The GAME Engineers

CG #2 – CPU vs. GPU

Inhalt

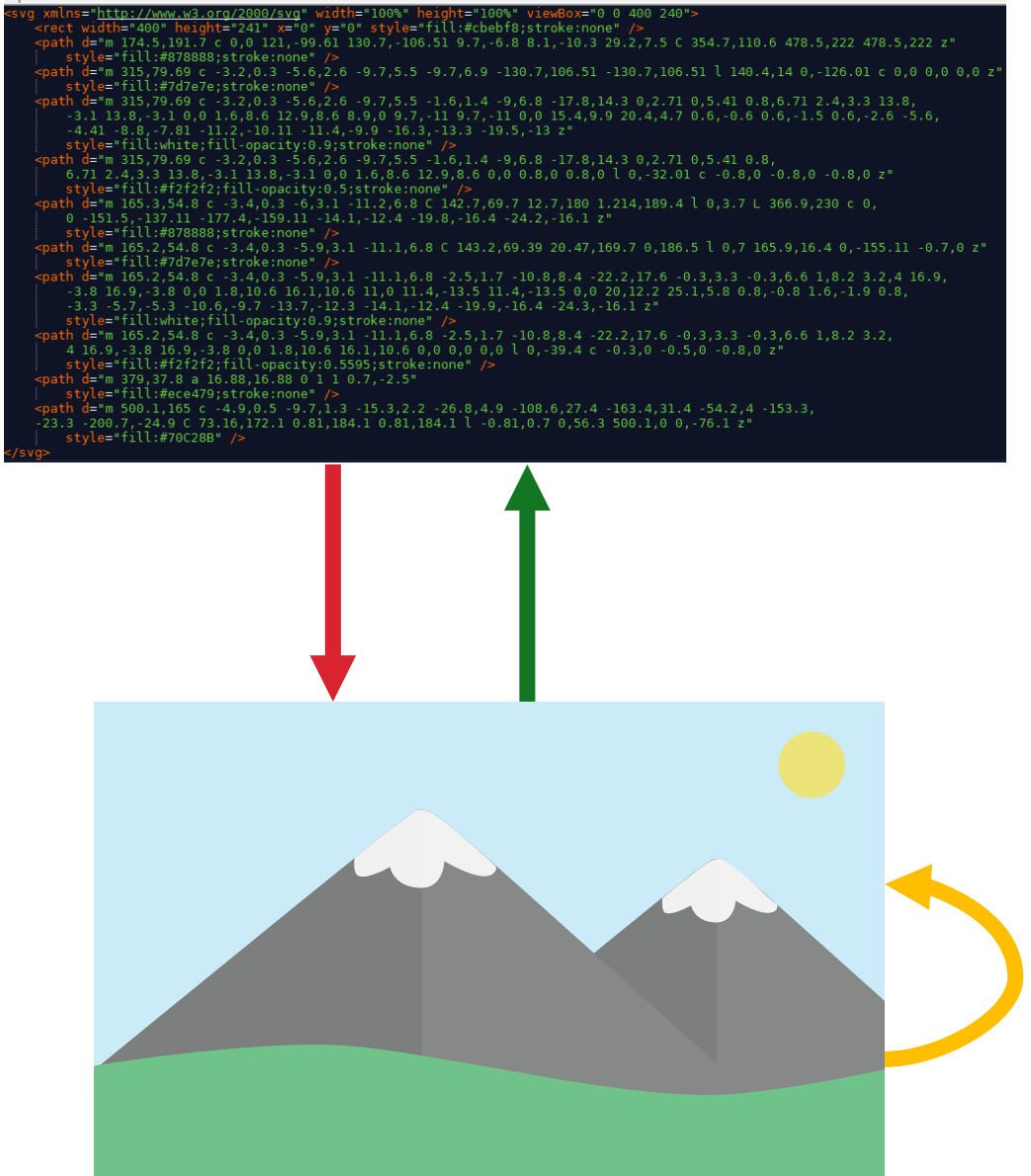
- Visual Computing

- Computergrafik #1
  - Was das?
  - Anwendungen
  - Vektor- vs. Rastergrafik

- CPU vs. GPU
  - Central Processing Unit
  - Pipelining und Superskalarität
  - SIMD und Vectorprocessing

Visual Computing

- computer graphics
  - description → image
- computer vision
  - image → description
- image processing
  - “bad” → “better” image



Visual Computing

- computer graphics
  - description → image
  - Descriptions?

```
niklas@Niklas-PC:/mnt/d/Niklas Birk/Downloads$ od -x bsp.png
0000000 5089 474e 0a0d 0a1a 0000 0d00 4849 5244
0000020 0000 0a00 0000 0a00 0208 0000 0200 5850
0000040 00ea 0000 7301 4752 0042 ceae e91c 0000
0000060 0400 4167 414d 0000 8fb1 fc0b 0561 0000
0000100 0900 4870 7359 0000 c30e 0000 c30e c701
0000120 a86f 0064 0000 493e 4144 2854 6353 8ff8
0000140 4017 57a5 58ac dad1 0ada 2361 a803 5034
