GA Gauß-Allianz

SPONSORED BY THE

Federal Ministry of Education and Research

High Performance Computing

Rekonfigurierbare Architekturen für HPC

Viktor Achter (RRZK), Florian Mayer (FAU)

Universität zu Köln

Fraunhofer SCAI

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

ZUSE INSTITUTE BERLIN

Assoziierter Partner:
Intel Deutschland GmbH

Motivation / Ziele

High Performance Computing

- Erweiterung der technologischen Alternativen für HPC
  - *Acceleratoren sind eine effektive Alternative zu CPUs*
  - *Pro: höhere Effizienz bei passender Problemstellung*
  - *Cons: hoher Spezialisierungsgrad*
    - → *möglichst große Auswahl unterschiedlichster Hardware Architekturen - Auswahl passend zur Problemstellung*

- Reduktion der Einstiegshürden
  - *Einstiegshürde zu diversen Beschleunigern gering, wenn diese mit OpenMP genutzt werden können.*
  - *OpenMP Pragmas Standard in HPC.*

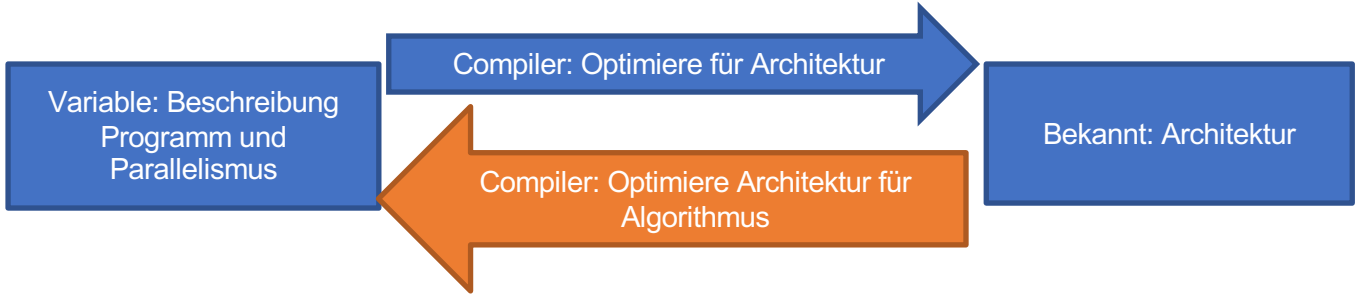
SPONSORED BY THE
Federal Ministry of Education and Research

Optimierungsprobleme

High Performance Computing

- Normaler Compiler:

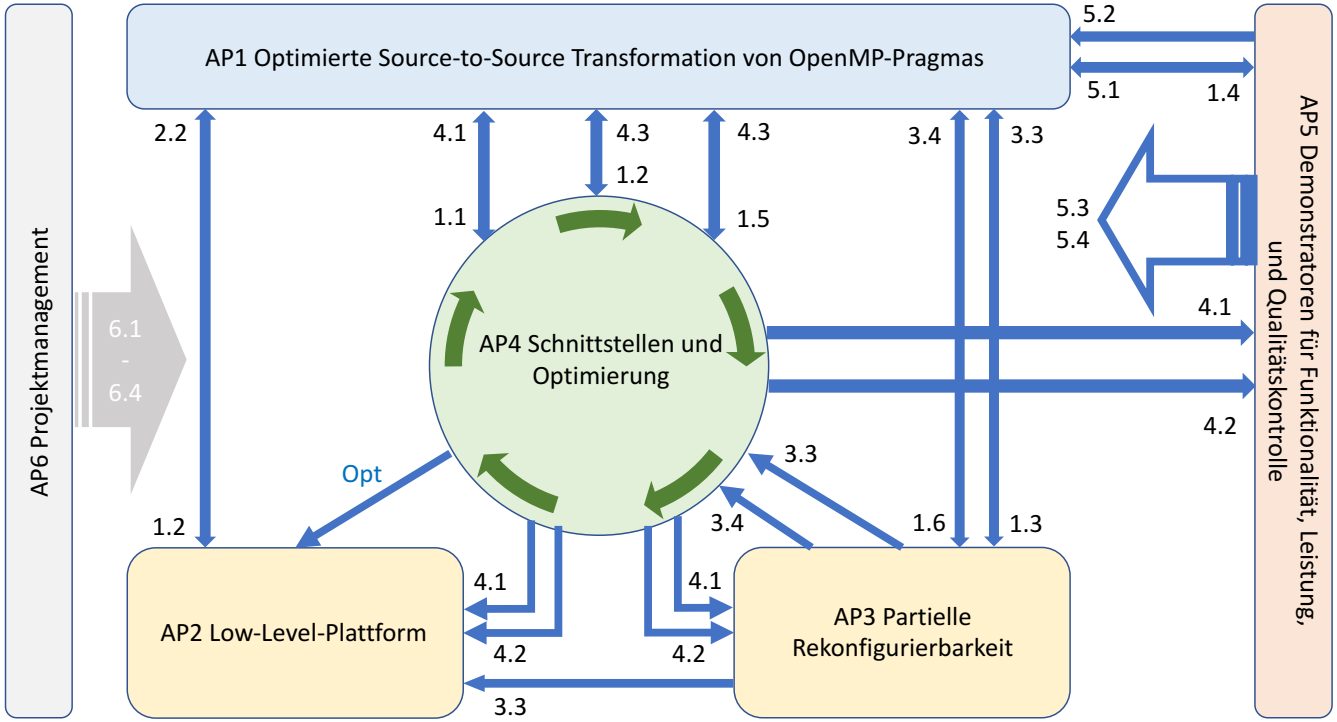
- ORKA-HPC Framework:



