



# *GPU Programming*

*Prof. Dr.-Ing. Richard Membarth*

15.03.23



- Computer Science background
  - Diploma and PhD from Friedrich-Alexander University Erlangen-Nürnberg
  - Postdoc at the Intel Visual Computing Institute (IVCI) at Saarland University
  - Senior researcher at the German Research Center for Artificial Intelligence (DFKI) in Saarbrücken
  - Research professorship for System on a Chip and AI for Edge Computing at THI
- Mostly working on
  - GPU computing
  - Parallel computing
  - Domain-specific languages
  - Compilers

# About Me

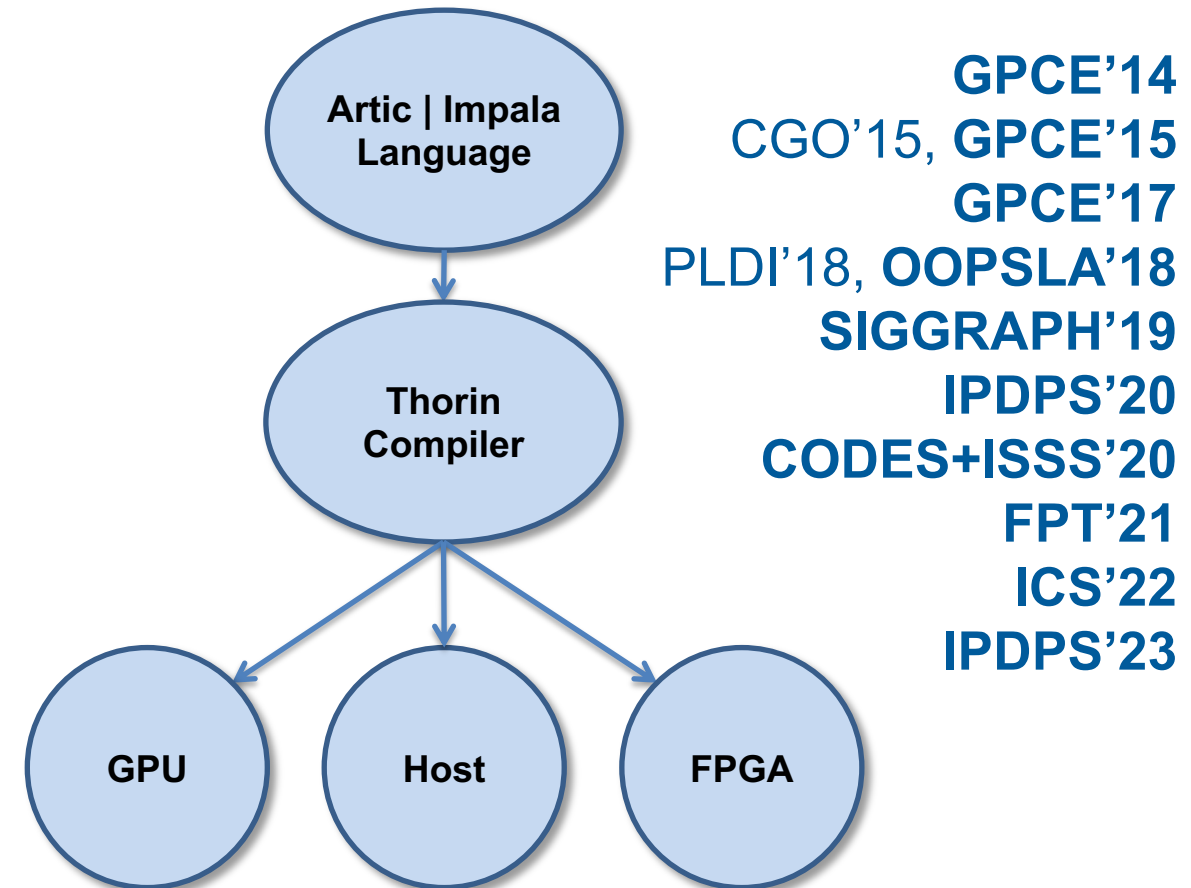
## AnyDSL Compiler Framework



<https://anydsl.github.io>

<https://github.com/AnyDSL>

- Artic | Impala language: Rust dialect
  - Functional & imperative
  - Filters to guide specialization
  - Triggered code generation
- Thorin compiler
  - Built-in partial evaluator
  - Multi-target code generation
    - Host: vectorization & parallelization
    - GPU: CUDA, OpenCL, AMDGPU, NVVM
    - FPGA: HLS