0000160 818e 8b01 c449 0e19 9801 a056 6348 415a
0000200 e18a 0070 05b7 8d21 316c 16dc 3484 c5a6
0000220 ffff 07ff 3100 0f2a 336c 523f 007c 0000
0000240 4900 4e45 ae44 6042 0082
0000251
```

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%" viewBox="0 0 400 240">
    <rect width="400" height="241" x="0" y="0" style="fill:#cbebf8;stroke:none" />
    <path d="m 174.5,191.7 c 0,0 121,-99.61 130.7,-106.51 9.7,-6.8 8.1,-10.3 29.2,7.5 C 354.7,110.6 478.5,222 478.5,222 z"
       style="fill:#878888;stroke:none" />
    <path d="m 315,79.69 c -3.2,0.3 -5.6,2.6 -9.7,5.5 -9.7,6.9 -130.7,106.51 -130.7,106.51 l 140.4,14 0,-126.01 c 0,0 0,0 0,0 z"
       style="fill:#7d7e7e;stroke:none" />
    <path d="m 315,79.69 c -3.2,0.3 -5.6,2.6 -9.7,5.5 -1.6,1.4 -9,6.8 -17.8,14.3 0,2.71 0,5.41 0.8,6.71 2.4,3.3 13.8,
       -3.1 13.8,-3.1 0,0 1.6,8.6 12.9,8.6 8.9,0 9.7,-11 9.7,-11 0,0 15.4,9.9 20.4,4.7 0.6,-0.6 0.6,-1.5 0.6,-2.6 -5.6,
