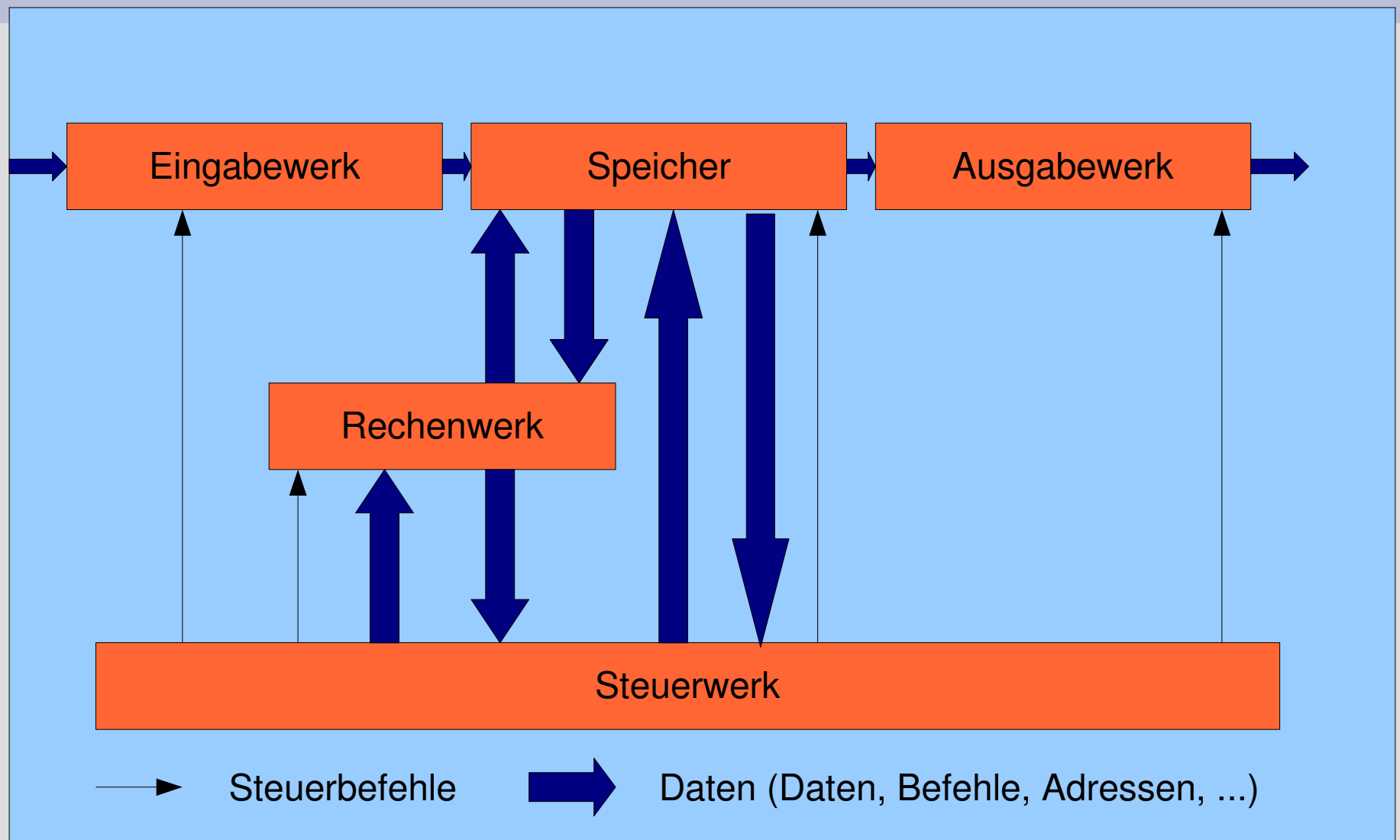


Ein- und Ausgabe

- von-Neumann Konzept enthält folgende Komponenten:
 - Rechenwerk
 - Steuerwerk
 - Speicher
 - Eingabewerk
 - Ausgabewerk(siehe 1. Vorlesung)

v.-Neumann-Architektur



Ein- und Ausgabe (2)

- Bisher behandelt:
 - Steuerwerk
 - Rechenwerk
 - Speicher
- aber **Wo kommen die Daten her, die zu verarbeiten sind und wie erfährt der Nutzer von den Ergebnissen?**

Ein- und Ausgabe (3)

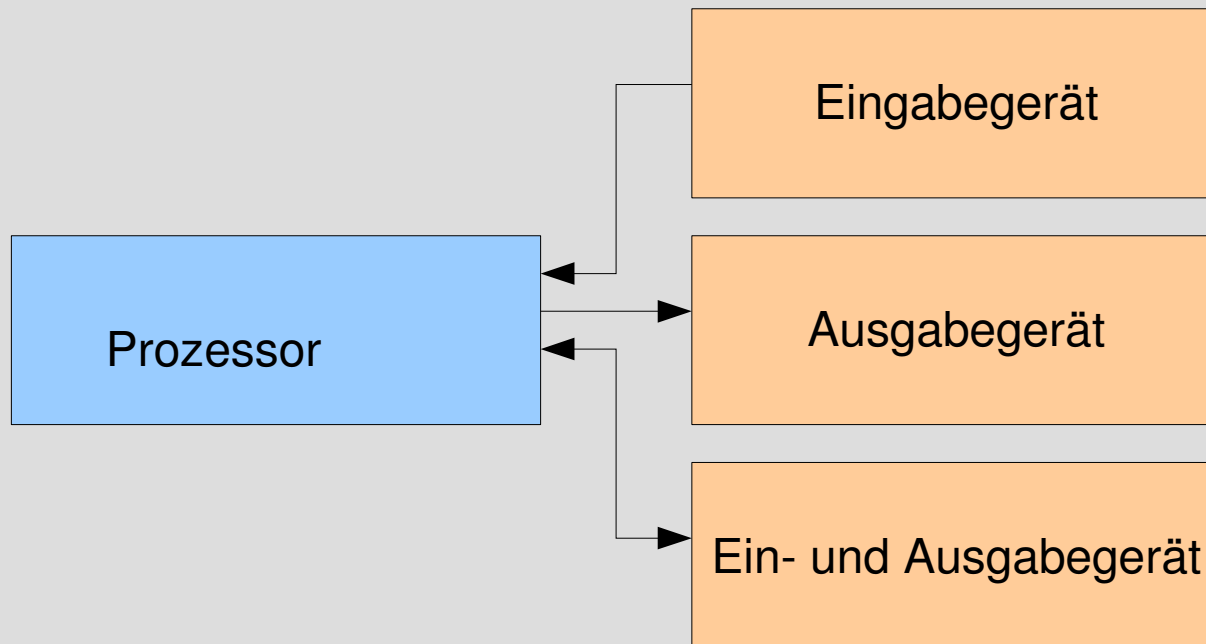
- Vielzahl von Geräten:
 - Eingabegeräte
 - Tastatur, Maus, Trackball, Joystick
 - Lochkartenleser, Lochbandleser
 - Scanner
 - CD-ROM-Laufwerk
 - Meßgeräte, Mikrofon, Kamera, Sensoren
 - Ausgabegeräte
 - Lochkartenstanzer, Lochbandstanzer
 - Drucker, Monitor, Beamer
 - Lautsprecher
 - Werkzeugmaschinen (NC), Aktoren (Relais, Motor,..)

