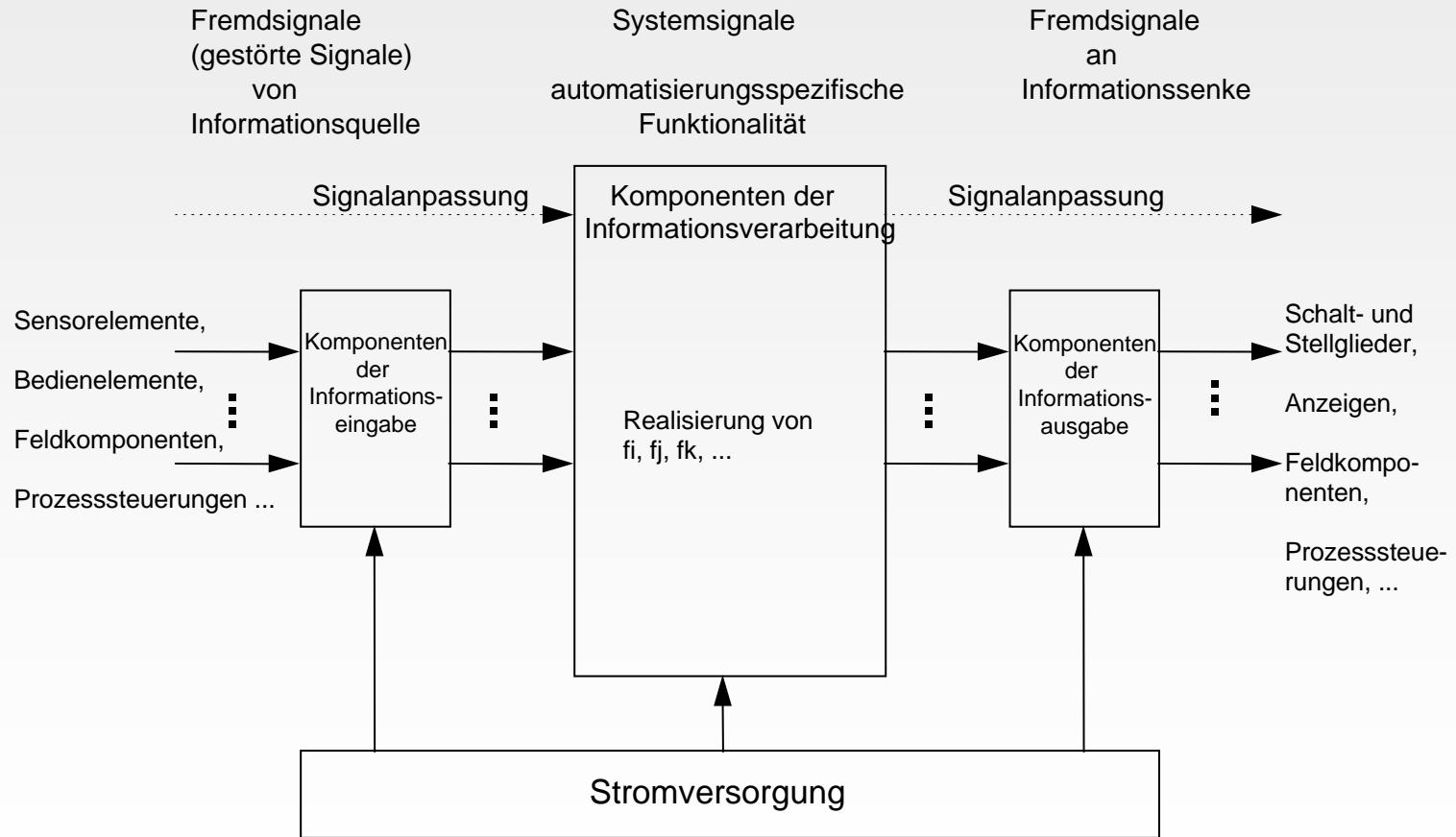
# Middleware für verteilte industrielle Umgebungen

Hardware in der Automatisierungstechnik



# Hardwarestruktur von Steuerungen

## Grundsätzliche Struktur einer Prozesssteuerung



( $f_i, f_j, f_k$  - elementare Funktionen der Informationsverarbeitung)



# Hardwarestruktur von Steuerungen

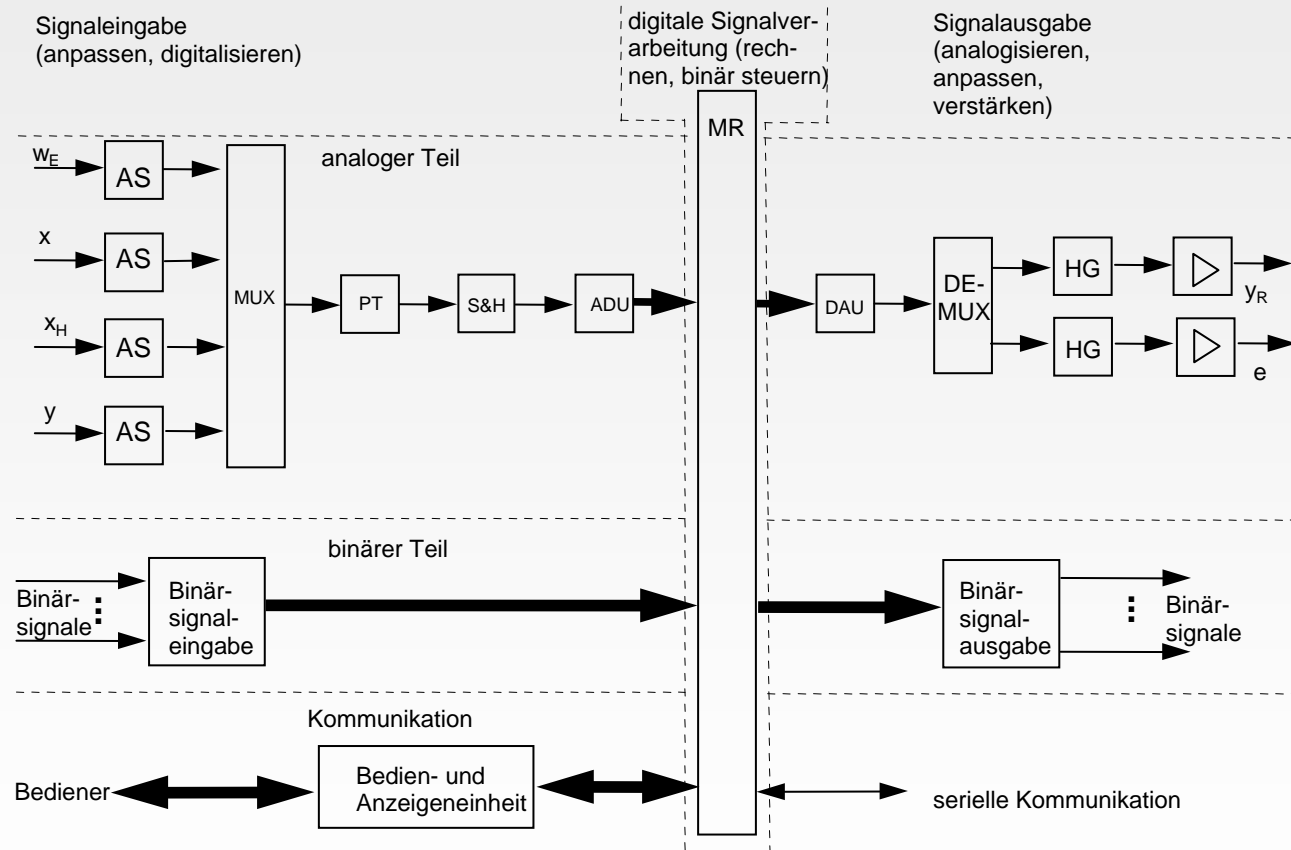
## Realisierungsvarianten von Prozesssteuerung

- ❑ Kompaktregler
  - Feste Anzahl und Typen von Signalein- und -ausgängen und Regelalgorithmen in einem Gerät
  - Konfigurierung und Parametrierung der Ein- und Ausgänge sowie der Algorithmen
  
- ❑ Speicherprogrammierbare Steuerungen (SPS)
  - Einsteckbare Signalein- und -ausgangsmodule
  - Programmierung der Steuerungs- und Regelalgorithmen
  - Programmierbare Ein- und Ausgangscharakteristiken
  
- ❑ Prozess-Controller von Leitsystemen (Prozessnahe Komponenten – PNK)
  - Einsteckbare Signalein- und -ausgangsmodule
  - Konfigurierung und Parametrierung vorgefertigter Steuerungs- und Regelalgorithmen
  - Konfigurierung und Parametrierung der Ein- und Ausgangscharakteristiken
  - Integration programmierter Signalcharakteristika und Algorithmen in die Systembibliotheken
  
- ❑ Prinzipiell weisen digitale Feldgeräte (Messumformer, Stellantriebe, umrichter gesteuerte Antriebe (Drives), Remote I/O-Systeme) gleiche Hardwarestrukturen auf. Diese Geräte unterscheiden sich durch ihre feste Funktionalität (ähnlich Kompaktregler)
- ❑ Die Grenzen zwischen diesen Gerätegruppen verlaufen mit steigender Verarbeitungsleistung in den Geräten (z.B. in Antriebe können wie bei Steuerungen anwendungsspezifischen Programmen geladen werden)



# Hardwarestruktur von Steuerungen

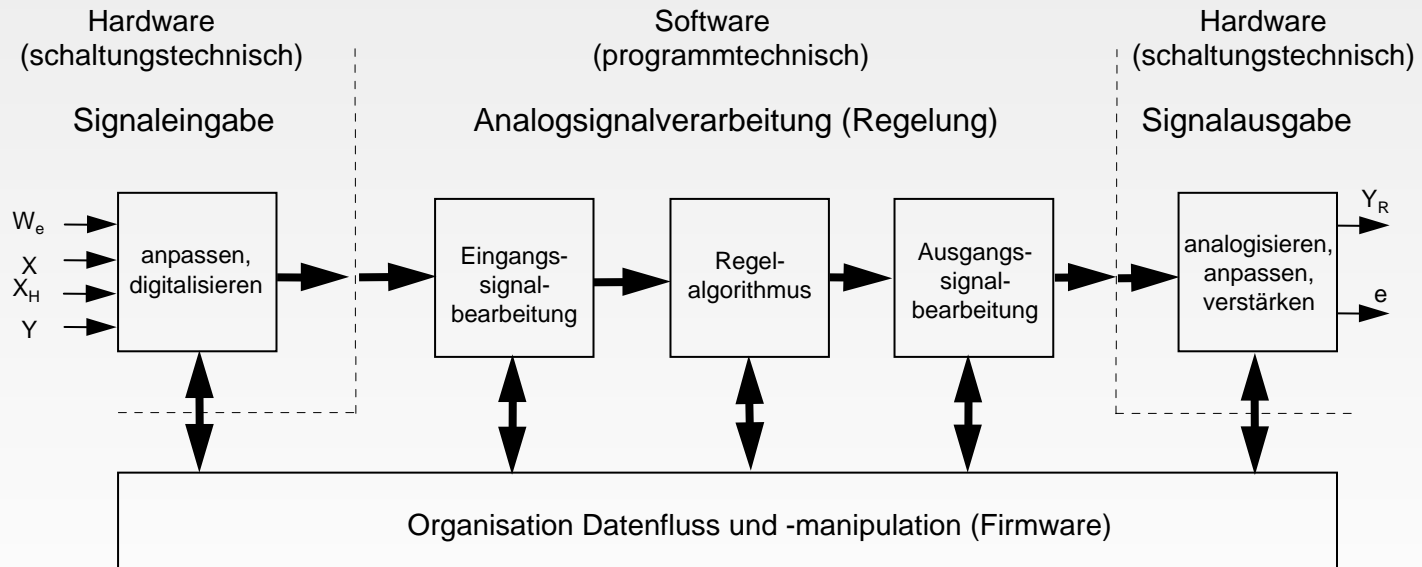
## Einkanaliger Kompaktregler



(MR - Mikrorechner, AS - Anpassschaltung, MUX - analoger Multiplexer, DE-MUX - analoger Demultiplexer, HG - Halteglied, S&H - Abtast- und Halteglied, ADU - Analog-Digital-Umsetzer, DAU - Digital-Analog-Umsetzer, PT - Potentialtrennung)



