

# Kapitel 11

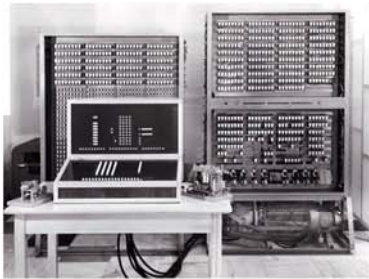
# Parallele Rechnerstrukturen

Parallele Prozessorarchitekturen  
Parallelrechner

# 11.1 Motivation

## ■ Entwicklung im Bereich der Höchstleistungsrechner

$$\text{MFlops} = \frac{\text{Anzahl der ausgeführten Gleitkommainstruktionen}}{10^6 \times \text{Ausführungszeit}}$$



**Zuse Z1 1 Flops ( $10^0$ )  
1941**



**GigaFlops ( $10^9$ )  
1985**

**Intel ASCI Red**



**TeraFlops ( $10^{12}$ )  
1997**

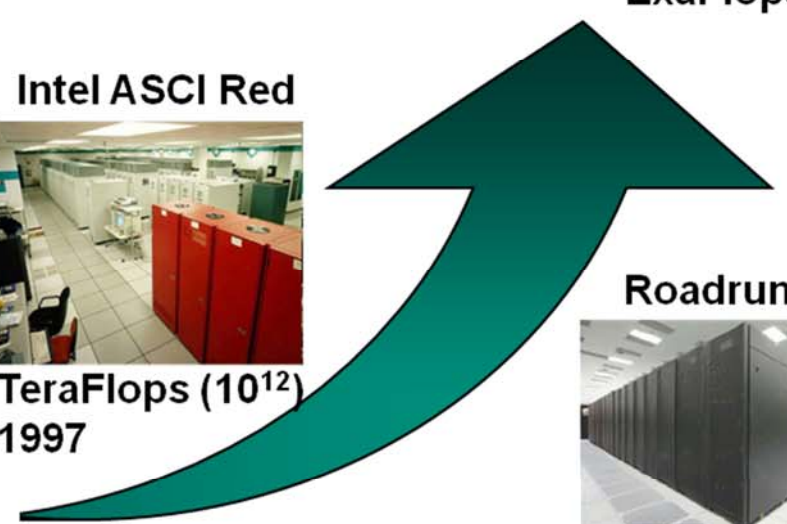
?

**ExaFlops ( $10^{18}$ )**

**Roadrunner**

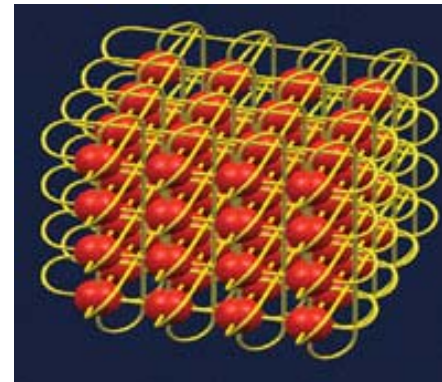
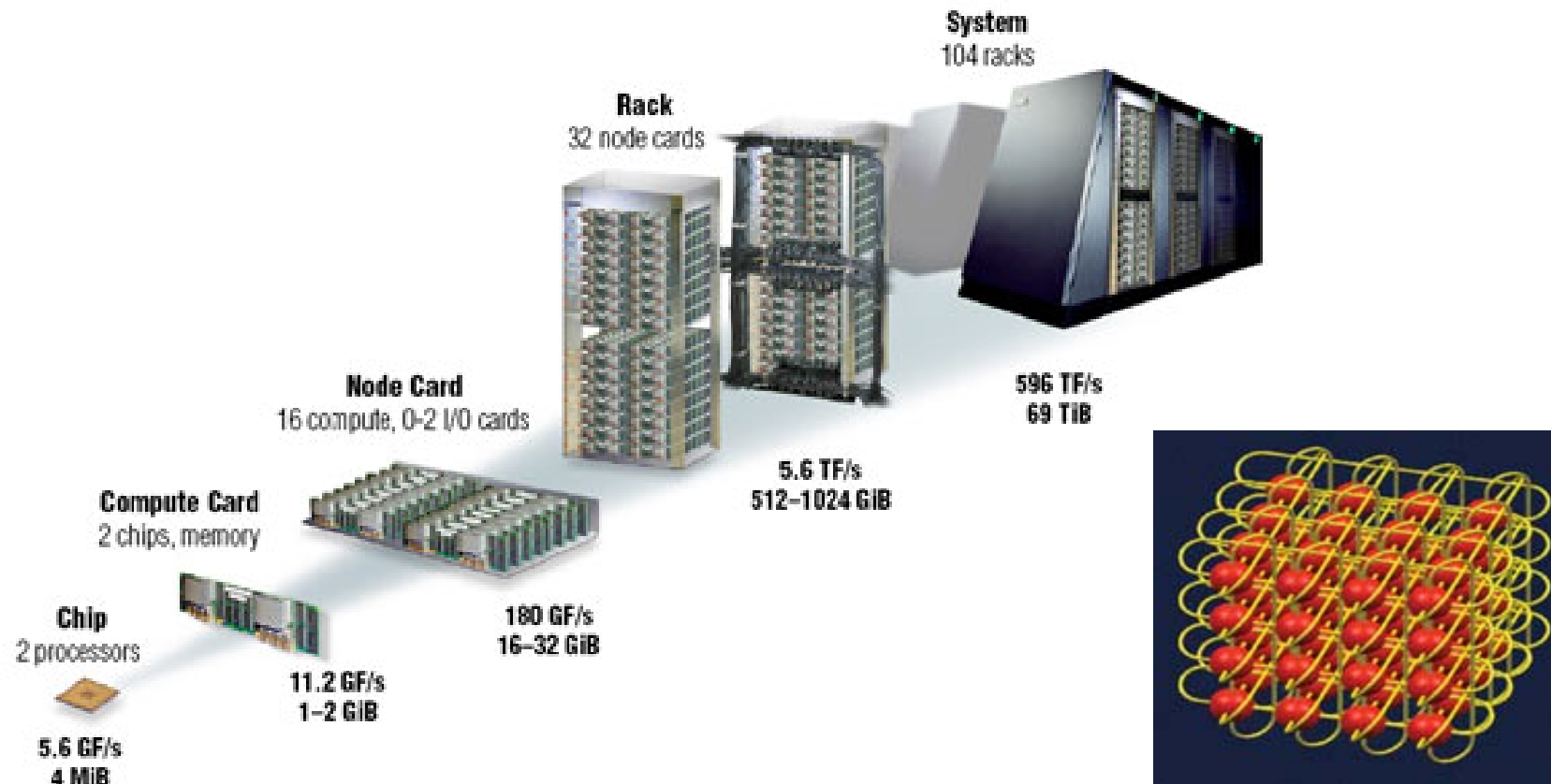