SPONSORED BY THE
Federal Ministry of Education and Research

Überblick Arbeitspakete

High Performance Computing



SPONSORED BY THE
Federal Ministry of Education and Research

GA Gauß-Allianz

orkan

High Performance Computing

SPONSORED BY THE

Federal Ministry of Education and Research

AP2+3 Low Level Platform

Universität zu Köln

Fraunhofer SCAI

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

ZUSE INSTITUTE BERLIN

Assoziierter Partner:
Intel Deutschland GmbH

Bereitstellung: Low Level Plattformen

High Performance Computing

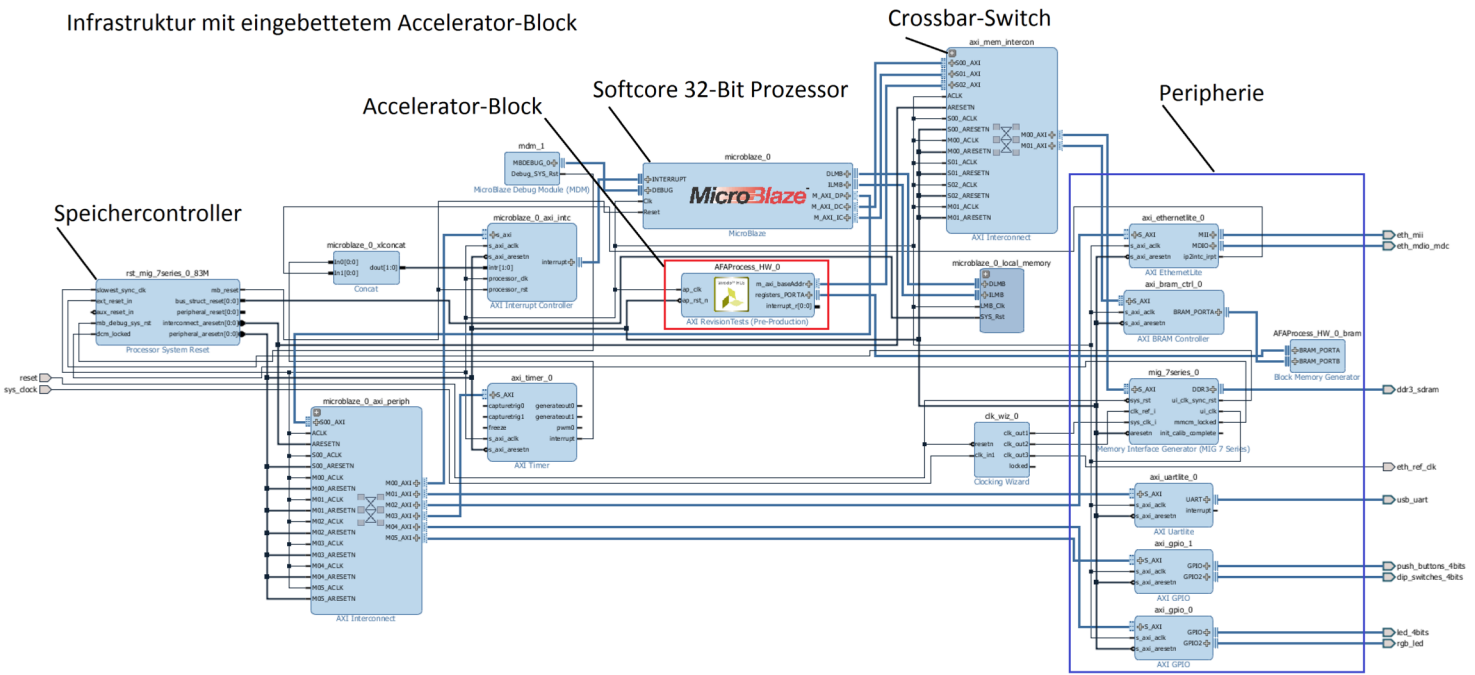
Anforderungen:

- Unterstützung einer repräsentativen FPGA-Karte
  - Xilinx, Altera
- Erstelle LL-Plattformen mit Optimierungen für unterschiedliche Code Charakteristiken
  - Maximale Speicherbandbreite
  - Cache
  - Streaming
- Jeweils optional mit Softcore und Debugging Funktionen
- Partielle Rekonfiguration
  - Nutzung größerer Chipflächen vs. Reprogrammierungszeiten

SPONSORED BY THE
Federal Ministry of Education and Research

Umfangreiches Design mit softcore Prozessor, Debugging, etc..

