

Grundlagen der Informatik

Rechnerarchitektur

Dr. Peter Jüttner

Übersicht / Zusammenfassung



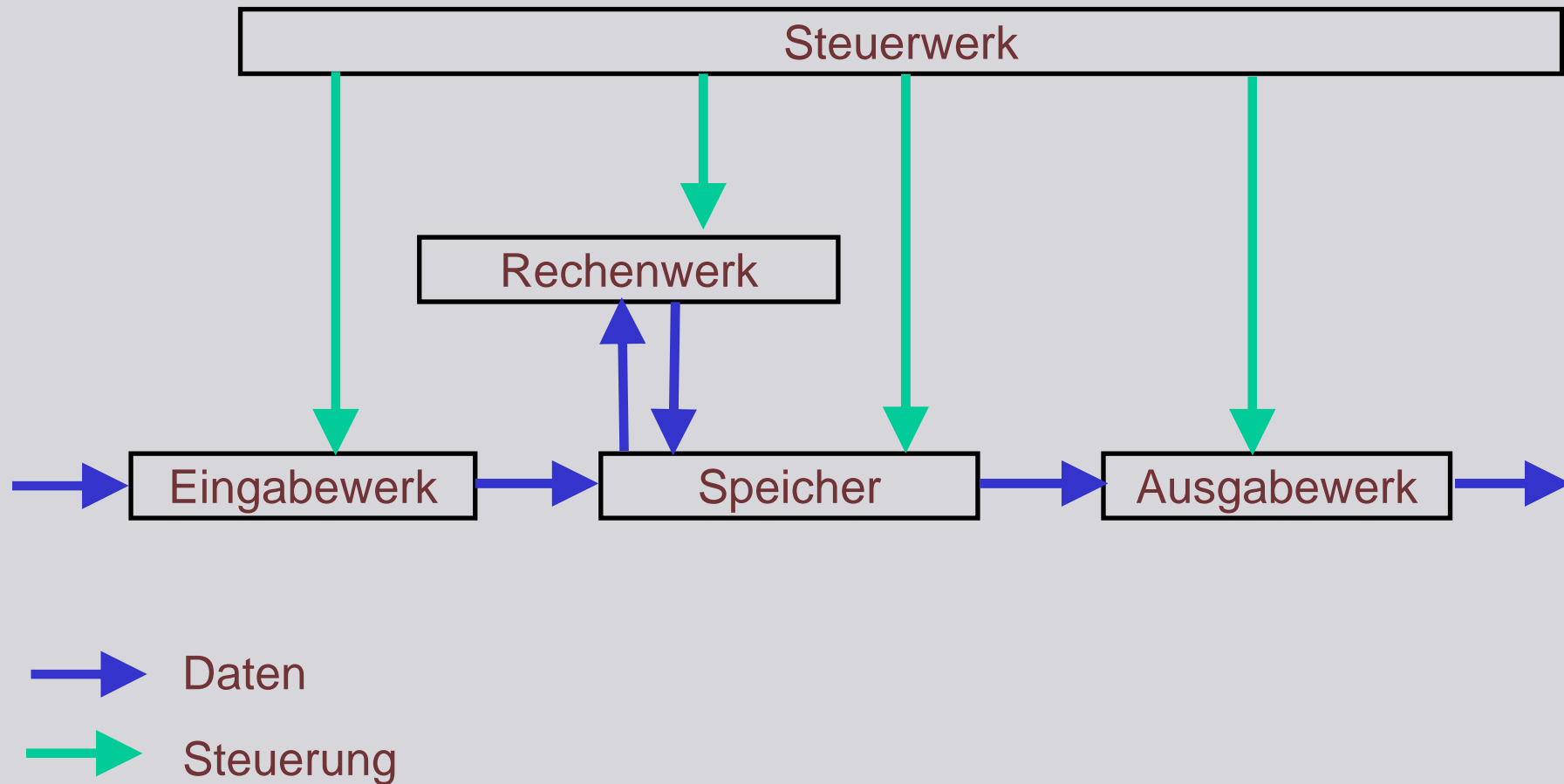
Rechnerarchitektur

Von Neumann Architektur

- benannt nach John von Neumann (1903 – 1957)
- abstrakte, allgemein einsetzbare Rechnerarchitektur (unabhängig von der Anwendung des Rechners)
- wird von heutigen Rechnern (mehr oder weniger) umgesetzt
- abstrakte Sichtweise
- dient dem grundlegenden Verständnis der Funktionsweise eines Computers
- dient dem grundlegenden Verständnis der Abarbeitung von Software