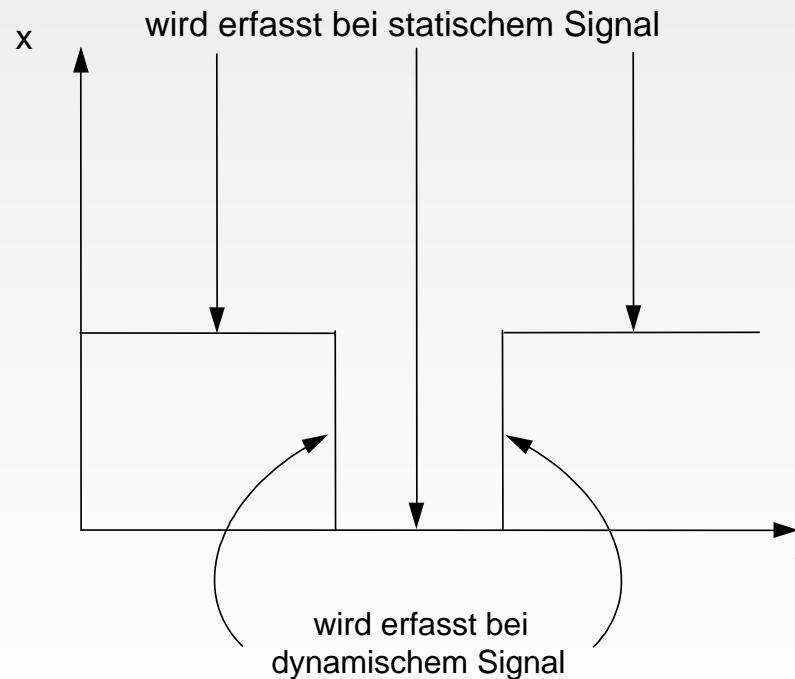
# Aufteilung der wichtigsten Aufgaben des Informationsmanagements eines (einkanaligen) SPR auf Hard- und Softwarekomponenten



# Hardwarestruktur von Steuerungen

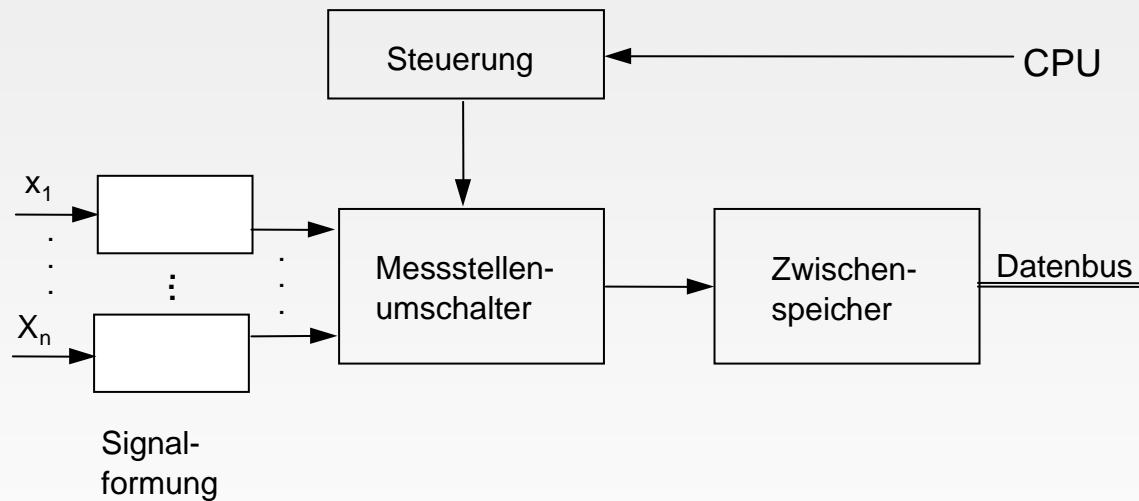
## Signalverarbeitung für binäre Eingänge

Erfassung statischer und dynamischer Binärsignale



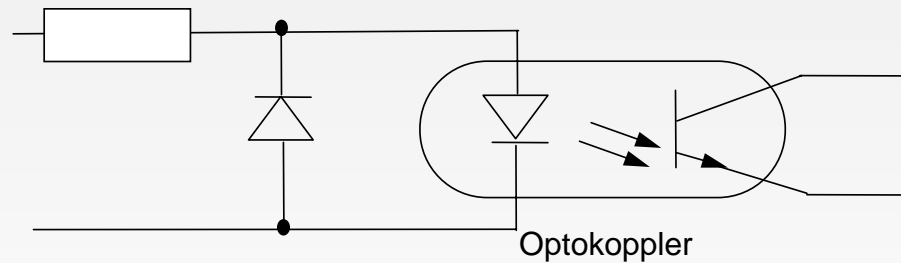
# Hardwarestruktur von Steuerungen

## Prinzipieller Aufbau einer Binäreingabe



# Hardwarestruktur von Steuerungen

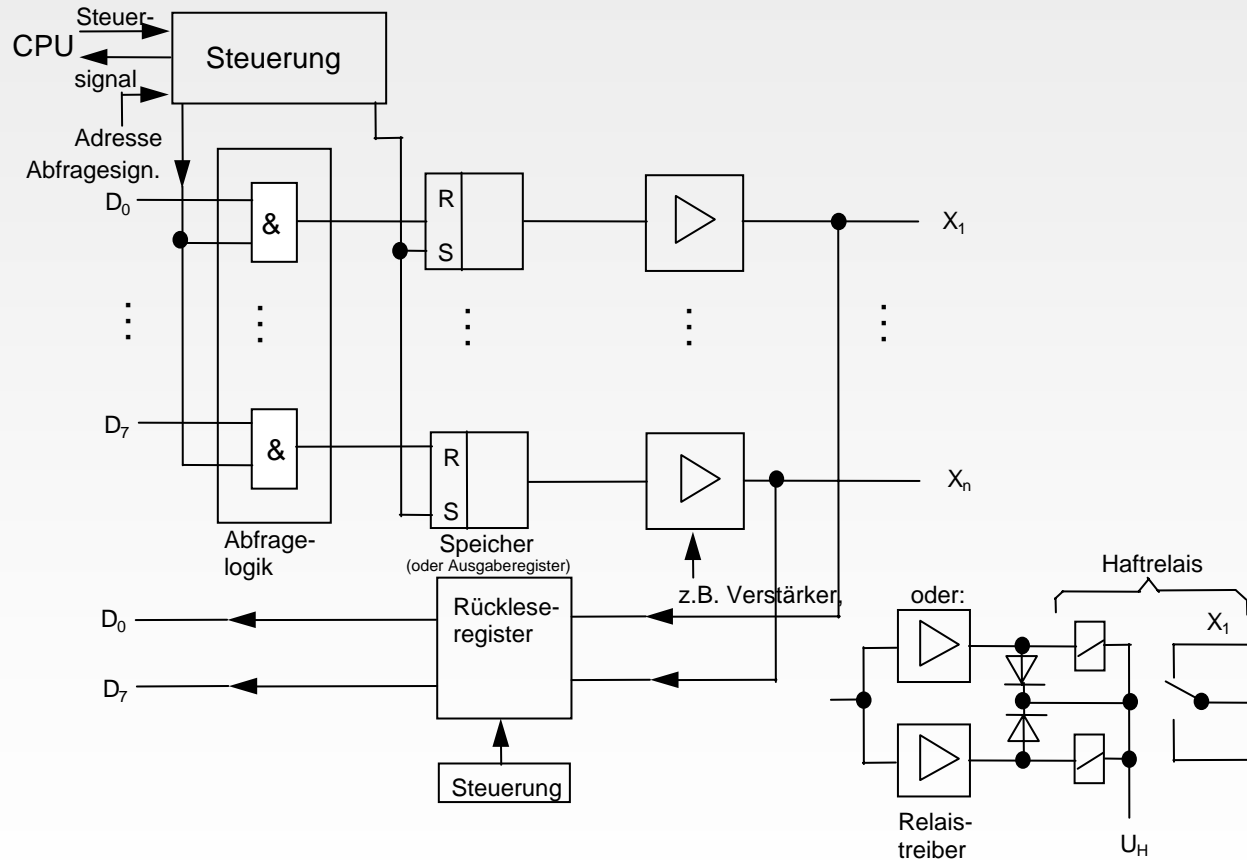
## Galvanische Trennung mittels Optokoppler