**PetaFlops ( $10^{15}$ )  
2008**



# 11.1 Motivation

## Multiprozessorsysteme:

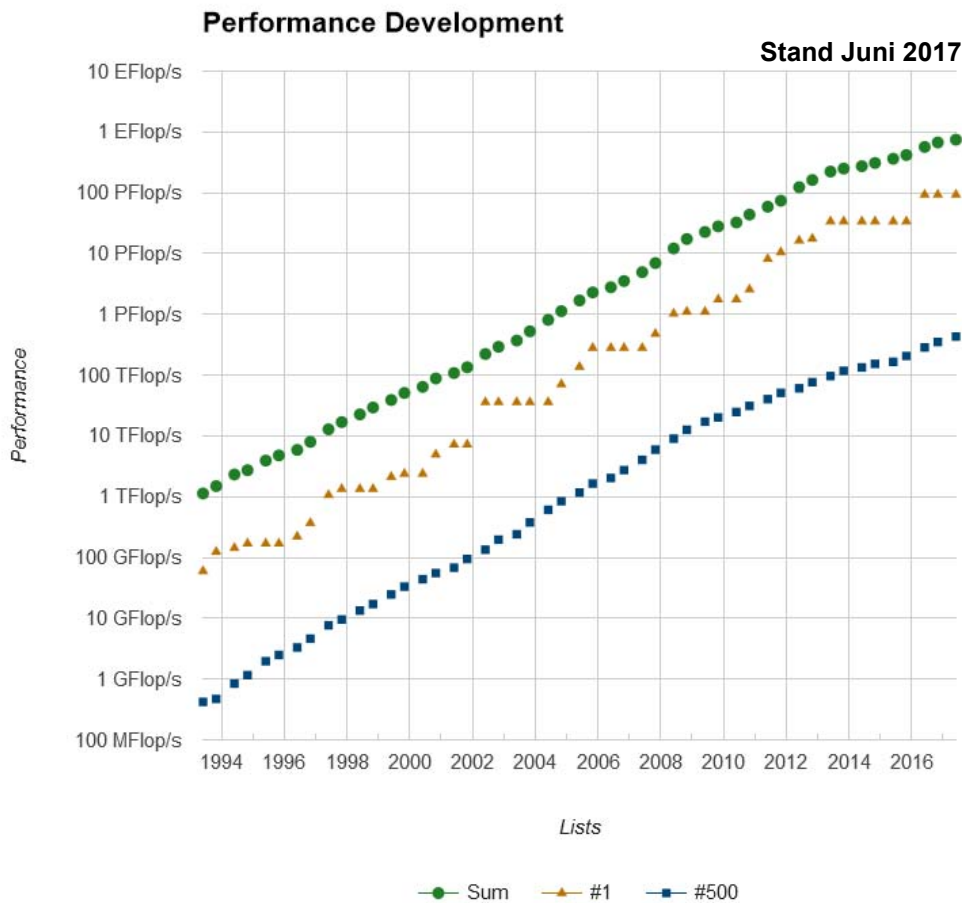


BlueGene/L: Interconnection Network

Quelle: [https://asc.llnl.gov/computing\\_resources/bluegenel/photogallery.html](https://asc.llnl.gov/computing_resources/bluegenel/photogallery.html)

# Entwicklung der Höchstleistungsrechner

## Top500 Liste



Vergleich mit Hilfe des  
LINPACK Benchmarks

Nr. 1 im November 2017:  
**Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCP**  
**National Supercomputing Center in Wuxi, China**  
**Daten:**  
**Anzahl Cores: 10,649,600**  
**Linpack Performance: 93,014.6 TFlops**  
**Power: 15,371.00 KW**

Quelle: <http://www.top500.org/statistics/perfdevel/>

# 11.1 Motivation

## Top500 Liste

- Top500 Liste Nr. 19, November 2017;
  - Hazel Hen - Cray XC40, Xeon E5-2680v3 12C 2.5GHz, Aries interconnect, Cray Inc.
  - HLRZ Stuttgart
- Daten:
  - Anzahl Cores: 185088
  - Linpack performance: 5640,2 TFLOPS
  - Power: 3615 kW



Quelle: <http://www.hlrz.de/systems/cray-xc40-hazel-hen/>

# 11.1 Motivation

## Top500 Liste

- Top500 Liste Nr. 16, November 2017:
  - MareNostrum - Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path
  - Barcelona Supercomputing Center, Spanien
  
- Daten:
  - Anzahl Cores: 153216
  - Linpack performance: 6470,8 TFLOPS
  - Power: 1,632 MW



By courtesy of Barcelona Supercomputing Center - [www.bsc.es](http://www.bsc.es)

# 11.1 Motivation

## Top500 Liste

- Top500 Liste Nr. 16, November 2017:
  - MareNostrum - Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path
  - Barcelona Supercomputing Center, Spanien
- **Wo ist der Rechner aufgestellt?**

# 11.1 Motivation

## Top500 Liste

- Top500 Liste Nr. 16, November 2017:
  - MareNostrum - Lenovo SD530, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path
  - Barcelona Supercomputing Center, Spanien
- **Wo ist der Rechner aufgestellt?**
- **HIER:**