Rechnerarchitektur

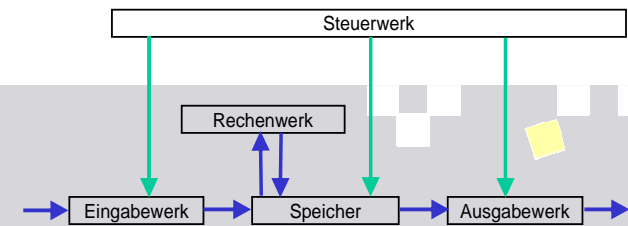
Übersicht Von Neumann Architektur



Rechnerarchitektur

Von Neumann Architektur

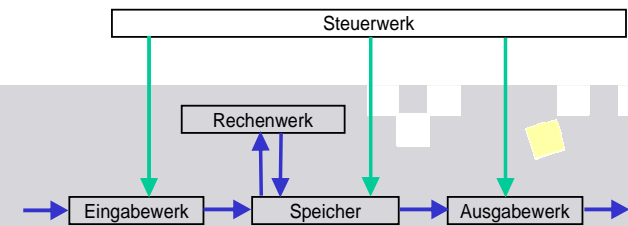
- Zur Lösung eines Problems muss von außen ein Programm eingegeben und im Speicher abgelegt werden.
- Ohne Programm ist der Rechner Maschine nicht arbeitsfähig.
- Programme, Daten inkl. Zwischen- und Endergebnisse nutzen den selben Speicher.
- Der Speicher ist in gleich große Zellen unterteilt, die fortlaufend nummeriert sind.



Rechnerarchitektur

Von Neumann Architektur

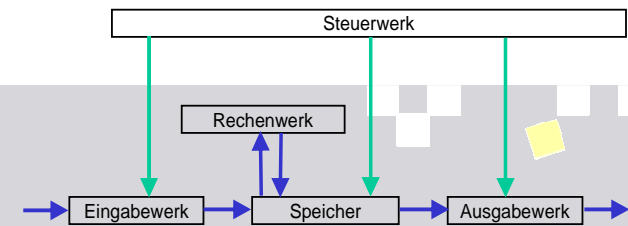
- Über die Adresse einer Speicherzelle kann deren Inhalt gelesen oder geschrieben werden.
- Die Ausführung eines Programms wird durch das Steuerwerk gesteuert.
- Aufeinander folgende Befehle eines Programms werden in aufeinander folgenden Speicherzellen abgelegt.
- Das Ansprechen des nächsten Befehls geschieht vom Steuerwerk aus durch Erhöhen der Befehlsadresse um eins.
- Durch Sprungbefehle kann von der Bearbeitung der Befehle in der gespeicherten Reihenfolge abgewichen werden.



