



# Technische Informatik 1

**Prof. Dr. Rolf Drechsler**  
**Christina Plump**

# Überblick

## Teil 1: Der Rechneraufbau (Kapitel 2-5)

- Rechner im Überblick
- Pipelining
- Speicher
- **Parallelverarbeitung**
  - **Parallele Rechnerarchitekturen**
  - **Grafikkarten**
  - **Verbindungsstrukturen**

## Teil 2: Der Funktionalitätsaufbau (Kapitel 6-12)

- Kodierung von Zeichen und Zahlen
- Grundbegriffe, Boolesche Funktionen
- Darstellungsmöglichkeiten
- Schaltkreise, Synthese, spezielle Schaltkreise



# Kapitel 5: Parallelverarbeitung

Parallele Rechnerarchitekturen

Exkurs: Grafikkarten

Exkurs: Verbindungsstrukturen

## Lernziele

- Unterschiedliche Formen der Parallelität kennenlernen
- Unterschiedliche Formen der Parallelverarbeitung kennenlernen und anhand unterschiedlicher Klassifikationssysteme unterscheiden können
- Klassifikationsprinzip nach Flynn kennen und anwenden können

# Formen der Parallelität

- Parallelität auf Bitebene: bis etwa 1985
  - Kombinatorische Addierer und Multiplizierer, etc.
  - *wachsende Wortbreite auf 64 Bit*
- Parallelität auf Instruktionsebene: 1985 bis heute
  - Pipelining der Instruktionsverarbeitung
  - Mehrere Funktionseinheiten (*superskalare Prozessoren*)  
bei mehr als vier Funktionseinheiten werden Datenabhängigkeiten oft zum Hindernis für eine effiziente Ausnutzung
  - Vektorprozessoren führen eine Operation parallel auf vielen Daten durch (Bsp.: Cray 1 [1974], C3880 [1992])
- Parallelität auf Prozessor-/Rechnerebene
  - nur so scheinen Beschleunigungen um Faktor 50 und mehr möglich

# Der neue Supercomputer: Frontier

 heise online

Trillionen Gleitkommaoperationen  
in der Sekunde

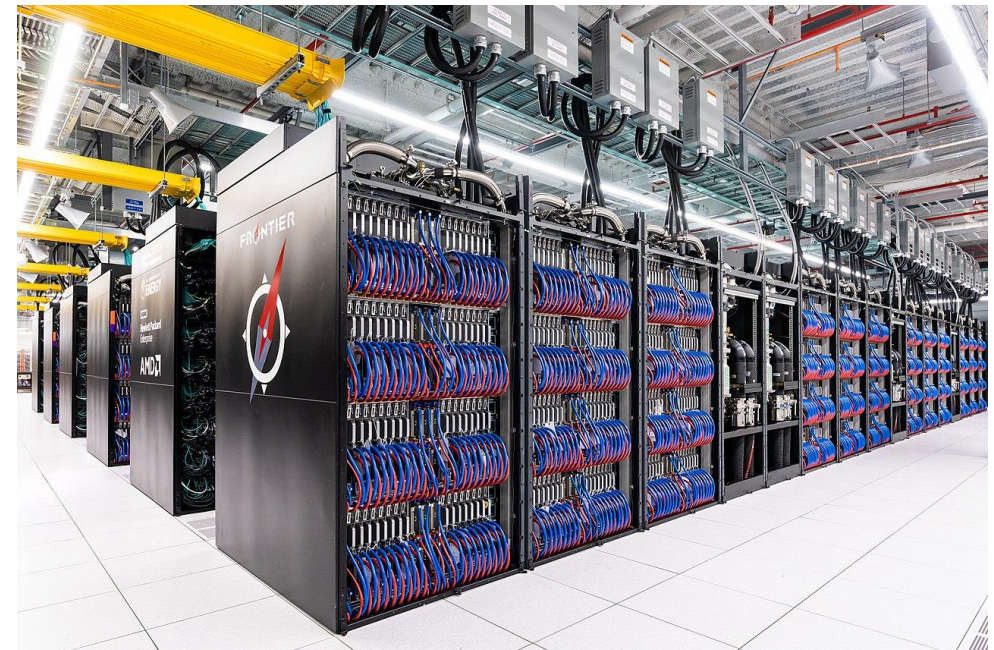
Der erste westliche **Exaflops-Supercomputer: Frontier** mit CPUs und GPUs von AMD

30.05.2022 12:47 Uhr Mark Mantel

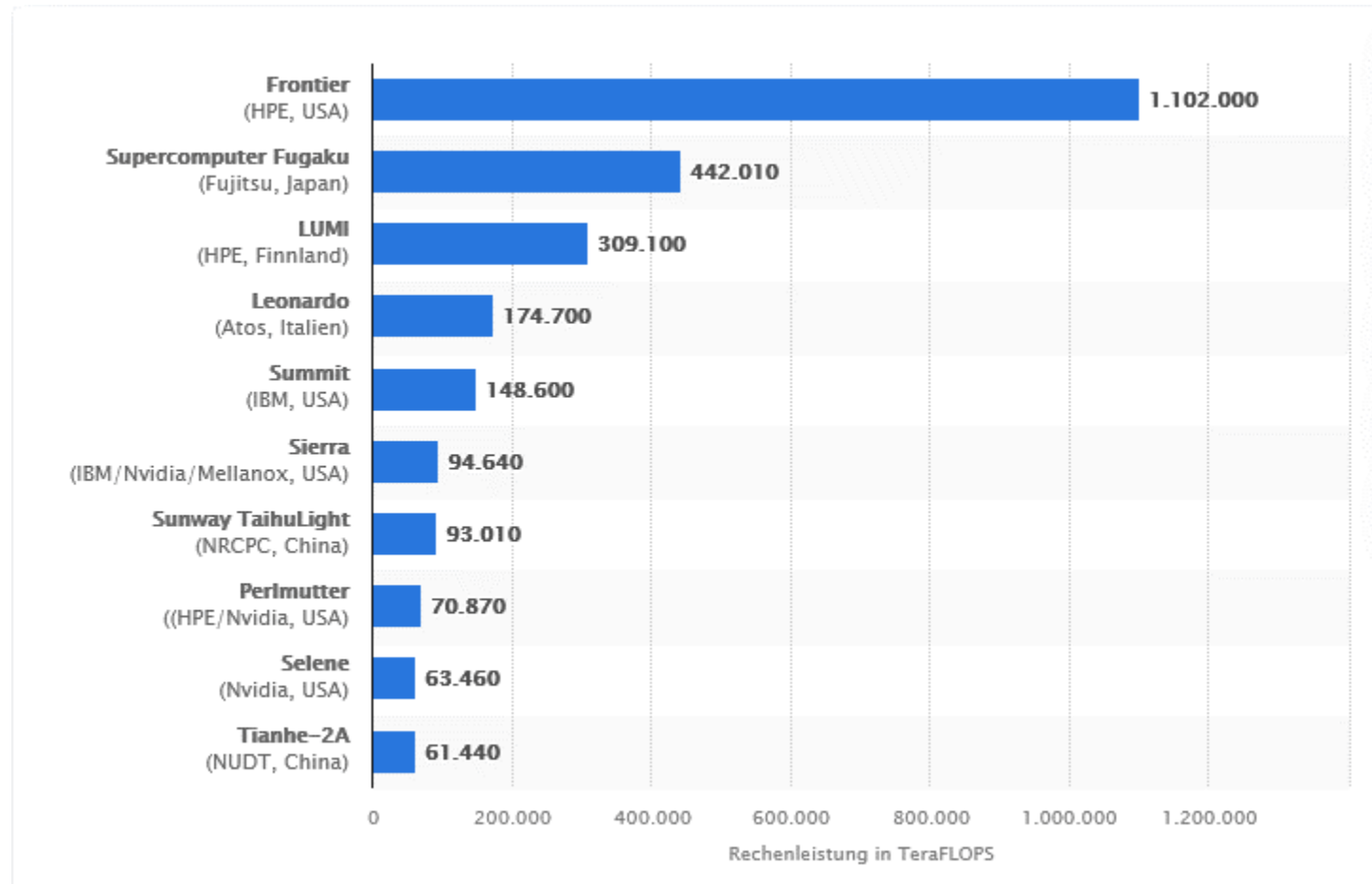
Die USA sind das zweite Land mit einem Supercomputer der Exascale-Leistungsklasse. Mehr als 1,1 Trillionen Berechnungen pro Sekunde schafft Frontier.