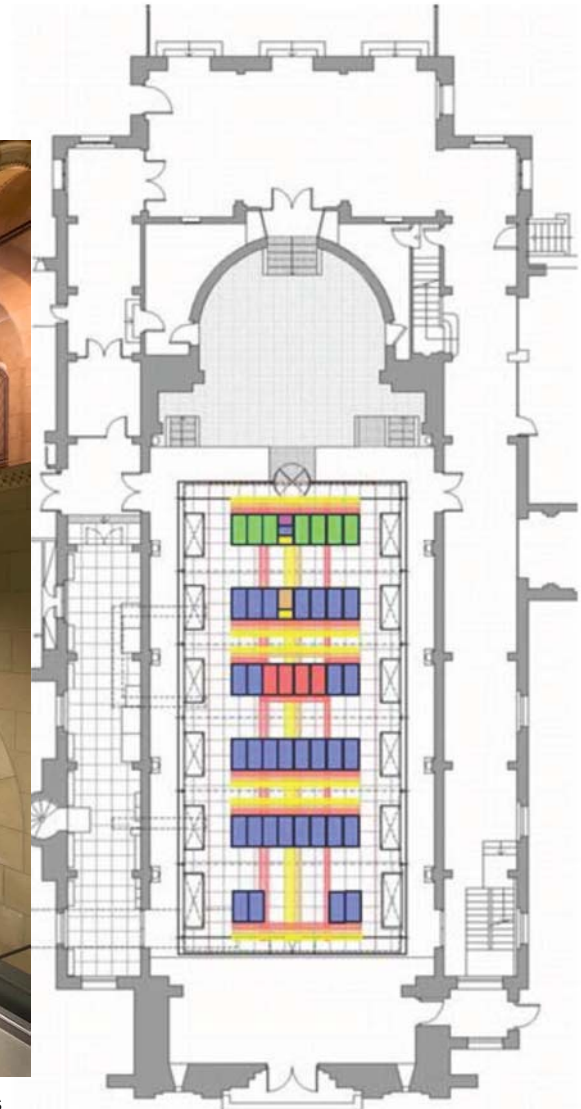
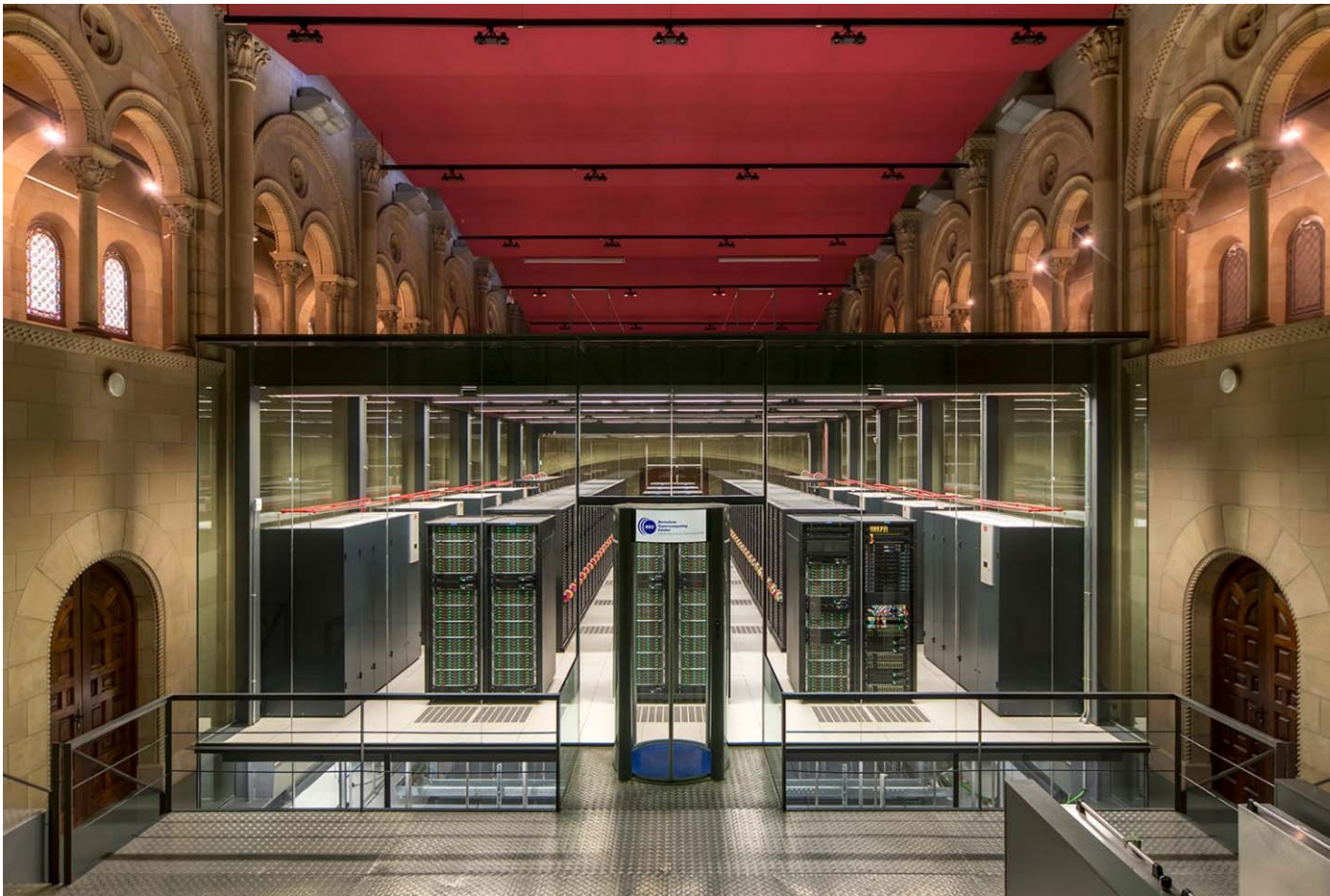
By courtesy of Barcelona Supercomputing Center - [www.bsc.es](http://www.bsc.es)



# 11.1 Motivation

## Top500 Liste

- MareNostrum



By courtesy of Barcelona Supercomputing Center - [www.bsc.es](http://www.bsc.es)