Rechnerarchitektur

Von Neumann Architektur

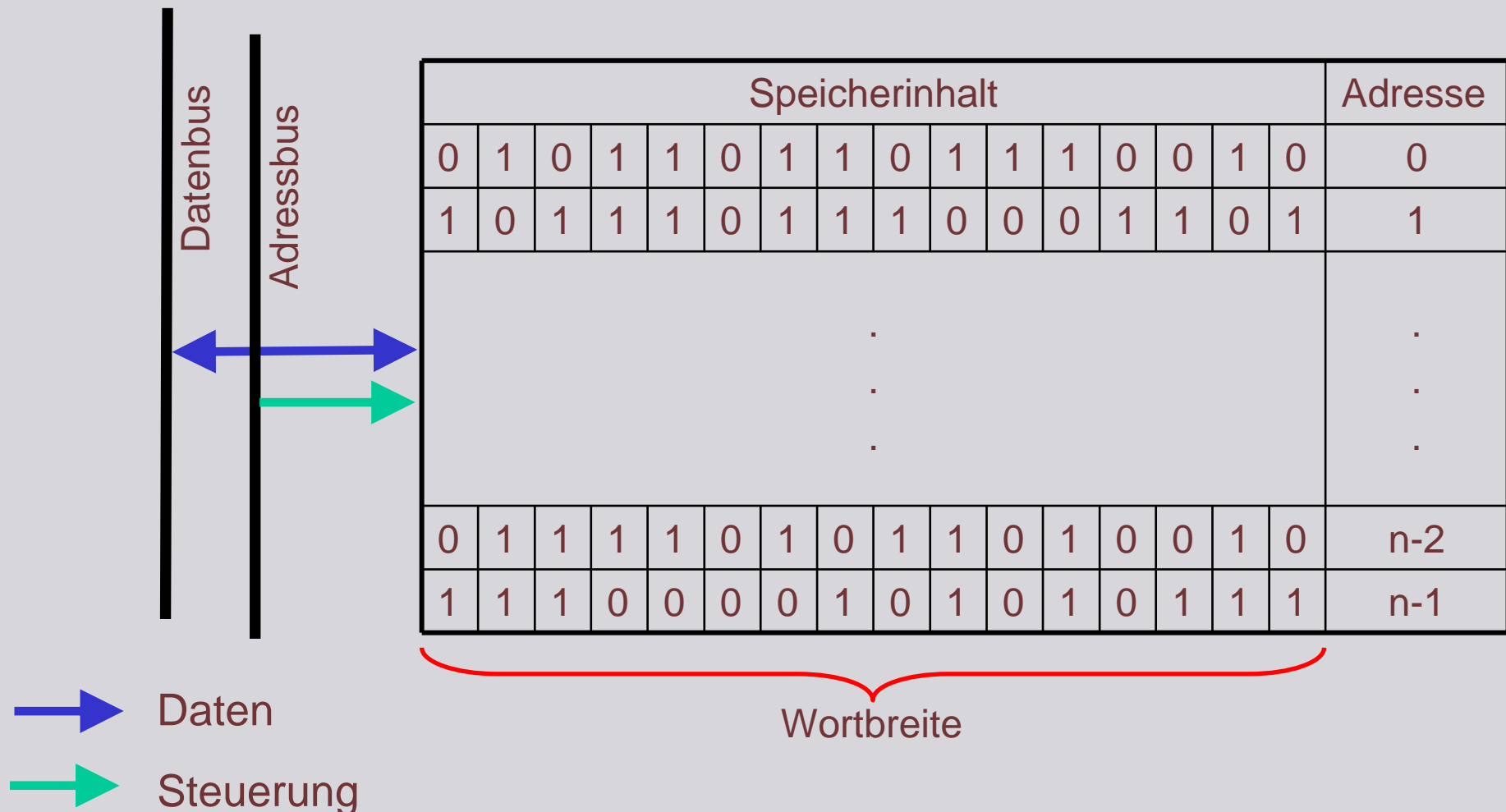
- Es gibt (mindestens)*)
 - arithmetische Befehle wie Addieren, Multiplizieren usw.
 - logische Befehle wie Vergleiche, logisches Und, Oder usw.
 - Transportbefehle, z. B. vom Speicher zum Rechenwerk und für die Ein-/Ausgabe
 - bedingte Sprünge (if 0 goto)