Frontier ist doch früher nutzbar als zuletzt in Aussicht gestellt – fast das komplette System ist inzwischen operabel. Somit schafft es Frontier auf eine Rechenleistung von 1,1 Exaflops im Linpack-Benchmark mit doppelter (FP64-)Genauigkeit, in der Spitze sind sogar fast 1,7 Exaflops drin und bei KI-Berechnung dank optimierter Matrizenrechner beinahe 6,9 Exaflops. Ein Exaflops entspricht einer Trillion Berechnungen pro Sekunde oder anders ausgedrückt: eine 1 mit 18 Nullen.

[www.top500.org](https://www.top500.org)



## Supercomputer in aller Welt (Stand 2022)



# Formen der Parallelverarbeitung auf Prozessor-/Rechnerebene (1)

- Arrayprozessoren
  - viele gleichartige Prozessoren, evtl. mit lokalem Speicher
  - eine Steuereinheit für alle Prozessoren
  - gleiche Instruktionsfolge auf allen Prozessoren
  - Beispiele:
    - Illiac IV (1972, 64 Knoten)
    - CM-2 (1987, 65.536 Knoten)

Nächster Schritt: Mache Prozessoren in der Verarbeitung unabhängiger



## Formen der Parallelverarbeitung auf Prozessor-/Rechnerebene (2)

- Multiprozessoren
  - bestehen aus mehreren Prozessoren und haben einen gemeinsamen Speicher ([shared memory](#))
  - Kommunikation erfolgt über den gemeinsamen Speicher
  - Problem: evtl. Performanz-Probleme, wenn viele Prozessoren gleichzeitig auf den Speicher zugreifen wollen

## Formen der Parallelverarbeitung auf Prozessor-/Rechnerebene (3)

- Multi-Core (Mehrkernprozessor)
  - Multiprozessor auf einem Chip
  - evtl. gemeinsamer Cache
  - evtl. direkte Kommunikation
- Koprozessoren
  - entlasten den Hauptprozessor
  - spezialisiert auf bestimmte Aufgaben
  - **Grafikprozessoren (GPUs)** für 2D- und 3D-Grafikberechnungen
    - große Ähnlichkeit zu Vektorprozessoren
  - **mathematische Koprozessoren** z.B. für Berechnungen mit Gleitkommazahlen

## Formen der Parallelverarbeitung auf Prozessor-/Rechnerebene (4)

- Multicomputer
  - bestehen aus mehreren Rechnern ohne gemeinsamen Speicher
  - jeder Prozessor mit eigenem Speicher (**local memory**)
  - Prozessoren tauschen Daten durch Nachrichten aus
  - Problem: evtl. hoher Aufwand für Nachrichtenaustausch

# Klassifikationsmerkmale bei parallelen Rechnerarchitekturen

1

## Speichermodell

- nicht-verteilter Speicher
- verteilter Speicher, gemeinsamer Adressraum
- verteilter Speicher, getrennter Adressraum

2

## Homogenität von Prozessoren

- homogene Prozessoren (alle gleich)
- heterogene Prozessoren (Hardware Unterscheidung möglich)

3

## Hierarchie von Prozessoren

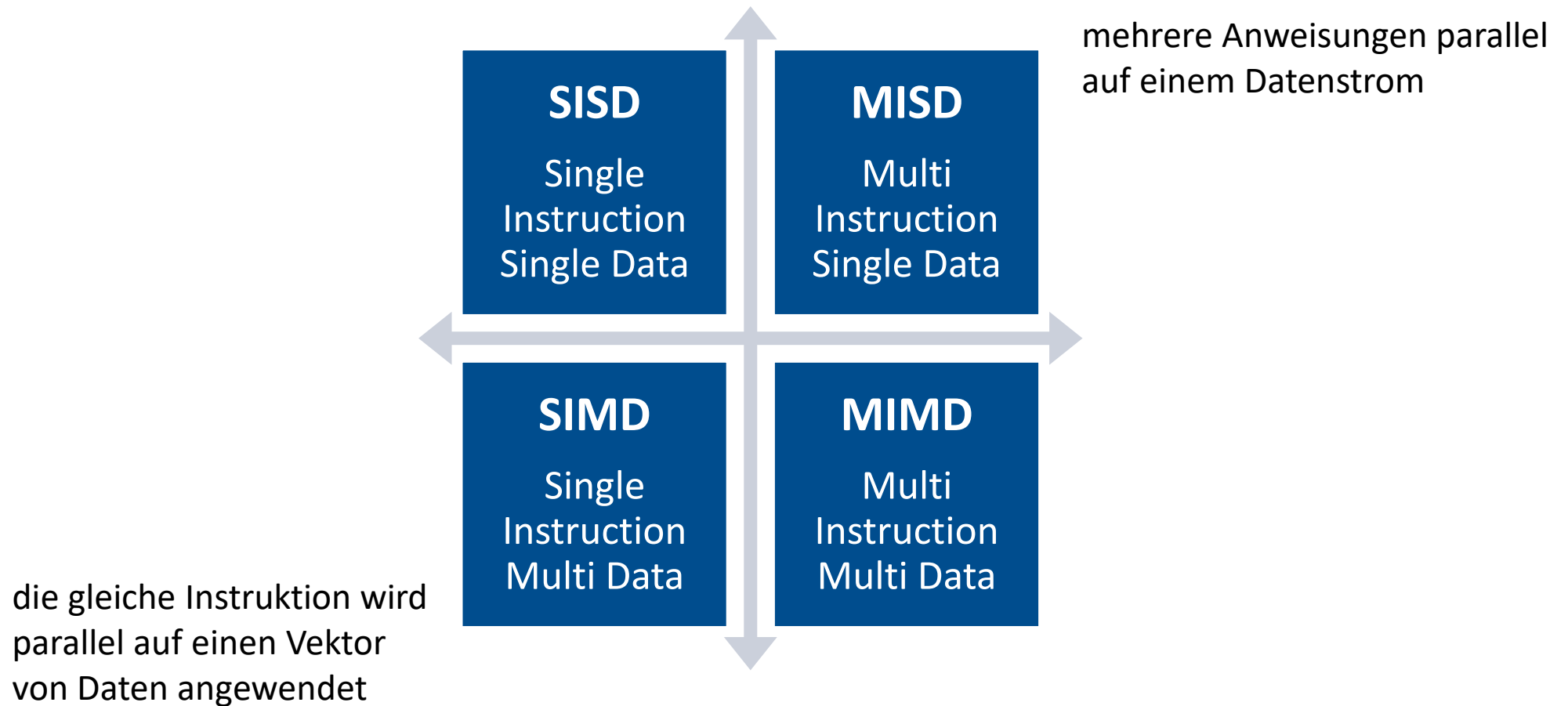
- symmetrische Rechner (alle Prozessoren sind bzgl. ihrer Rolle untereinander austauschbar)
- nicht-symmetrische Rechner (es gibt Leader und Follower)

4

## Eigenständigkeit von Prozessoren

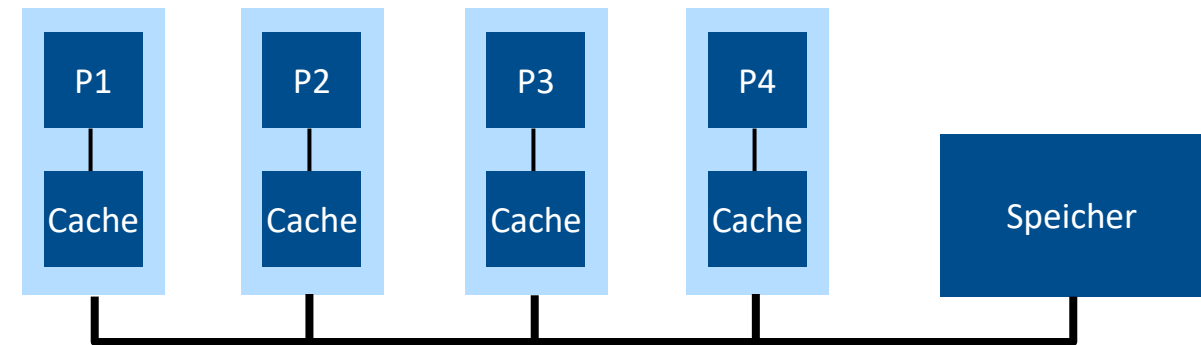
- lose gekoppelte Parallelrechner (Netzwerk aus selbstständigen Rechnern)
- eng gekoppelte Parallelrechner (physikalisch ein Rechner)

