

Bericht vom Workshop “FPGAs in (High-Performance) Computing... Quo Vadis?”

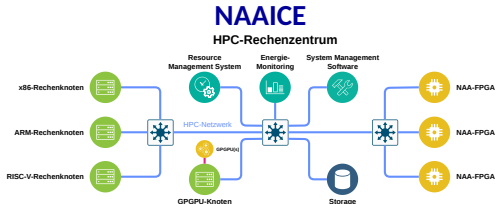
PARS-Workshop, 18. September 2026

Steffen Christgau (ZIB), Benno Stabernack (HHI)

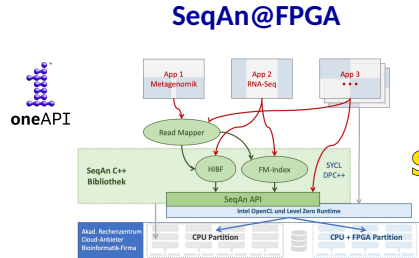
Zuse-Institut Berlin, Fraunhofer Fraunhofer Heinrich-Hertz-Institut



Motivation: “Demotivation” aus GreenHPC-Projekten



Lose gekoppelte FPGAs für deren skalierbaren Einsatz im HPC-RZ



Genomics auf FPGA

Endnutzer-Perspektive: Verwendung von HLS-Toolchain zur FPGA-Verwendung; keine Forschung an HLS selbst

Beide GreenHPC-Projekte haben im Jahr 2026 **Herausforderungen bei der Nutzung von FPGAs**, insbesondere mit HLS

1. HLS-Toolchain funktionieren zwar, aber FPGA-“Programmierung” bedarf Detailwissen zu FPGA-Konzepten
2. Performance von HLS ist im Vergleich zu HDL-Implementierung (VHDL, Verilog) schlecht.
3. Abhängigkeit von FPGA- und Board-Herstellern ist eine operative Herausforderung (Intel → Altera)

Offenbar weiterhin signifikante Hindernisse für die FPGA-Nutzung im HPC-Umfeld.

Idee: Workshop der Deutschen (HPC-)FPGA-Community um Status und Quo Vadis zu diskutieren.

Workshop am 13.3.2026 am ZIB

- 25 Teilnehmende aus 11 Einrichtungen, davon ein Industrievertreter
- 2 Kick-Off + 6 Invited Talks von C. Plessl (Paderborn), T. Kalkhoff (Darmstadt), H. Borrás (Heidelberg), D. Göhringer (Dresden), P. Holzinger (Erlangen), D. Passaretti (Magdeburg)
- Knapp zwei Stunden Diskussion zum *Status Quo* und *Quo Vadis*.



1. Programmiermodelle für FPGAs sind zu hardwarenah und für *Software*- oder gar *Anwendungs*-Entwickler ungeeignet, um FPGA-basierte Beschleuniger zu entwickeln.
2. Vorhandenes Wissen über effizientes FPGA-basiertes Beschleuniger-Design ist fragmentiert.
3. Es existiert kein nachhaltiger und quelloffener (Software)-Stack, der den HPC-Bedürfnissen gerecht wird.
4. FPGA-Hersteller sind heute an HPC nur wenig, wenn überhaupt, interessiert und Support ist in Summe nur unzureichend.

An High-Level-Synthese scheiden sich die Community-Geister

- Pro:
 - Diverse Arbeiten, die wiss. HPC-Anwendungen erfolgreich mit HLS umgesetzt haben
 - Schnellere Exploration des Entwurfsraums
- Contra:
 - Fehlannahme, dass Softwareentwickler mittels HLS zum Hardware-Designer wird.
 - Code muss compilverständlich und hardwarenah sein → Mehrwert für Entwickler?!
 - Unklare Performance und Effizienz im Vergleich zu HDL
- Keine “standardisierte” Compilerinfrastruktur für HLS vorhanden.
- Intransparente Entscheidungen entlang der langen Toolchain

Folgerung

Schaffung eines quelloffenen HLS-Frameworks mit Schnittstellen für Weiterentwicklung/Forschung (“LLVM für HLS”)

- Im Gegensatz zur Software-Welt fehlt es an Bibliotheken.
 - Herstellerangebot ist teils gering und kaum genutzt.
 - Output der wiss. Community nicht nachhaltig
- Einfacher gesagt, als getan:
 - keine ABI/Calling Convention
 - Freiheit im HW-Design vs. Vorgaben der Bibliotheken
 - Nutzung von vorhandenen IP-Blöcken in HLS-Designs
 - Parametrisierung nur bedingt Lösung

Folgerung

Wissen zur Erstellung effizienter Designs muss geteilt werden, bspw. durch Trainingsangebote. Bibliotheken müssen nachhaltig entwickelt werden.