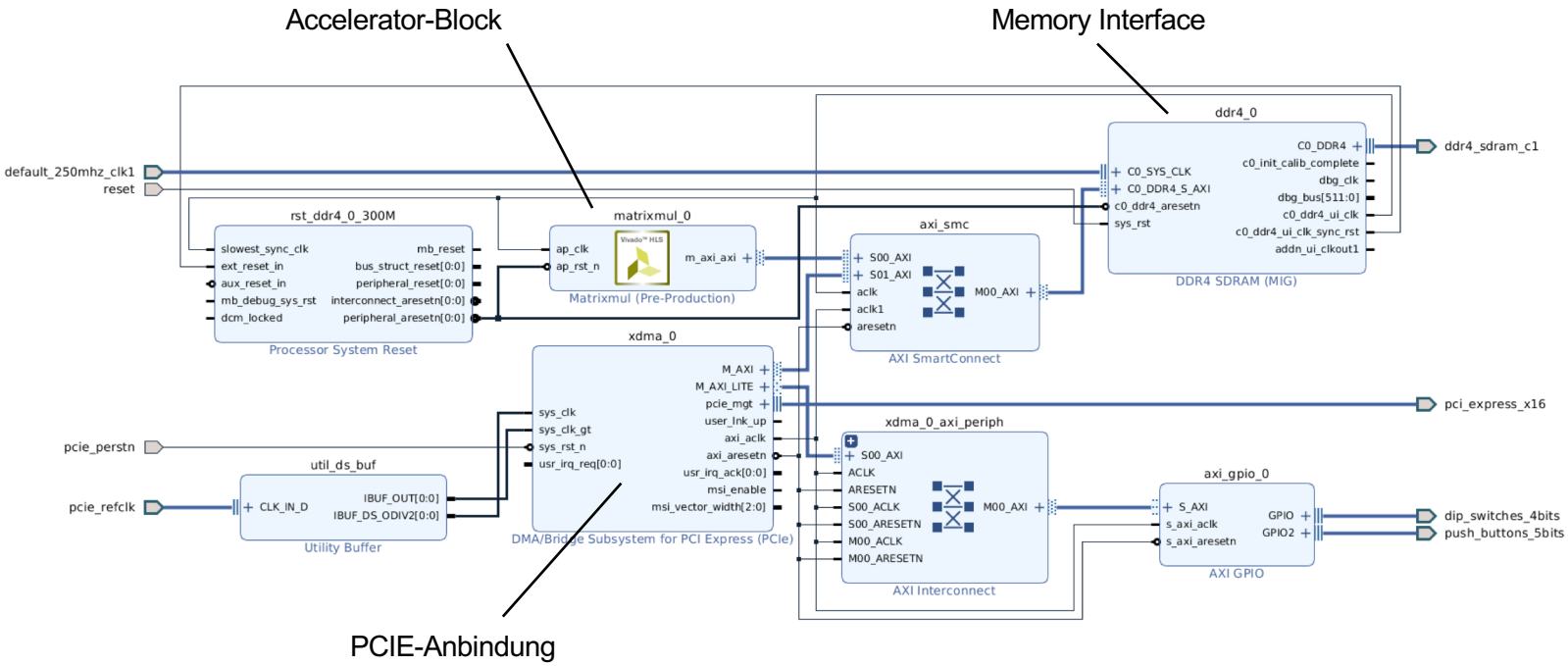
High Performance Computing



SPONSORED BY THE
Federal Ministry of Education and Research

Minimalistisches Design mit Speicherinterface und Pcie

High Performance Computing



-> Viel Platz für Accelerator-Block/Blöcke

SPONSORED BY THE
Federal Ministry of Education and Research

GA Gauß-Allianz

SPONSORED BY THE

Federal Ministry of Education and Research

High Performance Computing

AP4 Schnittstellen und Optimierung

Universität zu Köln

Fraunhofer SCAI

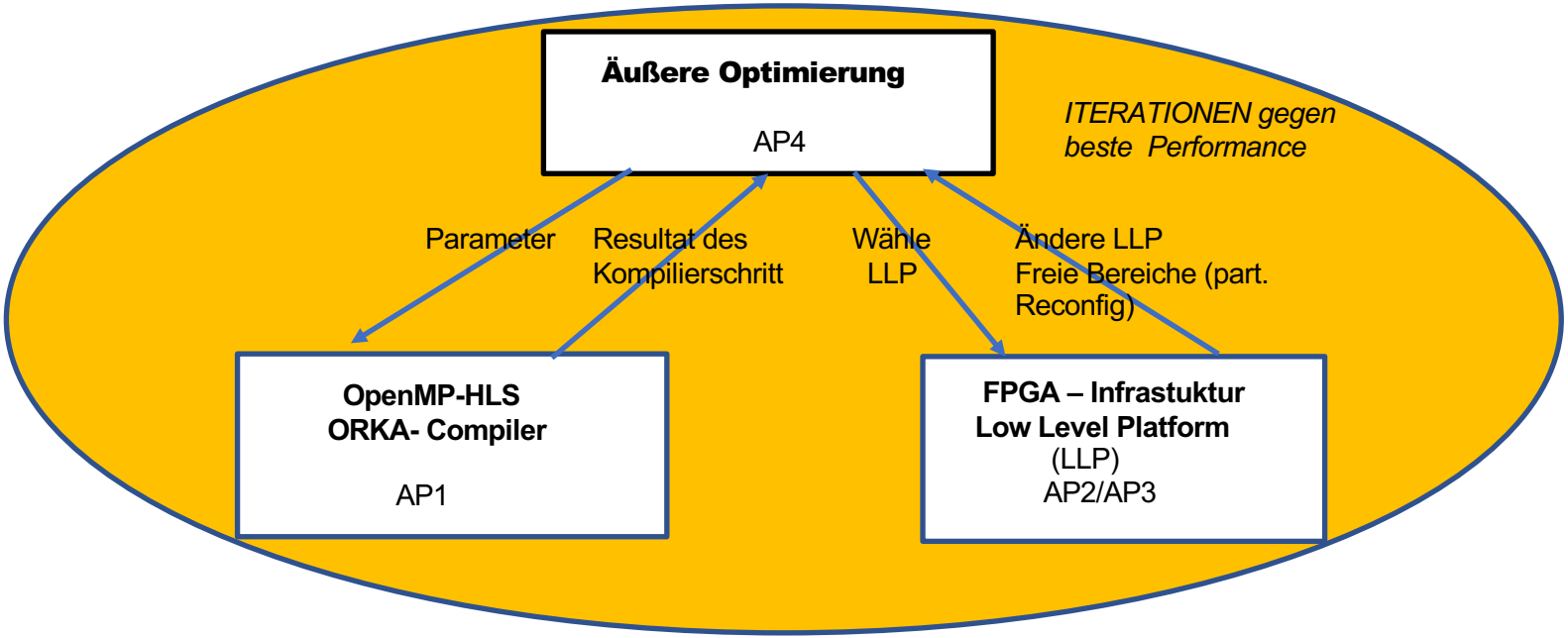
FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