Ein- und Ausgabe (4)

- Ein- und Ausgabegeräte
 - Magnetbandspeicher (z.B. Streamer)
 - Floppy Disk
 - Hard Disk
 - Telefon
 -

Ein- und Ausgabe (5)

- Alle diese höchst unterschiedlichen Geräte müssen nun mit den „Kernkomponenten“ des Rechners verbunden werden:

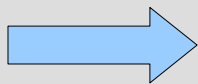


Ein- und Ausgabe (6)

- **Standard-Interface**

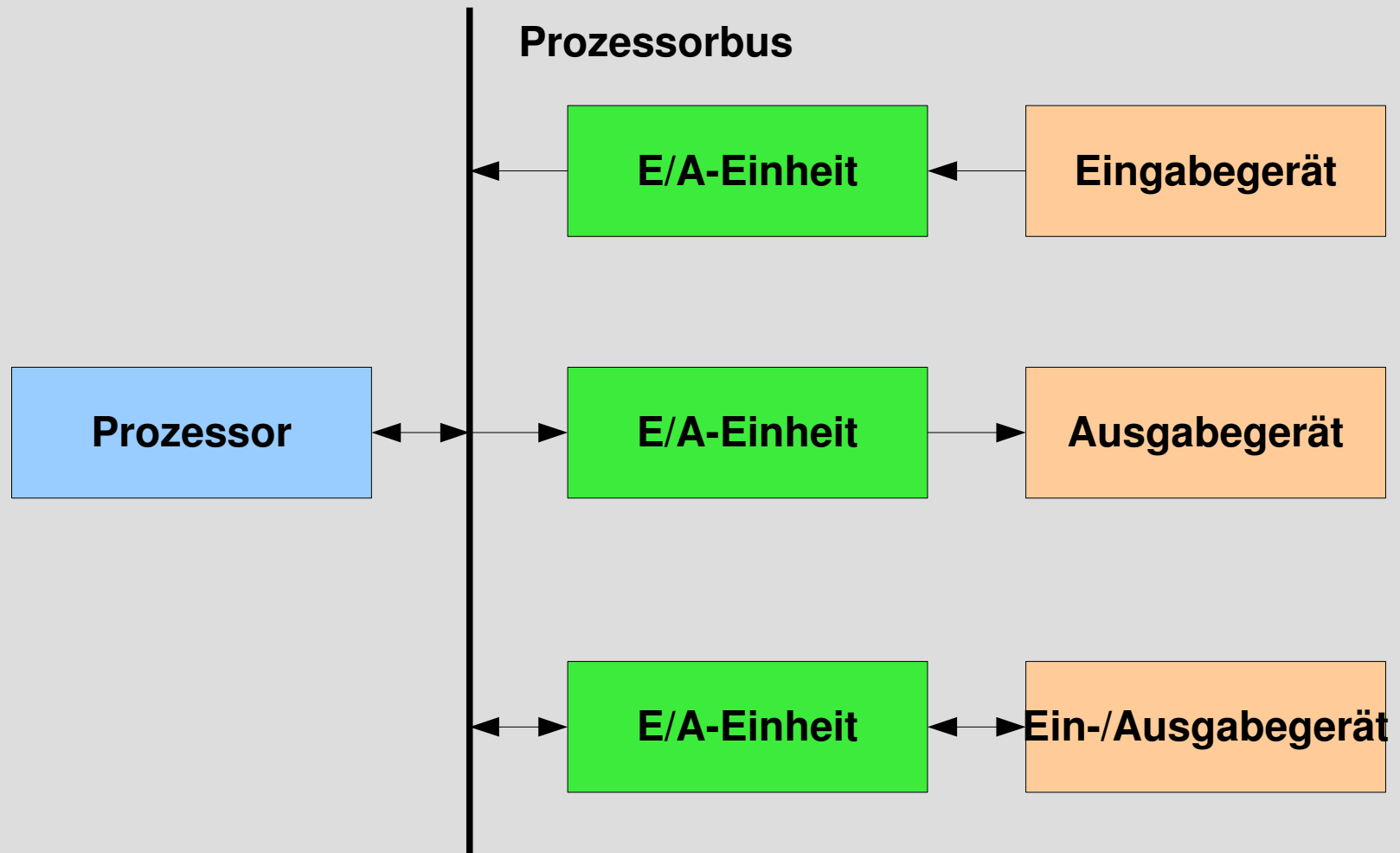
Gesamtheit elektrischer, konstruktiver, logischer und zeitlicher Bedingungen („Protokoll“), die die Koppelbarkeit zweier Einheiten (z. B. eines Prozessors und eines peripheren Gerätes) sichern

- Dynamik des Marktes bei peripheren Geräten deutlich größer als bei Prozessoren
- Ein peripheres Gerät muß mit unterschiedlichen Prozessoren arbeiten können