## Klassifikation nach Flynn [1966]



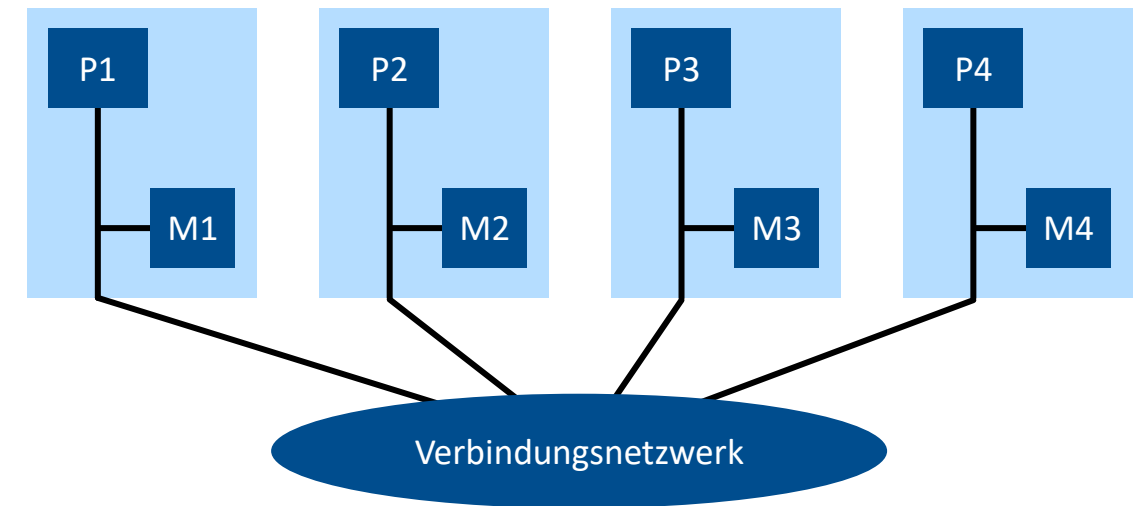
## Beispiele für parallele Rechnerarchitekturen

- symmetrische Multiprozessoren (SMP) mit gemeinsamem nichtverteilten Speicher
- oft busbasiert mit physikalisch gemeinsamem Speicher
- homogene Prozessoren
- Reduktion der Speicherzugriffslatenz durch lokale Caches
- Problem: Cache-Kohärenz



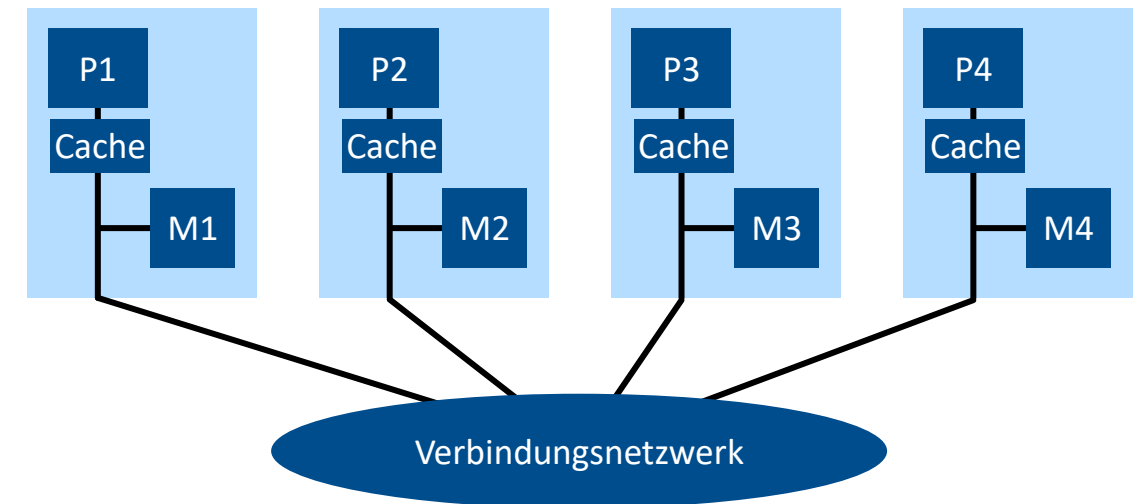
## Beispiele für parallele Rechnerarchitekturen

- Systeme mit Verbindungsnetzwerk und verteiltem gemeinsamen Speicher
- gemeinsamer Adressraum, lokale Speichermodule
- Beispiel: Carnegie Mellon Cm\* [1977]



## Beispiele für parallele Rechnerarchitekturen

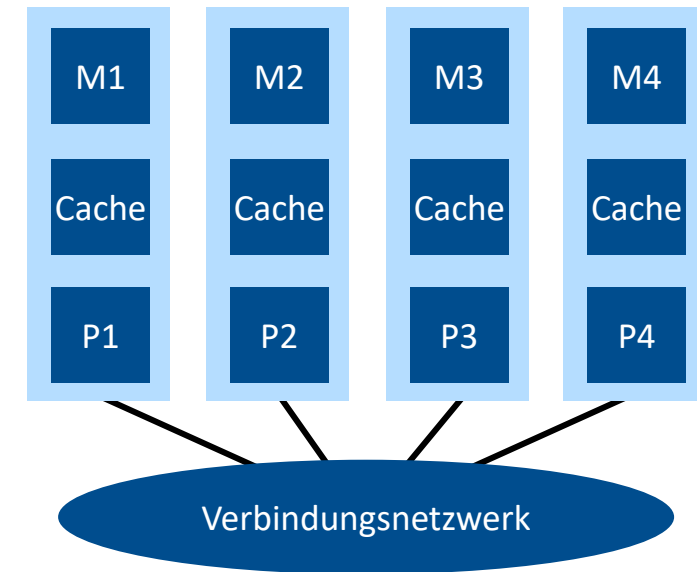
- Systeme mit Verbindungsnetzwerk, verteiltem gemeinsamen Speicher und lokalem Cache
- Verbindungsnetzwerk, z.B.: Crossbar Switch
- Beispiel: Sun Enterprise 10000 [1998], SGI Origin 2000 [1993]
- Realisierungen mit bis zu 256 CPUs





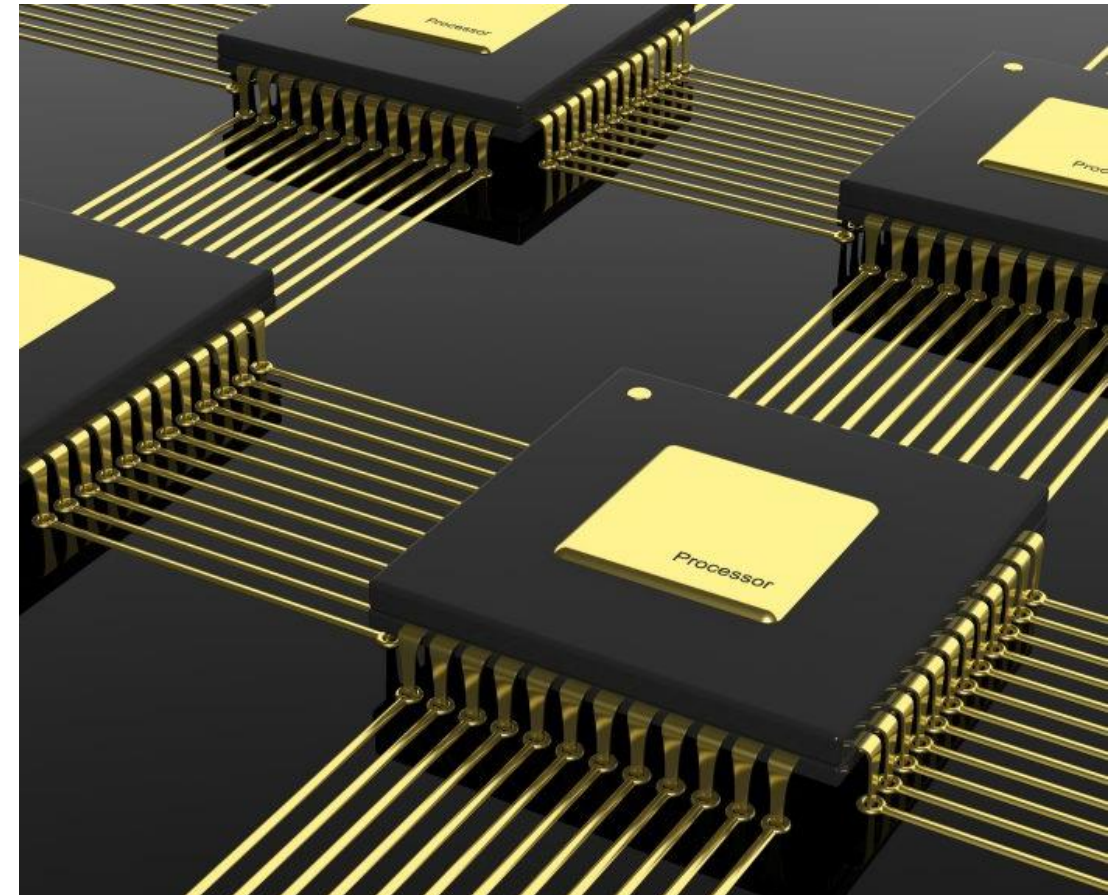
## Beispiele für parallele Rechnerarchitekturen