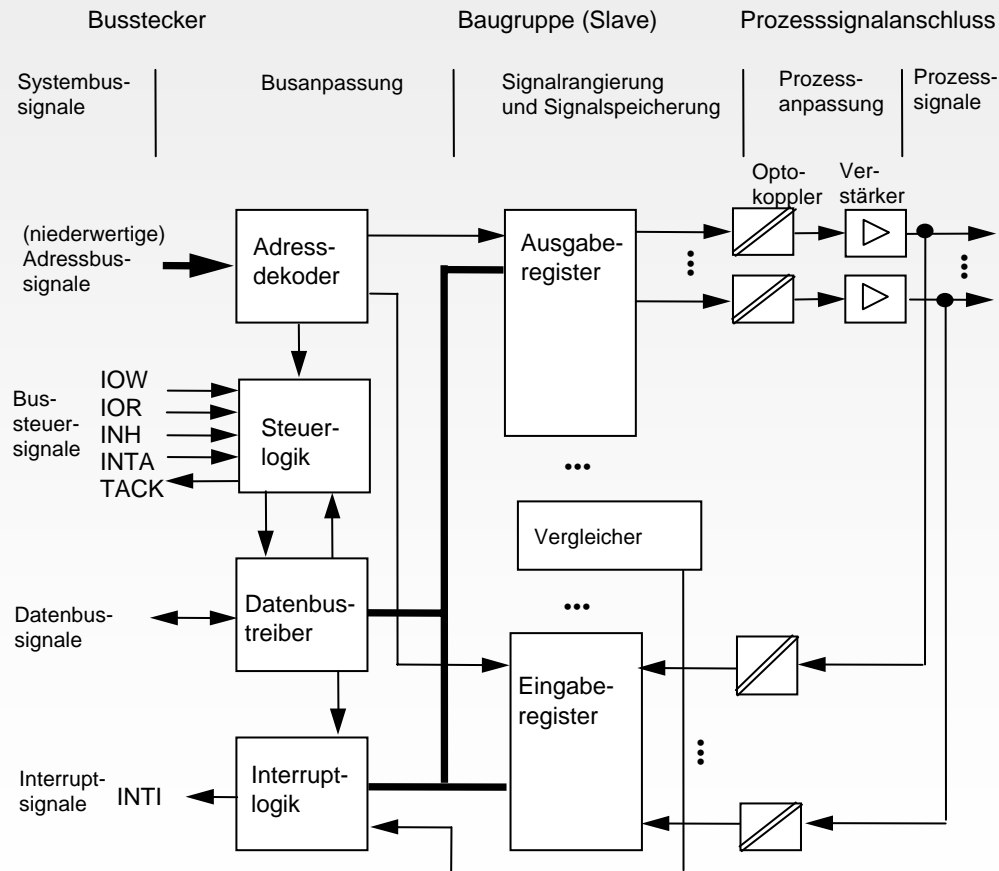
# Hardwarestruktur von Steuerungen

## Binärausgabe



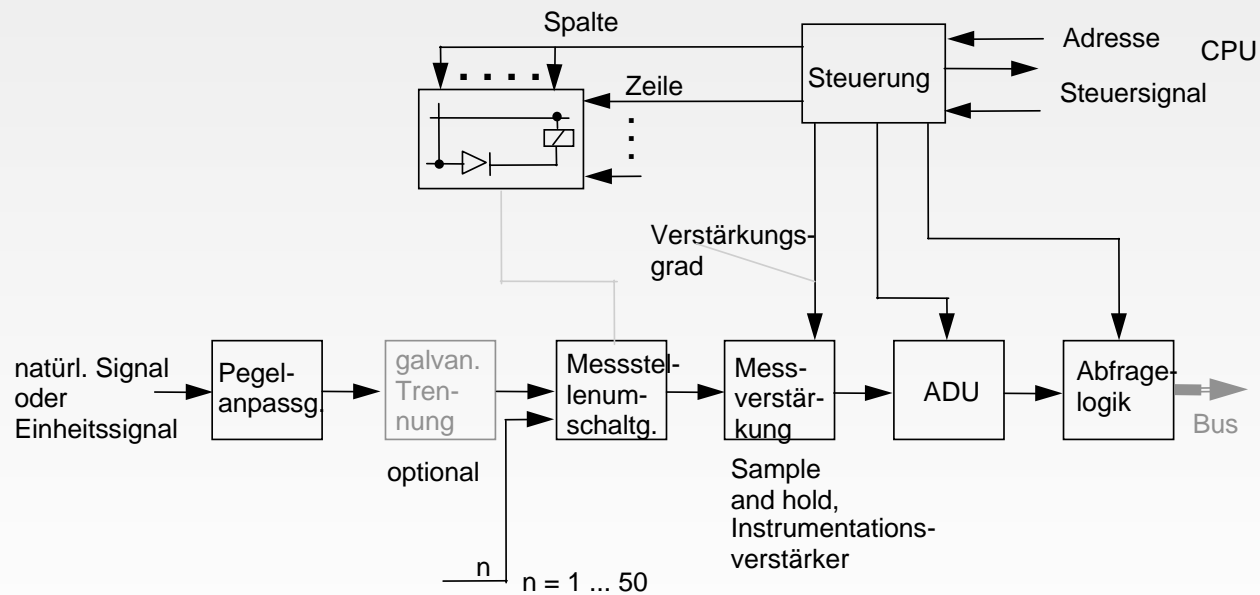
# Hardwarestruktur von Steuerungen

## Binärausgabe (mit Ausgabeüberwachung)



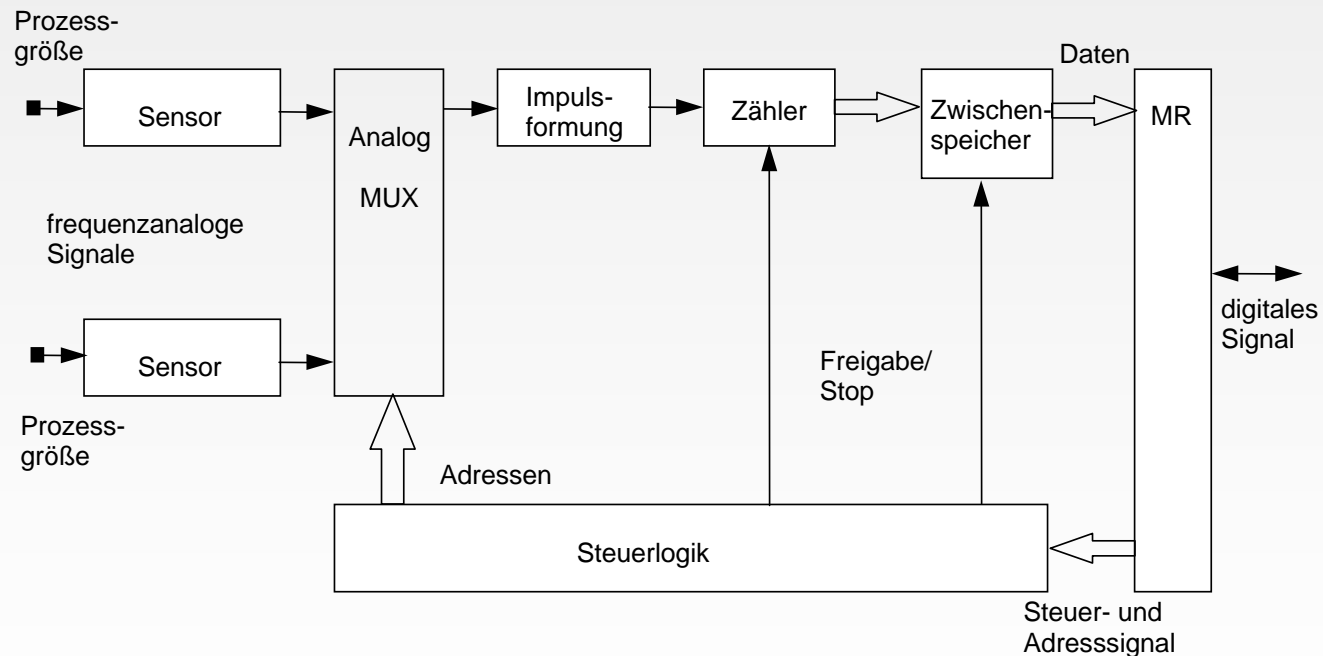
# Hardwarestruktur von Steuerungen

## Prinzipieller Aufbau einer Analogeingabe



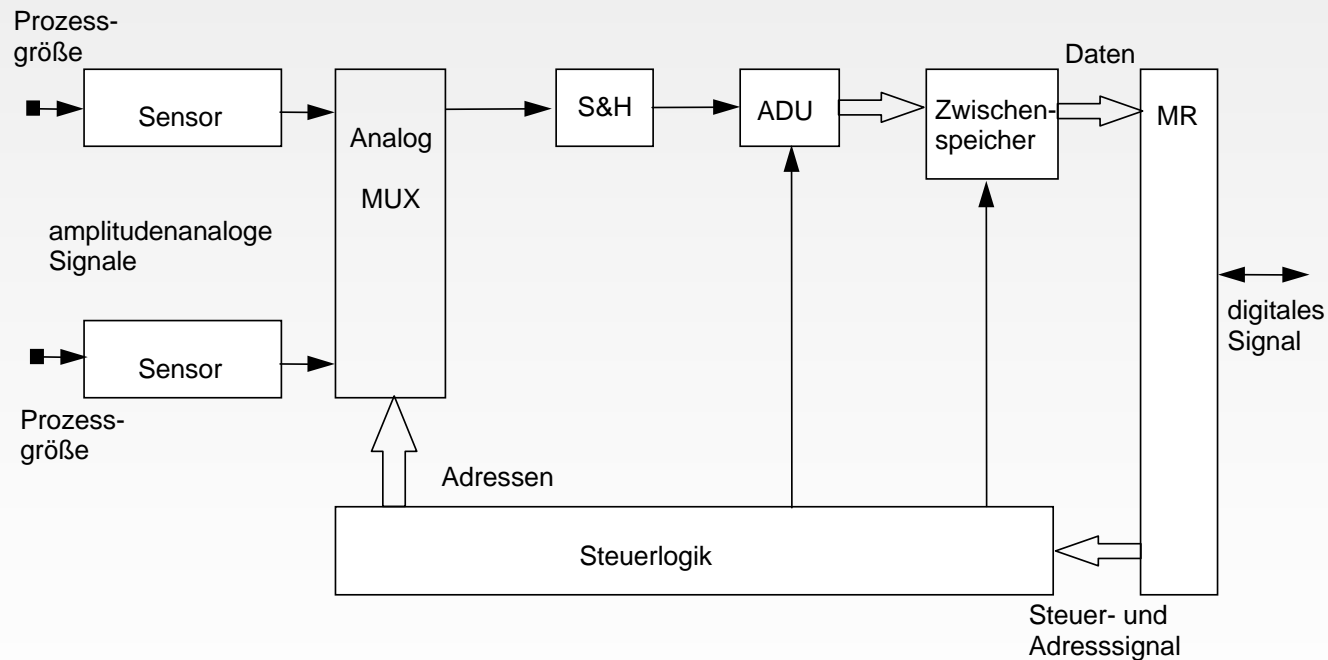
# Hardwarestruktur von Steuerungen

## Prinzipielle Struktur der frequenzanalogen Prozessgrößenabbildung



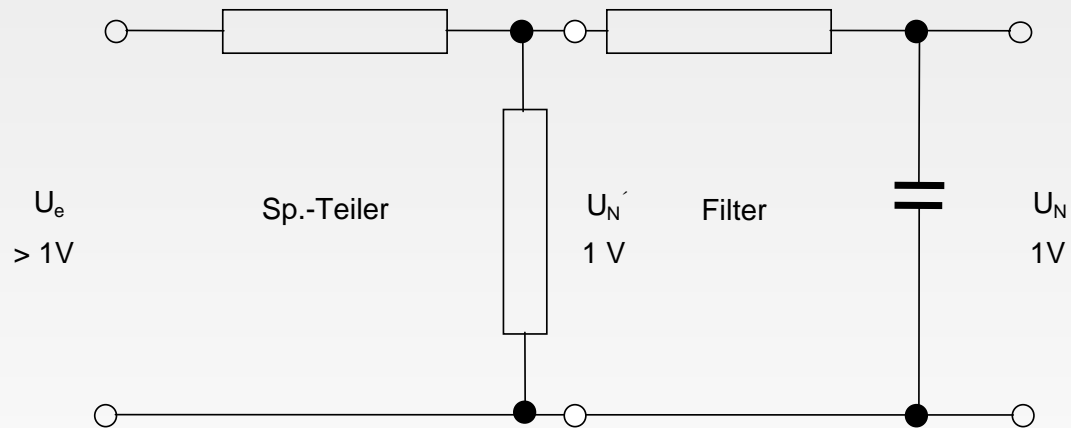
# Hardwarestruktur von Steuerungen

## Prinzipielle Struktur der amplitudenanalogen Prozessgrößenabbildung



# Hardwarestruktur von Steuerungen

## Pegelanpassung für aktive Geber

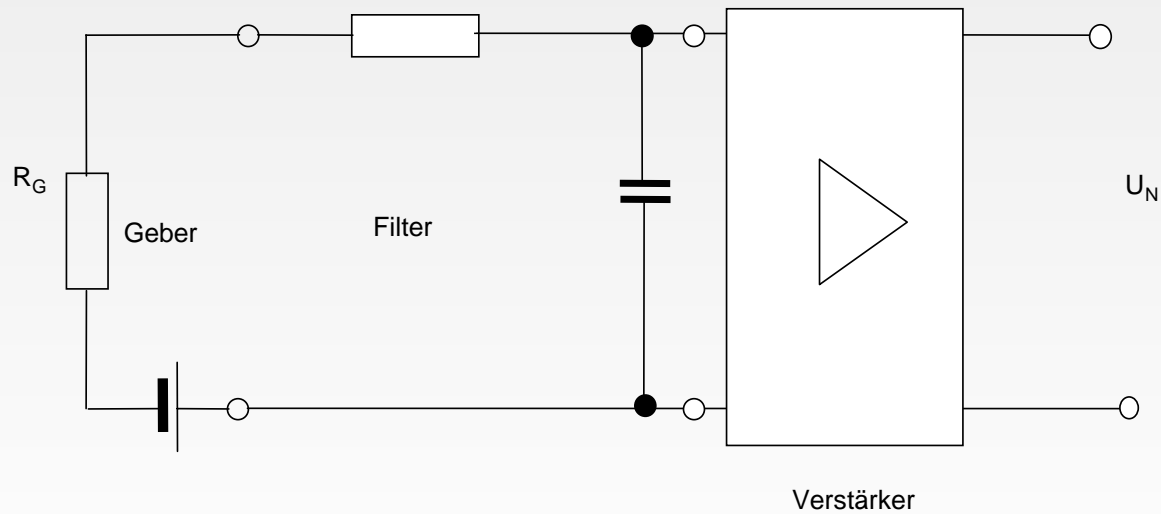


$$(U_{e \max} > U_N)$$



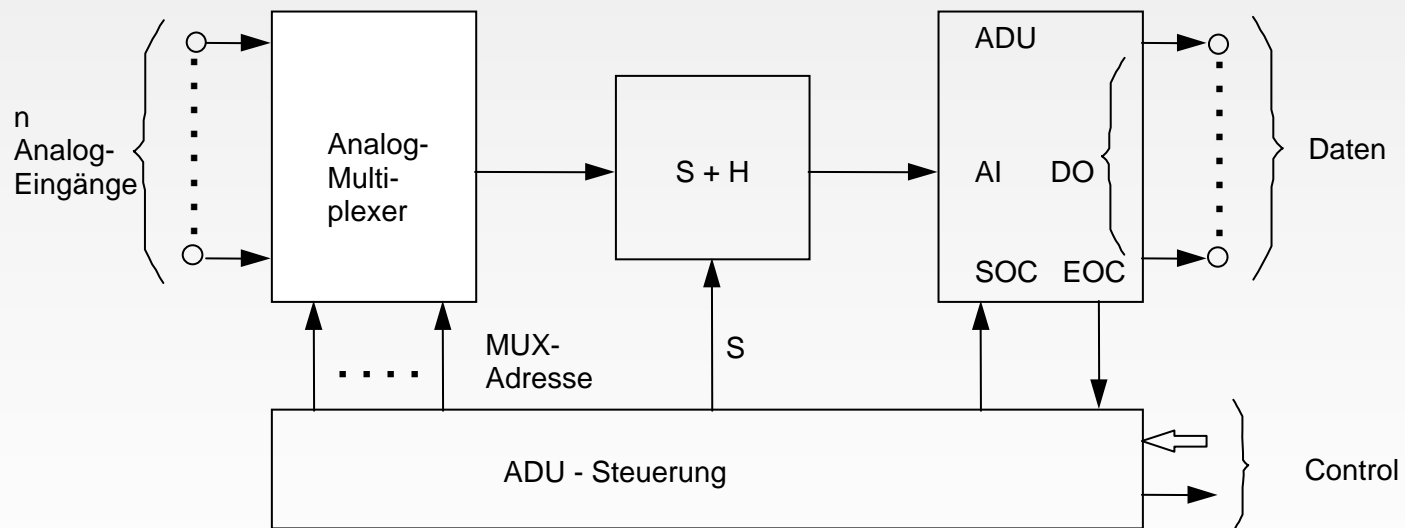
# Hardwarestruktur von Steuerungen

## Pegelanpassung für passive Geber



# Hardwarestruktur von Steuerungen

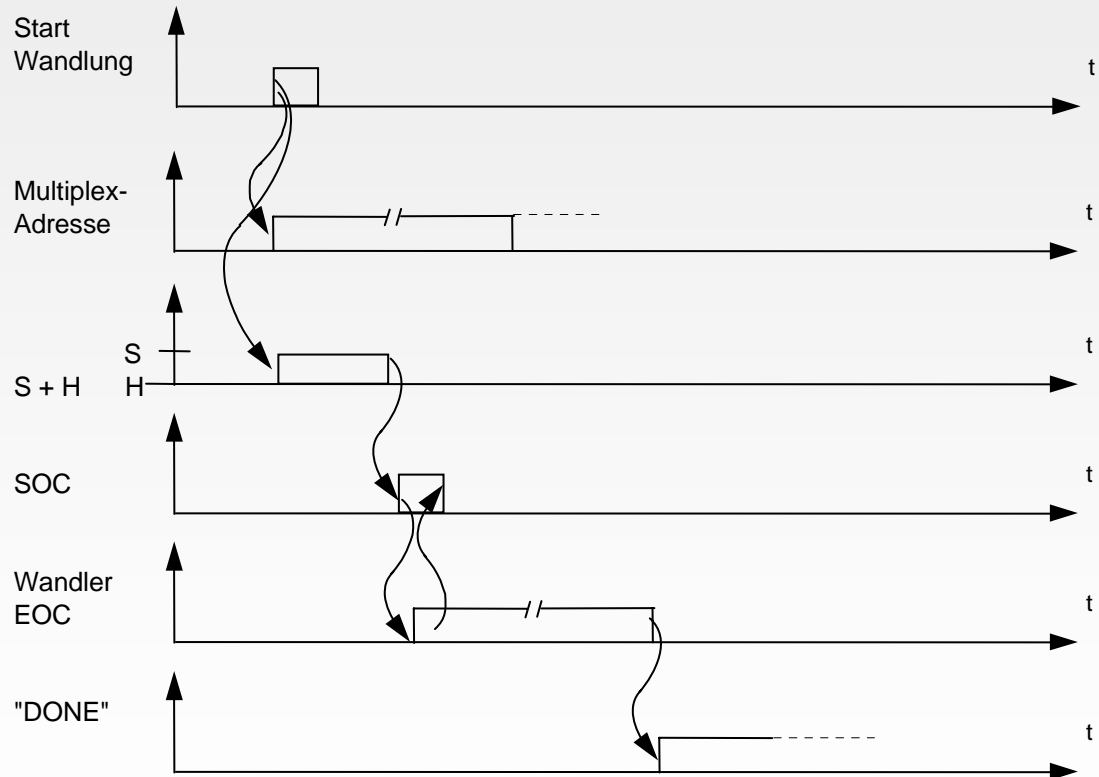
## Analog-Eingabe-System (Komponenten)