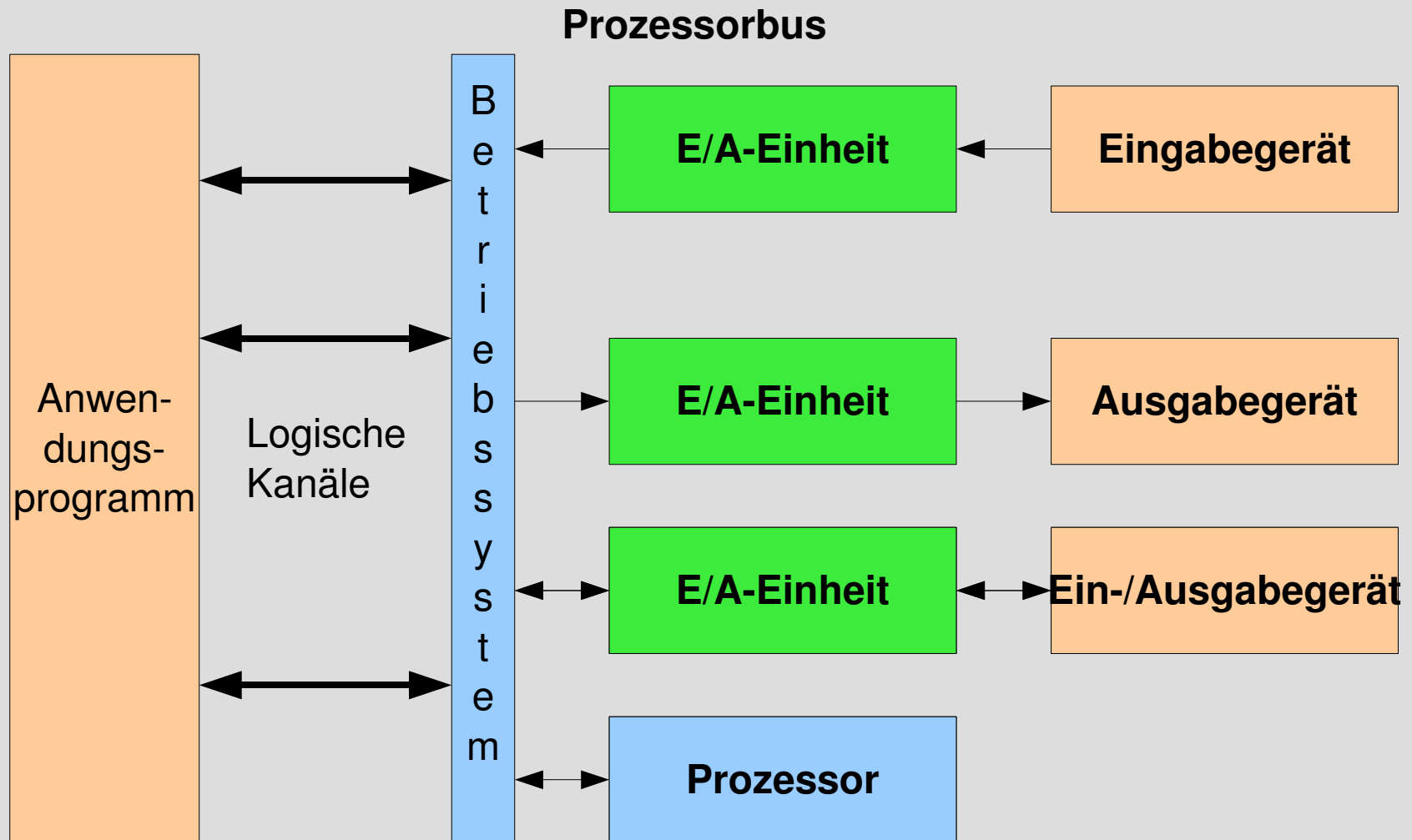


Ein- Ausgabeeinheiten zur Umsetzung der Schnittstelle des Prozessors auf die der E/A-Geräte

Ein- und Ausgabe (7)



Ein- und Ausgabe (8)



Ein- und Ausgabe (9)

- Wie wird die Ein/Ausgabeanforderung eines Benutzers in einen Gerätebefehl umgewandelt und an das Gerät übermittelt?
- Wie werden Daten zu oder von einer Speicherposition übertragen?
- Welche Rolle spielt das Betriebssystem?

Ein- und Ausgabe (10)

- **Rolle des Betriebssystems:**
- Schnittstelle zwischen Hardware und dem die Ein/Ausgabe anforderndem Programm
 - Mehrere Programme, die den Prozessor verwenden, nutzen das E/A-System gemeinsam
 - E/A-Systeme verwenden häufig Interrupts, um Informationen über E/A-Operationen zu übermitteln. Weil Interrupts einen Übergang in den Kernel- oder Supervisormodus verursachen, müssen sie vom Betriebssystem ausgeführt werden.

Ein- und Ausgabe (11)

- Low-level-Steuerung eines E/A-Gerätes ist komplex, weil dafür verschiedene nebenläufige Ereignisse verwaltet werden müssen und weil die Anforderungen für eine korrekte Gerätesteuerung häufig sehr detailliert sind.
- **weitere Behandlung: Lehrveranstaltung Betriebssysteme**

Verfahren

- **Polling**

Regelmäßige Abfrage eines Statusbits
(E/A-Gerät stellt Information in Statusregister,
Prozessor muss sich die Information holen)