       -4.41 -8.8,-7.81 -11.2,-10.11 -11.4,-9.9 -16.3,-13.3 -19.5,-13 z"
       style="fill:white;fill-opacity:0.9;stroke:none" />
    <path d="m 315,79.69 c -3.2,0.3 -5.6,2.6 -9.7,5.5 -1.6,1.4 -9,6.8 -17.8,14.3 0,2.71 0,5.41 0.8,
       6.71 2.4,3.3 13.8,-3.1 13.8,-3.1 0,0 1.6,8.6 12.9,8.6 0,0 0.8,0 0.8,0 l 0,-32.01 c -0.8,0 -0.8,0 -0.8,0 z"
       style="fill:#f2f2f2;fill-opacity:0.5;stroke:none" />
    <path d="m 165.3,54.8 c -3.4,0.3 -6,3.1 -11.2,6.8 C 142.7,69.7 12.7,180 1.214,189.4 l 0,3.7 L 366.9,230 c 0,
       0 -151.5,-137.11 -177.4,-159.11 -14.1,-12.4 -19.8,-16.4 -24.2,-16.1 z"
       style="fill:#878888;stroke:none" />
    <path d="m 165.2,54.8 c -3.4,0.3 -5.9,3.1 -11.1,6.8 C 143.2,69.39 20.47,169.7 0,186.5 l 0,7 165.9,16.4 0,-155.11 -0.7,0 z"
       style="fill:#7d7e7e;stroke:none" />
    <path d="m 165.2,54.8 c -3.4,0.3 -5.9,3.1 -11.1,6.8 -2.5,1.7 -10.8,8.4 -22.2,17.6 -0.3,3.3 -0.3,6.6 1,8.2 3.2,4 16.9,