# 11.1 Motivation

## ■ Parallelverarbeitung

### Parallelarbeitstechniken

	Programmebene	Prozessebene	Blockebene	Anweisungsebene	Suboperationsebene
<b>Techniken der Parallelarbeit durch Rechnerkopplung</b>					
Cluster	X	X			
<b>Techniken der Parallelarbeit durch Prozessorkopplung</b>					
Nachrichtenkopplung	X	X			
Speicherkopplung	X	X	X		
<b>Techniken der Parallelarbeit in der Prozessorarchitektur</b>					
Befehlspipelining				X	
Superskalar				X	
				X	
<b>SIMD-Techniken</b>					
Vektorrechnerprinzip					X
Feldrechnerprinzip					X
SIMD-Operationen					

Quelle: Ungerer, T. :  
 Skript Rechnerstrukturen,  
 SS 2000

# 11.2 Parallelverarbeitung

## Formen des Parallelismus

### ■ Nebenläufigkeit

- Eine Maschine arbeitet nebenläufig, wenn die Objekte vollständig gleichzeitig abgearbeitet werden.

### ■ Pipelining

- Pipelining auf einer Maschine liegt dann vor, wenn die Bearbeitung eines Objektes in Teilschritte zerlegt und diese in einer sequentiellen Folge (Phasen der Pipeline) ausgeführt werden. Die Phasen der Pipeline können für verschiedene Objekte überlappt abgearbeitet werden. (Bode 95)

# 11.2 Parallelverarbeitung

## Ebenen der Parallelität

### ■ Programmebene

- Parallele Verarbeitung verschiedener Programme
- Vollständig unabhängige Einheiten
  - ohne gemeinsame Daten
  - wenig oder keine Kommunikation und Synchronisation
- Parallelverarbeitung wird vom Betriebssystem organisiert

# 11.2 Parallelverarbeitung

## Ebenen der Parallelität

### ■ Prozessebene (Task-Ebene)

- Programm wird in Anzahl parallel ausführbarer Prozesse zerlegt
- **Prozess**: schwergewichtiger Prozess (heavy-weighted process), Beispiel: UNIX-Prozesse
  - Besteht aus vielen sequentiell ausgeführten Befehlen und umfasst eigene Datenbereiche
- **Synchronisation** und **Kommunikation**
- Betriebssystem unterstützt Parallelverarbeitung durch Primitive zur Prozessverwaltung, Prozess-Synchronisation, Prozesskommunikation

# 11.2 Parallelverarbeitung

## Ebenen der Parallelität

### ■ Blockebene

#### ■ leichtgewichtige Prozesse (Threads)

- Bestehen jeweils aus sequentiell ausgeführten Befehlen teilen sich gemeinsamen Adressraum
- Beispiel: Threads gemäß POSIX 1003.a Standard in mehrfädigen (multithreaded) Betriebssystemen
- Synchronisation über Schlossvariablen (mutex), und Bedingungsvariablen (condition variables) oder darauf aufbauenden Synchronisationsmechanismen
- Kommunikation über gemeinsame Daten
- Aufwand für Thread-Erzeugung und-Beendigung, Thread-Wechsel geringer

#### ■ Anweisungsblöcke

- Innere und äußere parallele Schleifen in FORTRAN-Dialekten
- Verwendung von Microtasking
- Hohes Parallelitätspotential durch parallel ausführbare Schleifeniterationen

# 11.2 Parallelverarbeitung

## ■ Ebenen der Parallelität

### ■ Anweisungs- oder Befehlsebene

- Parallele Ausführung einzelner Maschinenbefehle oder elementarer Anweisungen
- Optimierende (parallelisierende) Compiler für VLIW-Prozessoren oder Anwendung der Superskalartechnik in superskalaren Mikroprozessoren
- Analyse der sequentiellen Befehlsfolge
- Umordnen und Parallelisieren der Befehle
- Datenflusssprachen und funktionale Programmiersprachen erlauben explizite Spezifikation der Parallelität

# 11.2 Parallelverarbeitung

## ■ Ebenen der Parallelität

### ■ Suboperationsebene

- Elementare Anweisung wird durch Compiler oder durch die Maschine in Suboperationen aufgebrochen, die parallel ausgeführt werden

### ■ Vektoroperationen

- Überlappte Ausführung auf Vektorrechner in Vektorpipeline
- Komplexe Datenstrukturen (Matrizen, Vektoren, Datenströme) sind in höherer Programmiersprache verfügbar oder werden von einem parallelisierenden, vektorisierenden Compiler aus einer sequentiellen Programmiersprache generiert
- Beispiel: Vektoraddition  $C = A + B$  statt Abarbeitung einer Schleife



# 11. 2 Parallelverarbeitung

## Ebenen der Parallelität

### ■ Körnigkeit der Parallelität

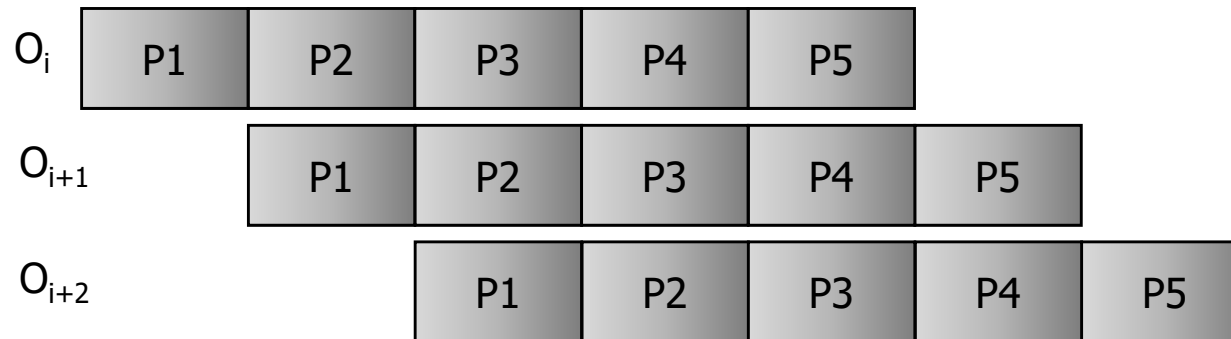
- Die **Körnigkeit** oder **Granularität (grain size)** ergibt sich aus dem Verhältnis von Rechenaufwand zu Kommunikations- oder Synchronisationsaufwand. Sie bemisst sich nach der Anzahl der Befehle in einer sequentiellen Befehlsfolge.
- Programm-, Prozess- und Blockebene werden häufig auch als **grobkörnige (large grained) Parallelität**,
- die Anweisungsebene **als feinkörnige (finely grained) Parallelität** bezeichnet.
- Seltener wird auch von **mittelkörniger (medium grained) Parallelität** gesprochen, dann ist meist die Blockebene gemeint.