ZUSE INSTITUTE BERLIN

Assoziierter Partner:
Intel Deutschland GmbH

Optimierung

High Performance Computing

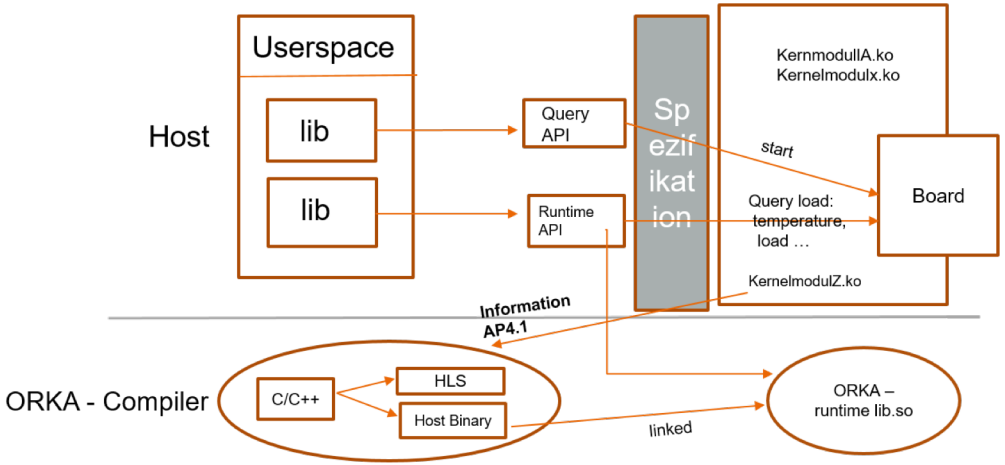


SPONSORED BY THE
Federal Ministry of Education and Research

Unterstützung Laufzeitumgebung

High Performance Computing

- Informationsschnittstelle
  - 1te Dokumentation und Schnittstellendefinition
- Runtime Library in Arbeit



SPONSORED BY THE
Federal Ministry of Education and Research

GA Gauß-Allianz

orka

High Performance Computing

SPONSORED BY THE

Federal Ministry of Education and Research

AP 5 Demonstrator Suite

Universität zu Köln

Fraunhofer SCAI

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

ZUSE INSTITUTE BERLIN

Assoziierter Partner:
Intel Deutschland GmbH

Demonstrator Application Suite

High Performance Computing

Anforderungen und Anwendungsfälle:

- Representation von Rechenmustern und OpenMP Annotationen
  - HPC und Data Analytic Workloads
  - Compute und Kontrollflüsse
- Demonstration von FPGA-bezogenen Optimierungen
  - Variable Repräsentation von Fixed und Floating-Point Daten über IEEE Standards hinaus (Double, Single, Half-Precision)
  - Ressourcen Optimierung
- Benchmarking und Qualitätssicherung
  - Mini-App und Kerneltypen von Demonstratoren
  - Performance und Ressourcennutzungs-Benchmarks
  - Referenzimplementierung auf CPUs und GPUs unter Nutzung von State-of-the-Art Features (e.g. SIMD, Multi-threading)

SPONSORED BY THE
Federal Ministry of Education and Research

Beispielanwendungen und Kernels

High Performance Computing

Code / Domain	OpenMP Feature	Anzahl	Anmerkungen
SWCluster / Ising Model	omp parallel omp for [collapse] omp parallel for [collapse] omp parallel for reduction(+) omp [parallel] simd [for] [reduction(+)]	(2x) (2x) (2x) (1x) (3x)	CPU: C++ GPU: CUDA
AMGmk (CORAL) / Solver	OpenMP code but no OpenMP 4.0 / 4.5 features		Platzhalter für SCAI AMG solver
DM-HEOM Kernel / open quantum systems	omp parallel for omp simd	(17x) (7x)	CPU: C++ / OpenCL GPU: OpenCL
Genomics: Deduplication			*Variable Datenrepräsentation*
Deep Learning Kernel Suite			*kompakte DL Kernel*
HLS Kernels			*Performance-Vergleich*