       -3.8 16.9,-3.8 0,0 1.8,10.6 16.1,10.6 11,0 11.4,-13.5 11.4,-13.5 0,0 20,12.2 25.1,5.8 0.8,-0.8 1.6,-1.9 0.8,
       -3.3 -5.7,-5.3 -10.6,-9.7 -13.7,-12.3 -14.1,-12.4 -19.9,-16.4 -24.3,-16.1 z"
       style="fill:white;fill-opacity:0.9;stroke:none" />
    <path d="m 165.2,54.8 c -3.4,0.3 -5.9,3.1 -11.1,6.8 -2.5,1.7 -10.8,8.4 -22.2,17.6 -0.3,3.3 -0.3,6.6 1,8.2 3.2,
       4 16.9,-3.8 16.9,-3.8 0,0 1.8,10.6 16.1,10.6 0,0 0,0 0,0 l 0,-39.4 c -0.3,0 -0.5,0 -0.8,0 z"
       style="fill:#f2f2f2;fill-opacity:0.5595;stroke:none" />
    <path d="m 379,37.8 a 16.88,16.88 0 1 1 0.7,-2.5"
       style="fill:#ece479;stroke:none" />
    <path d="m 500.1,165 c -4.9,0.5 -9.7,1.3 -15.3,2.2 -26.8,4.9 -108.6,27.4 -163.4,31.4 -54.2,4 -153.3,
    -23.3 -200.7,-24.9 C 73.16,172.1 0.81,184.1 0.81,184.1 l -0.81,0.7 0,56.3 500.1,0 0,-76.1 z"
       style="fill:#70C28B" />
</svg>
```

```
glColor3f(p->color->x, p->color->y, p->color->z);
glBegin(GL_POINTS);
glVertex3f(pos->x, pos->y, pos->z);