- Systeme mit Verbindungsnetzwerk und lokalem Speicher
- Zugriff auf fremden lokalen Speicher nicht möglich
- Kommunikation durch Austausch von Nachrichten
- gute physikalische Skalierbarkeit, mehrere Tausend CPUs möglich
- Beispiele:
  - MPP (massively parallel processors), Cray T3E, IBM SP/2
  - COW (cluster of workstations)



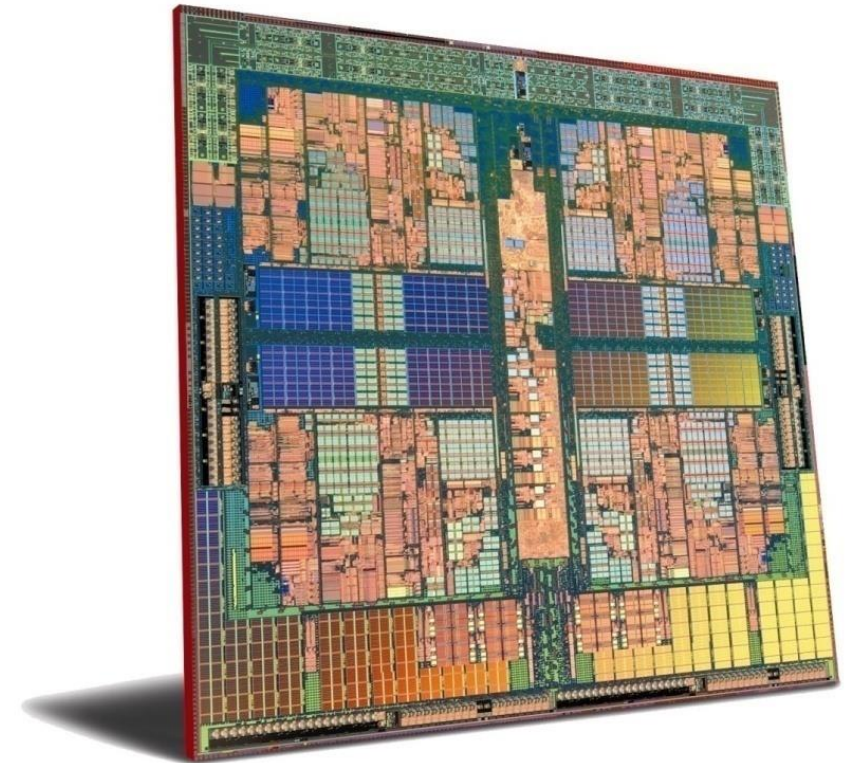
## Beispiele für parallele Rechnerarchitekturen

- Heutige Multi-Cores
  - Dual-Core:
    - Intel Core 2 Duo (2006)
    - AMD Athlon X2 (2007)
  - Quad-Cores:
    - AMD Phenom II X4 (2010)
    - Intel Core 2 Quad (2007)
    - Intel Core i7 (2009)
  - Octa-Cores (und darüber hinaus):
    - AMD Ryzen (bis zu 16 Kerne) (2017)
    - Intel Core i9 (bis zu 18 Kerne) (2017)



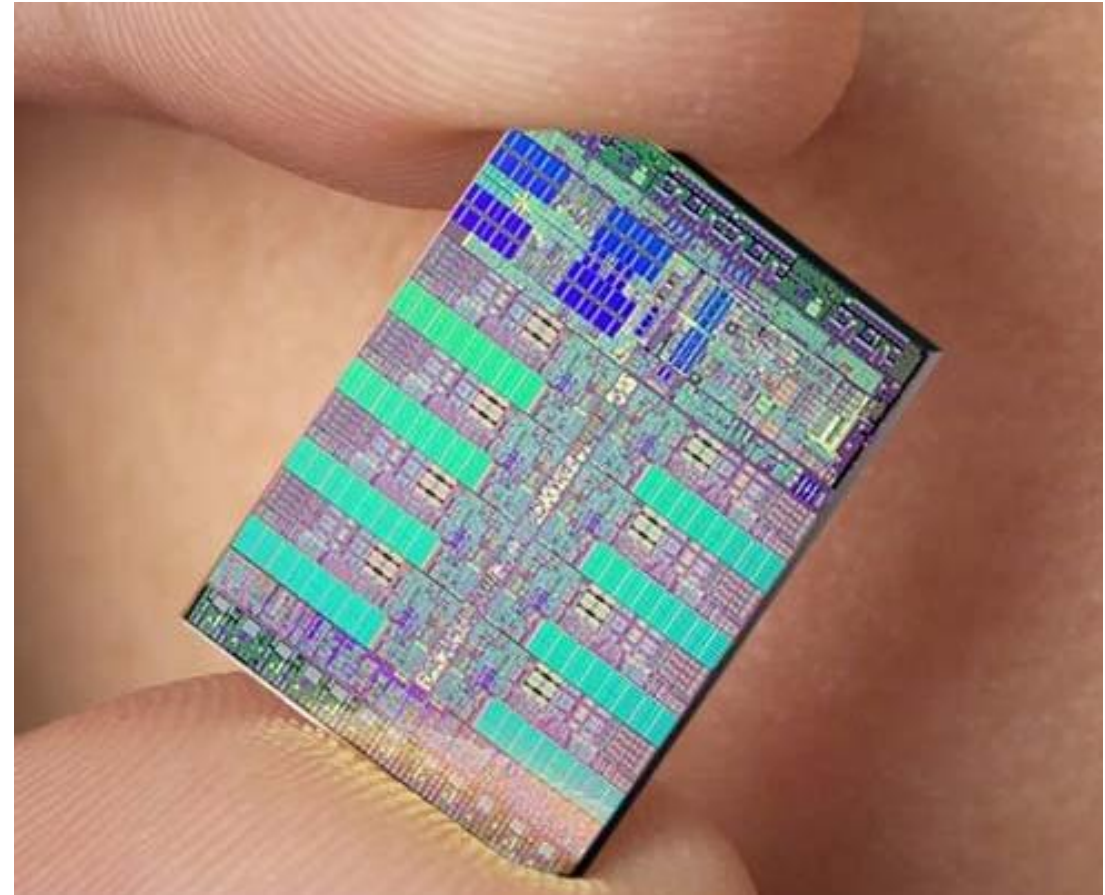
## Beispiel: AMD Phenom II X4 (2010)

- 4 identische Cores
- Caches
  - L1: 64 KB je Kern
  - L2: 512 KB je Kern
  - L3: 6144 KB (gemeinsam)
- 45 nm
- 2,8 – 3 GHz