SPONSORED BY THE
Federal Ministry of Education and Research

GA Gauß-Allianz

orka

SPONSORED BY THE

Federal Ministry of Education and Research

High Performance Computing

AP1 - Optimierte Source-to-Source Transformation von OpenMP-Pragmas

Universität zu Köln

Fraunhofer SCAI

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

ZUSE INSTITUTE BERLIN

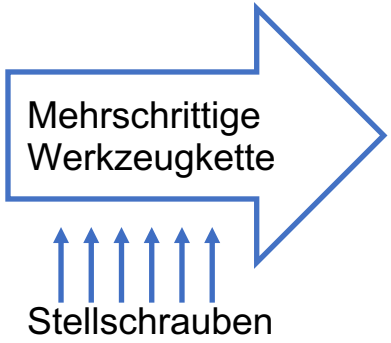
Assoziierter Partner:
Intel Deutschland GmbH

Stand der Kunst

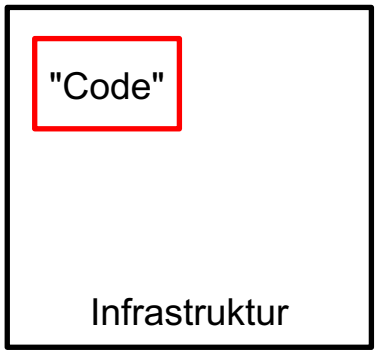
High Performance Computing

C-Code

```
for (int i=0; i<n; ++i){
    if (cond(i)){
        a[i] = 1;
    }
    b[i] = 0;
}
```



FPGA



SPONSORED BY THE
Federal Ministry
of Education
and Research

Stand der Kunst

High Performance Computing

C-Code

```
for (int i=0; i<n; ++i){
    if (cond(i)){
        a[i] = 1;
    }
    b[i] = 0;
}
```

Mehrschrittige Werkzeugkette

Stellschrauben

FPGA

"Code"

Infrastruktur

- Schaltung zu groß für FPGA
- Schaltung zu langsam für Timing
- ...

Umprogrammierung

Trial & Error – "Tuning"

SPONSORED BY THE
Federal Ministry of Education and Research

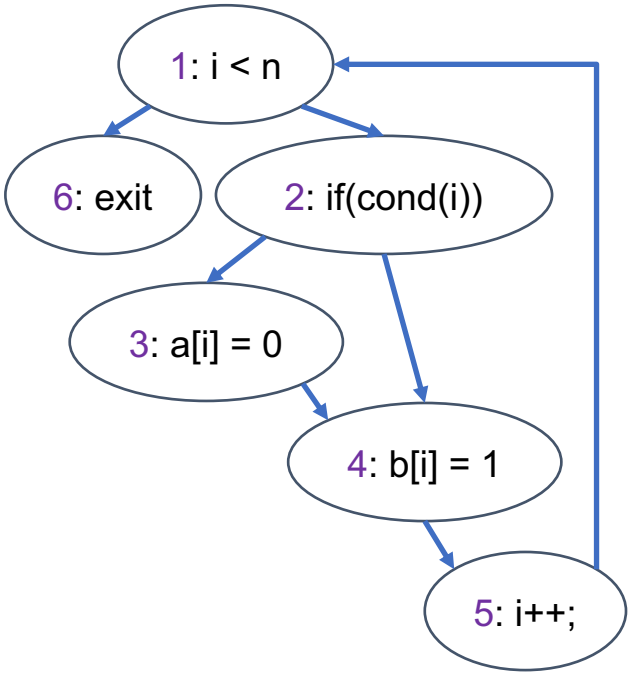
C-Code

```
for (int i=0; i<n; ++i){
    if (cond(i)){
        a[i] = 1;
    }
    b[i] = 0;
}
```

HLS, High-Level-Synthese (Teil 1)

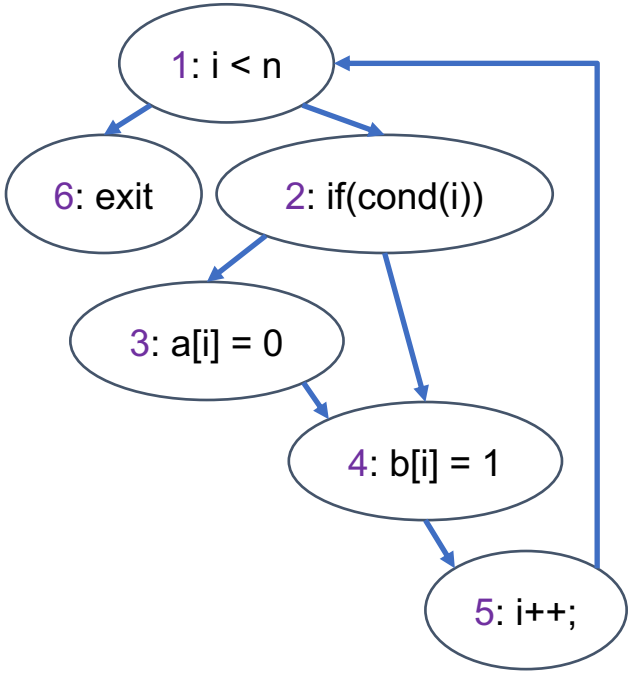
Stellschrauben

Zustandsautomat



SPONSORED BY THE
Federal Ministry of Education and Research

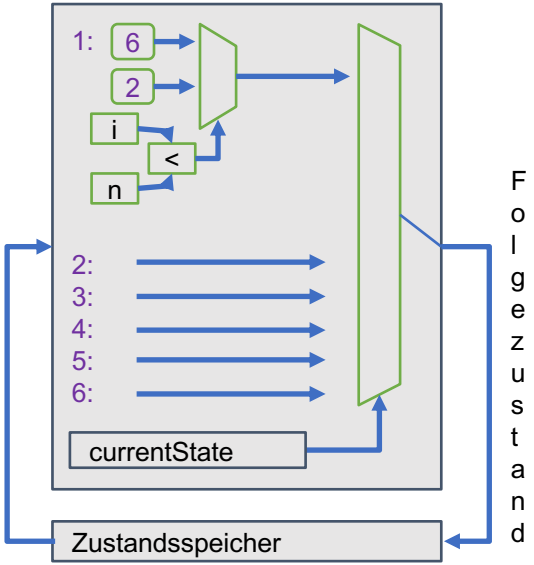
Zustandsautomat



HLS, High-Level-Synthese (Teil 2)