glEnd();
```

Computergrafik #1 – Was das?

- Vorgang von der Beschreibung bis zu einem Bild
  → Rendering
  - Bild kann gespeichert werden
  - Bild kann an einem Ausgabegerät angeschaut werden (Monitor, etc.)

- Aber auch noch mehr:
  - Geometrische Modellierung von Objekten
  - Kollisionserkennung
  - Schattierung
  - Animationen
  - ...

Computergrafik #1 – Anwendungen

- Filme
  - VFX (CGI, ...)
  - Trickfilme
  - …
- Industrie
  - CAD
  - AR / VR
  - ...
- Wissenschaft
  - Modellierung
  - …
- Computer
  - GUI
  - Videospiele
  - ...

Computergrafik #1 – Vektor- vs. Rastergrafik

- Vektorgrafik
  - geometrische / mathematische Repräsentation der Bilddaten
  - Schwer zu machen, dauert länger zum Darstellen
  - Größenänderung ohne Verlust möglich

Computergrafik #1 – Vektor- vs. Rastergrafik

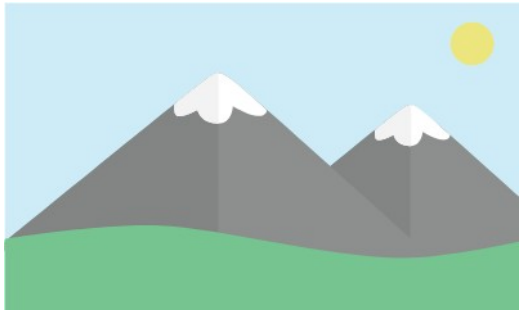
Computergrafik #1 – Vektor- vs. Rastergrafik

- Vektorgrafiken Anwendungen