# About Me

*AnyDSL: Selected Application Domains*



## Stincilla: Image Processing [OOPSLA'18]



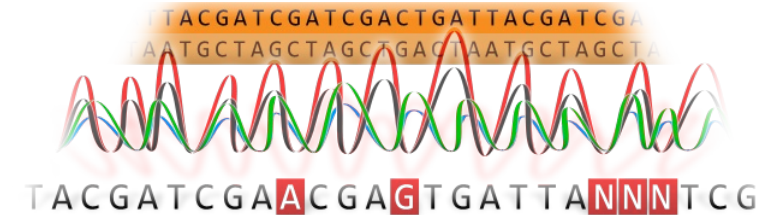
Blur (OpenCV): +40% (CPU) +50% (GPU)  
Blur (Halide): +12% (CPU) +7% (GPU)  
Harris (Halide): +37% (CPU) +44% (GPU)  
Runs also on FGPA!

## Rodent: Ray Tracing [SIGGRAPH'19]



Embree (CPU): +1% to +23% (WF)  
OptiX (GPU): +2% to +31% (MK)  
OptiX (GPU): +29% to +42% (WF)  
Runs also on ARM and AMD GPU!

## AnySeq: Genome Sequence Alignment [IPDPS'20]



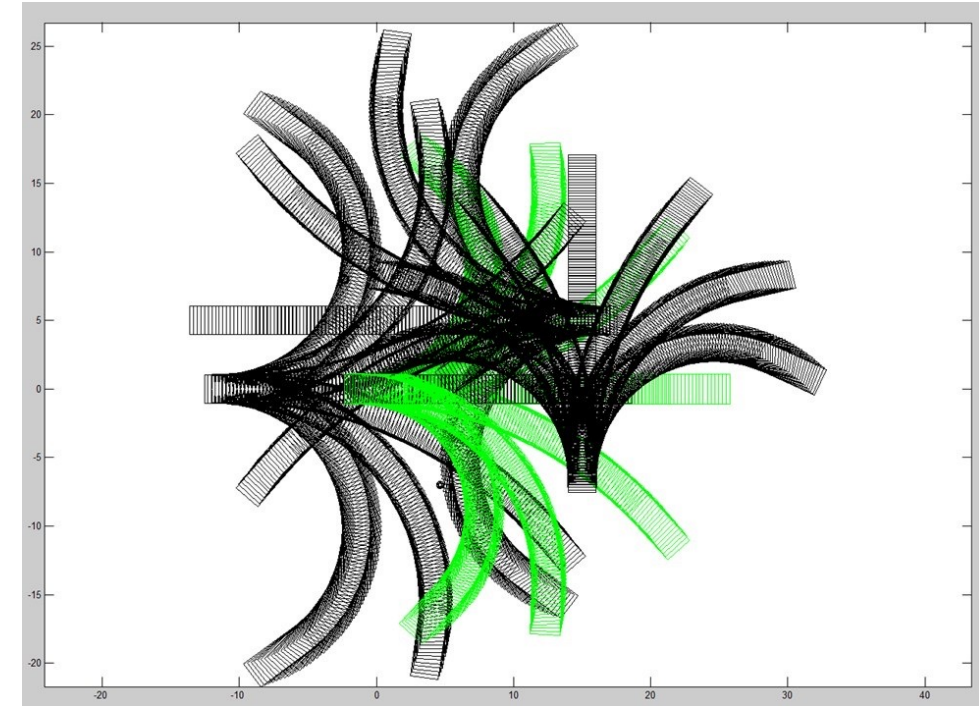
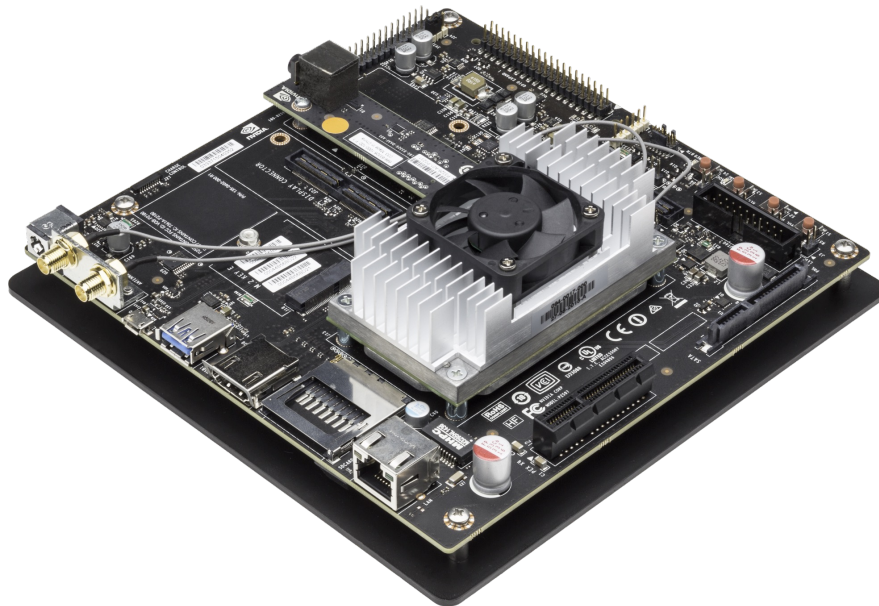
SeqAn (CPU): -7.2% to +7.8%  
Parasail (CPU): -2.1% to +90.0%  
NVBIO (GPU): +5.3% to +10.4%  
Runs also on FPGA!