Stellschrauben

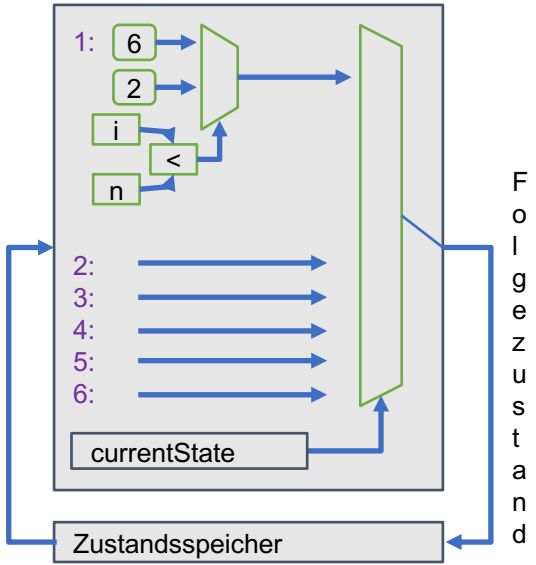
VHDL – Abstrakte Hardware-Beschreibung



SPONSORED BY THE
Federal Ministry of Education and Research

High Performance Computing

VHDL – Abstrakte Hardware-Beschreibung



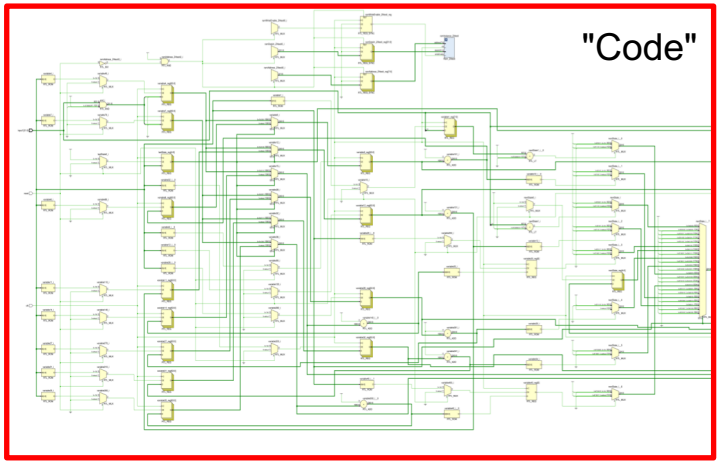
VHDL-Synthese

Logik-Synthese

Logik-Optimierung

etc.

Netzliste – Konkrete Hardware für FPGA



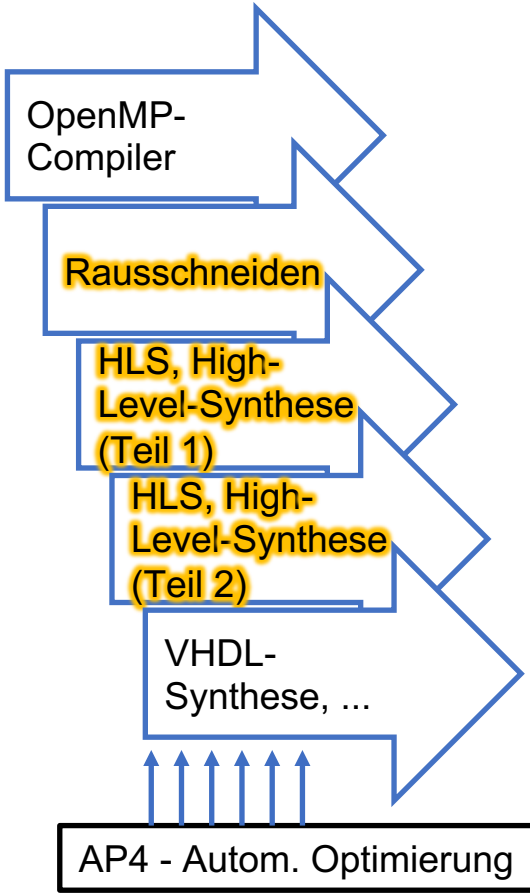
SPONSORED BY THE
Federal Ministry of Education and Research

Übertragung von C auf OpenMP

High Performance Computing

OpenMP-Code

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
        a[i] = 1;
    }
    b[i] = 0;
}
```



FPGA

"Code"

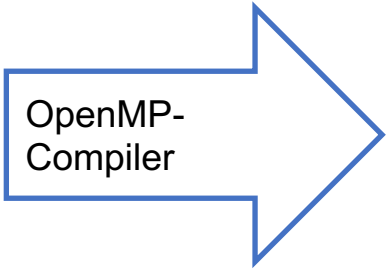
Interface

Infrastruktur

SPONSORED BY THE
Federal Ministry of Education and Research

OpenMP-Code

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
       a[i] = 1;
    }
    b[i] = 0;
}
```




```
{
    __omp_initialize_fpga(1);
    __omp_outlined(&n);
}

void __omp_outlined(...) {
    __make_threads();
    __run_threads(__loopBody);
    __destroy_threads();
}

void __loopBody(...) {
    for (...) {...}
}
```

AP1: Zusammensuchen der verstreuten Code-Stücke, die der OpenMP-Compiler für das `pragma`-Beispiel produziert.