Vorteil: sehr einfach

Nachteil: sehr viel Prozessorzeit

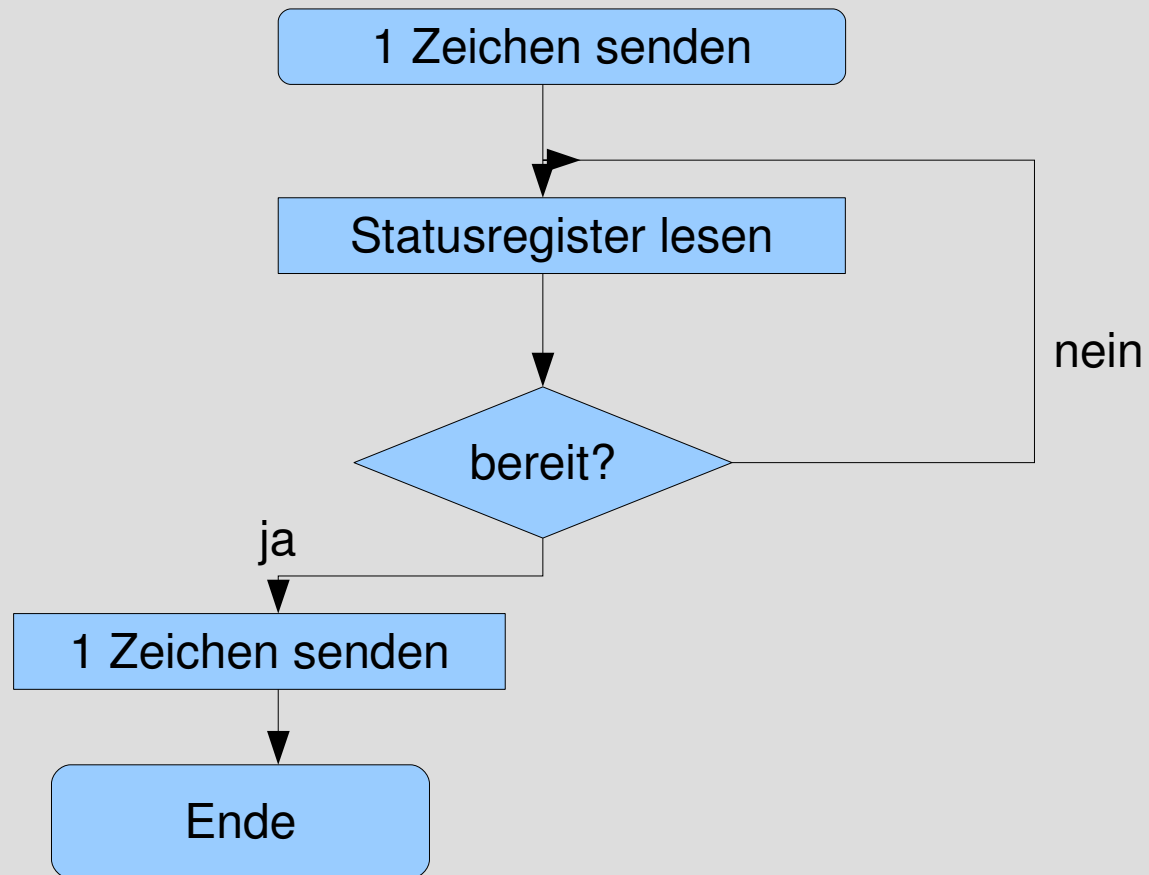
- **Interrupt**

Prozessor wird asynchron unterbrochen

- dies wird heute meist angewandt (siehe auch
4. Vorlesung)

Verfahren (2)

- Beispiel: serielle Schnittstelle



Verfahren (3)

- **Polling**
Regelmäßige Abfrage eines Statusbits
(E/A-Gerät stellt Information in Statusregister,
Prozessor muss sich die Information holen)
Vorteil: sehr einfach
Nachteil: sehr viel Prozessorzeit
- **Interrupt**
Prozessor wird asynchron unterbrochen
- **dies wird heute meist angewandt (siehe auch 4. Vorlesung)**

Ein- und Ausgabeeinheiten

- *PIO Parallel Input Output – Unit*
 - *Beispiele:*
 - *I8212 (8 bit Fang-Register mit Interruptsteuerung)*
 - *I8266 (3 Kanal PIO des 8086 Systems)*
 - *Z80-PIO (in der DDR: U855, 2 Kanal PIO des Z80 Systems)*
- *DMA Direct Memory Access*

PIO

- Blöcke von mehreren logischen Signalen können aus dem Rechner an die Umgebungselektronik abgegeben, oder aus der Umgebungselektronik in den Rechner eingelesen werden.
- Das geschieht häufig auf einer Breite von 8 Bit (neuere Entwicklungen: 32 Bit).

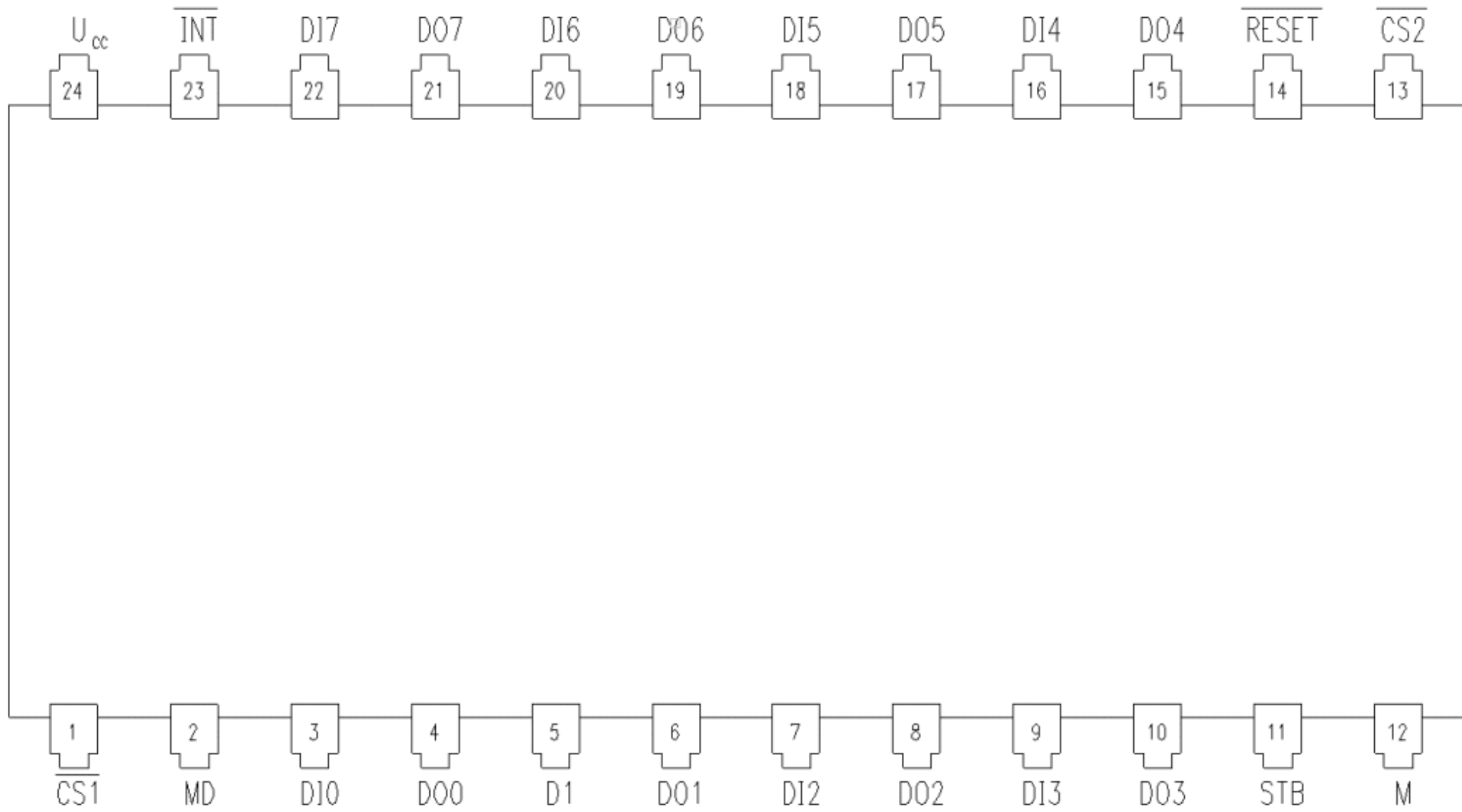
PIO (2)

- Byte- Word oder Double-Word organisierten Datenstrukturen dominieren.
- mehrere Bits mit einem "Kommando" auf die Reise zu einer Gegenstelle geschickt werden.
- diese sollte dazu auch in der Lage sein, die entsprechende Bitanzahl aufzunehmen und sollte dazu auch entsprechend programmiert sein.