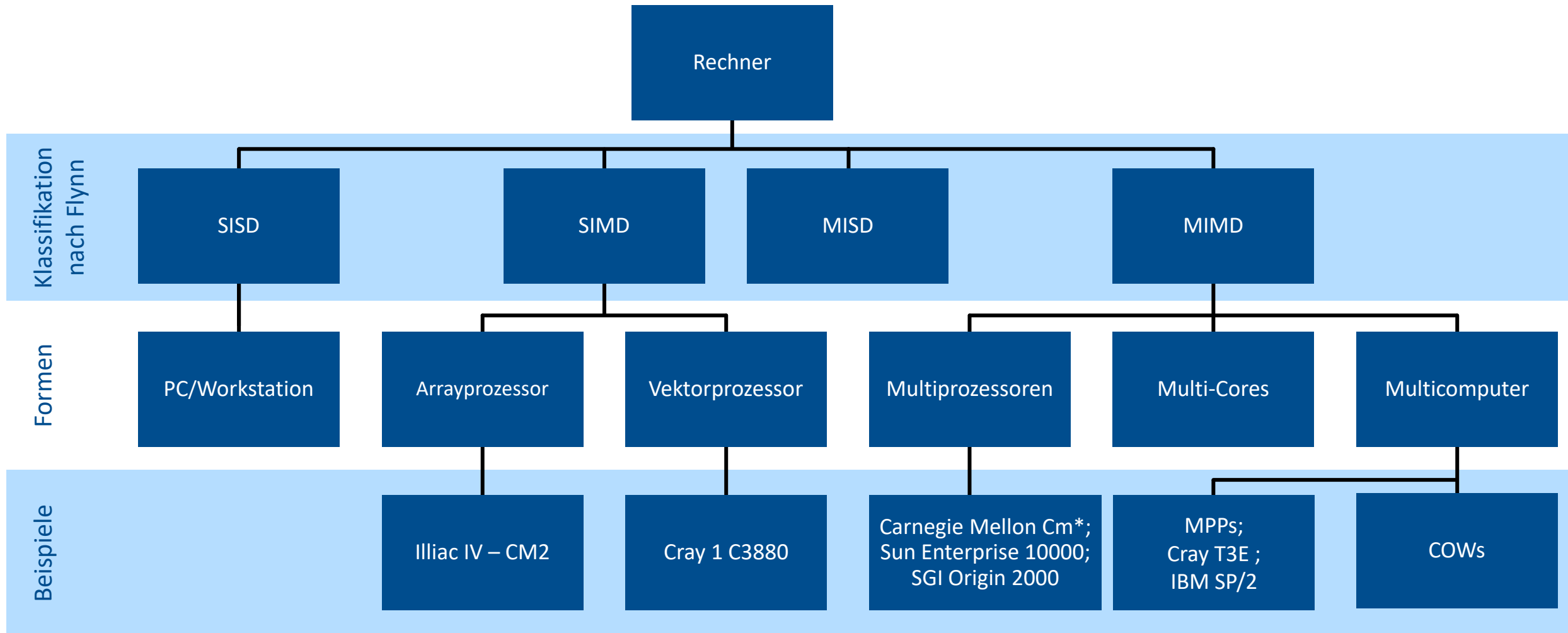


## Beispiel: Cell Chip (2005)

- Cell Chip (IBM, Sony, Toshiba)
- 234 Millionen Transistoren
- >256 GFlops (single prec.)
- CPU:
  - 1 x 64-Bit-Power-PC (PPE)
  - 8 x SPE



# Überblick





# Kapitel 5: Parallelverarbeitung

Parallele Rechnerarchitekturen

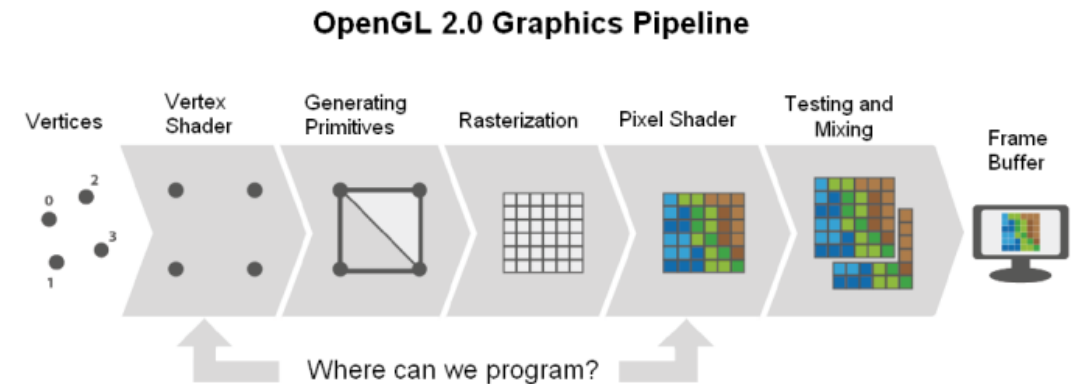
Exkurs: Grafikkarten

Exkurs: Verbindungsstrukturen



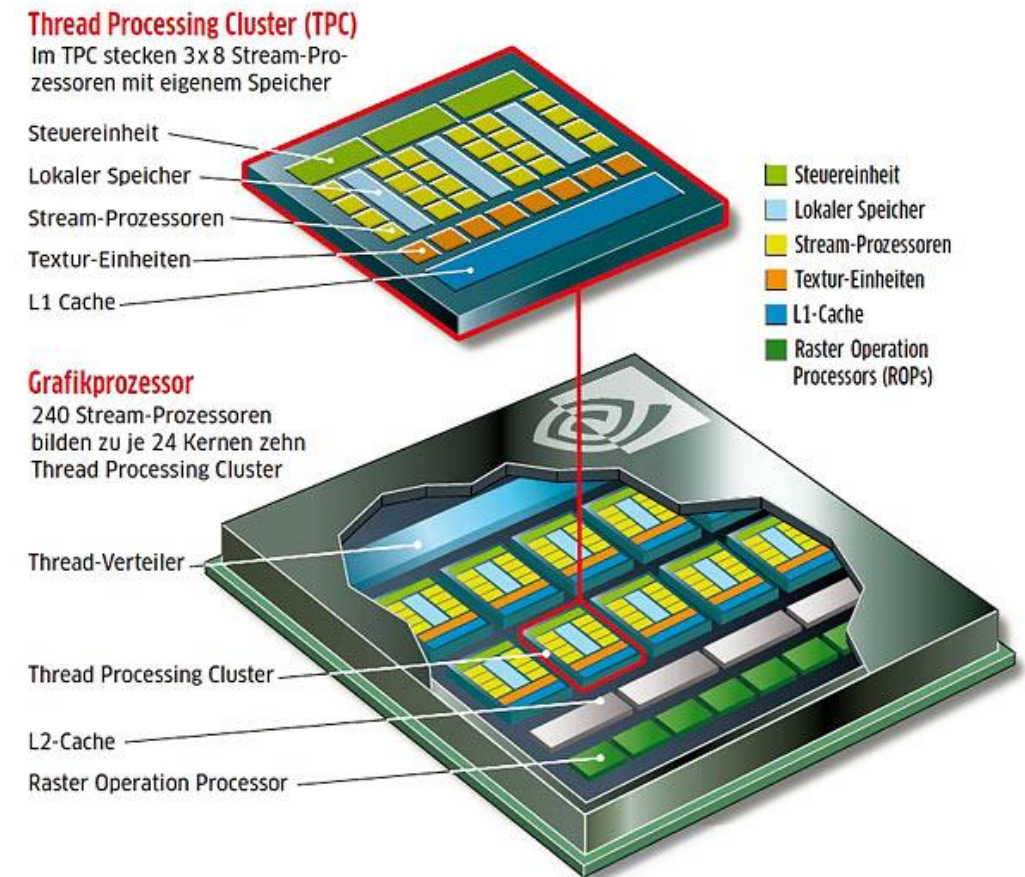
## Exkurs: Grafikprozessor (1)

- Beschreibung der Schritte für die Darstellung einer 3D-Szene:  
**Grafikpipeline**
- Umsetzung der Grafikpipeline direkt in Hardware durch **Grafikprozessor**



## Exkurs: Grafikprozessor (2)

- Darstellung von Bildpunkten durch Vektoren
- Auslegung der GPU auf parallele Ausführung einer Operation auf sehr vielen Bildpunkten
- Direkte Umsetzung spezieller Operationen, die in der Grafikverarbeitung häufig Verwendung finden, in Hardware
  - Skalarprodukt, Kreuzprodukt
  - Logarithmus, Sinus, Kosinus

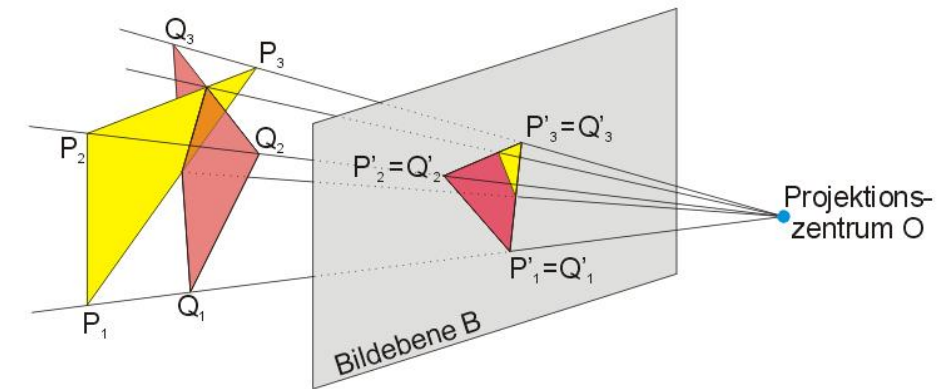




## Exkurs: Grafikprozessor (3)

- ein wichtiger Schritt in der Grafikpipeline: Projektion von 3D auf 2D
- Beispiel: Zentralprojektion
- für einen Punkt  $(x,y,z)$  und das Projektionszentrum  $(0,0,-z_0)$  ergibt sich die Projektion durch:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & z_0^{-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \\ 1 + z \cdot z_0^{-1} \end{pmatrix}$$



