

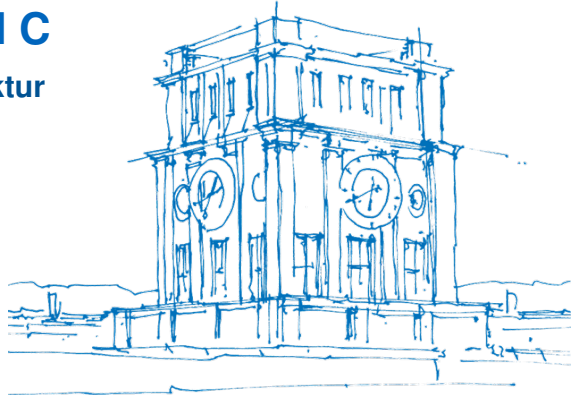
Übung 06: SIMD in x86 und C

Grundlagenpraktikum Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

24. Mai 2024



Keine Garantie für die Richtigkeit der Tutorfolien: Bei Unklarheiten/Unstimmigkeiten haben VL-Folien Recht!

- **Single Instruction, Multiple Data**
- Vektorisierung von Instruktionen, parallele Ausführung auf mehreren Datenströmen
- Achtung: SIMD ist nicht immer sinnvoll / einfach zu implementieren!
- XMM-Register mit 128 Bit (16 Byte), bspw.
 - ☐ 16 chars (1 Byte)
 - ☐ 4 ints (4 Byte)
 - ☐ 4 floats (4 Byte)
 - ☐ 2 doubles (8 Byte)
- Naming Convention: p für '**p**acked' (vgl. 'scalar')
- Achtung: Bei SIMD-Instruktionen wird 16-Byte-Alignment gefordert!
→ `movaps` vs. `movups`

- Nutzung von sog. *Intrinsics* (Liste siehe Intel Intrinsics Guide)
- im Praktikum werden SSE/SSE2/AVX verwendet → muss von der Zielarchitektur unterstützt werden
- neue Datentypen `__m128i`, `__m128d`, ...
- Funktionen für einzelne SIMD-Operationen
- `#include <immintrin.h>` nicht vergessen!
- Vergleiche in SIMD ergeben Bitmasken

Fragen?

- Zulip: „GRA Tutorium - Gruppe 20“ bzw. „GRA Tutorium - Gruppe 22“
- Praktikumswebsite
- x86 instruction reference by Félix Cloutier
- Intel Intrinsics Guide

Übung 06: SIMD in x86 und C

Grundlagenpraktikum Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

24. Mai 2024