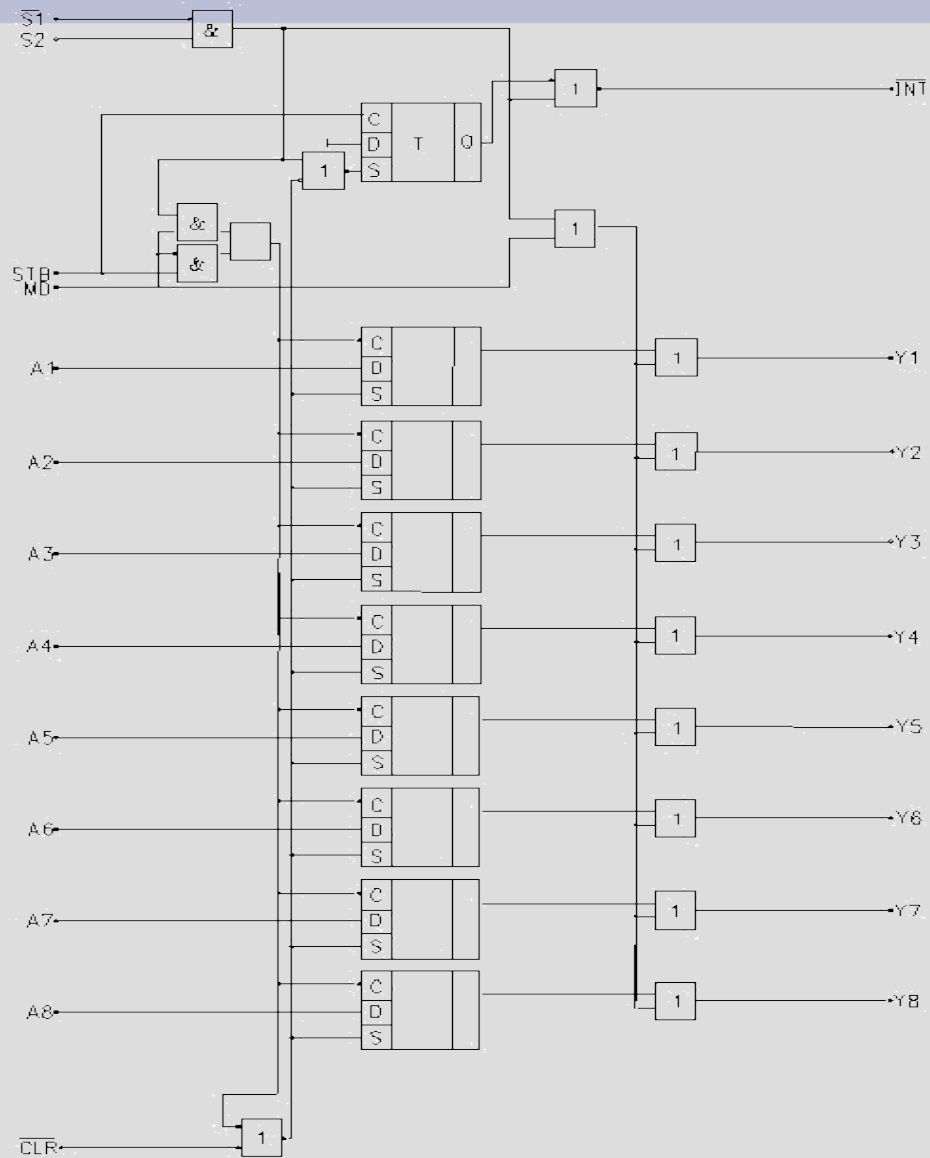
PIO (3)

- Die Verfügbarkeit muss also entweder langfristig und in einem garantierten Zeitraum liegen (wobei ms hier schon extrem langfristig sind).
- Der PIO-Kanal hält Daten auch noch dann bereit, wenn auf dem internen Daten-BUS bzw. auf den externen Verbindungsleitungen schon lange andere Daten liegen.

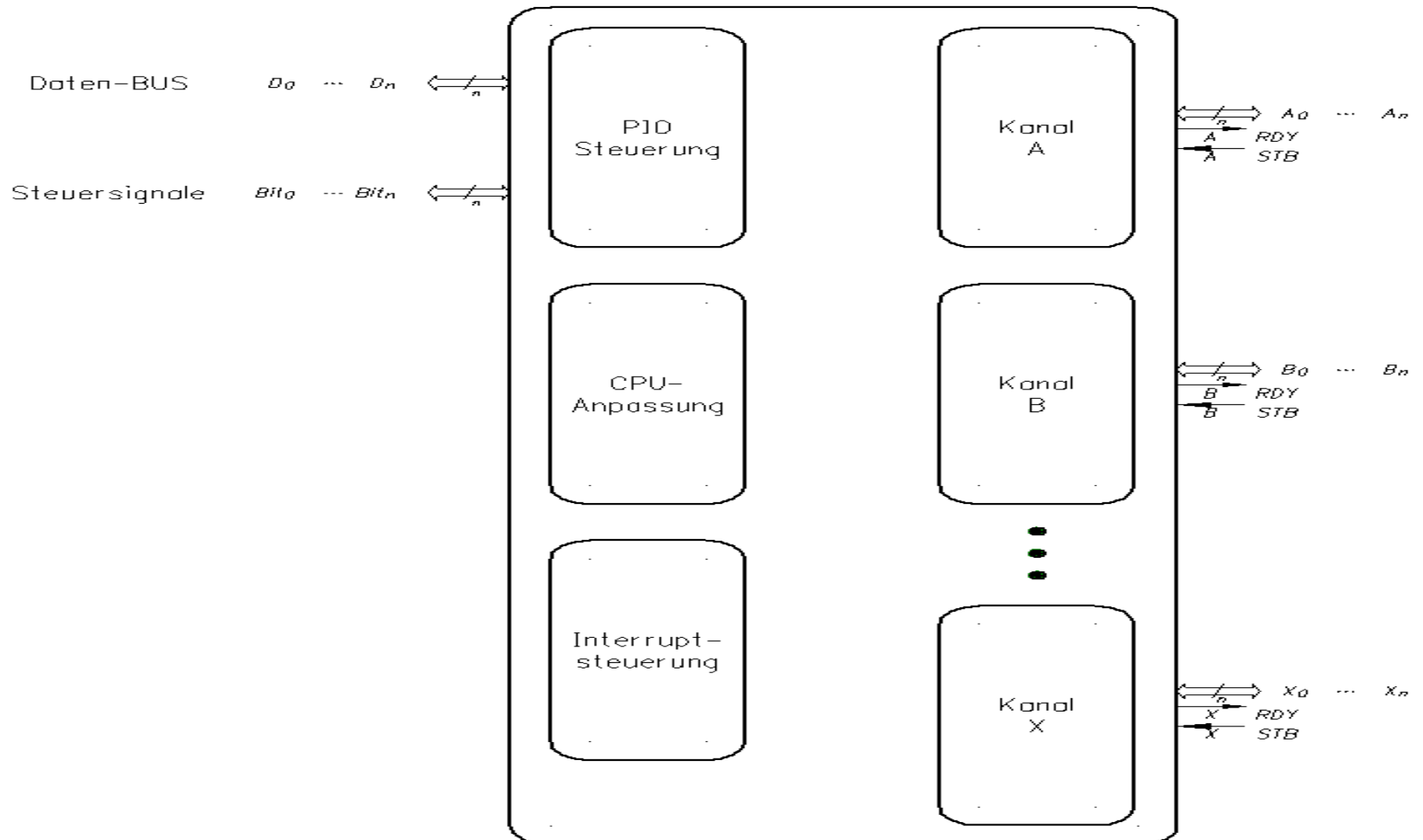
PIO (I8212)