# 11.3 Parallelismus auf Befehlsebene

## Pipelining

### ■ Definition

- *Pipelining auf einer Maschine liegt dann vor, wenn die Bearbeitung eines Objektes in Teilschritte zerlegt und diese in einer sequentiellen Folge (Phasen der Pipeline) ausgeführt werden. Die Phasen der Pipeline können für verschiedene Objekte überlappt abgearbeitet werden. (Bode 95)*



# 11.3 Parallelismus auf Befehlsebene

## Pipelining

### ■ Befehlspipelining (Instruction Pipelining):

- Zerlegung der Ausführung einer Maschinenoperation in Teilphasen, die dann von hintereinander geschalteten Verarbeitungseinheiten taktsynchron bearbeitet werden, wobei jede Einheit genau eine spezielle Teiloperation ausführt.

### ■ Pipeline:

- Gesamtheit der Verarbeitungseinheiten

### ■ Pipeline-Stufe:

- Stufen der Pipeline, die jeweils durch Pipeline-Register getrennt sind

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Einfache Maschinenbefehle

- Einheitliches und festes Befehlsformat

### ■ Load/Store Architektur

- Befehle arbeiten auf Registeroperanden
- Lade- und Speicherbefehle greifen auf Speicher zu

### ■ Einzyklus-Maschinenbefehle

- Effizientes Pipelining des Maschinenbefehlszyklus
- Einheitliches Zeitverhalten der Maschinenbefehle, wovon nur Lade- und Speicherbefehle sowie die Verzweigungsbefehle abweichen

### ■ Optimierende Compiler

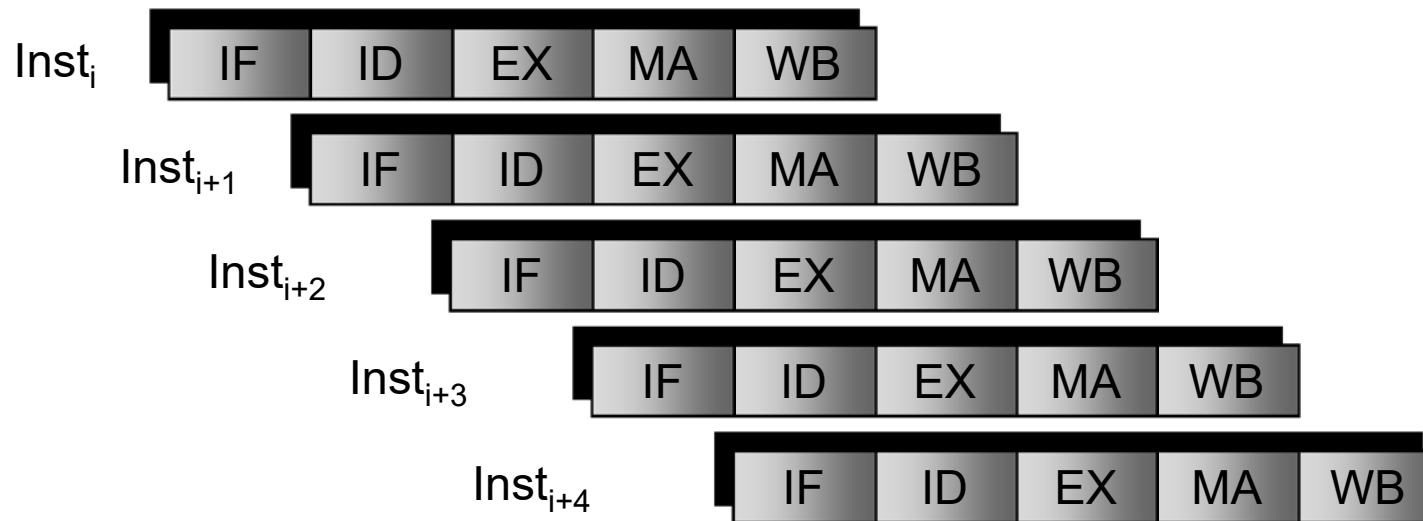
- Reduzierung der Befehle im Programm

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Pipelining des Maschinenbefehlszyklus

- k-stufige Befehlspipeline (k=5)



#### Logische Phasen:

IF: Befehl holen  
 ID: Befehl dekodieren  
 EX: Befehl ausführen  
 MA: Speicherzugriff  
 WB: Zurückschreiben