*) Sprachumfang von Assemblersprachen

Rechnerarchitektur

Von Neumann Architektur – Der Speicher



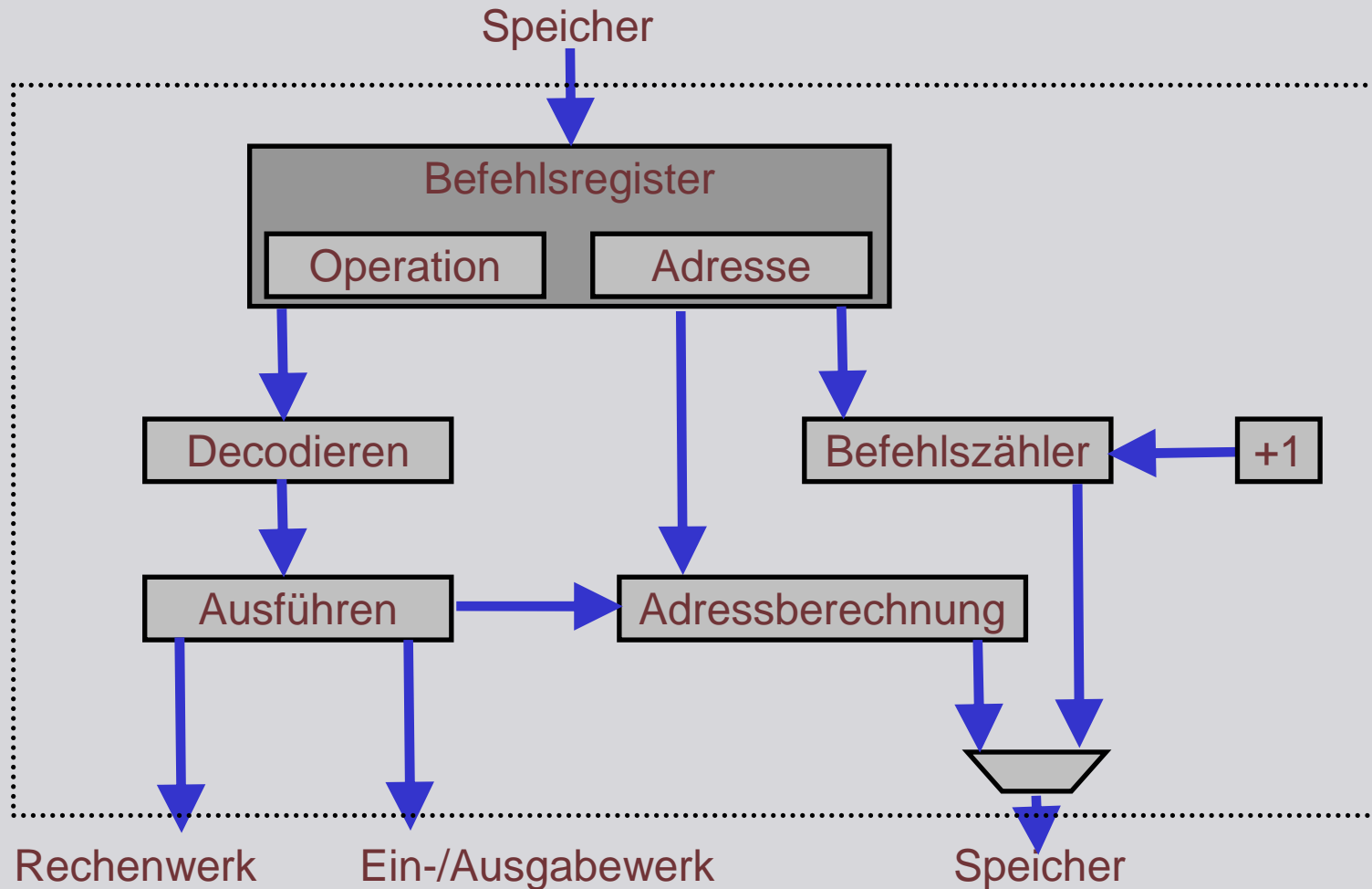
Rechnerarchitektur

Von Neumann Architektur – Der Speicher

- Für den Zugriff auf den Speicher existieren zwei Leitungen (Busse): Adressbus und Datenbus
- Jede Speicherzelle umfasst w (die Wortbreite) Bits.
- Auf dem Adressbus wird die Adresse (Nummer) a der Speicherstelle übermittelt, auf die zugegriffen werden soll.
- Soll im Speicher geschrieben werden, wird die adressierte Speicherzelle mit dem Wert auf dem Datenbus überschrieben
- Soll im Speicher gelesen werden, wird der Wert der adressierten Speicherzelle auf den Datenbus geschrieben

Rechnerarchitektur

Von Neumann Architektur – Das Steuerwerk



Rechnerarchitektur

Von Neumann Architektur – Das Steuerwerk

- steuert die Ausführung eines Programms
- holt Befehle nacheinander aus dem Speicher (**fetch**) in das Befehlsregister.
- Jeder Programmbefehl besteht aus zwei Teilen:
 - Der Operationsteil legt fest, was gemacht werden soll
 - Der Adressteil bestimmt, auf welche Daten der Befehl ggf. anzuwenden ist.

Rechnerarchitektur

Von Neumann Architektur – Das Steuerwerk

- Jeder Programmbefehl wird decodiert (**decode**)
- danach ausgeführt (**execute**) indem entsprechende Funktionseinheiten (z.B. Rechenwerk, Ein-Ausgabe) aktiviert werden.

Rechnerarchitektur

Von Neumann Architektur – Das Steuerwerk

- Das Befehlszählerregister enthält die Speicheradresse des aktuell ausgeführten Befehls
- Nach Ausführung eines Befehls wird das Befehlsregister auf die Adresse des nächsten auszuführenden Befehls geändert und der nächste Befehl wird geholt.
 - Befehlsregister + 1 (falls kein Sprungbefehl)
 - Sprungziel sonst