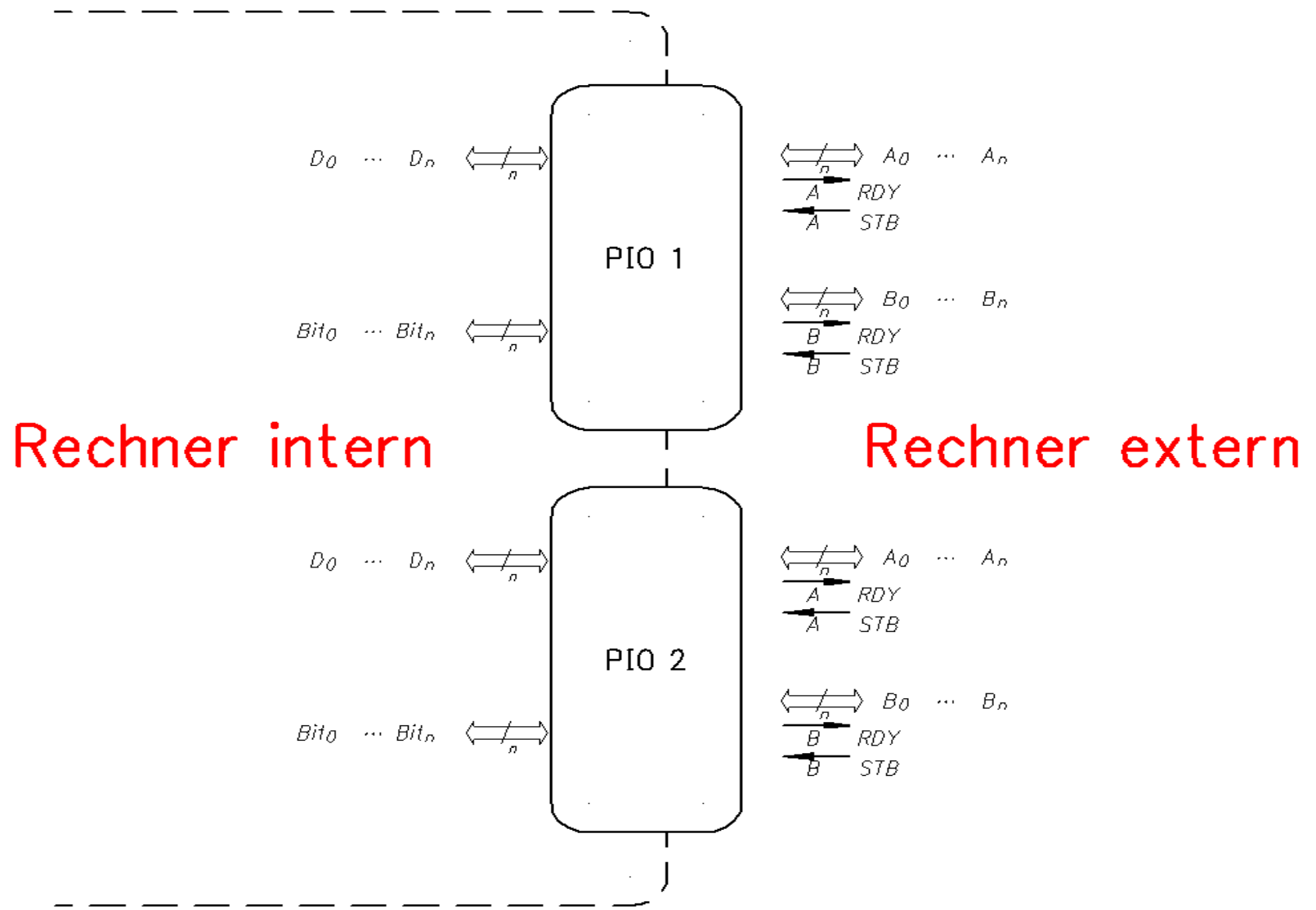
PIO (I8212)



PIO (4)



PIO (5)



DMA

- Interruptgesteuerte E/A ==> befreien Prozessor davon, auf jedes einzelne E/A-Ereignis zu warten
- Verwendung dieser Methode zum Festplattenzugriff ==> großer Teil der Prozessorzeit wird mit Datenübertragung verbraucht.