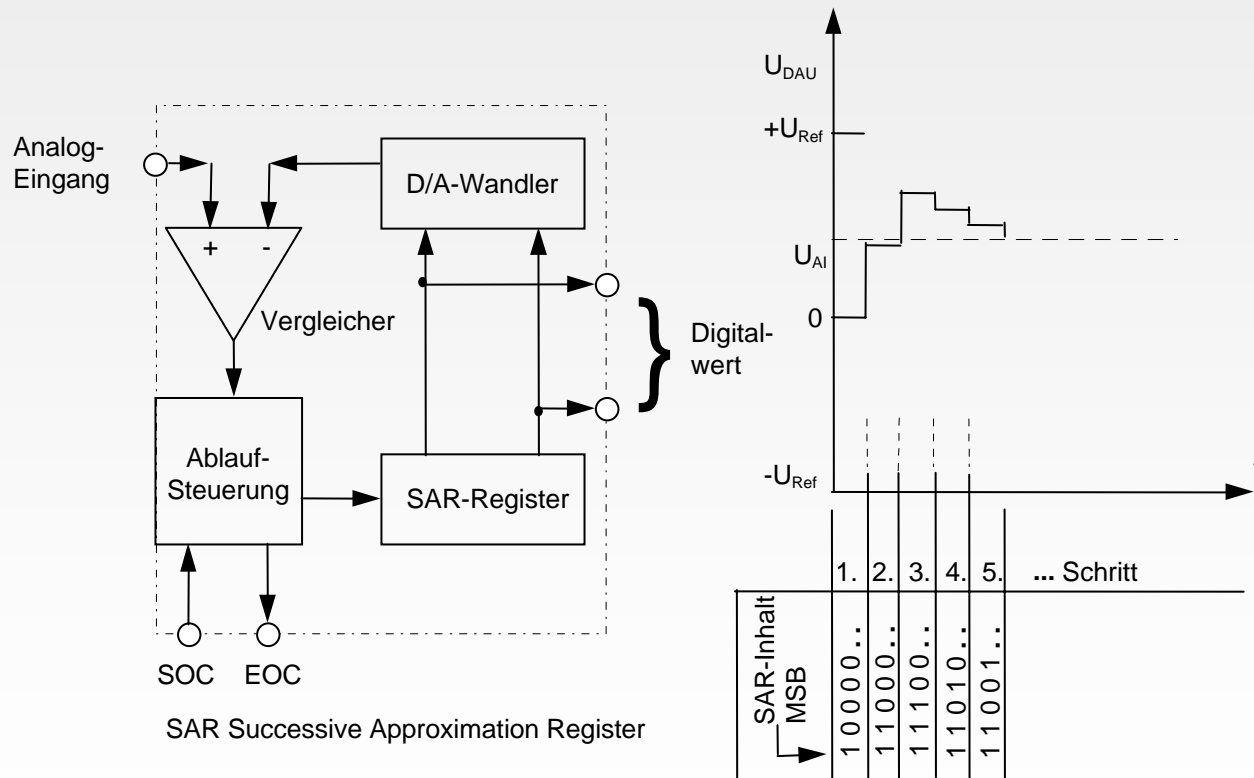
# Hardwarestruktur von Steuerungen

## Analog-Eingabe-System (zeitlicher Verlauf)



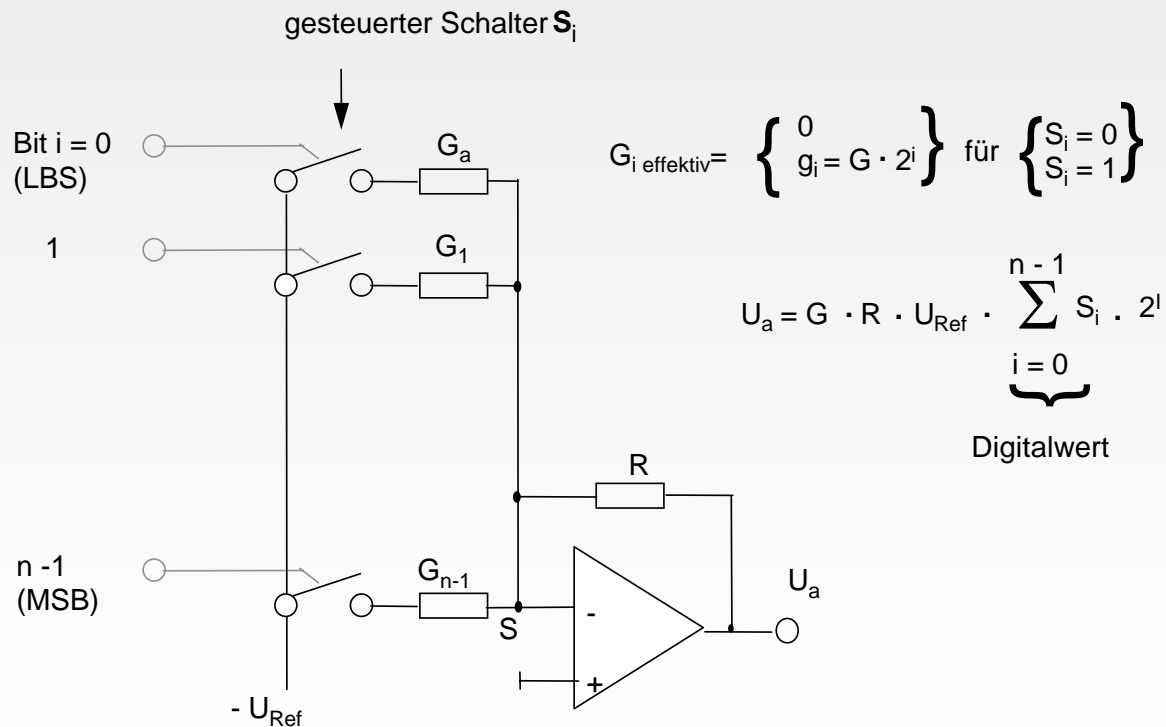
# Hardwarestruktur von Steuerungen

## Prinzip des Wäge-Kodierers (sukzessive Approximation)



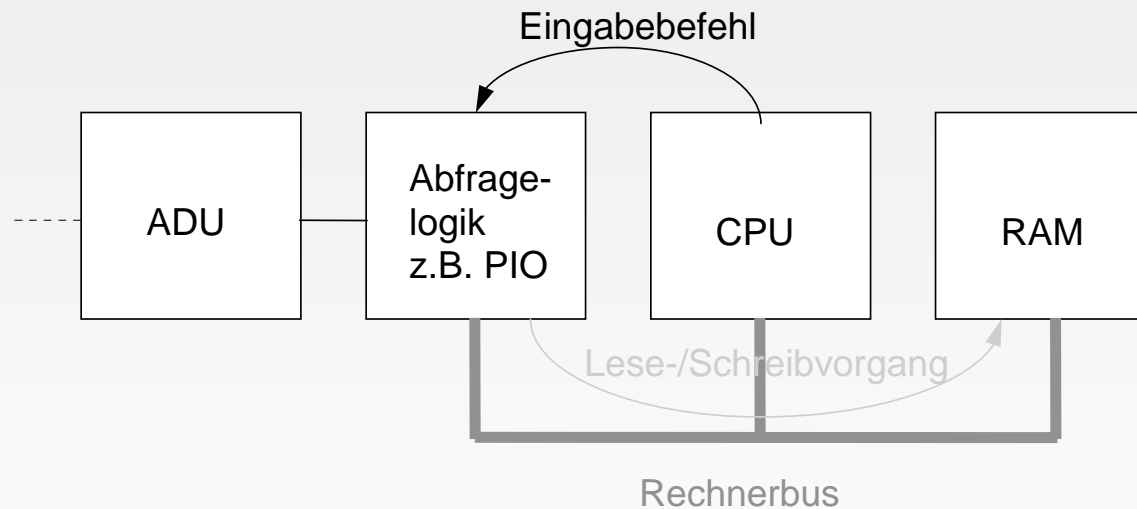
# Hardwarestruktur von Steuerungen

## Digital - Analog - Wandler



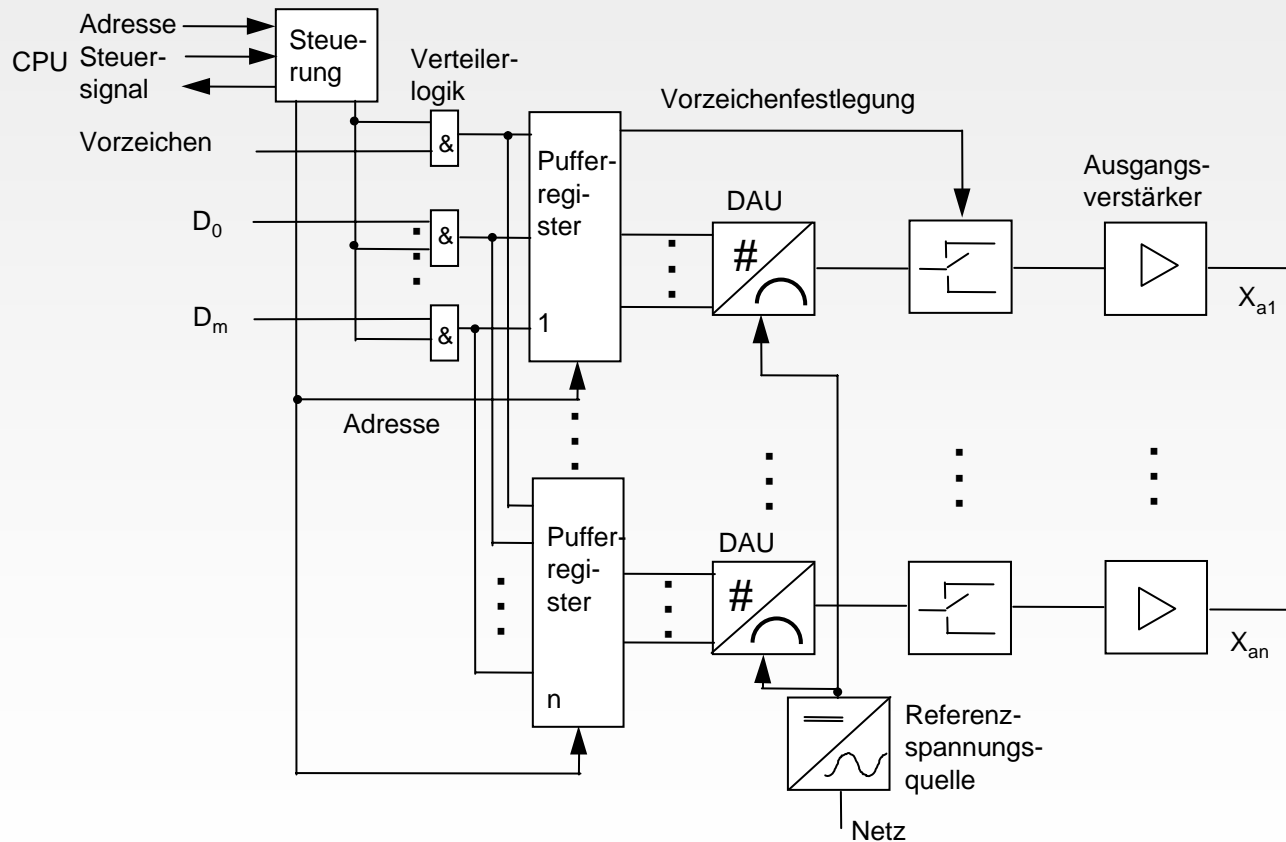
# Hardwarestruktur von Steuerungen

## Übernahme von ADU- Werten in den RAM



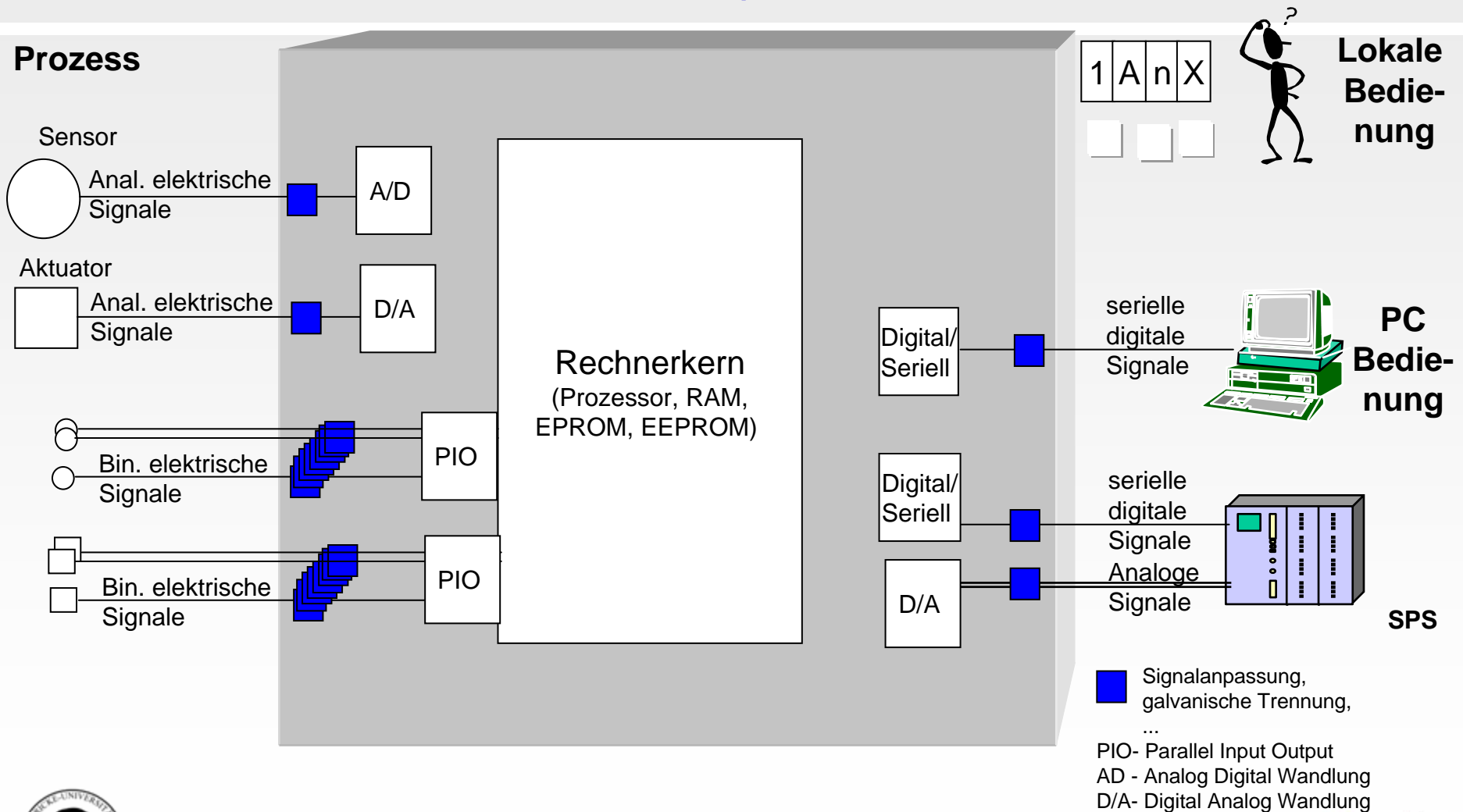
# Hardwarestruktur von Steuerungen

## Prinzipieller Aufbau einer Analogausgabe



# Hardwarestruktur von Steuerungen

## Rechnerstruktur mit externer Peripherie



# Signalverarbeitung

- ❑ Unterschiede in der Signalverarbeitung sind einmal durch die Aufgabenstellung bedingt, z.B. erzwingt diese sowohl Verknüpfungs- als auch Ablaufsteuerung, zum anderen aber auch durch die technische Realisierung. So lassen sich viele Aufgaben sowohl mit synchronen als auch mit asynchronen Steuerungen erfüllen. Bei den verschiedenen Arten der Signalverarbeitung ist es nicht so, dass die Arten Alternativen darstellten, eine Steuerung kann durchaus mehrere Merkmale haben.
  
- ❑ Synchrone Steuerung
  - In einer synchronen Steuerung werden Signale **zeitgleich** zu einem Taktsignal verarbeitet. Charakteristische Beispiele dafür sind von Mikroprozessoren geführte oder unterstützte Steuerungen, bei denen ein Programm alle 10 ms gestartet wird, die Eingangsgrößen abfragt, sie gemäß vorgegebener Gesetzmäßigkeiten verarbeitet und dementsprechend die Ausgangssignale stellt.
  - Die Arbeitsweise synchroner Steuerungen **entspricht weitgehend dem menschlichen Handeln**, sie ist deshalb besser verständlich als die asynchroner Steuerungen.
  
- ❑ Asynchrone Steuerung