Matrix  $A \cdot$  Vektor  $x =$

$$\begin{pmatrix} \sum_{j=1}^n a_{1j} x_j \\ \vdots \\ \sum_{j=1}^n a_{mj} x_j \end{pmatrix}$$

die Operation  $a' = a + (b \cdot c)$  wird sehr häufig verwendet, daher wird sie durch einen Multiplikationsakkumulator (MAC) direkt in Hardware realisiert

## Beispiel: AMD Radeon RX 580 (2017)

- GPU “Polaris 20”
  - 2304 shading units
  - 36 computing units
  - 144 Textureinheiten
- 1,257 GHz
- 6200 Gflops
- 2 x 1024 KB L2-Cache





# Kapitel 5: Parallelverarbeitung

Parallele Rechnerarchitekturen

Exkurs: Grafikkarten

Exkurs: Verbindungsstrukturen

## Lernziele

- Ungerichtete Graphen als abstrakte Form der Verbindungsstruktur kennenlernen
- Eigenschaften Durchmesser, Grad und Anzahl Verbindungen von ungerichteten Graphen kennen lernen und an unterschiedlichen Graphen bestimmen können
- Unterschiedliche Verbindungsstrukturtopologien kennenlernen und ihre Vor- und Nachteile verstehen

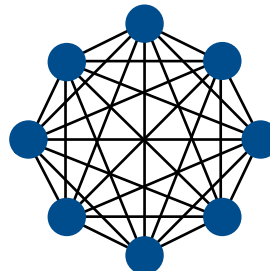
# Kurzer Exkurs über Verbindungsstrukturen

- **Kommunikationsaufwand** zwischen den Prozessoren mitentscheidend für Leistung paralleler Rechnersysteme.

➡ Nicht jedes Problem ist für Parallelisierung geeignet.

- Verschiedene Kommunikationsstrukturen unterscheiden sich hinsichtlich ihrer **Kosten** und ihrer **Leistung**.
  - Kosten: Wie viele Verbindungen sind notwendig?
  - Leistung: Wie schnell gelangt eine Nachricht von A nach B? Wie robust ist die Kommunikationsstruktur gegenüber Ausfällen?

Beispiel: Vollständige Verbindungsstruktur



# Modellierung von Verbindungen (1)

- Modellierung:

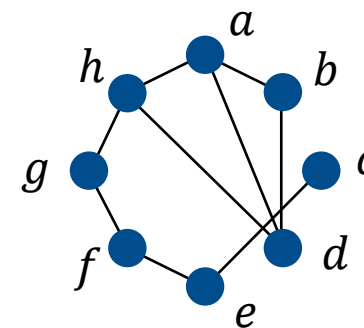
Die Topologie eines Parallelrechners wird durch einen abstrakten Graphen  $G = (V, E)$  dargestellt mit

- $V = \{1, \dots, n\}$  Menge der Knoten, sie repräsentieren die Prozessoren oder Switches
- $E \subseteq \{ \{a, b\} | a, b \in V \}$  Menge der Kanten, sie repräsentieren die Verbindungen

Beispiel:

$$V = \{a, b, c, d, e, f, g, h\}$$

$$E = \{ \{a, b\}, \{b, d\}, \{a, d\}, \{c, e\}, \{e, f\}, \{f, g\}, \{g, h\}, \{d, h\}, \{a, h\} \}$$



## Modellierung von Verbindungen (2)

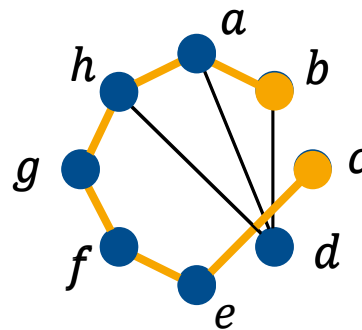
- Charakteristika einer Verbindungsstruktur
  - $\text{Durchmesser}(G) = \max_{v,w \in V} \{\min\{\text{Länge des Pfades von } v \text{ nach } w\}\}$

Beispiel:

$$V = \{a, b, c, d, e, f, g, h\}$$

$$E = \{\{a, b\}, \{b, d\}, \{a, d\}, \{c, e\}, \{e, f\}, \{f, g\}, \{g, h\}, \{d, h\}, \{a, h\}\}$$

$$\text{Durchmesser}(G) = 6$$



Länge	a	b	c	d	e	f	g	h
a	0	1	5	1	4	3	2	1
b	1	0	6	1	5	4	3	2
c	5	6	0	5	1	2	3	4
d	1	1	5	0	4	3	2	1
e	4	5	1	4	0	1	2	3
f	3	4	2	3	1	0	1	2
g	2	3	3	2	2	1	0	1
h	1	2	4	1	3	2	1	0

# Modellierung von Verbindungen (3)

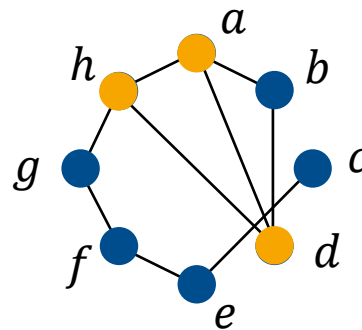
- Charakteristika einer Verbindungsstruktur
  - $Durchmesser(G) = \max_{v,w \in V} \{\min\{Länge\ des\ Pfades\ von\ v\ nach\ w\}\}$
  - $Grad(G) = \max_{v \in V} |\{w \in V | \{v, w\} \in E\}|$

Beispiel:

$V = \{a, b, c, d, e, f, g, h\}$

$E = \{\{a, b\}, \{b, d\}, \{a, d\}, \{c, e\}, \{e, f\}, \{f, g\}, \{g, h\}, \{d, h\}, \{a, h\}\}$

$Grad(G) = 3$



Knoten	Grad (v)
<i>a</i>	3
<i>b</i>	2
<i>c</i>	1
<i>d</i>	3
<i>e</i>	2
<i>f</i>	2
<i>g</i>	2
<i>h</i>	3



# Modellierung von Verbindungen (4)

- Charakteristika einer Verbindungsstruktur

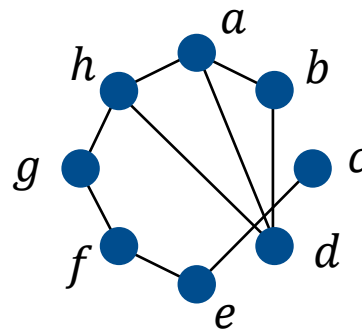
- $Durchmesser(G) = \max_{v,w \in V} \{\min\{\text{Länge des Pfades von } v \text{ nach } w\}\}$
- $Grad(G) = \max_{v \in V} |\{w \in V | \{v, w\} \in E\}|$
- $Verbindungen(G) = |E|$

Beispiel:

$V = \{a, b, c, d, e, f, g, h\}$

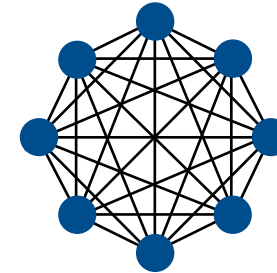
$E = \{\{a, b\}, \{b, d\}, \{a, d\}, \{c, e\}, \{e, f\}, \{f, g\}, \{g, h\}, \{d, h\}, \{a, h\}\}$

$Verbindungen(G) = 9$



# Vollständiger Verbindungsgraph

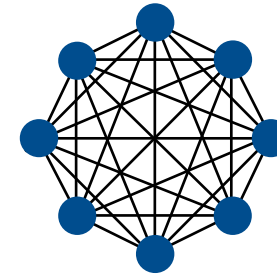
- Eigenschaft: Alle Knoten sind miteinander verbunden



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )			

# Vollständiger Verbindungsgraph

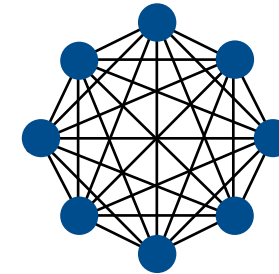
- Eigenschaft: Alle Knoten sind miteinander verbunden