Pipeline-  
 Stufe | 1 Takt-  
 | zyklus |

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Diskussion

#### ■ Verfeinerung der Pipeline-Stufen

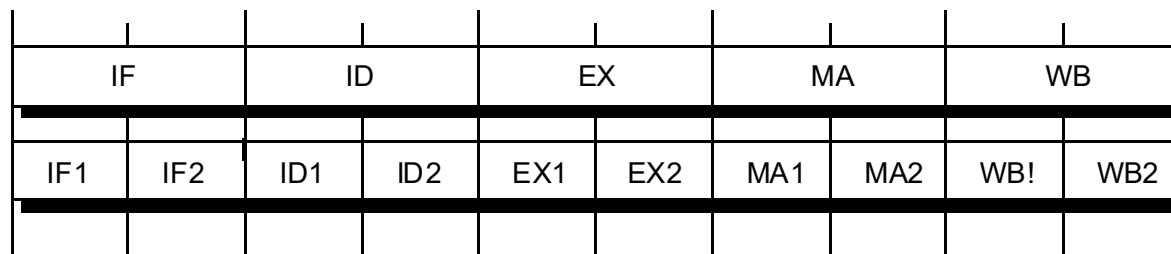
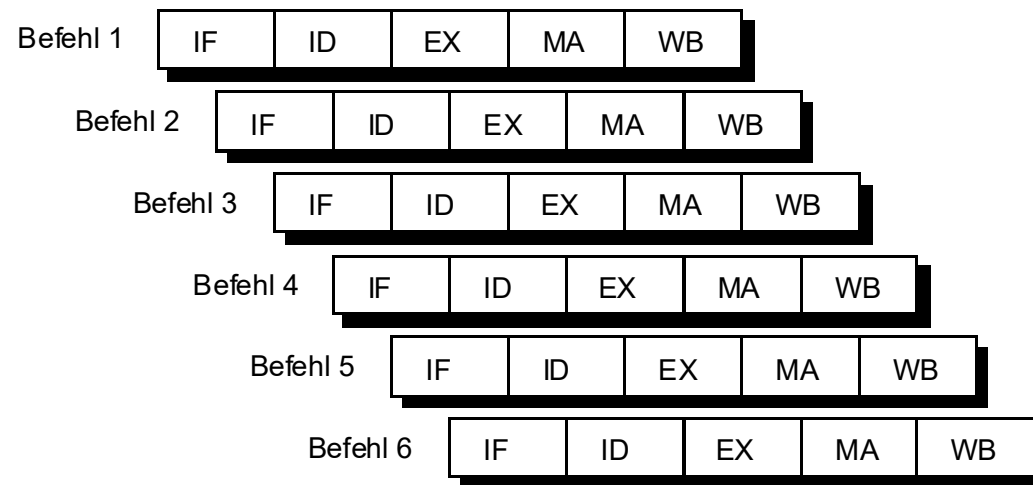
- Weitere Unterteilung der Pipeline-Stufen
- Weniger Logik-Ebenen pro Pipeline-Stufe
- Erhöhung der Taktrate
- Führt aber auch zu einer Erhöhung der Ausführungszeit pro Instruktion

#### ■ „Superpipelining“

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Verfeinerung der Pipeline-Stufen

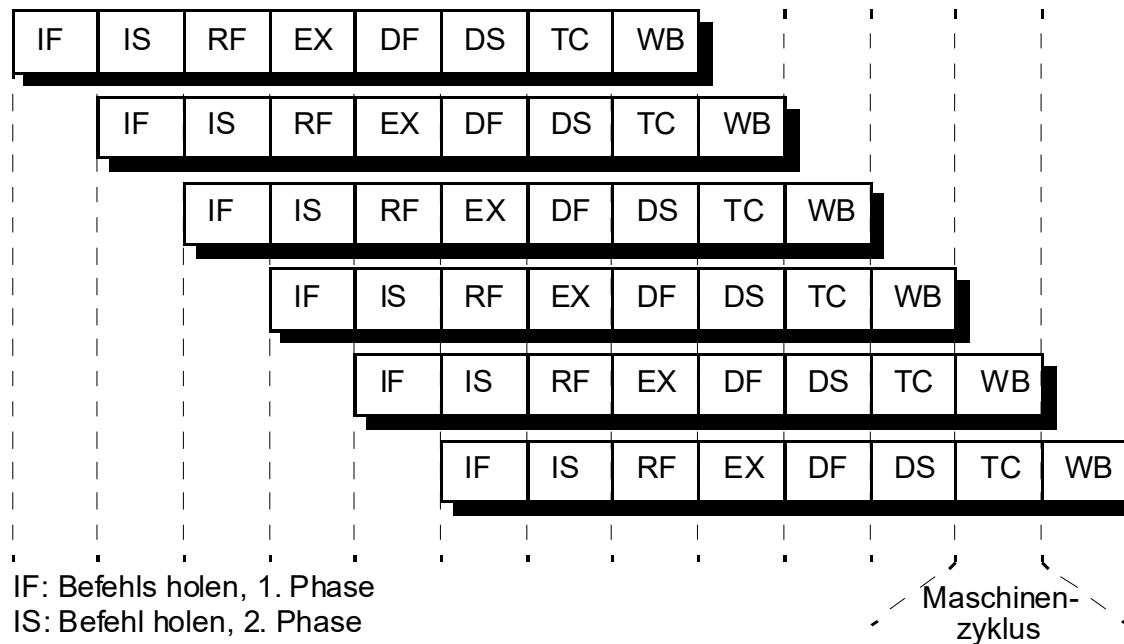


Maschinenzyklus  
 = 1/2 Grundtakt  
 der RISC-Maschine

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Verfeinerung der Pipeline-Stufen: Beispiel MIPS R4000 (~1991)





# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Pipeline-Konflikte (Pipeline Hazards, Pipeline-Hemmnisse)

- Situationen, die verhindern, dass die nächste Instruktion im Befehlsstrom im zugewiesenen Taktzyklus ausgeführt wird
  - Unterbrechung des taktsynchronen Durchlaufs durch die einzelnen Stufen der Pipeline
- Verursachen Leistungseinbußen im Vergleich zum idealen Speedup
- Einfaches Verfahren zur Auflösung von Konflikten:
  - Anhalten der Pipeline (Pipeline stall)
- Bei einfacher Pipeline:
  - Wenn eine Instruktion angehalten wird, werden auch alle Befehle, die nach dieser Instruktion zur Ausführung angestoßen wurden, angehalten
  - Alle Befehle, die vor dieser Instruktion zur Ausführung angestoßen wurden, durchlaufen weiter die Pipeline

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Pipeline-Konflikte (Pipeline Hazards, Pipeline-Hemmnisse)

#### ■ Strukturkonflikte

- Ergeben sich aus **Ressourcenkonflikten**
- Die Hardware kann nicht alle möglichen Kombinationen von Befehlen unterstützen, die sich in der Pipeline befinden können
- Beispiel:
  - Gleichzeitiger Schreibzugriff zweier Befehle auf eine Registerdatei mit nur einem Schreibeingang

#### ■ Datenkonflikte

- Ergeben sich aus **Datenabhängigkeiten** zwischen Befehlen im Programm
- Instruktion benötigt das Ergebnis einer vorangehenden und noch nicht abgeschlossenen Instruktion in der Pipeline
  - D.h. ein Operand ist noch nicht verfügbar