Rechnerarchitektur

Von Neumann Architektur – Das Steuerwerk

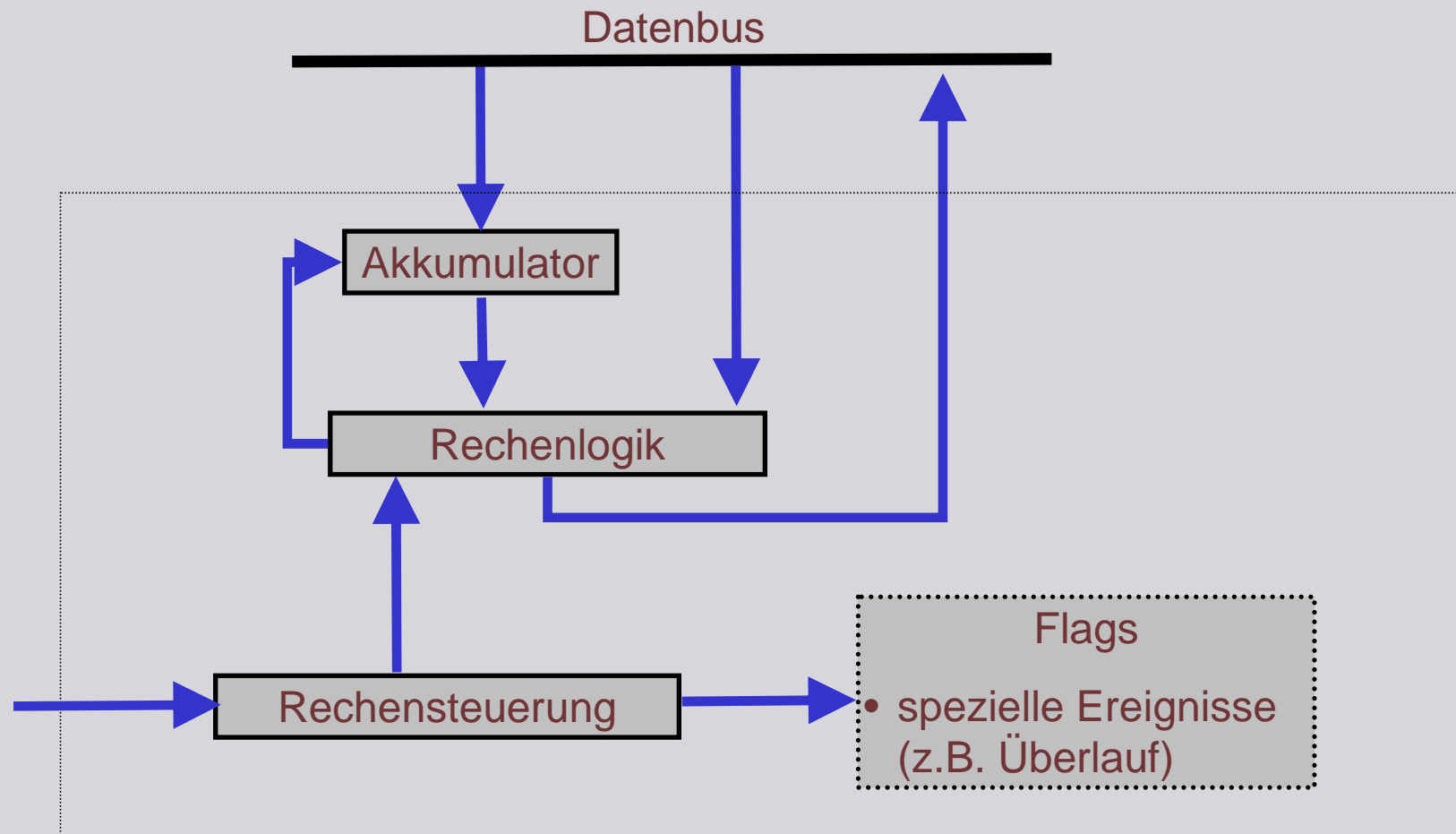
- Sind Daten für eine Befehlsausführung im Speicher abgelegt, so müssen diese vor Ausführung geholt bzw. nach Ausführung wieder gespeichert werden:
 - Zuerst wird bestimmt, auf welche Speicherstelle zugegriffen werden muss (Adressberechnung: **address**)
 - Dann wird der Speicherzugriff (**read / write**) durchgeführt. (der Befehl kann auch zwei oder oder mehr Worte umfassen)

➔ Vollständiger Befehlszyklus:

fetch ➔ decode ➔ address ➔ read ➔ excute ➔ write

Rechnerarchitektur

Von Neumann Architektur – Das Rechenwerk



Rechnerarchitektur

Von Neumann Architektur – Das Rechenwerk

- Akkumulator dient als Speicher für Zwischenergebnisse.
- Beim klassischen Von-Neumann-Modell enthält der Akkumulator immer den ersten Operanden (evtl. vorher aus dem Speicher laden!) und das Ergebnis.
- Die Rechenlogik führt arithmetische Operationen aus

