

# **Halbleitertechnologie von A bis Z**

Digitaltechnik

|                                                |   |
|------------------------------------------------|---|
| <b>CPU vs. GPU</b> .....                       | 3 |
| <b>Einordnung</b> .....                        | 3 |
| <b>Architekturunterschied</b> .....            | 3 |
| <b>Speicherhierarchie und Bandbreite</b> ..... | 4 |
| <b>Anwendungsfelder</b> .....                  | 4 |
| <b>Vergleichstabelle</b> .....                 | 5 |

# CPU vs. GPU

---

## Einordnung

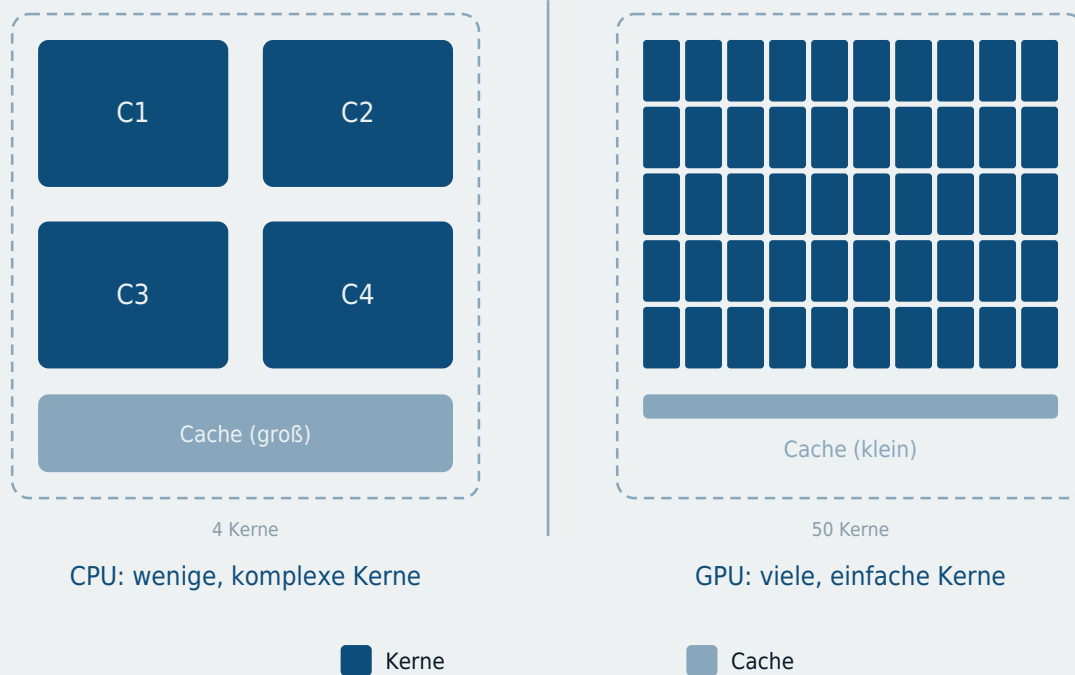
CPU und GPU sind beides Prozessoren im Sinne des [vorherigen Kapitels](#) – beide bestehen aus Steuerwerk, Rechenwerken, Registern und einer Cache-Hierarchie, verbunden über einen Bus. Der entscheidende Unterschied liegt in der Zielsetzung: Die CPU (Central Processing Unit) ist darauf ausgelegt, wenige, oft voneinander abhängige Aufgaben so schnell wie möglich nacheinander abzuarbeiten. Die GPU (Graphics Processing Unit) ist darauf ausgelegt, sehr viele gleichartige, voneinander unabhängige Aufgaben gleichzeitig zu bearbeiten. Beide Ansätze sind sinnvoll – nur eben für unterschiedliche Arten von Problemen.