  - In asynchronen Steuerungen werden Signaländerungen nur durch **Änderungen der Eingangssignale** ausgelöst. Zu den asynchronen Steuerungen gehören Relaissteuerungen, aber auch elektronische Steuerungen, die nicht von Mikroprozessoren geführt werden (dabei können sie durchaus frei-programmierbar sein).



# Signalverarbeitung

## □ Verknüpfungssteuerung

- Eine Verknüpfungssteuerung **ordnet den Signalzuständen der Eingangsgrößen bestimmte Zustände der Ausgangsgrößen** im Sinne Boolescher Verknüpfungen eindeutig zu. Eindeutig heißt hier, daß die Ausgangssignale bestimmte Werte nicht nur bei bedingungsgemäßer Kombination der Eingangssignale annehmen, sondern auch, dass diese Werte nur bei der bedingungsgemäßen Kombination eingenommen werden.
- Reine Verknüpfungssteuerungen, also in der Art, dass die Ausgangssignale statisch nur vom Zustand der Eingangssignale abhängen, sind selten. Schon in einer Neuwertmeldeeinrichtung sind für die Eingangssignale, die sich seit der letzten Quittierung geändert haben, Speicherglieder erforderlich. Auch für andere Aufgaben müssen in Verknüpfungssteuerungen meist **Speicher- und Zeitglieder** integriert werden.

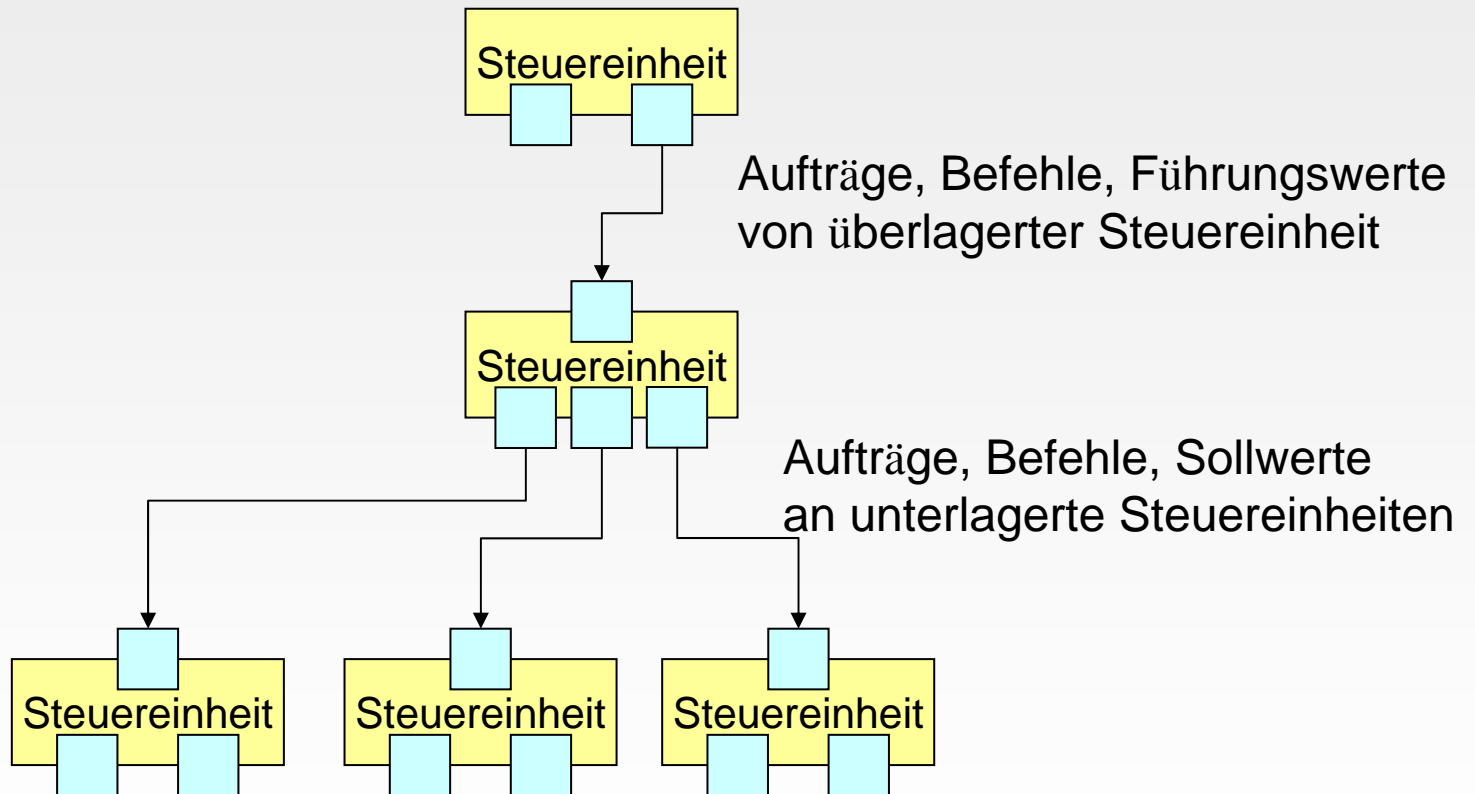
## □ Ablaufsteuerung

- Eine Ablaufsteuerung ist eine Steuerung, die **zwangsläufig schrittweise** abläuft. Für das Weiterschalten zum programmgemäß nächsten Schritt ist es erforderlich, dass gegebene **Weiterschaltbedingungen** erfüllt sind. Die Schritte entsprechen meistens dem technologisch bedingten Verfahrensablauf. Die technologischen Ablaufschritte müssen sehr genau spezifiziert werden. So müssen vor vielen Schritten Start- und Verriegelungsbedingungen überprüft und diesbezügliche Programmschritte mit Sprüngen, Schleifen und Verzweigungen mehrfach genutzt in das Ablaufprogramm integriert werden.
- Es ist zwischen **zeitgeführter und prozessabhängiger** Ablaufsteuerung zu unterscheiden. Beide Alternativen kommen jedoch selten in reiner Form vor.