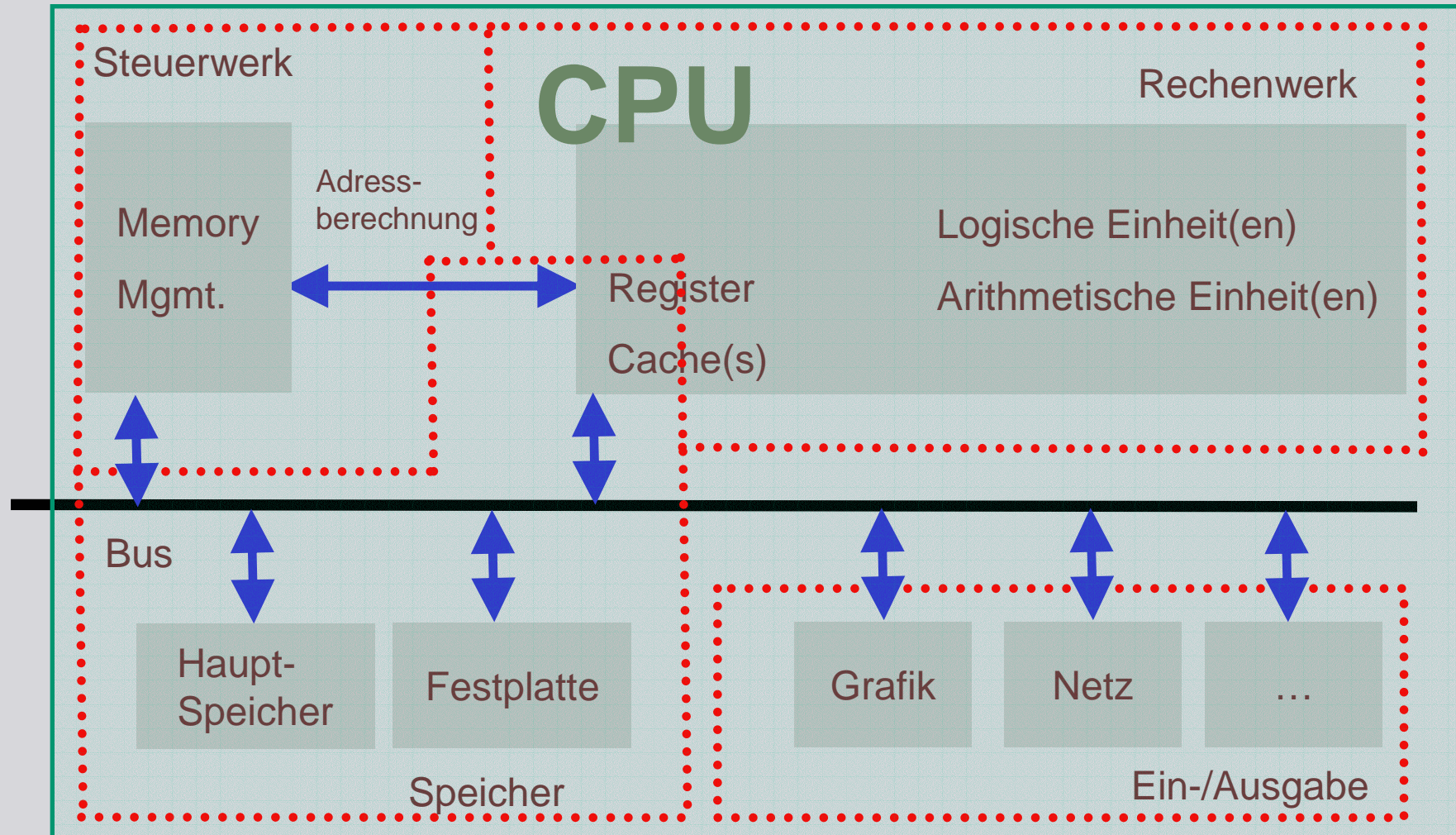
Rechnerarchitektur

Reale Rechnerarchitekturen



Rechnerarchitektur

Reale Rechnerarchitekturen



Rechnerarchitektur

Reale Rechnerarchitekturen

- Laden größerer Speicherbereiche in speziellen Speicher (**Cache**)
 - ➔ Ausnutzen der „Lokalität“ zusammengehöriger Daten
- Bearbeitung mehrerer Befehle gleichzeitig (**Pipelining**: Parallelität innerhalb der Befehlsausführung durch Überlagerung von Phasen aufeinanderfolgender Befehle)