#### ■ Steuerkonflikte

- Treten bei **Verzweigungsbefehlen** und anderen **Instruktionen, die den Befehlszähler verändern**, auf

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Pipeline-Konflikte (Pipeline Hazards, Pipeline-Hemmnisse)

#### ■ Auflösung der Pipeline-Konflikte

- Einfache Lösung: Anhalten der Pipeline (**Pipeline stall**)
- Einfügen eines Leerzyklus (**Pipeline Bubble**)
  
- Führt zu Leistungseinbußen
  - Verschiedene Maßnahmen in der Hardware und in der Software, um Auswirkungen auf die Leistungsfähigkeit möglichst zu vermeiden

# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Diskussion

#### ■ Ausführungsphase

##### ■ Integer-Verarbeitung

- Ausführung von arithmetischen und logischen Befehlen dauert einen Taktzyklus (Ausnahme: Division)

##### ■ Gleitkomma-Verarbeitung:

- Zerlegung in weitere Stufen
- Eingliederung an der Stelle der Ausführungsstufe in der Befehlspipeline
- Mehrere Gleitkommarechenwerke (Floating-Point Units)

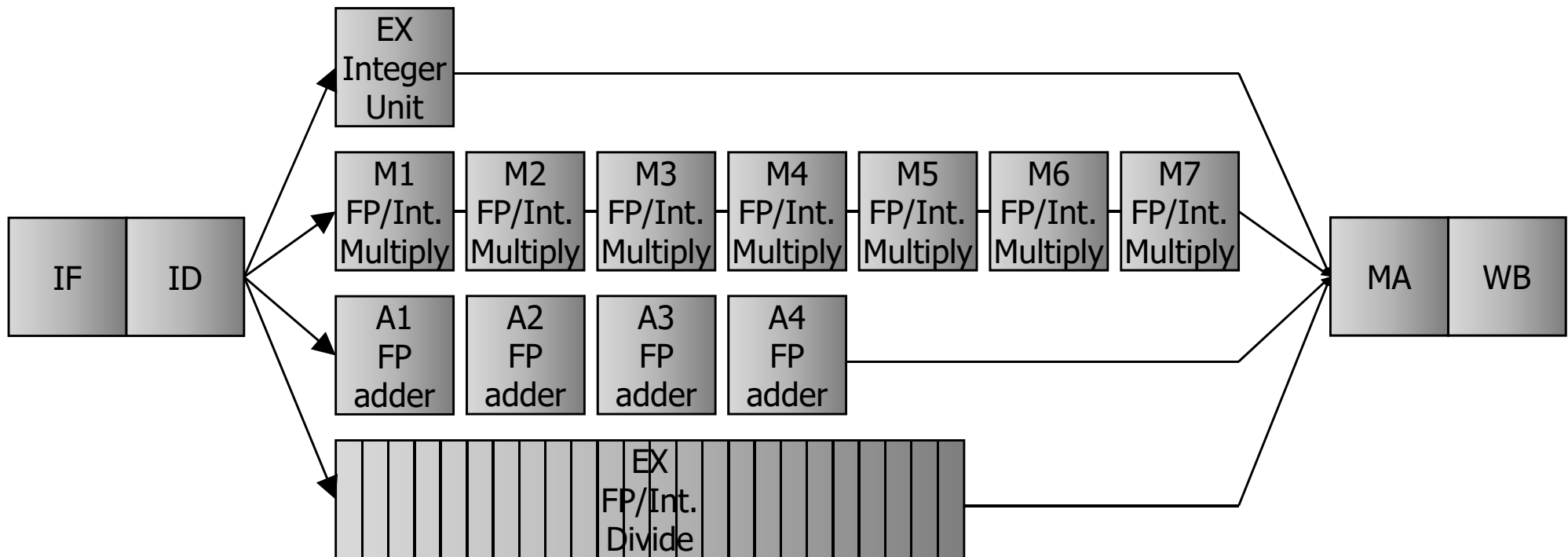
# 11.3 Parallelismus auf Befehlsebene

## RISC (Reduced Instruction Set Computers)

### ■ Diskussion

#### ■ Ausführungsphase

#### ■ Gleitkomma-Verarbeitung: weitere Rechenwerke



# 11.3 Parallelismus auf Befehlsebene

## Nebenläufigkeit

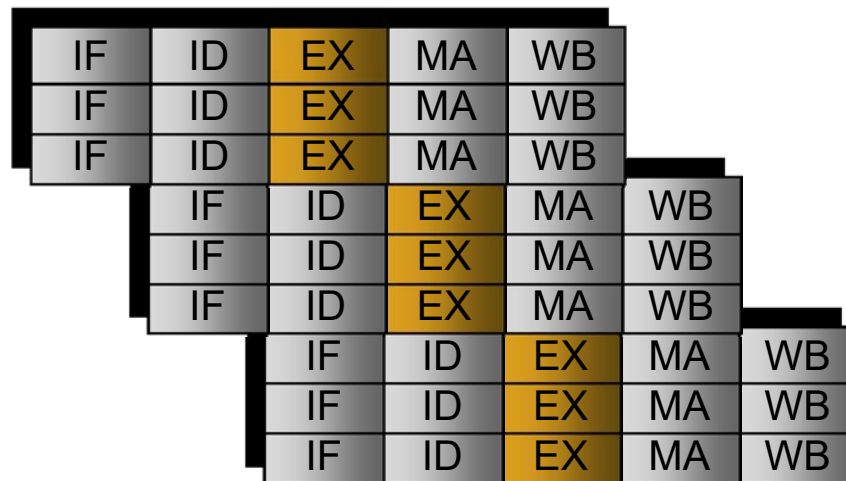
- Superskalartechnik

# 11.3 Parallelismus auf Befehlsebene

## Nebenläufigkeit

### ■ Superskalartechnik

- Anstoßen (Issue) von n Befehlen pro Zyklus
- Max. IPC = n Befehle pro Zyklus



# 11.3 Parallelismus auf Befehlsebene

## Nebenläufigkeit

### ■ RISC → Superskalar

#### ■ Mehrfachzuweisungsmethoden (multiple issue)

- Die Superskalar-Technik ermöglicht es, pro Takt mehrere Befehle den Ausführungseinheiten zuzuordnen und eine gleiche Anzahl von Befehlsausführungen pro Takt zu beenden.