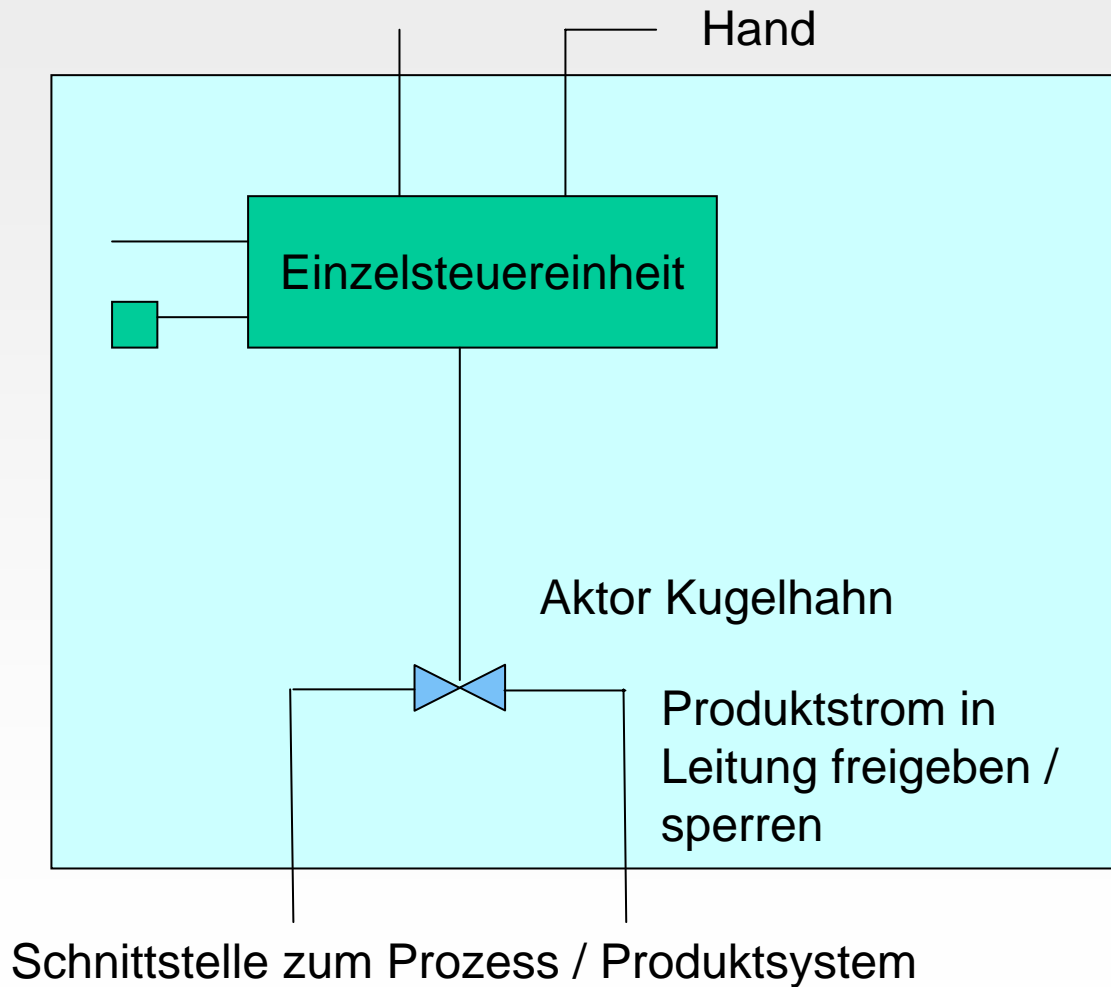




# Hierarchisches Steuermodell



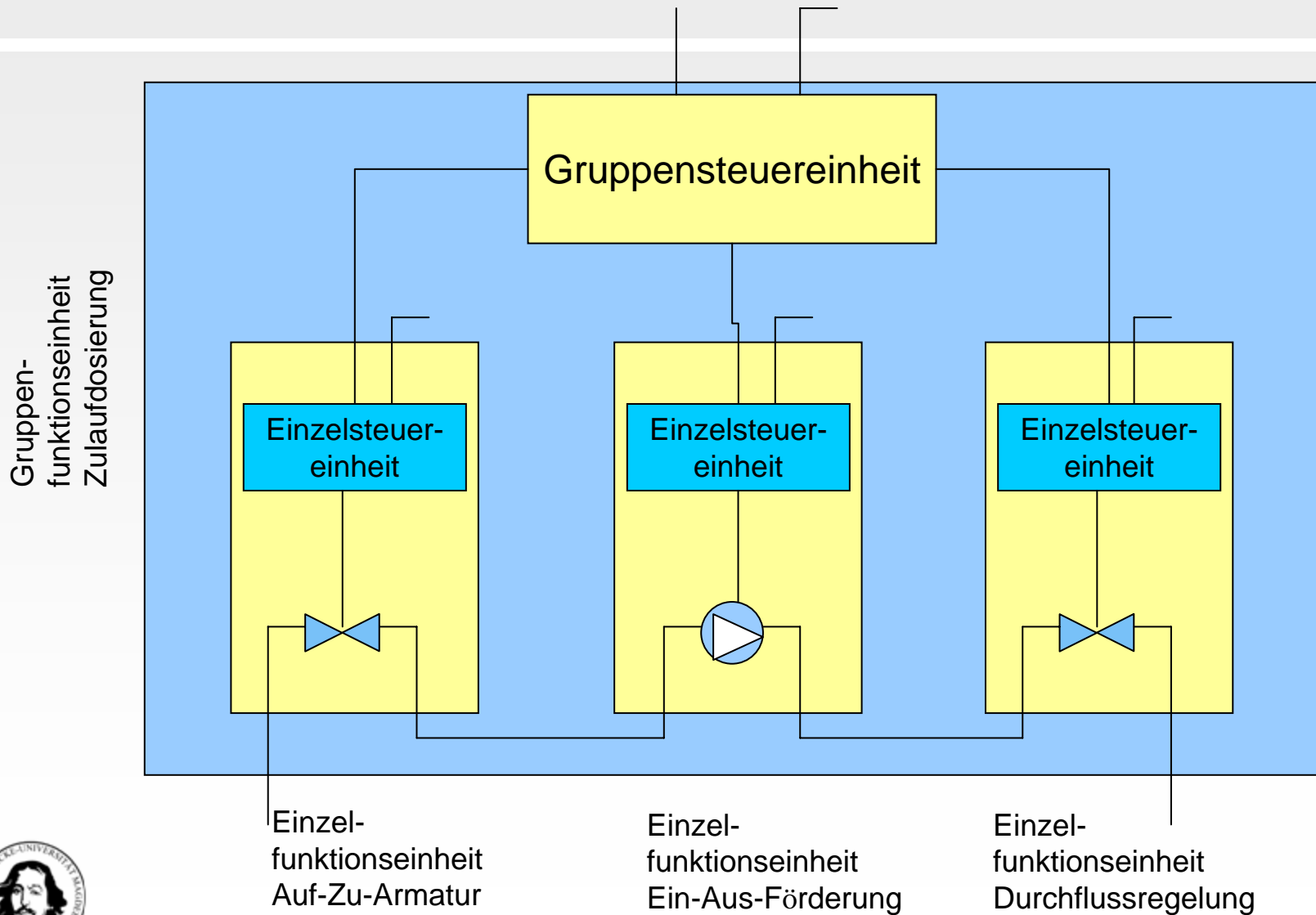
# Einzelsteuereinheit



Baustein:  
Auf-Zu Armatur



# Gruppenfunktionseinheit



# Hierarchischer Aufbau

- ❑ Um **komplex zusammenwirkende Steuerungen** handhabbar betreiben zu können, ist es oft zweckmäßig, sie so zu strukturieren, dass sich mit **autarken Untersystemen** der Prozess auch dann führen lässt, wenn Teile der Steuerung gestört sind oder der Prozess sich in Ausnahmezuständen befindet, wobei es sich auch um An- und Abfahrvorgänge handeln kann. Es bietet sich eine **hierarchische Struktur** mit folgenden Ebenen an.
  
- ❑ Einzelsteuerungs- oder Antriebssteuerungsebene
  - Die unterste Hierarchieebene ist die der Einzel- oder Antriebssteuerungen, die alle Funktionseinheiten zum **Steuern einzelner Stellglieder oder Antriebe** umfasst. Bevorzugte Aufgaben dieser Ebene sind **Hand-Automatik-Umschaltung, Speicherung, Verriegelung, Überwachung, Meldung und Verstärkung**. Zu speichern sind z.B. ein Auf- oder ein Zu-Befehl, zu verriegeln die gleichzeitige Betätigung eines anderen Stellgliedes, zu überwachen eine Laufzeit und zu melden eine Endstellung.
  
- ❑ Gruppensteuerungsebene
  - Eine oder mehrere Gruppensteuerungsebenen bilden die nächst höhere Hierarchieebene. Sie umfasst alle Funktionseinheiten zum **Steuern von Teilprozessen**, z.B. An- und Abfahrsteuerung eines größeren Verdichters oder die Steuerung von Teilschritten eines Reaktors, wie Inertisieren, Beschicken, Reagieren und Austragen.
  - **Sinn der Gruppensteuerung** ist es, Einzelsteuerungen so zusammenzufassen, dass **bei Ausfall** der Gesamtsteuerung wenigstens **Teilbetriebe automatisch laufen** können, dass z.B. die In- oder Außerbetriebnahmemodalitäten von Maschinen nicht in umständlichen, unübersichtlichen und oft nicht automatisch gesicherten Einzeloperationen ablaufen müssen.



# Hierarchischer Aufbau (ff)

- Leitsteuerungsebene
  - Die oberste Hierarchieebene ist die der Leitsteuerung. Die Leitsteuerung ist die Funktionseinheit zur **Führung des Gesamtprozesses**.



# Programmverwirklichung

- ❑ Das letzte, aber für uns sehr wichtige Ordnungsmerkmal ist die Art der Programmverwirklichung. DIN 19 237 nennt das **Anpassen** einer jeden Steuerung **an die Aufgabenstellung** eine Programmierung und unterscheidet in der Hauptsache zwischen **verbindungsprogrammierten Steuerungen**, zu denen z.B. auch Relaissteuerungen zählen, und **speicherprogrammierten Steuerungen**. Ein Programm ist dabei die Gesamtheit aller Anweisungen und Vereinbarungen für die Signalverarbeitung, die erforderlich sind, um eine Verfahrensanlage aufgabengemäß zu beeinflussen.
  
- ❑ Speicher einer Steuerung
  - Ein **Speicher** ist eine Funktionseinheit, die sowohl **Programme** (Programmspeicher) als auch **Daten** (Datenspeicher) **in digitaler Darstellung aufbewahrt**. Programm- und Datenspeicher können verschiedener Art sein, z.B. können Programme fest eingebrannt sein, während Daten natürlich laufend eingeschrieben und ausgelesen werden müssen. Es sind folgende Speicherarten und Abkürzungen dafür gebräuchlich:
    - Schreib-Lese-Speicher (Random Access Memory, RAM)
    - Nur-Lese-Speicher (Read Only Memory, ROM)
      - einmalig programmierbar: Programmable Read Only Memory (PROM)
      - mehrfach programmierbar: Re-Programmable Read Only Memory (RPROM)
        - Erasable Programmable Read Only Memory (EPROM)
        - Electrical Erasable Programmable Read Only Memory (EEPROM)



# Programmverwirklichung (ff)

## □ Verbindungsprogrammierte Steuerung

- Verbindungsprogrammierte Steuerungen werden durch die **Auswahl der Funktionsglieder und deren Verbindungen** an Aufgabenstellungen angepasst oder normgemäß **programmiert**. Dabei kann noch weiter zwischen **fest programmierten** und **umprogrammierbaren** unterschieden werden, also z.B. zwischen solchen, die durch **Löt-, Schraub- oder Wickelverbindungen** Aufgaben fest zugeordnet sind, und solchen, die sich durch Umstecken von Leitungen oder Schläuchen, durch **Diodenmatrizen oder durch Austausch von Baugruppen** an geänderte Aufgaben anpassen lassen.

## □ Speicherprogrammierbare Steuerungen

- In speicherprogrammierbaren oder speicherprogrammierten Steuerungen, mit denen wir uns im folgenden in erster Linie befassen wollen, sind die **Programme** zur Steuerung **in Speichern hinterlegt**. Auch dabei gibt es noch Unterscheidungskriterien, nämlich zwischen **frei programmierbaren**, in denen die Programme in **RAM-Speichern** hinterlegt sind, und **austauschprogrammierbaren** mit ROM-, PROM- oder EPROM-Programmspeichern, die beim Hersteller programmiert und beim Anwender erforderlichenfalls ausgetauscht werden können.
- Für speicherprogrammierbare Steuerungen hat sich die Abkürzung **SPS** (im englischsprachigen Raum zuerst PC und später zur Vermeidung von Verwechslungen **PLC** für Programmable Logic Controller) durchgesetzt, die auch wir im folgenden gebrauchen wollen. Nach den vorangegangenen grundsätzlichen Erläuterungen soll jetzt konkret auf die technische Realisierung von Steuerungen eingegangen werden.



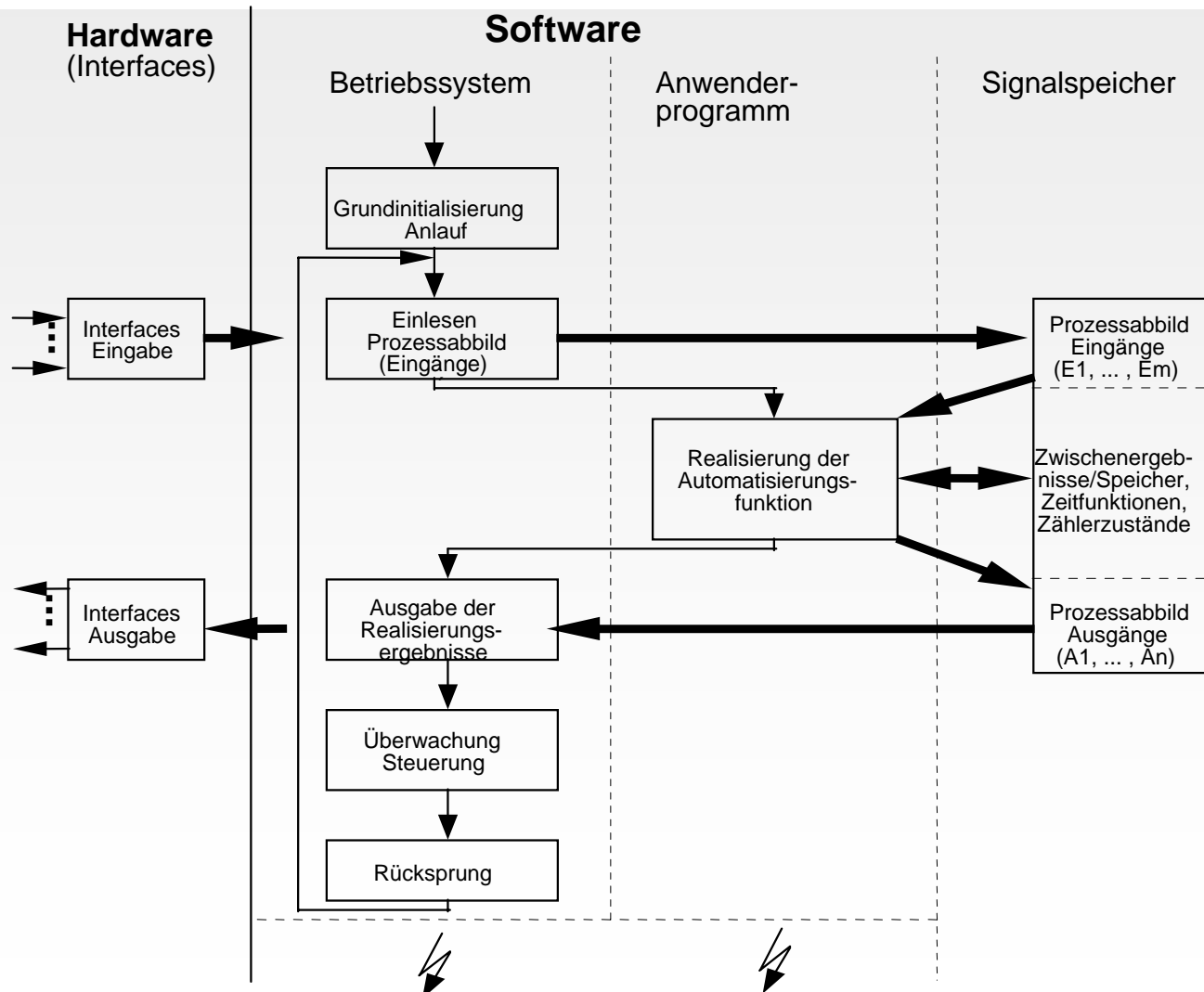
# Was ist eine SPS? Wie funktioniert sie?