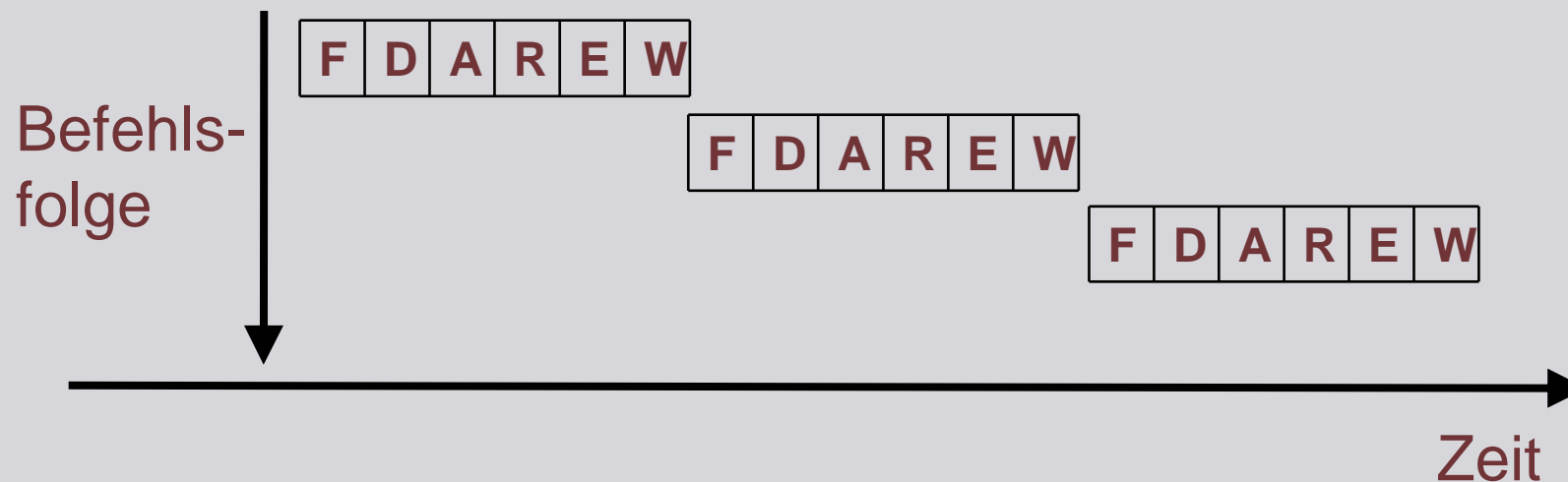
Rechnerarchitektur

Reale Rechnerarchitekturen

- ohne Pipelining

Befehlszyklus:

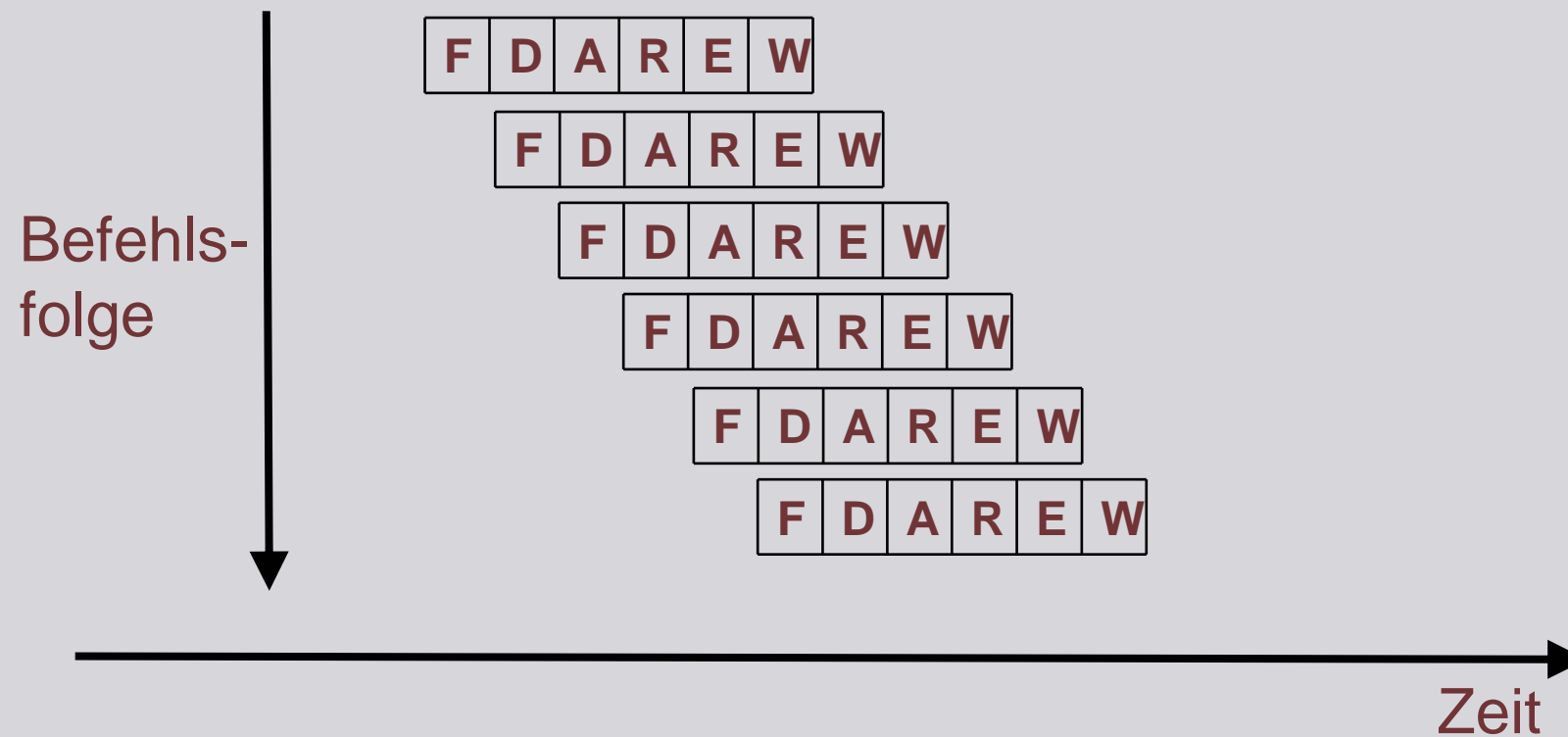
fetch → decode → address → read → excute → write