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1		

# Vollständiger Verbindungsgraph

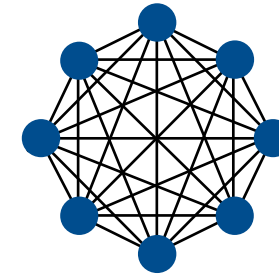
- Eigenschaft: Alle Knoten sind miteinander verbunden



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	

# Vollständiger Verbindungsgraph

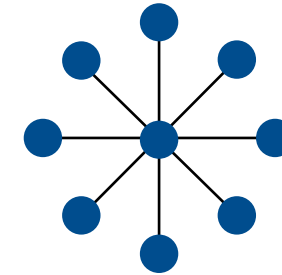
- Eigenschaft: Alle Knoten sind miteinander verbunden
- Vorteil: Keine Kollisionen, kurze Übertragungszeit
- Nachteil: sehr teuer



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$

# Stern

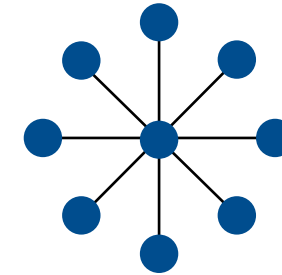
- Eigenschaft: Zentraler Knoten, über den alles läuft



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )			

# Stern

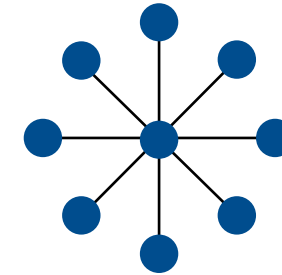
- Eigenschaft: Zentraler Knoten, über den alles läuft



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2		

# Stern

- Eigenschaft: Zentraler Knoten, über den alles läuft

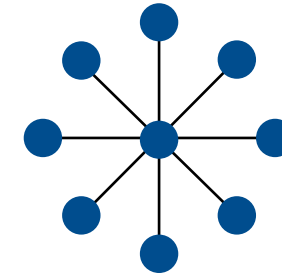


Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	



## Stern

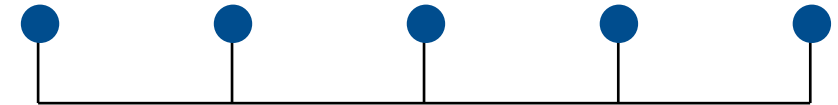
- Eigenschaft: Zentraler Knoten, über den alles läuft
- Vorteil: geringe Kosten, geringe Übertragungszeit
- Nachteil: Ausfall des zentralen Knotens kritisch



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$

# Bus

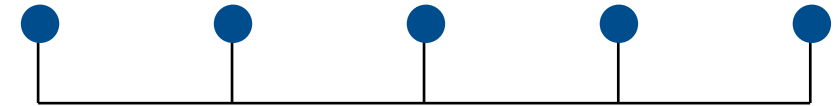
- Eigenschaft: ähnlich wie Stern, Bus übernimmt zentrale Rolle



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )			

## Bus

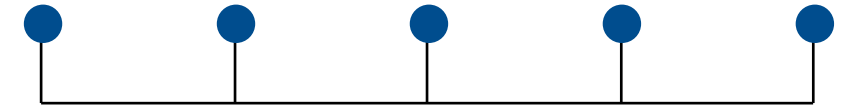
- Eigenschaft: ähnlich wie Stern, Bus übernimmt zentrale Rolle



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1		

# Bus

- Eigenschaft: ähnlich wie Stern, Bus übernimmt zentrale Rolle

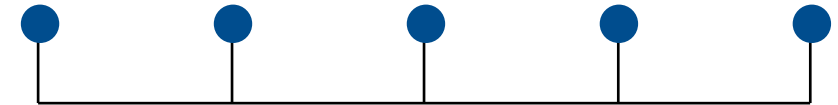


Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	

# Bus

- Eigenschaft: ähnlich wie Stern, Bus übernimmt zentrale Rolle
- Vorteil: geringe Kosten, geringe Übertragungszeit
- Nachteil: Bus zentrales Element, Überlastung möglich

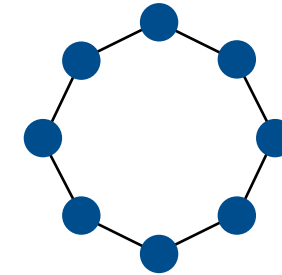
Topologie versagt bei den heutigen Technologien bei großem Datentransfer zwischen den Prozessoren



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$

# Ring

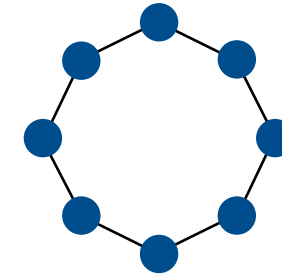
- Eigenschaft: Prozessoren sind im „Kreis“ angeordnet



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )			

# Ring

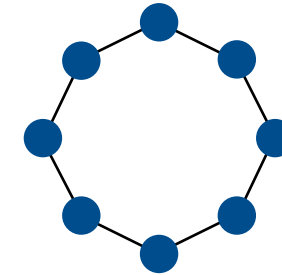
- Eigenschaft: Prozessoren sind im „Kreis“ angeordnet



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$		

# Ring

- Eigenschaft: Prozessoren sind im „Kreis“ angeordnet



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	

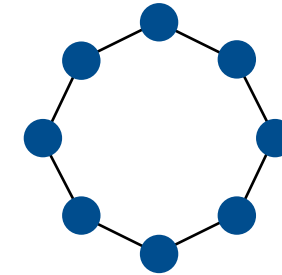


# Ring

- Eigenschaft: Prozessoren sind im „Kreis“ angeordnet
- Vorteil: geringe Kosten, „doppelter“ Boden
- Nachteil: längere Übertragungszeit

Beispiel für Ring: „Token-Ring“

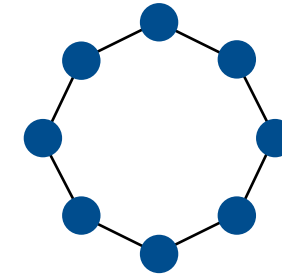
Es kreist ein sogenanntes Token (spezielles Paket).  
Ein Rechner darf nur dann senden, wenn er das Token besitzt.



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$

## Ring - CDDI/FDDI-Ring (Copper/Fiber Distributed Data Interconnect)

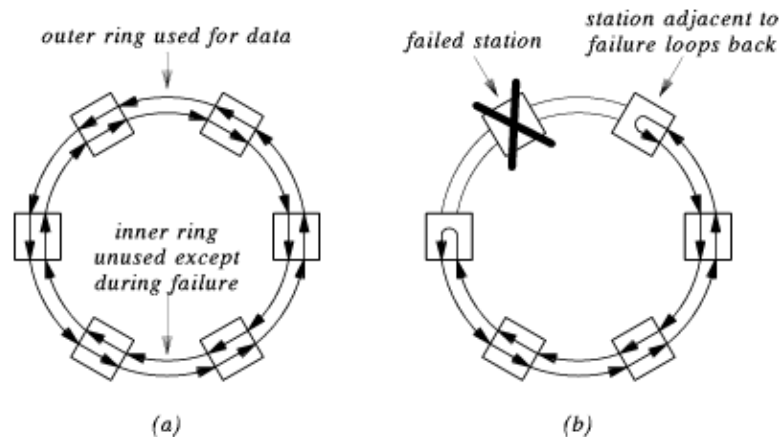
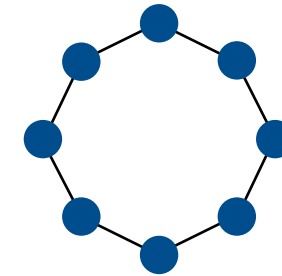
- Eigenschaft: wie beim Ring, aber doppelte Verbindung



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$

## Ring - CDDI/FDDI-Ring (Copper/Fiber Distributed Data Interconnect)

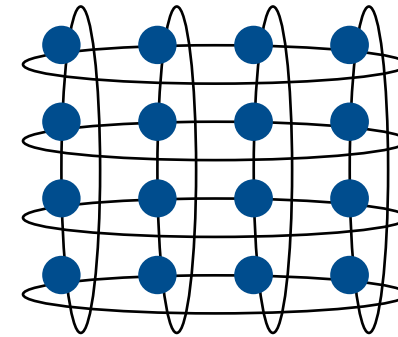
- Eigenschaft: wie beim Ring, aber doppelte Verbindung
- Vorteil: geringe Kosten, fehlertolerant!
- Nachteil: längere Übertragungszeit