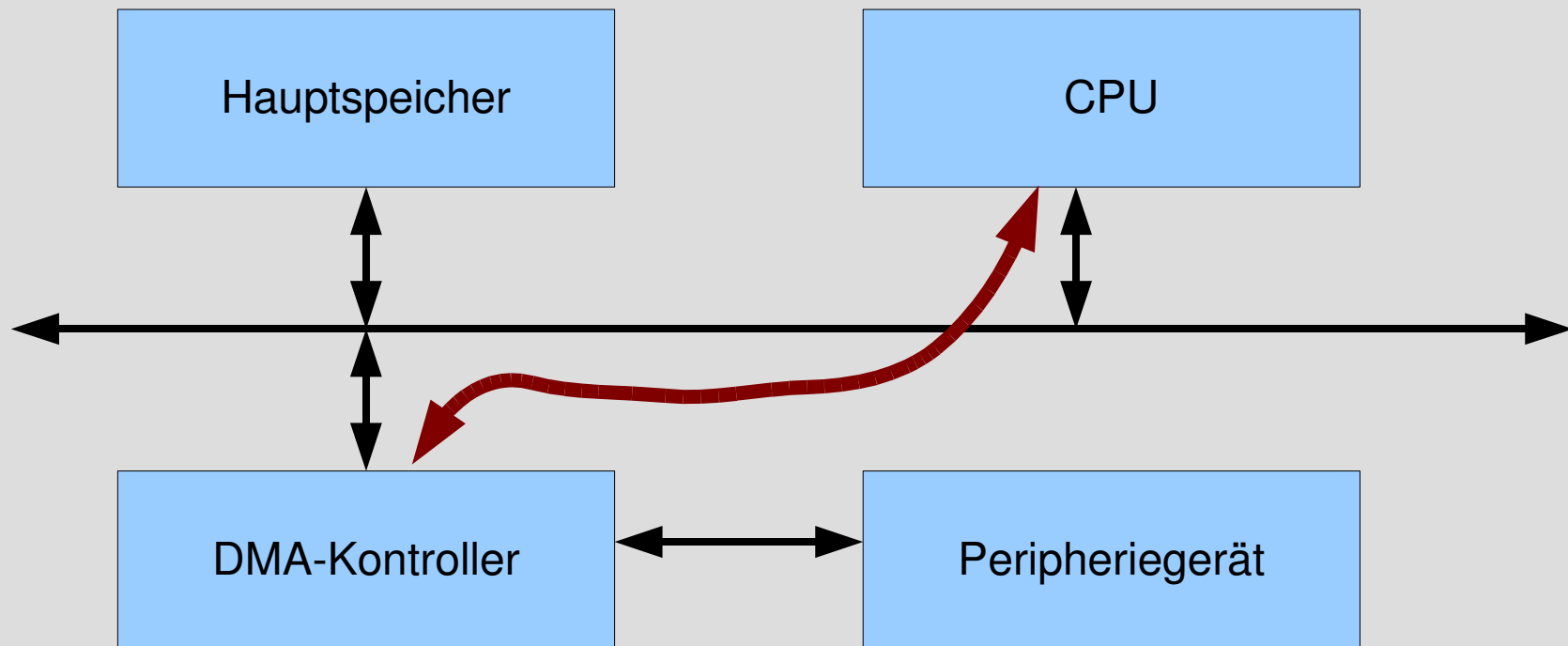
DMA (2)

- Mechanismus, der Last vom Prozessor nimmt und es dem Gerätekontroller überläßt, Daten direkt zum oder vom Speicher zu transportieren, ohne den Prozessor daran zu beteiligen ==> ***Direct Memory Access (DMA)***
- Interrupt wird weiter benutzt, um mit dem Prozessor zu kommunizieren, aber nur, um Ende der Kommunikation bzw. Fehler anzuzeigen.

DMA (3)

- **Ablauf:**
 - **Prozessor richtet DMA ein (Angabe von:**
 - **Identität des Gerätes**
 - **auszuführende Operation**
 - **Quelle/Ziel der Daten**
 - **Anzahl der Bytes**
 - DMA startet Operation und teilt Bus zu
 - Nach Abschluß unterbricht DMA Prozessor mit Interrupt, Prozessor stellt Korrektheit der Übertragung fest.

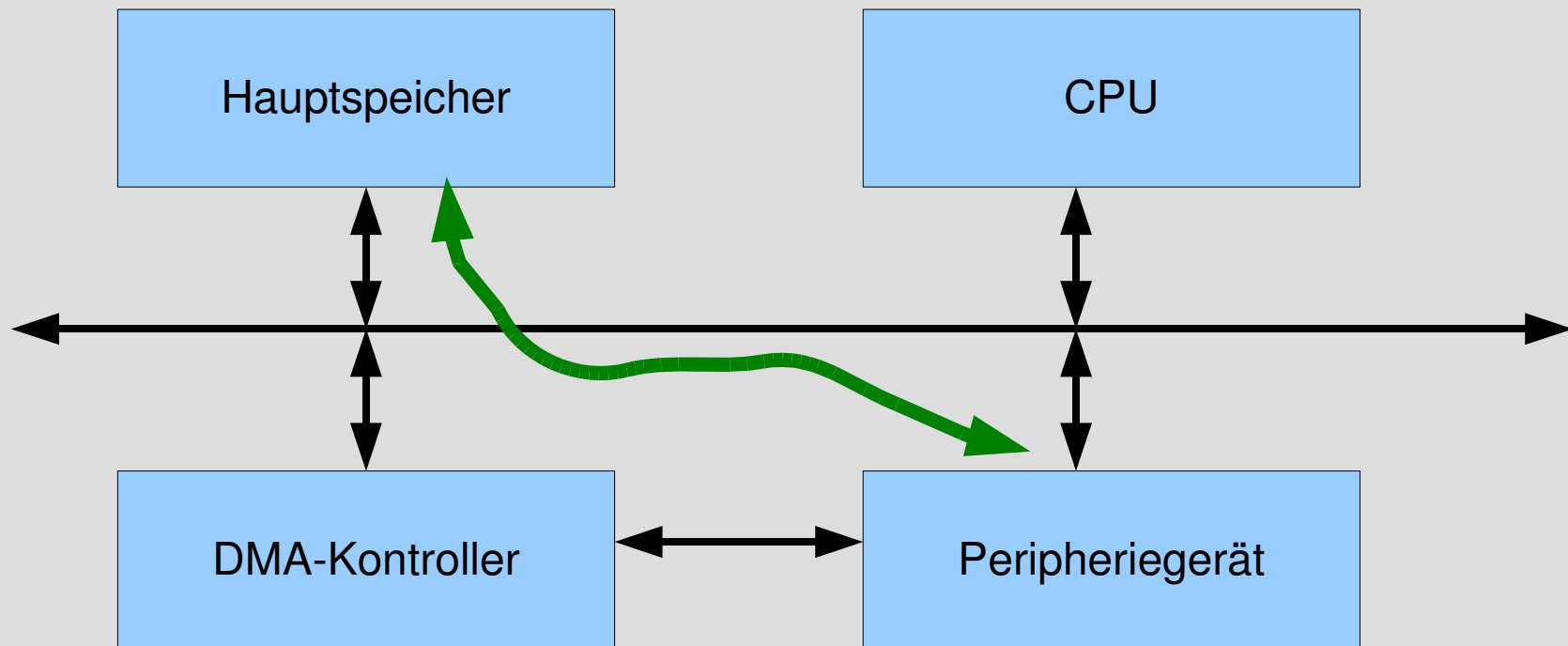
DMA (4)