  - In Software, in denen man Grafiken erstellen kann (Illustrator, …)
  - Schriftarten (Drum kannst du Schriftgrößen ohne Verlust ändern

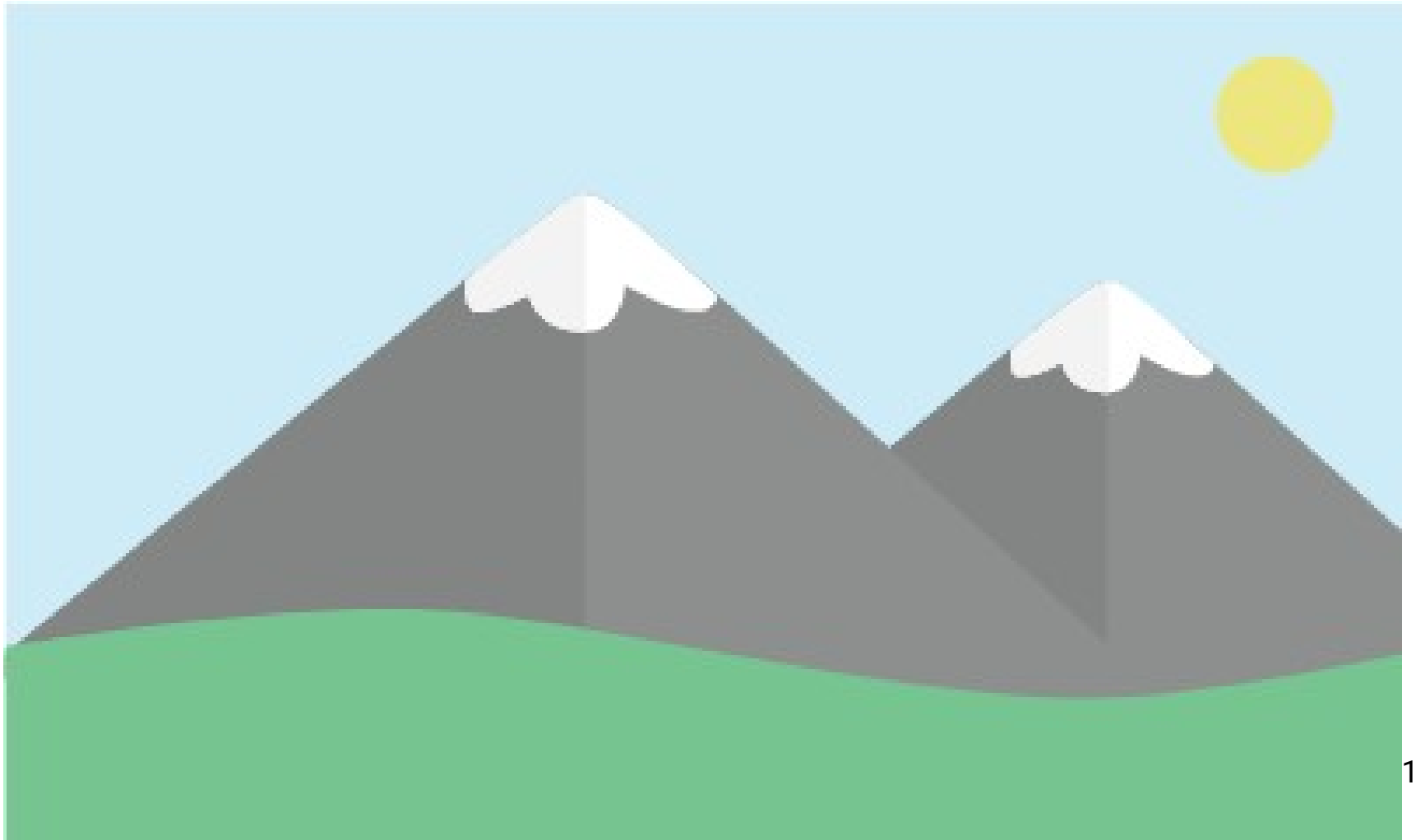
72 72

Computergrafik #1 – Vektor- vs. Rastergrafik

- Rastergrafik
  - Rechteckiges Gitter von Farbelementen
  - Schwer effektiv zu speichern und sinnvoll zu bearbeiten
  - Größenänderung nicht ohne Verlust möglich

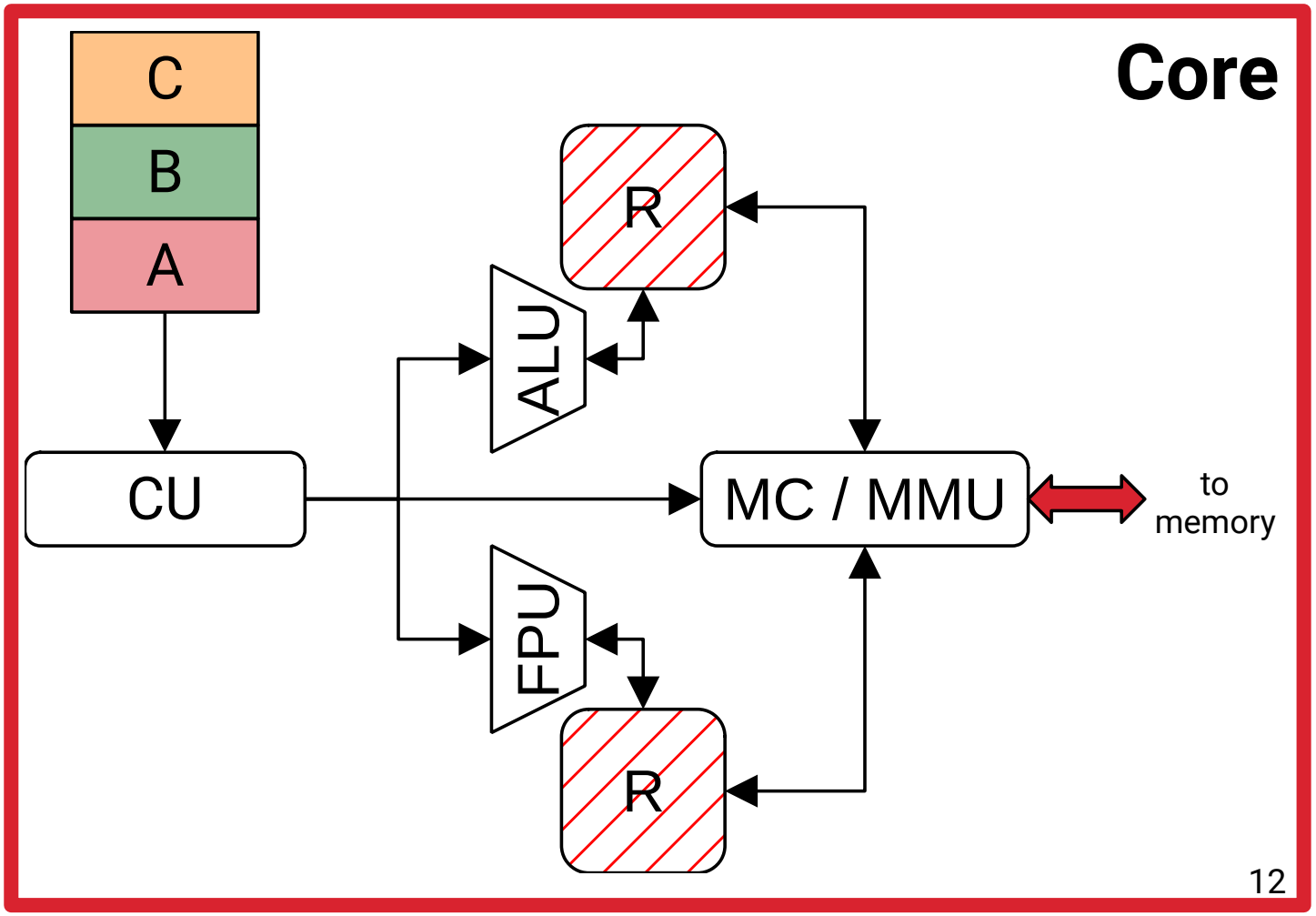


Computergrafik #1 – Vektor- vs. Rastergrafik

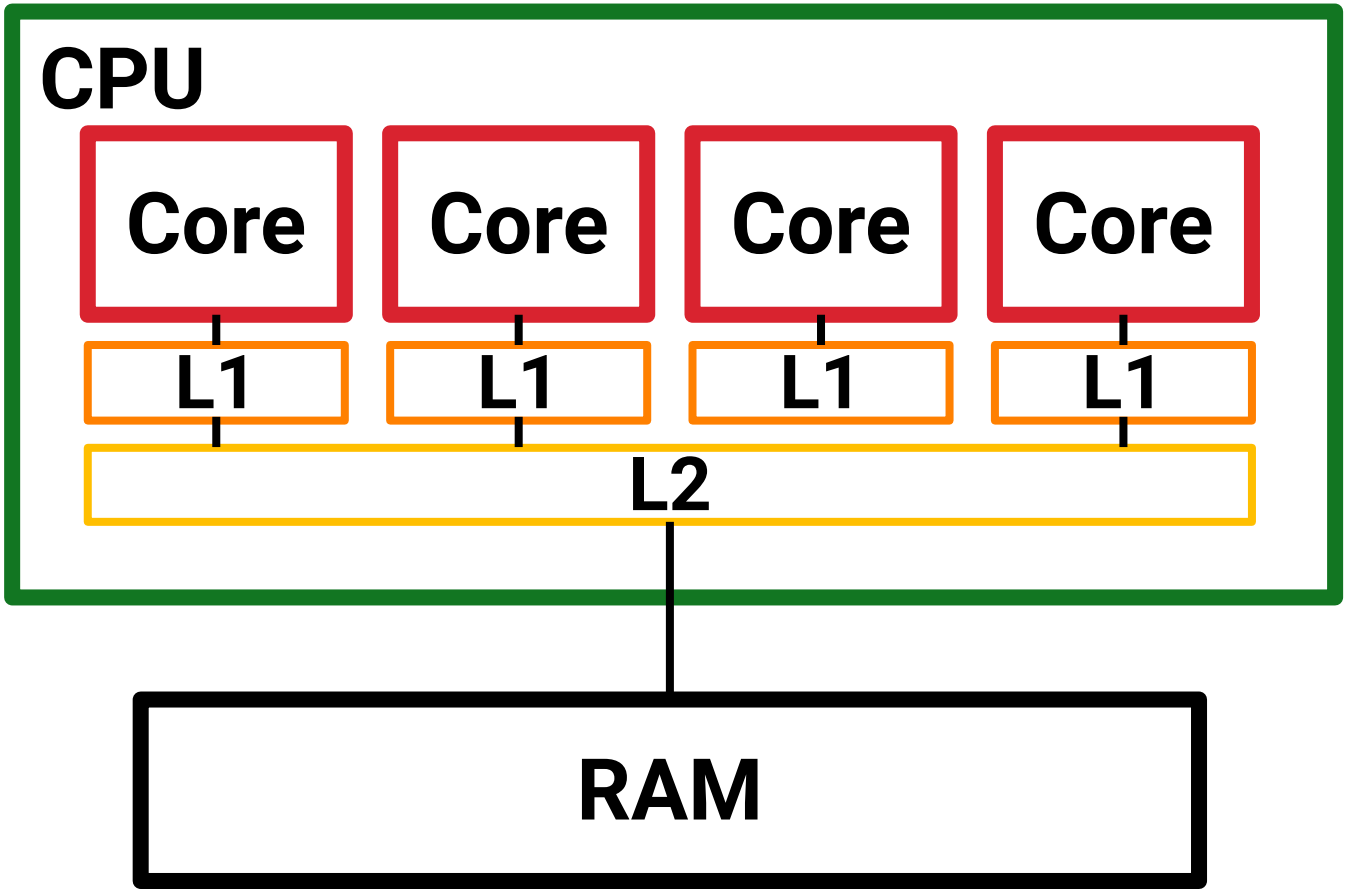


CPU vs. GPU – Central Processing Unit

- CU
  - Steuerung
- ALU / FPU
  - +, -, *, /
  - and, or, xor
  - Registerspeicher
- MC / MMU
  - Speicher

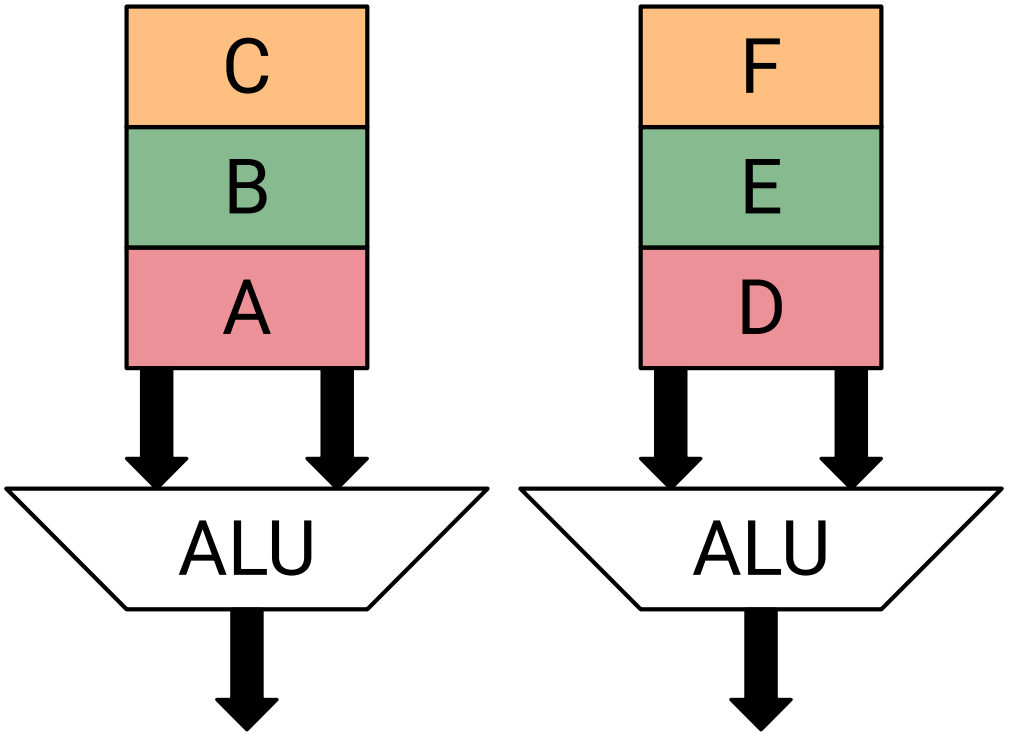


CPU vs. GPU – Central Processing Unit

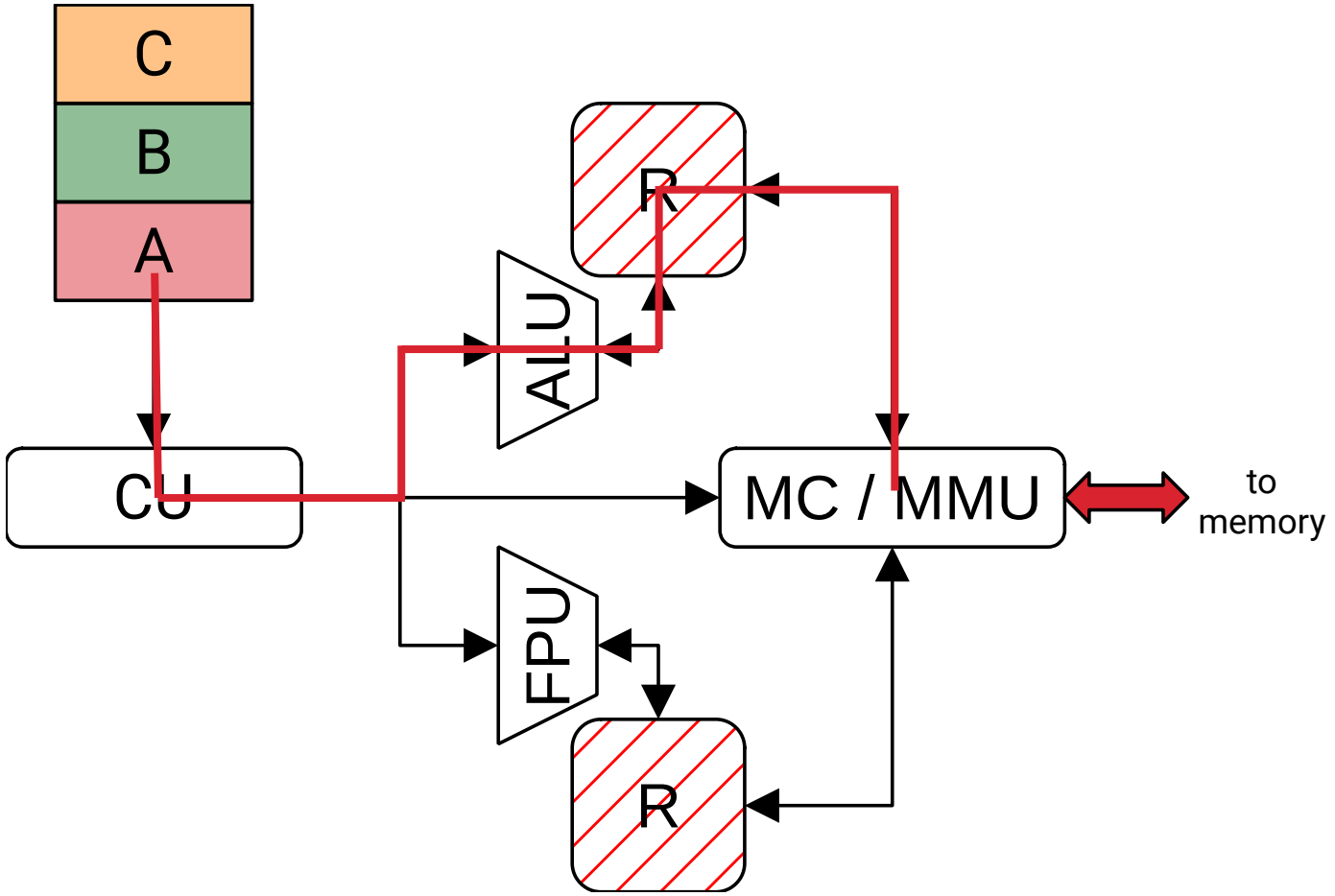


CPU vs. GPU – Pipelining und Superskalarität

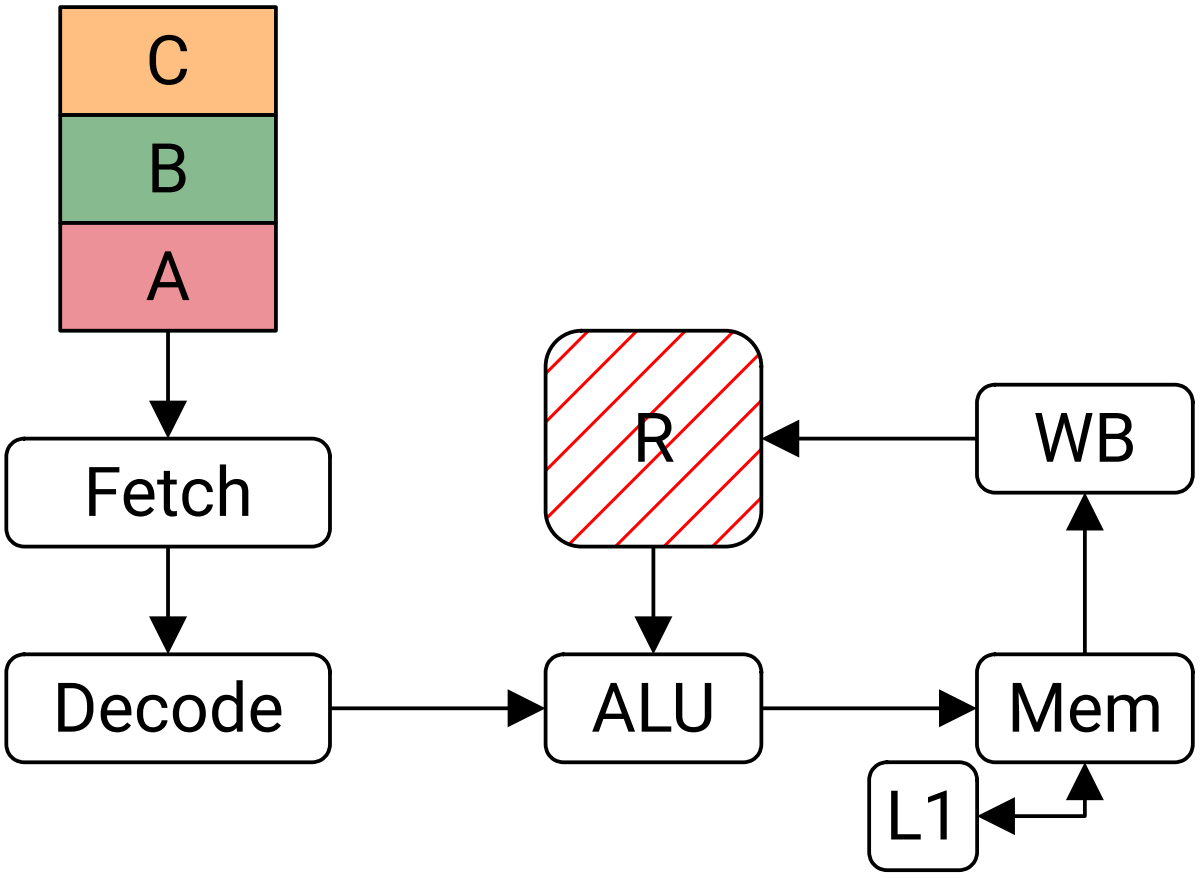
- Parallelisierung (nach Flynn)
  - SISD
  - SIMD