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$

## MESH (Torusähnliches Gitter)

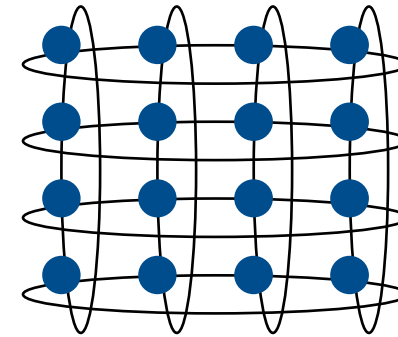
- Eigenschaft: Gitter mit Verbindung „hintenrum“



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )			

## MESH (Torusähnliches Gitter)

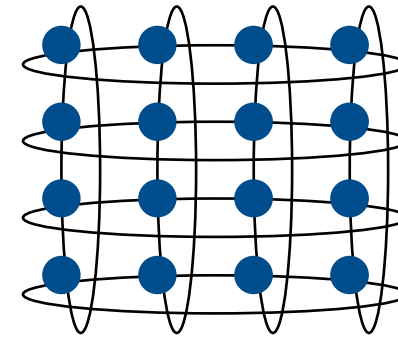
- Eigenschaft: Gitter mit Verbindung „hintenrum“



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$		

## MESH (Torusähnliches Gitter)

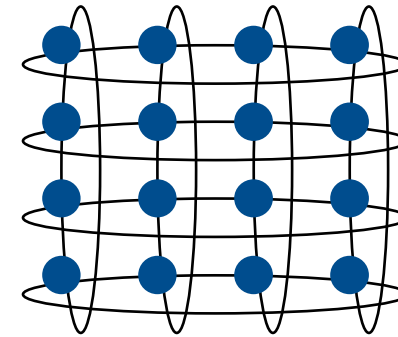
- Eigenschaft: Gitter mit Verbindung „hintenrum“



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	

## MESH (Torusähnliches Gitter)

- Eigenschaft: Gitter mit Verbindung „hintenrum“
- Vorteil: akzeptable, viele mögliche Wege
- Nachteil: längere Wege als beim Stern, Bus

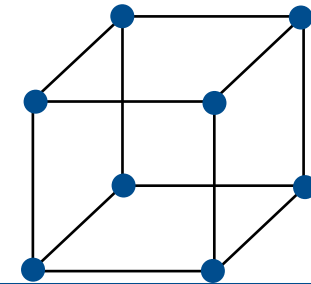


wird z.B. bei MPPs benutzt

Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$

## Hypercube (d-dimensionaler Würfel)

- Eigenschaft: Würfel mit  $d = \log_2 n$  Dimensionen

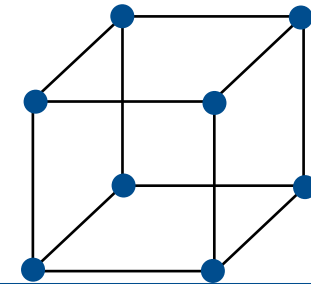


Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )			



## Hypercube (d-dimensionaler Würfel)

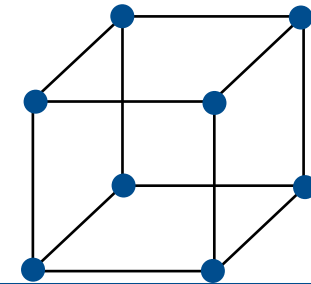
- Eigenschaft: Würfel mit  $d = \log_2 n$  Dimensionen



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$		

## Hypercube (d-dimensionaler Würfel)

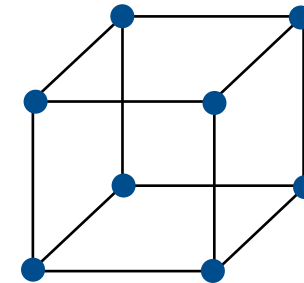
- Eigenschaft: Würfel mit  $d = \log_2 n$  Dimensionen



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$	$\log_2 n$	

## Hypercube (d-dimensionaler Würfel)

- Eigenschaft: Würfel mit  $d = \log_2 n$  Dimensionen
- Vorteil: kürzere Wege als beim Mesh
- Nachteil: zu hoher Grad

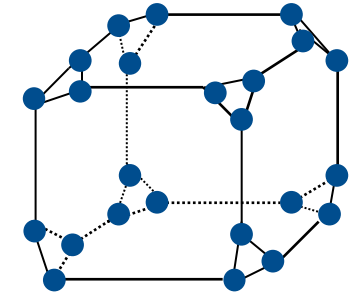


Bei vielen Prozessoren wegen des hohen Grads nicht verwendbar

Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$	$\log_2 n$	$(n \log_2 n)/2$

## Cube Connected Cycle (CCC)

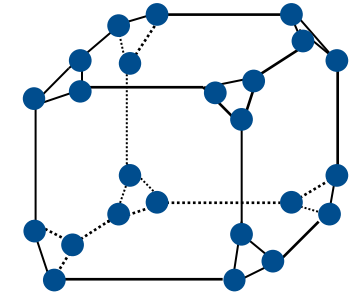
- Eigenschaft: Würfel mit „engeren Zellen“



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$	$\log_2 n$	$(n \log_2 n)/2$
CCC ( $ V  = n \log_2 n$ )			

## Cube Connected Cycle (CCC)

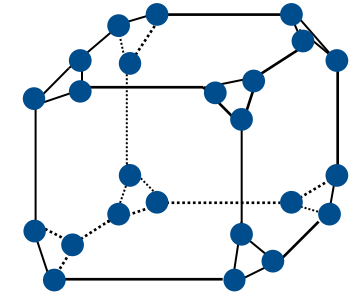
- Eigenschaft: Würfel mit „engeren Zellen“



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$	$\log_2 n$	$(n \log_2 n)/2$
CCC ( $ V  = n \log_2 n$ )	$\lfloor 5/2 \log_2 n \rfloor - 1$		

## Cube Connected Cycle (CCC)

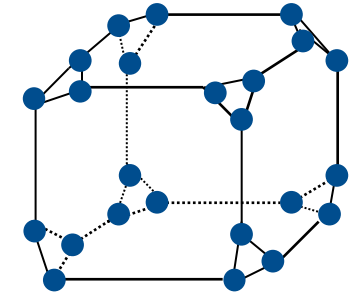
- Eigenschaft: Würfel mit „engeren Zellen“



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$	$\log_2 n$	$(n \log_2 n)/2$
CCC ( $ V  = n \log_2 n$ )	$\lfloor 5/2 \log_2 n \rfloor - 1$	3	

## Cube Connected Cycle (CCC)

- Eigenschaft: Würfel mit „engeren Zellen“
- Vorteil: Cluster von schnellen Verbindungen
- Nachteil: längere Wege als beim Stern, Bus



Verbindungstyp	Durchmesser	Grad	Verbindungen
Vollständig ( $ V  = n$ )	1	$n - 1$	$n(n - 1)/2$
Stern ( $ V  = n$ )	2	$n - 1$	$n - 1$
Bus ( $ V  = n$ )	1	1	$n$
Ring ( $ V  = n$ )	$\lfloor n/2 \rfloor$	2	$n$
MESH ( $ V  = n$ )	$n^{1/2}$	4	$2n$
Hypercube ( $ V  = n$ )	$\log_2 n$	$\log_2 n$	$(n \log_2 n)/2$
CCC ( $ V  = n \log_2 n$ )	$\lfloor 5/2 \log_2 n \rfloor - 1$	3	$3/2 n \log_2 n$

# Überblick

## Teil 1: Der Rechneraufbau (Kapitel 2-5)

- Rechner im Überblick
- Pipelining
- Speicher
- Parallelverarbeitung

## Teil 2: Der Funktionalitätsaufbau (Kapitel 6-12)

- Kodierung von Zeichen und Zahlen
- Grundbegriffe, Boolesche Funktionen
- Darstellungsmöglichkeiten
- Schaltkreise, Synthese, spezielle Schaltkreise



# Überblick

## Teil 1: Der Rechneraufbau (Kapitel 2-5)

- Rechner im Überblick
- Pipelining
- Speicher
- Parallelverarbeitung

## Teil 2: Der Funktionalitätsaufbau (Kapitel 6-12)

- Kodierung von Zeichen und Zahlen
- Grundbegriffe, Boolesche Funktionen
- Darstellungsmöglichkeiten
- Schaltkreise, Synthese, spezielle Schaltkreise