SPONSORED BY THE
Federal Ministry of Education and Research

OpenMP-Code

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
       a[i] = 1;
    }
    b[i] = 0;
}
```

OpenMP-Compiler

```
{
    __omp_initialize_fpga(1);
    __omp_outlined(&n);
}

void __omp_outlined(...) {
    __make_threads();
    __run_threads(__loopBody);
    __destroy_threads();
}

void __lo
    for (
}
```

Rausschneiden

AP1: Zusammensuchen der verstreuten Code-Stücke, die der OpenMP-Compiler für das `pragma`-Beispiel produziert.

SPONSORED BY THE
Federal Ministry of Education and Research

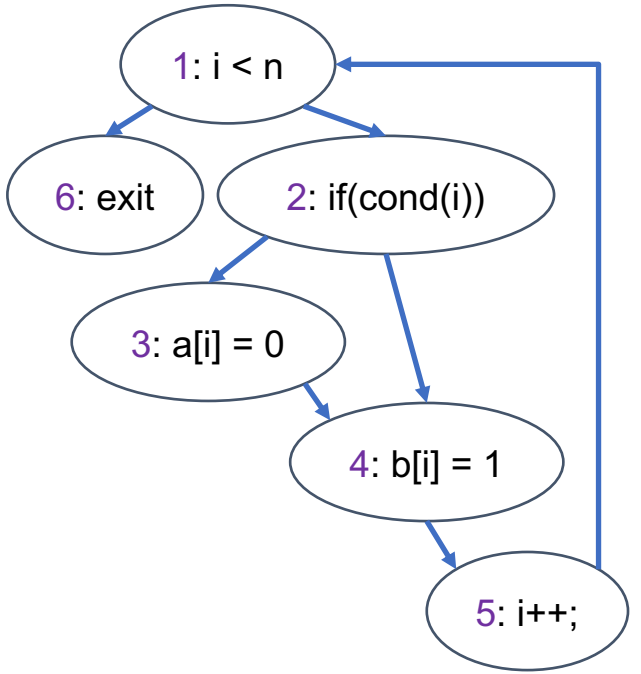
Zustandsautomat

Kommerzielle HLS-Werkzeuge:

- können nicht mit den vielen Code-Schnipseln umgehen, die OpenMP-Compiler produzieren,
- kennen das Datenabhängigkeitswissen nicht,

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
       a[i] = 1;
    }
    b[i] = 0;
}
```

HLS, High-Level-Synthese (Teil 1)



SPONSORED BY THE
Federal Ministry of Education and Research

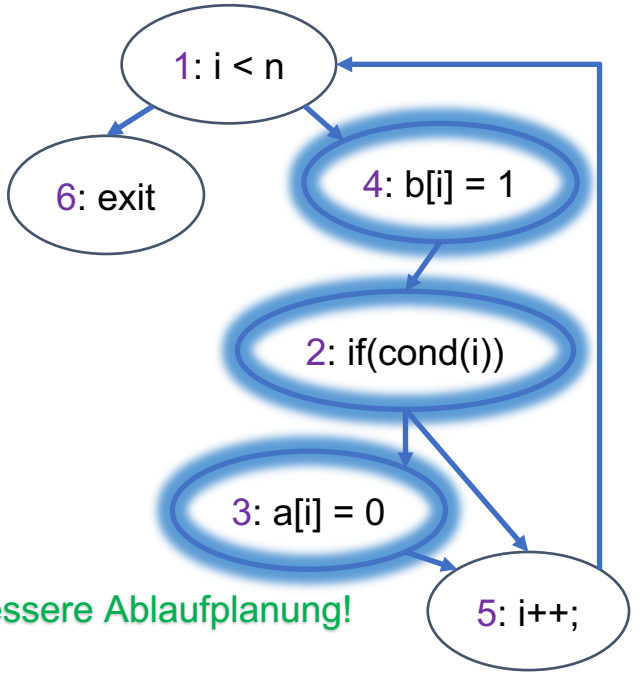
Kommerzielle HLS-Werkzeuge:

- können nicht mit den vielen Code-Schnipseln umgehen, die OpenMP-Compiler produzieren,
- kennen das Datenabhängigkeitswissen nicht,