## Architekturunterschied

Eine CPU besitzt typischerweise nur wenige Kerne (aktuell etwa 4 bis 16 im Consumer-Bereich), dafür ist jeder einzelne Kern hochkomplex: ausgefeiltes Steuerwerk, tiefes Pipelining, Sprungvorhersage und Out-of-Order-Execution (Befehle werden nicht zwingend in Programmreihenfolge ausgeführt, sondern sobald ihre Daten bereitstehen), um auch bei komplexen, verzweigten Programmen eine hohe Geschwindigkeit pro Kern zu erreichen.

Eine GPU geht den entgegengesetzten Weg: Statt weniger komplexer Kerne enthält sie hunderte bis tausende einfacher, stark vereinfachter Kerne. Diese arbeiten nach dem SIMT-Prinzip (Single Instruction, Multiple Threads) – dieselbe Instruktion wird gleichzeitig auf viele unterschiedliche Datenelemente angewendet, etwa auf alle Pixel eines Bildausschnitts oder alle Einträge einer Matrix. Ein einzelner GPU-Kern ist dabei deutlich langsamer und unflexibler als ein CPU-Kern – die Stärke liegt in der schieren Anzahl.

## CPU vs. GPU – gleiche Chipfläche im Vergleich



Gleiche Chipfläche – CPU investiert in wenige starke Kerne, GPU in viele einfache

## Speicherhierarchie und Bandbreite

Beide Architekturen nutzen die im vorherigen Kapitel beschriebene **Cache-Hierarchie**, aber mit unterschiedlicher Zielrichtung. Die CPU optimiert auf niedrige Latenz: Jeder Kern besitzt einen vergleichsweise großen Cache, um einzelne Speicherzugriffe so schnell wie möglich zu beantworten – wichtig, wenn ein sequenzieller Programmablauf auf ein Ergebnis wartet.

Die GPU optimiert dagegen auf hohen Durchsatz: Der Cache pro Kern ist klein, dafür ist der Hauptspeicher (meist GDDR6 oder HBM statt gewöhnlichem DDR-DRAM) extrem breit angebunden und liefert ein Vielfaches der Bandbreite einer CPU-Speicheranbindung. Statt Wartezeiten durch großen Cache zu vermeiden, versteckt die GPU sie: Während ein Thread auf Daten wartet, schaltet die Hardware verzögerungsfrei auf einen anderen, bereits wartenden Thread um – bei tausenden gleichzeitig aktiven Threads fällt eine einzelne Wartezeit kaum ins Gewicht.

## Anwendungsfelder

Die CPU ist die richtige Wahl für sequenzielle Programmlogik mit vielen Verzweigungen: Betriebssysteme, Anwendungssoftware, Datenbankabfragen –

überall dort, wo der nächste Schritt vom Ergebnis des vorherigen abhängt und sich schlecht vorab parallelisieren lässt.

Die GPU wurde ursprünglich für die Grafikpipeline entwickelt, bei der dieselbe Berechnung (Beleuchtung, Texturierung, Transformation) auf Millionen Pixel oder Vertices gleichzeitig angewendet wird – ein klassisches datenparalleles Problem. Genau diese Eigenschaft macht GPUs auch für andere datenparallele Aufgaben interessant: Matrixmultiplikationen, wissenschaftliche Simulationen und vor allem das Training neuronaler Netze, das im Kern aus enorm vielen gleichartigen, parallel ausführbaren Matrixoperationen besteht. Das ist der Hauptgrund, warum GPUs heute die dominierende Hardware für KI-Training sind.

## Vergleichstabelle

| <b>Merkmal</b>     | <b>CPU</b>                 | <b>GPU</b>                      |
|--------------------|----------------------------|---------------------------------|
| Kernanzahl         | wenige (4–16)              | viele (hunderte bis tausende)   |
| Kerntyp            | komplex, hochgetaktet      | einfach, niedriger getaktet     |
| Ausführungsmodell  | Out-of-Order, spekulativ   | SIMT (parallel, gleichartig)    |
| Cache pro Kern     | groß                       | klein                           |
| Speicherbandbreite | moderat                    | sehr hoch (GDDR6/HBM)           |
| Optimierungsziel   | niedrige Latenz            | hoher Durchsatz                 |
| Typische Anwendung | sequenzielle Programme, OS | Grafik, KI-Training, Simulation |