## AnySeq/GPU [ICS'22]

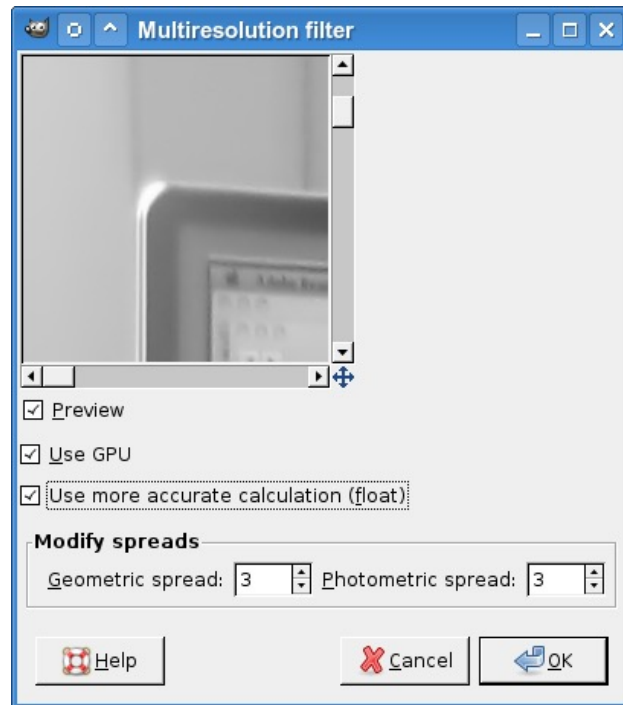
Median speedup of 19.2x vs. fastest  
available GPU implementations!



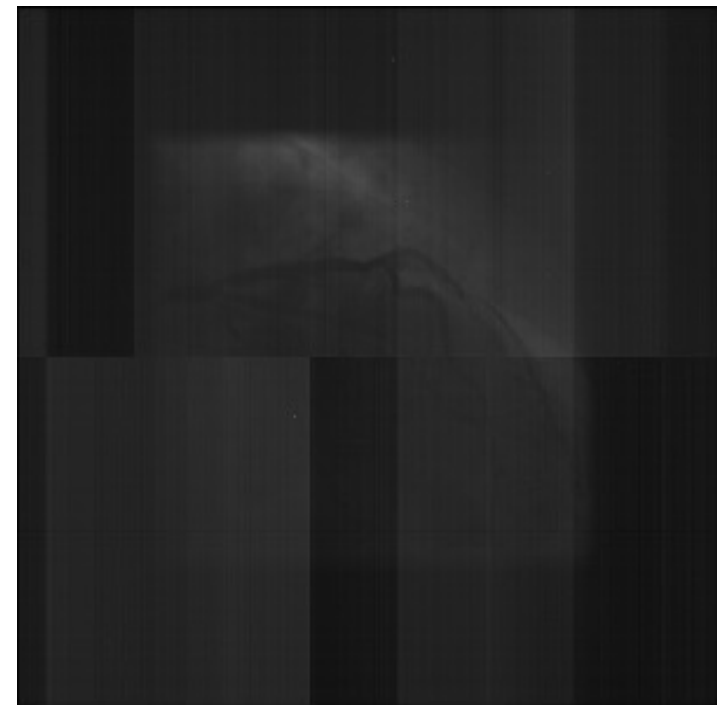
- Collision Avoidance & Crash Impact Point Optimization
  - 26 million hypothesis combinations
  - 6 mins Matlab → 36 ms Jetson TX1  
→ 15 ms Drive PX2