Rechnerarchitektur

Reale Rechnerarchitekturen

- mit Pipelining



Rechnerarchitektur



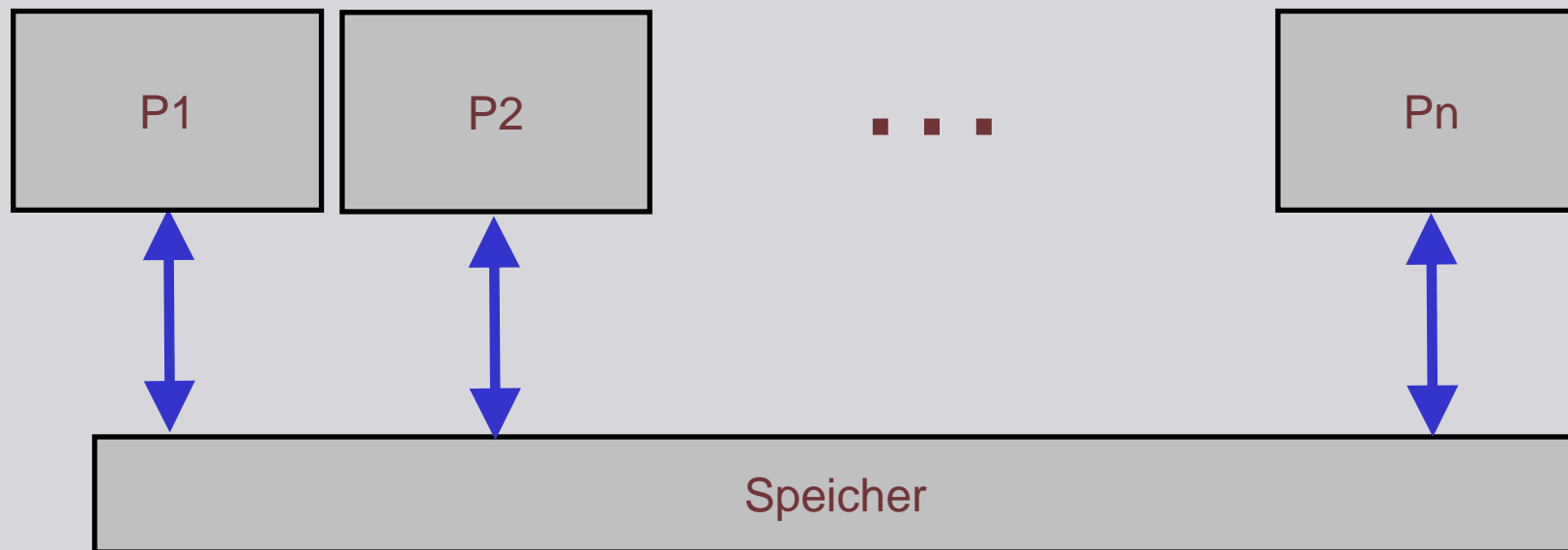
Reale Rechnerarchitekturen

- Funktionale Parallelität
 - gleichzeitiges Nutzen „disjunkter“ Teile des Rechenwerks (Ganzzahlarithmetik / Gleitkommarithmetik)
- Speicherhierarchien
 - sehr schnelle Cache Speicher
 - „langsamerer“ Hauptspeicher
 - langsamer Plattenspeicher

Rechnerarchitektur

Reale Rechnerarchitekturen

- Parallele Prozessoren mit gemeinsamem Speicher (**shared memory**)



Pi Prozessor ohne Speicher

Rechnerarchitektur

Reale Rechnerarchitekturen

- Parallele Prozessoren mit verteiltem Speicher (**distributed memory**)



PMi Prozessor mit Speicher

Zum Schluss dieses Abschnitts ...

Noch Fragen ??