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
       a[i] = 1;
    }
    b[i] = 0;
}
```

Orka-HLS (Teil 1)

Zustandsautomat



Bessere Ablaufplanung!

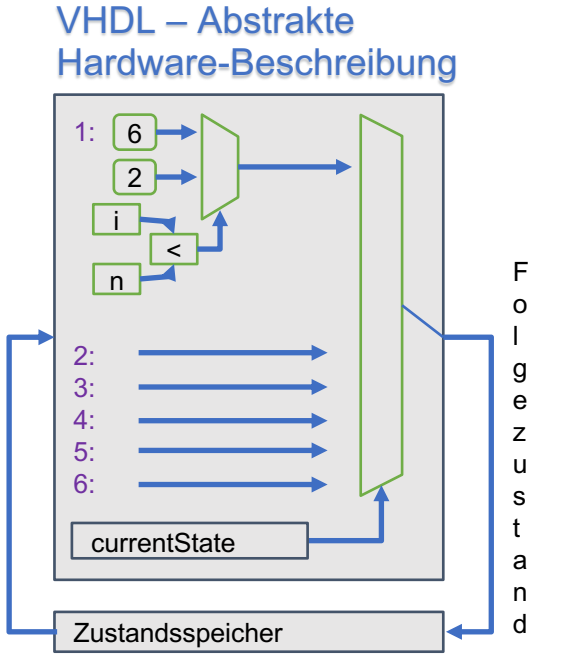
SPONSORED BY THE
Federal Ministry of Education and Research

Kommerzielle HLS-Werkzeuge:

- können nicht mit den vielen Code-Schnipseln umgehen, die OpenMP-Compiler produzieren,
- kennen das Datenabhängigkeitswissen nicht,
- verstehen `#pragma`s nicht.

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
        a[i] = 1;
    }
    b[i] = 0;
}
```

HLS, High-Level-Synthese (Teil 2)



SPONSORED BY THE
Federal Ministry of Education and Research

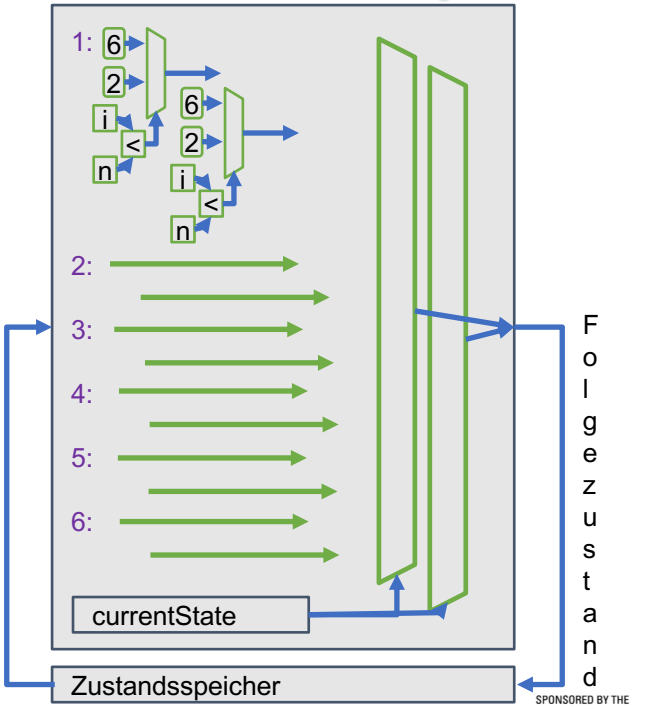
VHDL – Abstrakte Hardware-Beschreibung

Kommerzielle HLS-Werkzeuge:

- können nicht mit den vielen Code-Schnipseln umgehen, die OpenMP-Compiler produzieren,
- kennen das Datenabhängigkeitswissen nicht,
- verstehen `#pragma`s nicht.

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
        a[i] = 1;
    }
    b[i] = 0;
}
```

Orka-HLS (Teil 2)



Parallele Hardware!

SPONSORED BY THE
Federal Ministry of Education and Research

High Performance Computing

Kommerzielle HLS-Werkzeuge:

- können nicht mit den vielen Code-Schnipseln umgehen, die OpenMP-Compiler produzieren,
- kennen das Datenabhängigkeitswissen nicht,
- verstehen `#pragma`s nicht.

```
#pragma omp target fpga1
#pragma omp parallel for
#pragma omp simd
for (int i=0; i<n; ++i){
    if (cond(i)){
       a[i] = 1;
    }
    b[i] = 0;
}
```

AP1:

- Technischer Durchstich des
- Orka-HLS Frameworks
  - Einfache Bit-Optimierung
  - SDC-Ablaufplanung
- Interface Generators
- uvm.

SPONSORED BY THE
Federal Ministry of Education and Research

Status

High Performance Computing

- Technischer Durchstich vom Kompilat zum laufenden FPGA Bitstream
- Diverse Schnittstellendiskussionen mit Partnern und den anagestrebten HLS Herstellern
- Erste Low-Level-Plattformen erstellt sowie verskriptet
- Erste Beschreibung von Runtime Library Schnittstellen erstellt
- Einige Beispielcodes bestimmt und in Teilen vorbereitet
- Erste Version des Treibers für die Host-FPGA Kommunikation vorbereitet
- Für den Durchstich wurde eine teilfunktionale ORKA-HLS (ZIB, FAU) erstellt

SPONSORED BY THE
Federal Ministry of Education and Research

Vielen Dank!

High Performance Computing

- RRZK
  - Prof. Dr. Ulrich Lang
  - Viktor Achter
  - Lech Nieroda
- FhG SCAI
  - Horst Schwichtenberg
  - Olga Rodikow
- Fridrich-Alexander Universität Erlangen (FAU)
  - Prof. Dr. Michael Philippsen
  - Florian Mayer
- Konrad Zuse Institut
  - Dr. Thomas Steinke
  - Marius Knaust
- Unterauftragnehmer / Assoziierte Partner
  - Jesko Schwarzer
  - Dr. Michael Klemm (Intel Deutschland GmbH)

SPONSORED BY THE
Federal Ministry of Education and Research