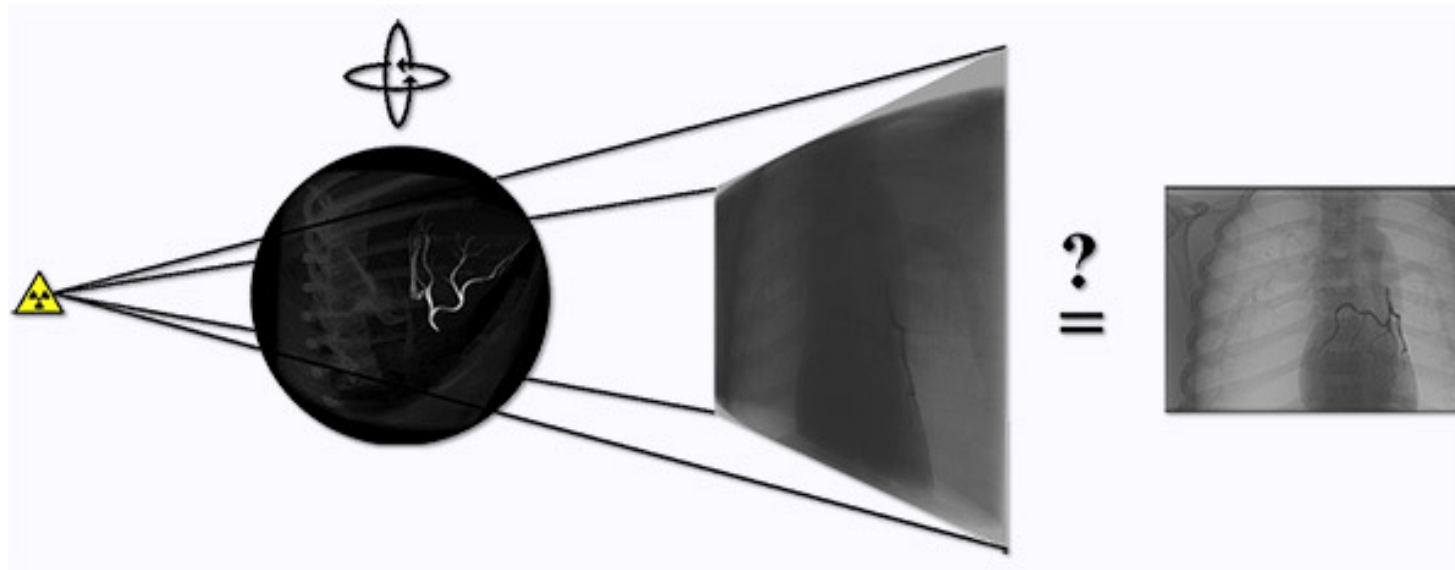
## ■ CUDA Challenge 2008



## ■ Detector Defect Correction on Graphics Processors

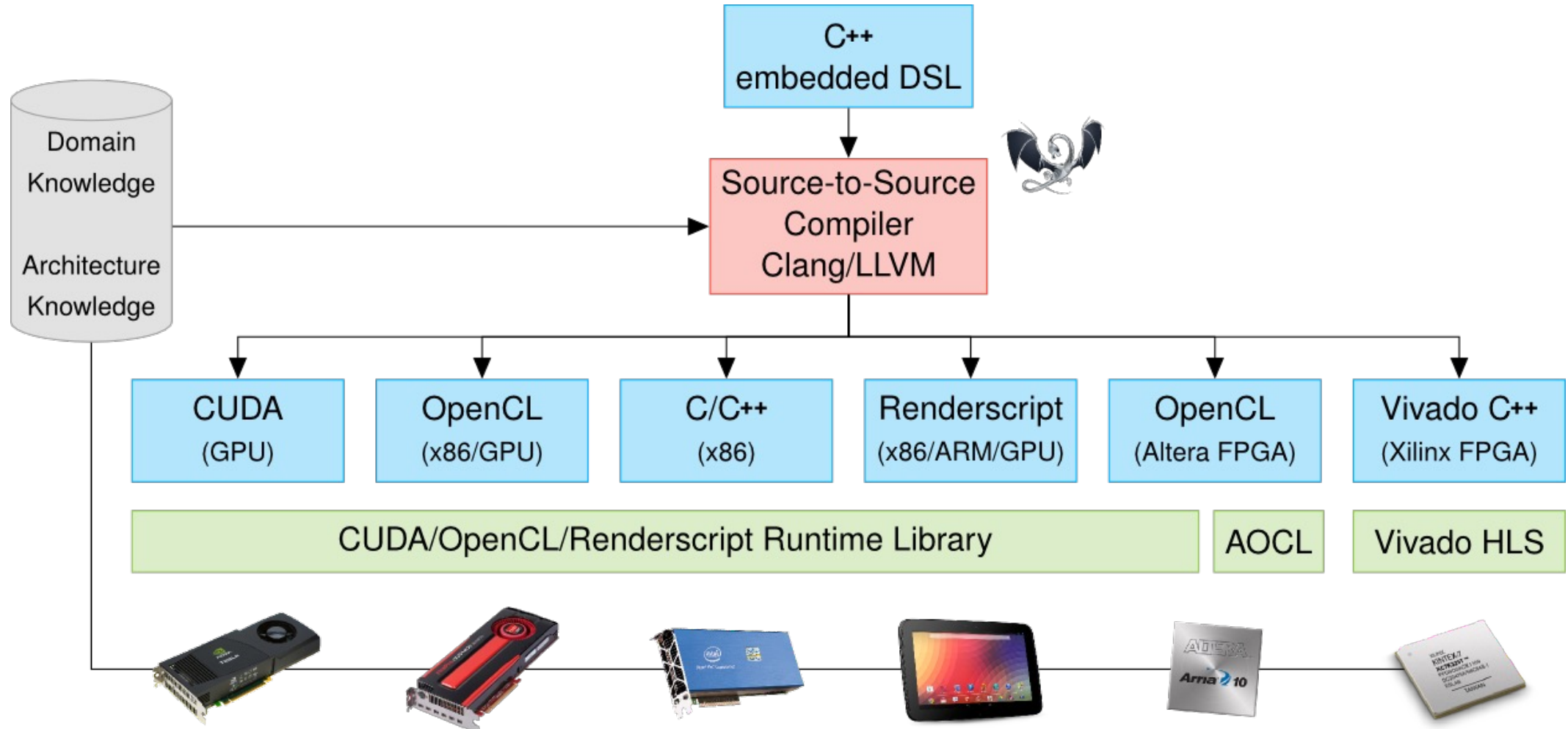


- 2D/3D Image Registration on Multi-core and Many-core Architectures



# About Me

## Hipacc: A Domain-Specific Language and Compiler for Image Processing







## ■ Objectives

- Understand the architecture of current graphics processing units (GPUs)
- Map applications and algorithms to the parallel execution units of a GPU
- Programming of NVIDIA GPUs using CUDA
- Optimization and debugging of GPU applications



- Course syllabus
  - History of GPU programming
  - CUDA programming model
  - The CUDA API in detail
  - Parallel programming patterns
  - Memory hierarchy
  - Performance optimization case study
  - Hardware / warp scheduling
  - Advanced programming techniques
  - Related programming models
  - AnyDSL compiler framework



- Tutorial
  - Development environment
  - Toolchain
  - Debugging
  - Assignments

- Lecture (Wahlpflichtfach)
  - Automatisiertes Fahren und Fahrzeugsicherheit (AUF)
  - Cloud Applications und Security Engineering (CASE)
  - Künstliche Intelligenz (KI)
  - Lecture in German
- ECTS
  - 5 credit points
- Lecture
  - Mon, 09:55 – 11:25, G308
  - Wed, 11:35 – 13:05, K010
- Exam
  - Oral exam (20 minutes)
- Consultation hours
  - Wed, after the lecture