- Zweifache Abhängigkeit:
 1. FPGA-Hersteller: Chip, Synthese-Tools, HLS-Compiler
 2. Board-Hersteller: Board, Interfaces, Treiber für OS-Integration, Board-Infrastruktur
- Toolchains sind üblicherweise *closed source*
- Wirtschaftliche Interessen der Hersteller dominieren
 1. Intel: Abtrennung von Altera; Abkündigung von FPGA-Support in oneAPI
 2. AMD: Entfernung des Linux-Supports aus freier Vivado-Edition
 3. bevorzugte Programmiermodelle der jeweiligen Hersteller
- Open-Source-Projekte aus der wiss. Community haben (noch) keinen Durchbruch erzielt

Folgerung

Zur Lösung der Hersteller-Abhängigkeiten sollten Open-Source-Lösungen auf nationaler und europäischer Ebene gemeinsam entwickelt werden.

- Zusätzliche Abstraktionsschicht auf der physischen FPGA-Hardware
- Virtuelle Architektur für Anwendungsentwickler
- Programmierung des Overlays statt direkter Konfiguration von LUTs, DSPs, Registern und Routing
- Ziele: höhere Produktivität und bessere Portierbarkeit
- Vorimplementierte Infrastruktur kann Synthese- und Implementierungsaufwand reduzieren

Folgerung

Overlays können FPGA-spezifische Komplexität verbergen und den Zugang zu FPGA-Beschleunigung erleichtern.

- Statisches Overlay — vordefinierte programmierbare Architektur; geringe bis mittlere Flexibilität
- Dynamisches anwendungsspezifisches Overlay — Rechenwerke werden zur Laufzeit per DPR ausgetauscht
- Dynamisches Overlay mit regulärer Infrastruktur — rekonfigurierbare Tiles mit standardisierter Kommunikation
- Beispiele: CNN-Beschleunigung, CGRA-basierte Overlays und dynamische Dataflow-Architekturen

Folgerung

Je mehr Flexibilität zur Laufzeit benötigt wird, desto komplexer wird die Overlay-Infrastruktur.

- Traditionelle DSE untersucht Architektur, Performance, Energieverbrauch und Ressourcennutzung
- Für HPC muss auch der Entwicklungsaufwand ein zentraler DSE-Parameter werden
- FPGA-Beschleunigung kann sehr energieeffizient sein, erfordert aber oft deutlich mehr Engineering-Aufwand
- Entscheidend: Wie viel Aufwand ist nötig, um die gewünschte Effizienz zu erreichen?

Folgerung

HPC-orientierte DSE = Anwendung → FPGA-Architektur → Performance + Energie + Entwicklungsaufwand

- Agentisches Programmieren mit modernen LLMs könnte HLS, Overlays und andere Abstraktionsansätze ergänzen
- Bibliothek von Hardware-Beschleuniger-Templates in hardwarenahen HDLs
- Automatische Anpassung und Erweiterung von Beschleunigern an konkrete Anwendungen
- Unterstützung bei Laufzeit- und Leistungsanalysen
- Automatisierte Design Space Exploration und Parameteroptimierung

Vision

Von manueller Hardwareentwicklung hin zur KI-gestützten Generierung von Beschleunigern.

- HEUTE: HPC-Anwendung → HLS/HDL → herstellerspezifische Tools → FPGA-Expertise → hoher Entwicklungsaufwand
- ZUKUNFT? HPC-Anwendung → offenes Ökosystem + Bibliotheken + HLS + Overlays + DSE + KI-Agenten
- Ziel: Entwicklungsaufwand senken und gleichzeitig die Energieeffizienzvorteile von FPGAs erhalten

Vision

Die Abstraktionsschicht soll den HPC-Entwickler näher an die Anwendung und weiter weg von FPGA-Details bringen.

1. Open-Source-HLS — gemeinschaftlich getrieben und unabhängig von Herstellern
2. Wiederverwendbare Kernel und Bausteine für wissenschaftliches Rechnen
3. Integrierte Plattform und Toolchain für Entwicklung, Deployment und DSE
4. Hardware-Abstraktionstechniken wie Overlays
5. Ausbildung und Wissenstransfer an spezialisierten Zentren

Ziel

Ein offenes europäisches FPGA-Ökosystem für HPC.

- Status quo für FPGAs im HPC-Umfeld ist nur bedingt zufriedenstellend
- Gemeinsame Anstrengungen nötig, um vielseitige Herausforderungen zu bewältigen
- **Save the date:** Follow-Up Workshop zur ARCS-Konferenz, 8./9.-11. März 2027, Magdeburg

Fragen? Diskussion!