  - MISD
  - MIMD



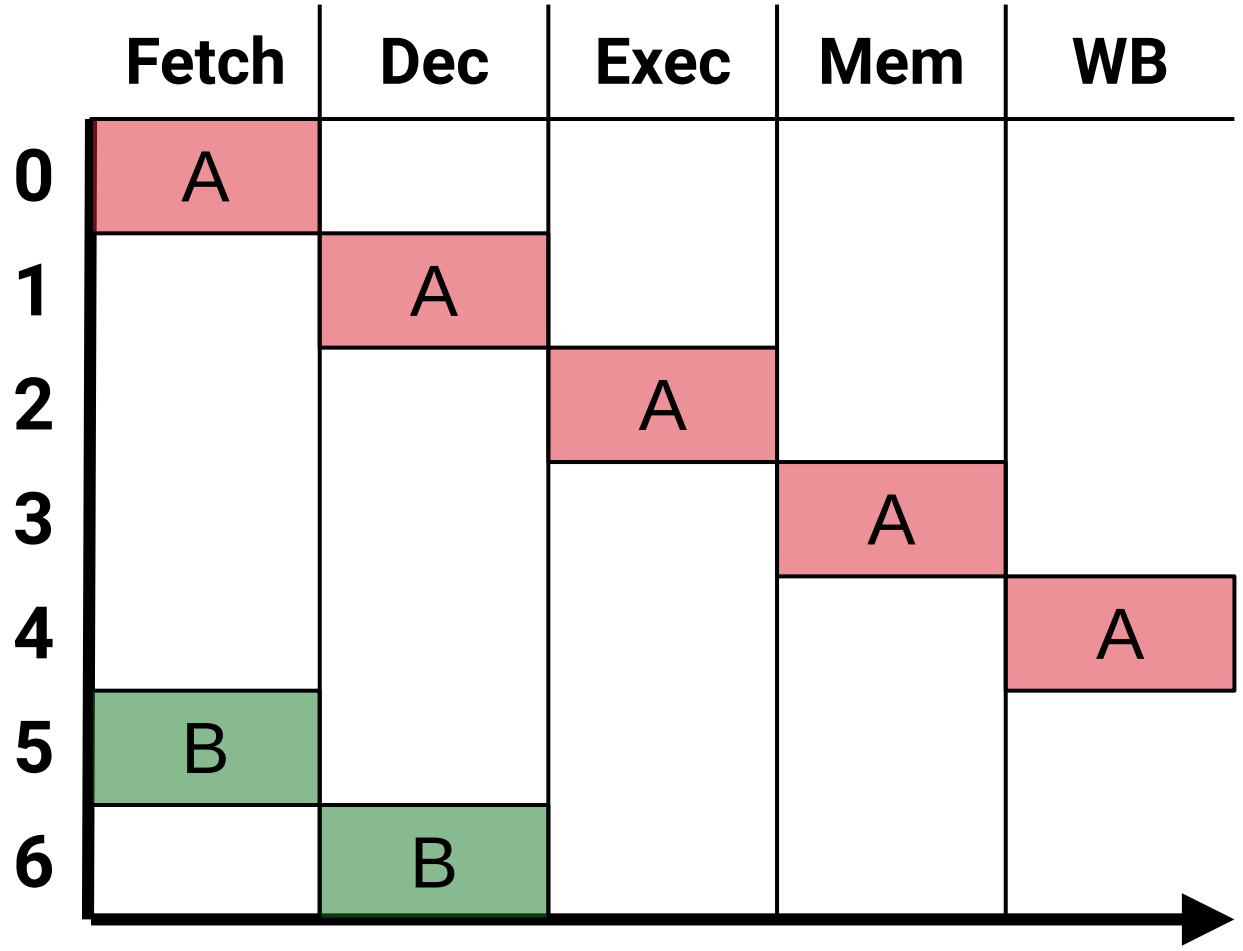
CPU vs. GPU – Pipelining und Superskalarität



CPU vs. GPU – Pipelining und Superskalarität



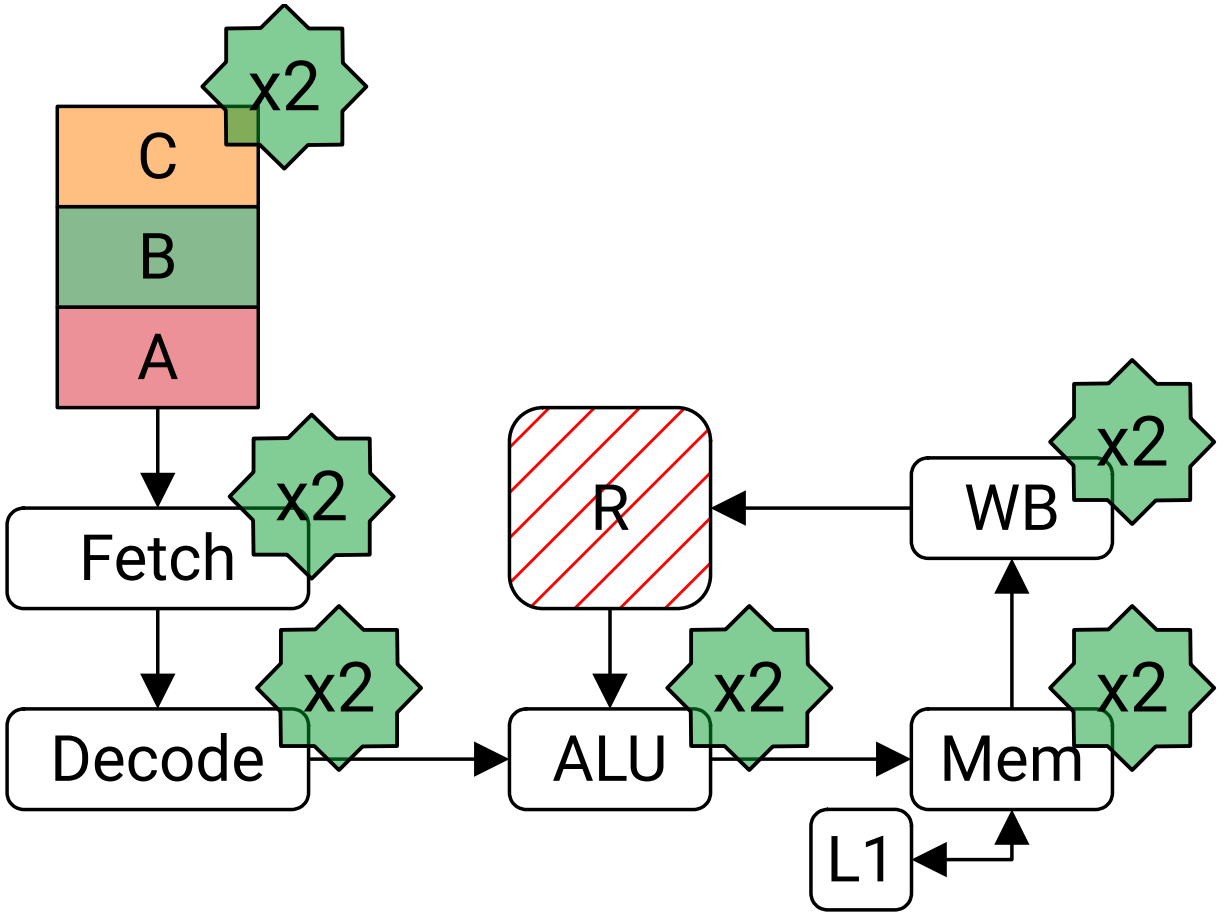
CPU vs. GPU – Pipelining und Superskalarität



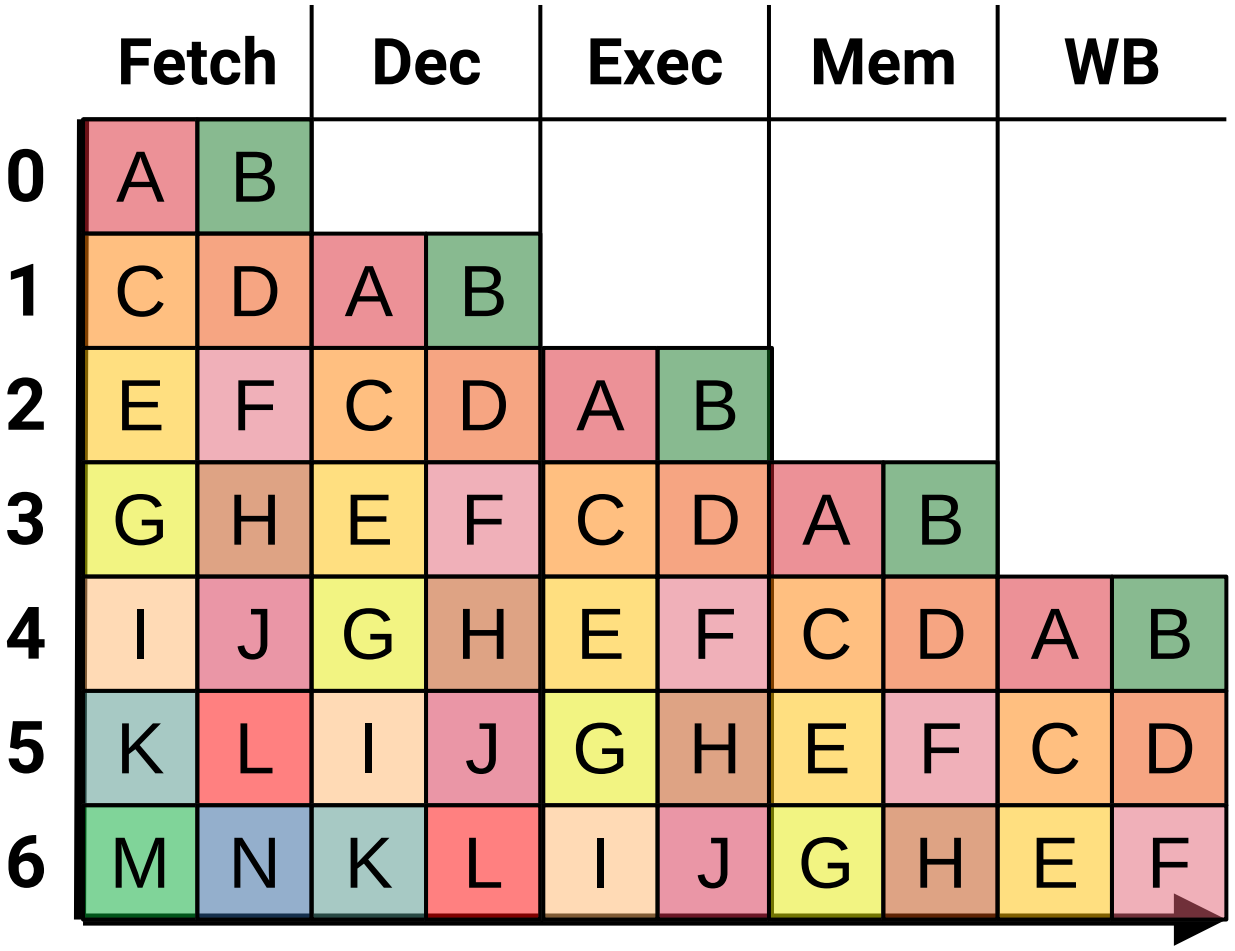
CPU vs. GPU – Pipelining und Superskalarität

	Fetch	Dec	Exec	Mem	WB
0	A				
1	B	A			
2	C	B	A		
3	D	C	B	A	
4	E	D	C	B	A
5	F	E	D	C	B
6	G	F	E	D	C

CPU vs. GPU – Pipelining und Superskalarität

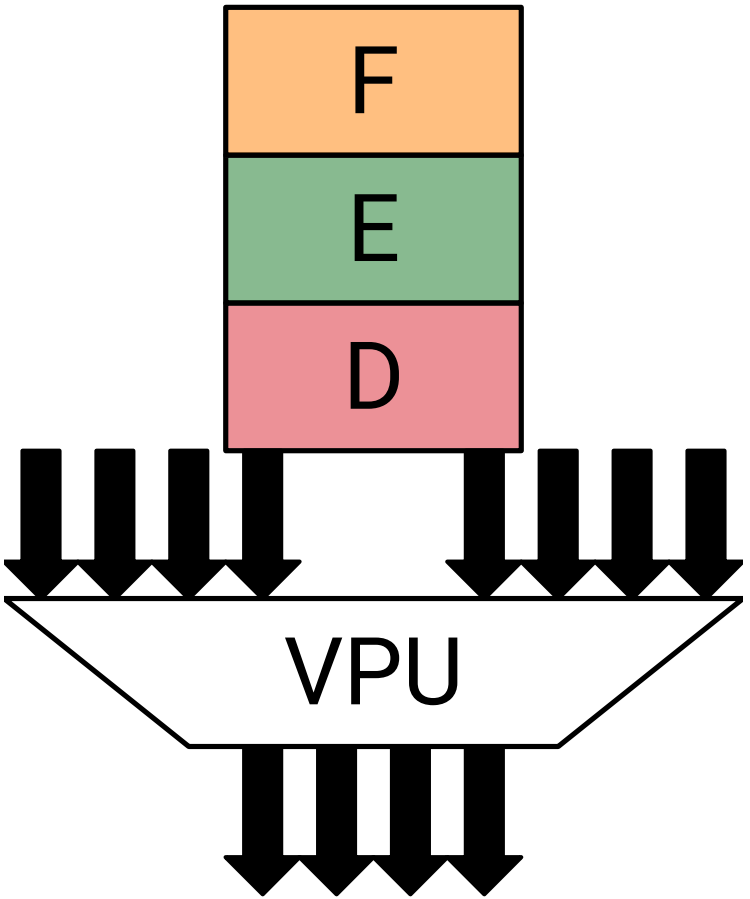


CPU vs. GPU – Pipelining und Superskalarität

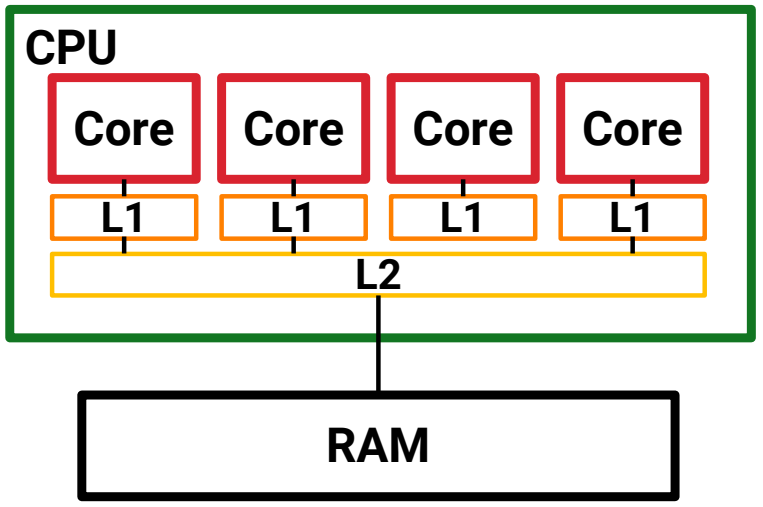


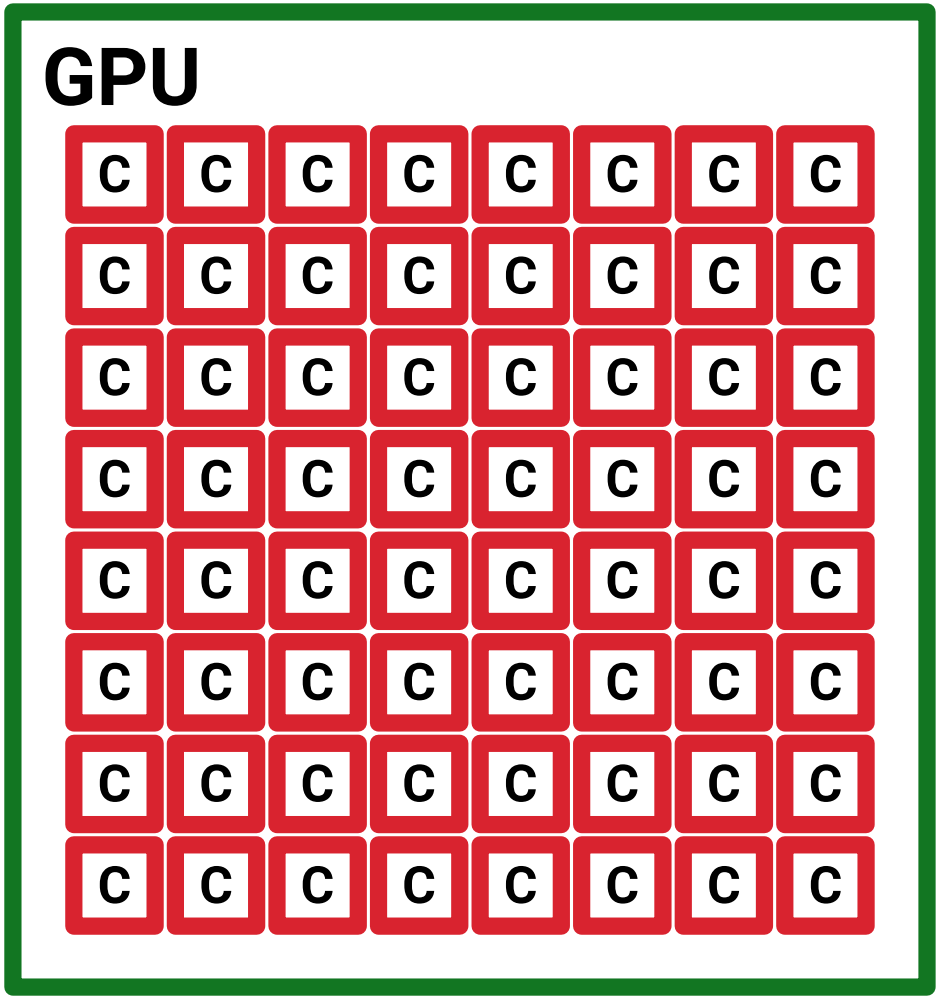
CPU vs. GPU – SIMD und Vectorprocessing

- SIMD
  - Bsp:
    - Auf viele Daten
      soll 5 addiert werden
      → Spielfigur bestehend aus 1000
      Eckpunkten wird um 5 nach rechts
      verschoben



CPU vs. GPU – SIMD und Vectorprocessing

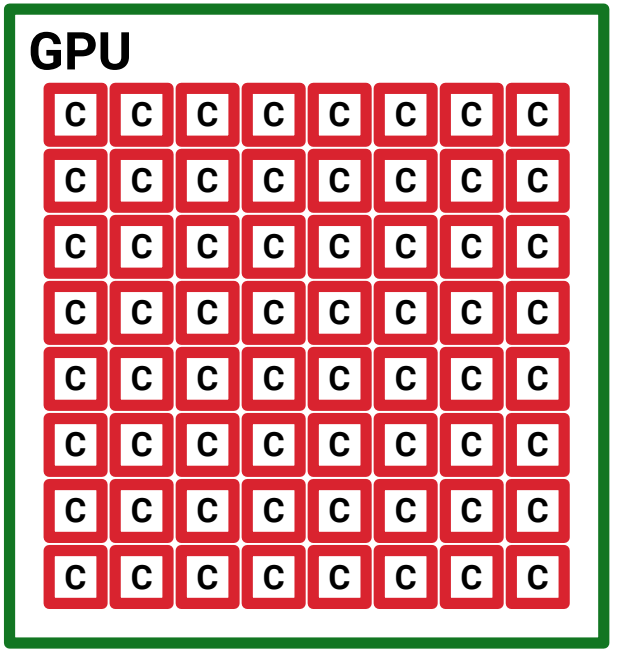