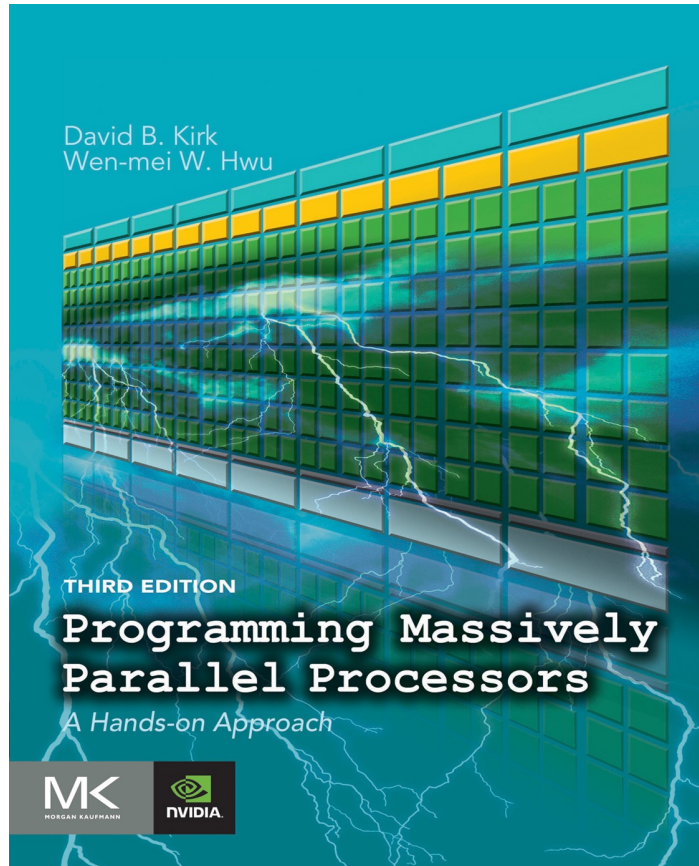


- Monday lecture will be in the CIP pool G308
  - OS: Ubuntu 22.04 LTS
  - Intel Core i9-10900K CPU
  - NVIDIA Quadro RTX 4000
- Programming exercises using NVIDIA CUDA
  - CUDA C/C++
  - CMake
  - OpenACC / OpenMP
- Fallback: remote access to a NVIDIA GTX 970 via ssh

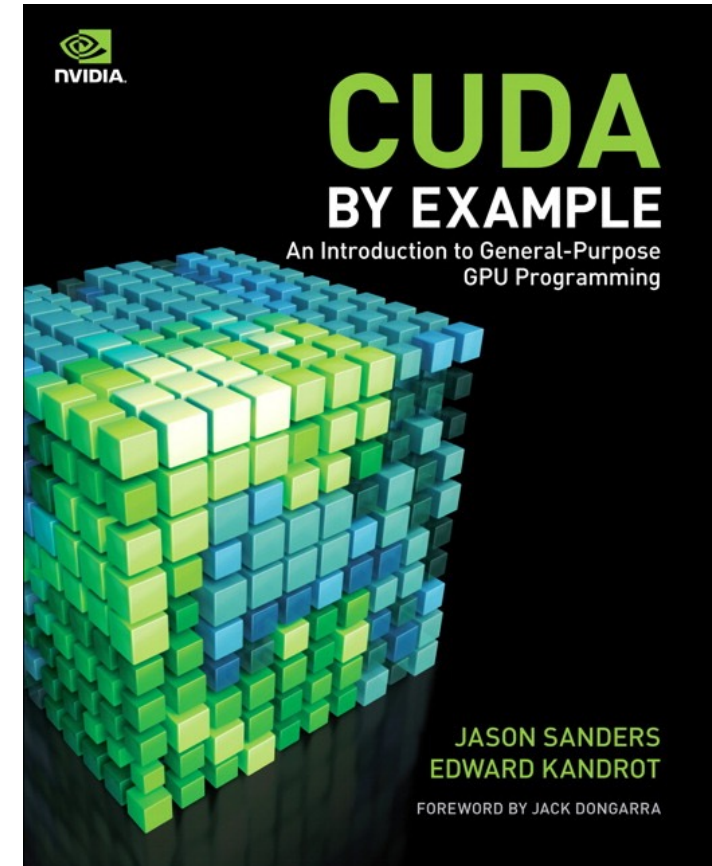




- Slides will be provided on Moodle after the lecture
- Exercises will be provided on Moodle and worked on during the Monday lecture in G308
- Discord channel for questions and discussions
- Questions concerning the lecture: Richard Membarth ([richard.membarth@thi.de](mailto:richard.membarth@thi.de))



Programming Massively Parallel Processors  
Kirk, Hwu



CUDA by Example  
Sanders, Kandrot

- Lecture based on material by
  - GPU Programming lecture at Saarland University
  - Real-Time Graphics 2 lecture at TU Graz
  
- NVIDIA teaching kit
  - 30% discount and free shipping for “Programming Massively Parallel Processors” from <https://www.store.elsevier.com>
  - Ask for the discount code!