DMA (5)

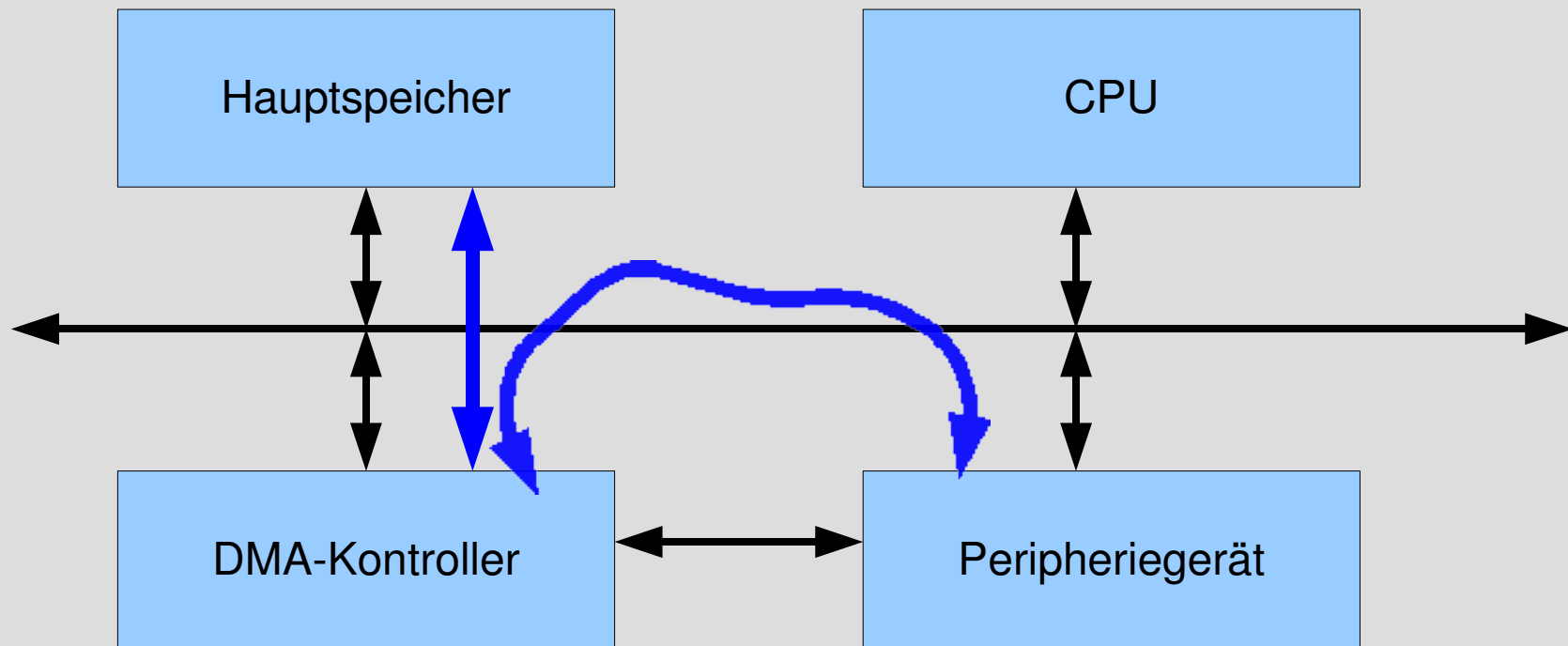
- Ablauf:
 - Prozessor richtet DMA ein (Angabe von:
 - Identität des Gerätes
 - auszuführende Operation
 - Quelle/Ziel der Daten
 - Anzahl der Bytes
 - **DMA startet Operation und teilt Bus zu**
 - Nach Abschluß unterbricht DMA Prozessor mit Interrupt, Prozessor stellt Korrektheit der Übertragung fest.

DMA (6)



Direkter Transfer

DMA (7)



Indirekter Transfer

DMA (8)

- Ablauf:
 - Prozessor richtet DMA ein (Angabe von:
 - Identität des Gerätes
 - auszuführende Operation
 - Quelle/Ziel der Daten
 - Anzahl der Bytes
 - DMA startet Operation und teilt Bus zu
 - **Nach Abschluß unterbricht DMA Prozessor mit Interrupt, Prozessor stellt Korrektheit der Übertragung fest.**

Vergleich E/A-Verfahren

Polling

**Interrupt
gesteuert**

DMA

Initialisierung

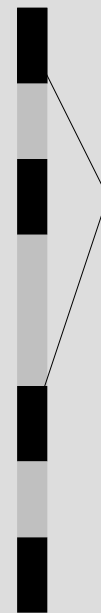
1. Wert bereit

2. Wert bereit

3. Wert bereit

...

**letzter Wert
bereit**



CPU belegt



Amdahls's Gesetz

- Ausführungszeit nach der Verbesserung =
(von der Verbesserung betroffene
Ausführungszeit) / Verbesserungsfaktor +
von Verbesserung nicht betroffene
Ausführungszeit

E/A-Leistung im Vergleich mit Prozessorleistung

- Beispiel:
 - Benchmark, der in 100 Sekunden Gesamtausführungszeit ausgeführt wird. 90 % Prozessorzeit, 10 % E/A Zeit
 - Um wieviel schneller wird ein Programm ausgeführt, wenn sich die CPU-Zeit in den nächsten 5 Jahren jährlich um 50 %, die E/A-Zeit jedoch gar nicht verbessert?

E/A-Leistung im Vergleich mit Prozessorleistung (2)

Jahre	CPU-Zeit	E/A Zeit	Gesamtzeit	Anteil E/A in %
0	90	10	100	10
1	60	10	70	14
2	40	10	50	20
3	27	10	37	27
4	18	10	28	36
5	12	10	22	45