CPU vs. GPU – SIMD und Vectorprocessing

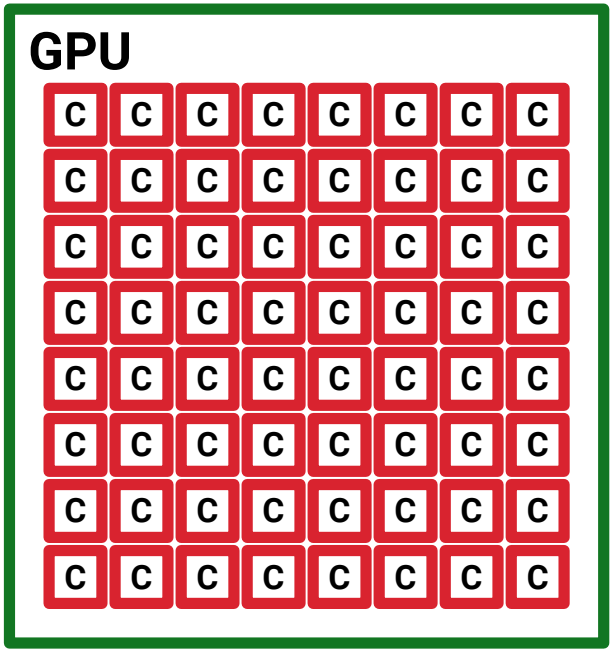
- NVIDIA RTX 3090
  - 5248 Cores
    → 10496 als Marketing, weil intern noch etwas verdoppelt wurde pro Core
    → verpackt in 64 Stream-Multiprozessoren

- AMD Radeon RX 6900 XT
  - 5120 Cores
    → verpackt in 80 Stream-Multiprozessoren



CPU vs. GPU – SIMD und Vectorprocessing

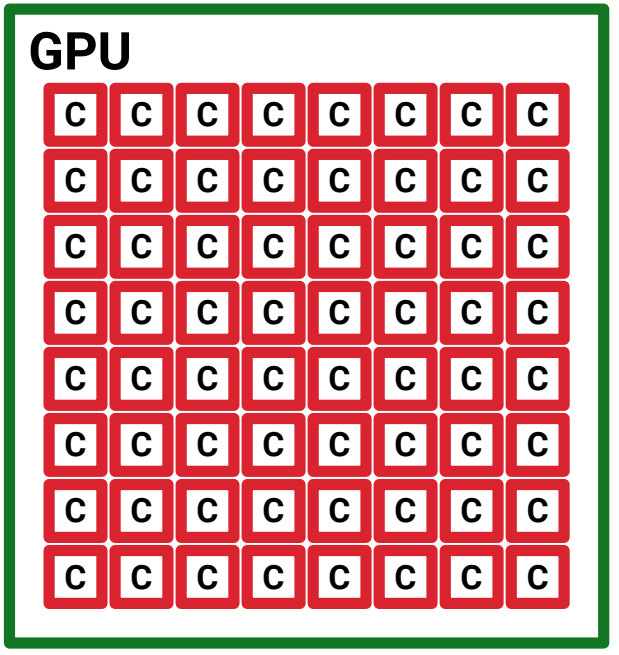
- Graphics Processing Unit
  - Cores werden auch Shader genannt
  - Nicht jeder Core tut das Gleiche
    - Vertex Shader
    - Tesellation Shader
    - Geometry Shader
    - Fragment Shader
    - Rasterisierung
  - Hierbei sind einige fest vorgegeben und andere können durch Shader programmiert werden



CPU vs. GPU – SIMD und Vectorprocessing

- Shader
  - Sind sowohl also auch die Hardware-Cores, als auch die Programme, die wir auf den Hardware-Shadern noch extra laufen lassen können

- Graphics- und Computeshader



Ende

Fragen?