#### ■ Superskalare RISC-Prozessoren:

- RISC-Charakteristika werden auch heute noch weitgehend beibehalten
  - Lade-/Speicher-Architektur
  - Festes Befehlsformat (z. B.: Befehlslänge: 32 Bit)
- Entwurfsziel: Erhöhung des IPC (Instruction per Cycle)
- Heutige Mikroprozessoren nutzen Parallelismus auf Maschinenbefehlsebene durch die Pipelining- und Superskalartechnik



# 11.3 Parallelismus auf Befehlsebene

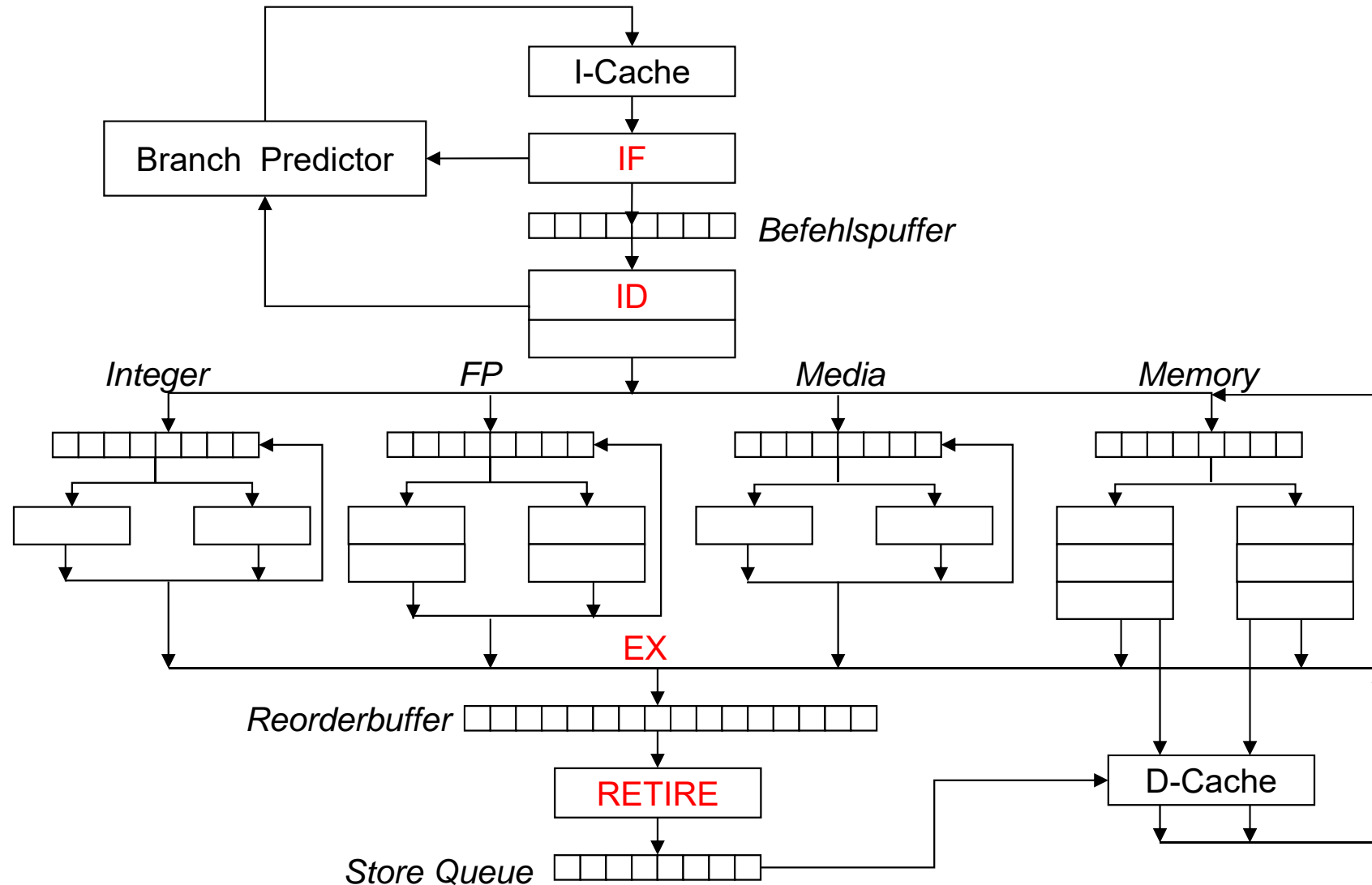
## Nebenläufigkeit

### ■ Superskalarer Prozessor

- Nützt den Parallelismus auf Befehlsebene aus
  - **Vielstufige Befehlspipeline**
  - **Superskalartechnik**
- **Eigenschaften:**
  - Mehrere voneinander unabhängige Ausführungseinheiten
  - Zur Laufzeit werden pro Takt mehrere Befehle aus einem sequentiellen Befehlsstrom den Verarbeitungseinheiten zugeordnet und ausgeführt
  - **Dynamische Erkennung und Auflösung von Konflikten** zwischen Befehlen im Befehlsstrom ist Aufgabe der Hardware

# 11.3 Parallelismus auf Befehlsebene

## Superskalarer Prozessor



# 11.3 Parallelismus auf Befehlsebene

## Superskalarer Prozessor

### ■ Komponenten

- **Befehlsholeinheit** (Instruction Fetch)
- **Dekodiereinheit** (Instruction Decode) mit **Registerumbenennung** (register renaming)
- **Zuordnungseinheit** (Instruction Issue)
- Unabhängige **Verarbeitungseinheiten** (Functional Units)
- **Rückordnungseinheit** (Retire Unit)
- **Register:**
  - Allzweckregister
  - Multimediaregister
  - Spezialregister
- *Anmerkung: Die Bezeichnungen der Einheiten sind bei den verschiedenen Prozessoren nicht einheitlich!*