- Einsatz für industrielle Anwendungen
- Arbeitet ein Programm ab, das den Prozess z.B.:
  - Steuerung eines Transportprozesses (Förderband, Fahrstuhl, ...)
  - Mischen von Chemikalien
  - Bearbeiten von Werkstücken
- in gezielter Weise beeinflusst

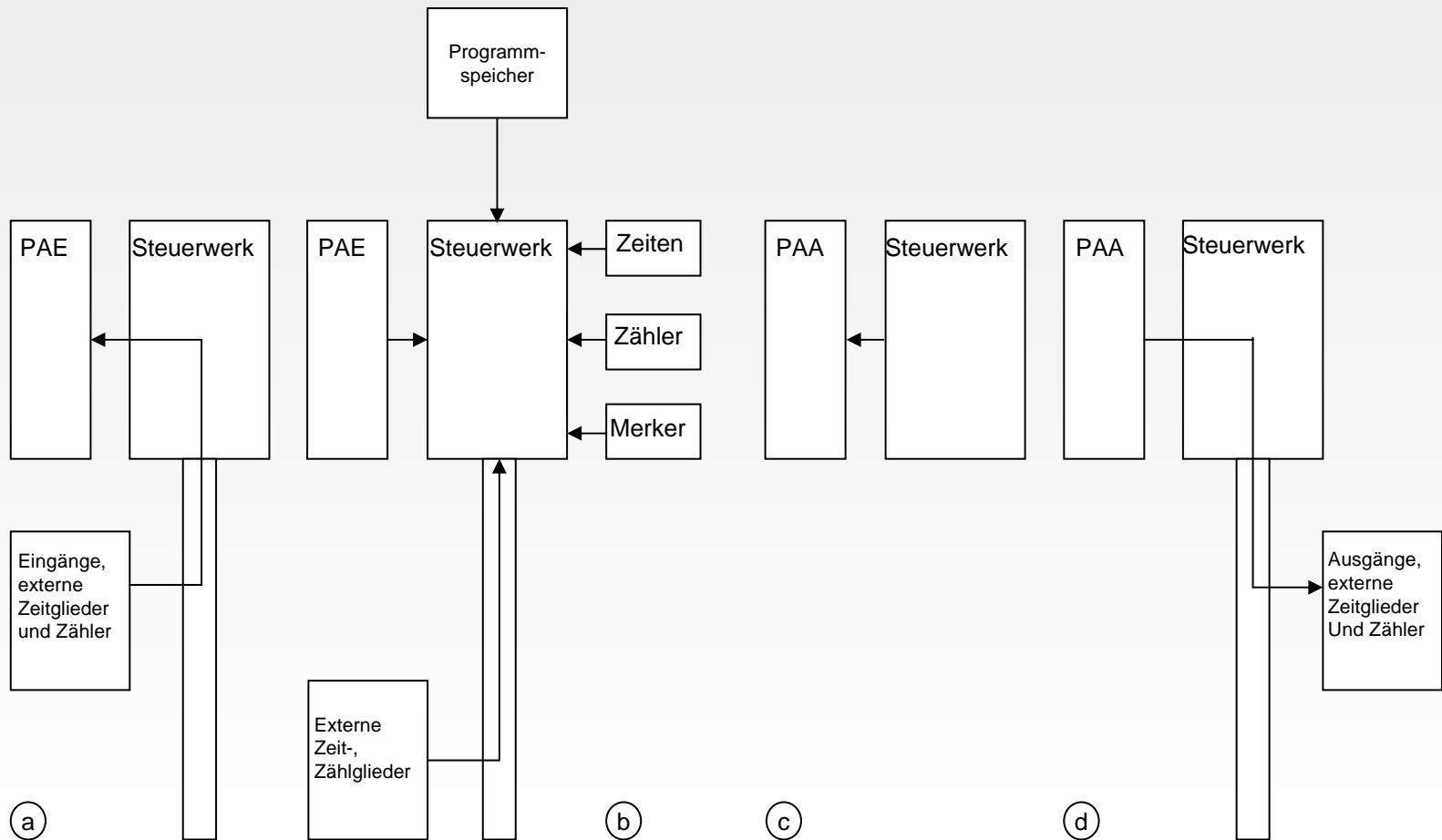




# Abarbeitungssteuerung in einer SPS (allg. Prinzipip)



# Arbeitsweise einer SPS (Siemens)



# Arbeitsweise speicherprogrammierbarer Steuerungen

- ❑ Die vier Teilbilder a bis d der vorherigen Folie zeigen die wesentlichen Ablaufschritte eines Programmdurchlaufs, auch **Zyklus** genannt. Zu **Beginn** eines jeden Zyklusses fragt das Steuerwerk die **Signalzustände an den Eingängen** der Steuerung sehr schnell hintereinander ab und **setzt im Prozessabbild für Eingänge** (PAE) die jedem Eingang zugeordnete Speicherzelle so auf 0 oder 1 bzw. auf den digitalen Wert des analogen Signals, dass diese Speicherzellen nach Ablauf dieses Bearbeitungsschrittes ein Abbild der Signalzustände der Eingänge enthalten (siehe vorherige Folie a).
- ❑ Während der nun folgenden **Programmbearbeitung greift** der Prozessor des Steuerwerkes **auf dieses Prozessabbild** und nicht direkt auf die Eingänge zurück und **bearbeitet** abhängig davon die im Programmspeicher stehenden **Steuerungsanweisungen** (siehe vorherige Folie b). Diese sagen, sehr vereinfacht ausgedrückt, aus, welcher Eingang mit welchem anderen wie zu einem bestimmten Ausgangssignal zu verknüpfen ist. Diese Operationen werden mitunter auch Befehle genannt.
- ❑ Die Operandenteile der Anweisungen sind nach Operandenkennzeichen (Eingänge E, Ausgänge A, Merker M usw.) und Parametern (Adressen) unterteilt. Üblicherweise werden die Parameter byte-weise, d.h. in Gruppen zu 8, strukturiert. Die Beschreibung "**Eingang E 2.3**" ist so zu interpretieren: die Ziffer vor dem Punkt ist die Byte-Nummer, die Ziffer dahinter gibt einen der acht Eingangskanäle 0 bis 7 dieses Bytes an. Im Verlauf dieses Arbeitsganges muss das Steuerwerk auch, wie im Bild dargestellt, Verbindung zu externen Zeit- und Zählgliedern haben.



# Arbeitsweise speicherprogrammierbarer Steuerungen (ff)

- ❑ Die Verknüpfungsergebnisse, also die Sollsignalzustände von Ausgängen, werden während der Programmbearbeitung in den Speicherzellen des Prozessabbildes der Ausgänge (PAA) hinterlegt (siehe vorherige Folie c). So entsteht Schritt für Schritt ein Prozessabbild für den neuen Sollzustand der Ausgänge der Steuerung.
- ❑ Erst nach Abschluss der Programmbearbeitung, also am Ende des Bearbeitungszyklus', überträgt das Steuerwerk den Inhalt (die Sollsignalzustände) des Prozessabbildes der Ausgänge zu den Ausgabebaugruppen (siehe vorherige Folie d). Danach beginnt das Steuerwerk den nächsten Bearbeitungszyklus mit der Übernahme der Signalzustände der Eingänge, anschließender Programmbearbeitung und nachfolgender Ausgabe des Prozessabbildes der Ausgänge an die Ausgaben. Diese periodische Bearbeitung geht nun mit typischen Zykluszeiten von 1 bis 10 msec bei Ausführung von 1.000 Anweisungen für Binäroperationen so schnell vor sich, dass sequentiell arbeitende speicherprogrammierbare Steuerungen nach außen wie parallel arbeitende verbindungsprogrammierte Steuerung wirken.
- ❑ Die permanent zyklische Bearbeitung ist eine Eigenart der meisten Steuerungen. Obwohl die Zykluszeiten extrem kurz sein können, stellen sich doch bei der Prozessbearbeitung unterschiedliche Zykluszeiten ein. Abhängig vom Prozesszustand durchlaufen Steuerungen zum einen unterschiedliche Bearbeitungswege und zum anderen können die Prozessantwortzeiten unterschiedlich sein, z.B. beim Öffnen oder Schließen von Stellgeräten.



# Abarbeitung eines SPS-Programms

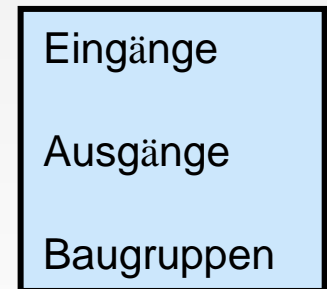
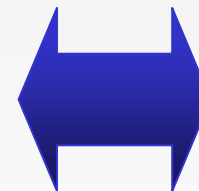
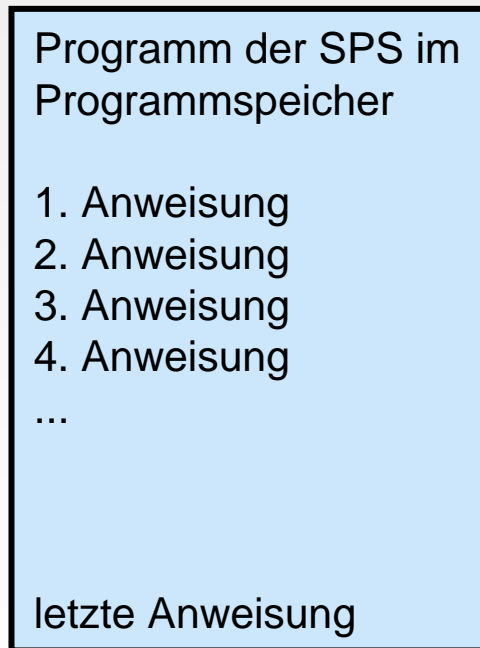
1. Status der Eingänge  
im PAE speichern.



2. Abarbeiten des  
Programms  
Anweisung für  
Anweisung mit Zugriff  
auf PAE und PAA,  
sowie Zeiten Zähler  
und Merker



3. Status aus dem PAA an  
die Ausgänge übertragen.



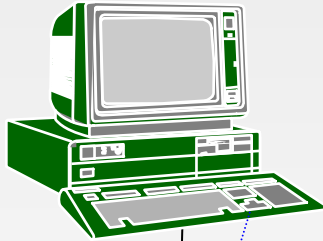
# Programmierung und Projektierung

- ❑ Die Prozessoren **speicherprogrammierbarer Steuerungen** haben wie alle Digitalrechner die von-Neumann-Architektur, d.h. **Programme werden in Speichern abgelegt**. Wie wir wissen und wie auch schon der Name sagt, ist ein **Programm eine Folge der Anweisungen**, bei SPSen in Form von Bitmustern, die eine Steuerung zu bearbeiten hat. Man kann sich das so vorstellen, dass die z.B. 64 Anweisungen oder Befehle eines Prozessors durch unterschiedliche Kombinationen von sechs Schaltern eingestellt werden, wobei ein bestimmtes Bit-Muster einen bestimmten Befehl repräsentiert. Auch Operandenadressen müssen Prozessoren in Form von Bit-Mustern mitgeteilt werden.
- ❑ Nun wird niemand eine SPS mit Bit-Mustern programmieren, also für eine UND-Verknüpfung z.B. jedesmal 011001 einschreiben. Gerätehersteller werden vielmehr **andere Möglichkeiten der Programmierung und Adressierung** bieten z.B. in der Art, für die UND-Verknüpfung ein U auf einem alpha-numerischem Datensichtgerät einzugeben oder Kontaktsymbole in einem Kontaktplan bzw. Funktionsplanelemente in einem **Funktionsplan auf einem graphischen Bildschirm** zu kombinieren. **Adressen** können bei komfortablen Systemen **symbolisch** vorgegeben werden, z.B. als ISTWERT 1 bis ISTWERT n, zumindest aber als Dezimalzahlen. Das **Übersetzen der Informationen** auf einer graphischen Programmieroberfläche in die Maschinensprache einer SPS **besorgt ein Programmiergerät**. (Es wäre technisch möglich, auch diese Aufgaben direkt der SPS zuzuordnen. Besonders kleine SPSen müssten dafür aber sehr aufwendig aufgerüstet werden, was Einfluss auf ihr Zeitverhalten haben kann und besonders deshalb unwirtschaftlich ist, da sich i.a. mit einem einzigen Programmiergerät sehr viele SPSen programmieren lassen, die nur bei Bedarf mit dem PG zu verbinden sind.) So erfolgt die Programmierung der meisten speicherprogrammierbaren Steuerungen mit Programmiergeräten, was eines ihrer kennzeichnenden Merkmale ist.



# Systemstruktur eines SPS-gesteuerten Prozesses

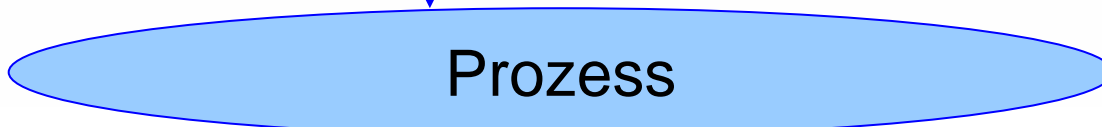
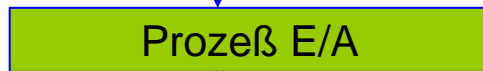
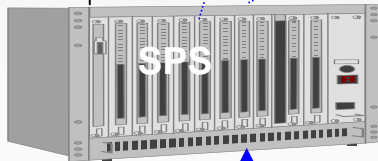
Bedienen und Beobachten  
(Visualisierungssystem/Leitsystem  
z.B. Wonderware, Fix)



Steuerungsprogrammierungssystem  
(z.B. für IEC 61131, S7, MultiProgt)



Kommunikationssystem (z.B. Ethernet, TCP/IP)



Herstellerspezifische  
Punkt-zu-Punkt-Verbindung



# Bausteine zur Rationalisierung des Programmierens

- ❑ Zur Rationalisierung des Programmierens können **häufiger vorkommende Folgen von Steuerungsanweisungen zu Funktionsbausteinen zusammengefasst** werden. So gibt es z.B. Funktionsbausteine zur Ablaufsteuerung, PID-Schrittregelung, Extremwertauswahl, Protokollierung oder Codeumsetzung. Aber der **Anwender kann auch selbst Steuerungsanweisungen zu Funktionsbausteinen zusammenstellen**, z.B. für die Ansteuerung eines Stellgerätes mit zugehörigen Verriegelungen und Überwachung von Laufzeiten.
- ❑ Ein **Funktionsbaustein** entsteht, indem man die Steueranweisungen der zu programmierenden Funktion nicht mit absoluten Parametern angibt, sondern diese allgemein bezeichnet, z.B. mit X1...n, Y1...n, die mit **Formalparameter** bezeichnet werden. Immer dann, wenn man vom Anwenderprogramm her einen Funktionsbaustein **in Anspruch nehmen** will, wird nach dem Aufruf des Bausteins nur noch angegeben, welche **aktuellen Parameter** den formalen zugeordnet sind, d.h. beispielsweise, mit welchen Ein- und Ausgängen, Merkern und Zählern der Funktionsbaustein an dieser Stelle des Programms arbeiten soll. **Dieses in Anspruch nehmen wird mit Instanziierung bezeichnet.**
- ❑ Der Anwender braucht deshalb auch die Befehlsfolge innerhalb eines Standardbausteins nicht zu kennen und wird sie bei komplizierten Bausteinen i.a. auch nicht verstehen. Ein **Baustein** wird von ihm als **"schwarzer Kasten"** betrachtet. Die Hersteller von SPSen liefern Funktionsbausteinen für ihre Systeme, die dann mit Hilfe von Programmiergeräten wie oben beschrieben in Programme eingebunden werden können.





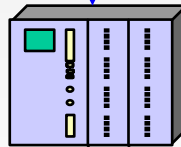
# Problemstellungen



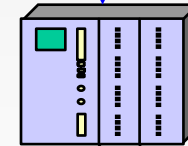
Maschinen-  
funktion



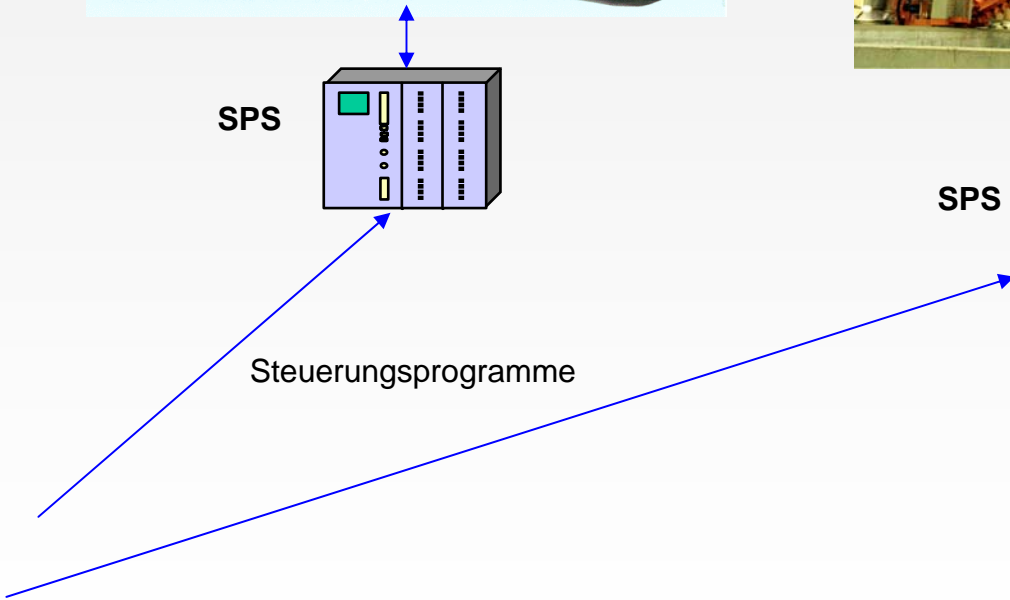
SPS



SPS



Steuerungsprogramme



# Problemstellungen (ff)

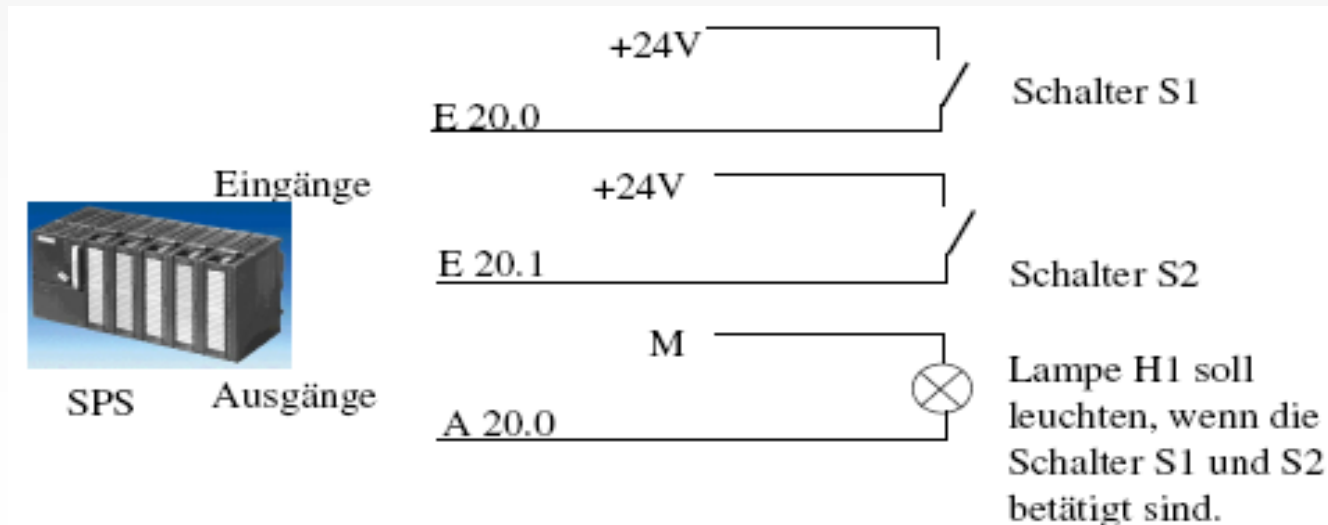
- Maschinen-/Anlagen-/Prozessfunktionen (beispielhaft)
  - Antrieb von beweglichen Komponenten
  - Regelungen von Prozessgrößen
  - Verriegelungen
  - Zeitliche und funktionale Abläufe
  - Sicherheitsfunktionen
  
- Funktionen an der Mensch-Maschine-Schnittstelle (beispielhaft)
  - Anzeige des Maschinen-/Anlagen-/Prozessstatus
  - Anzeige von Istwerten
  - Veränderungen von Sollwerten
  - Anzeige von Alarmen
  - Auswertungen von Maschinen-/Anlagen-/Prozessereignissen
  - Auswertung von Maschinen-/Anlagen-/Prozessdurchsätzen/ und -leistungen



# Informationsverarbeitung, UND-Verknüpfung

## □ Beschaltung der SPS

- Beide Schalter werden an Eingängen der SPS angeschlossen
  - S1 an Eingang E 20.0
  - S2 an Eingang E 20.1
- Die Lampe wird an einem Ausgang angeschlossen
  - H1 an Ausgang A 20.0



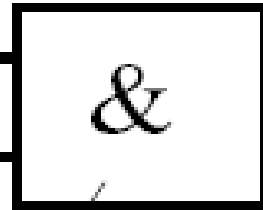
# Informationsverarbeitung, UND-Verknüpfung (ff)

Eingänge der UND-Verknüpfung.

Hier können auch mehr als 2 Eingänge stehen!

E 20.0

E 20.1



A 20.0



Ausgang, dem die Zuweisung zugeordnet wird!

Bildliche Darstellung der logischen UND-Verknüpfung!

Zuweisung des Ergebnisses der logischen Verknüpfung!

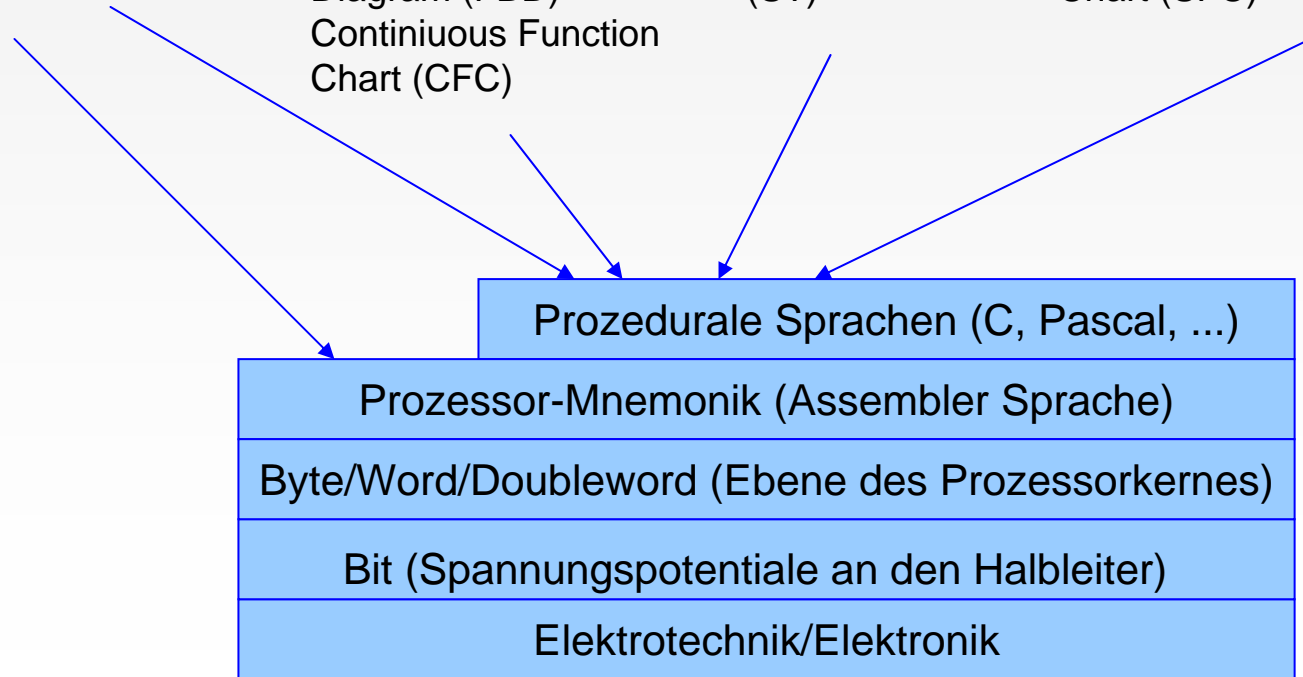


# Beschreibungshierarchie

## Technologisch-orientierte Sprachen

Boolsche Algebra	Datenfluß-orientiert	Prozedural	Ablauforientiert
Anweisungsliste (AWL) Instruction List (IL)	Funktionsblockplan (FUP) Function Block Diagram (FBD) Continuous Function Chart (CFC)	C, Pascal Structured Text (ST)	Perti Netze Sequential Function Chart (SFC)

**Funktions-  
beschreibung**



**Software/  
(Hardware)  
Realisierung**

