Drei Jahre, zwei Projekte, ein Problem (?!)… Erfahrungen mit HLS

13. März 2026

Steffen Christgau, Thomas Steinke

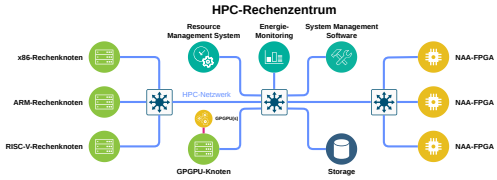
Abteilung «Verteilte Algorithmen und Supercomputing»
Zuse-Institut Berlin

ZIB
ZUSE INSTITUTE BERLIN

GreenHPC-Projekte am ZIB

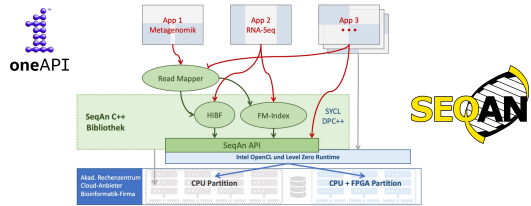
Laufzeit: 9’2022 - 9/12’2025

NAAICE



Lose gekoppelte FPGAs für deren skalierbaren Einsatz im HPC-RZ

SeqAn@FPGA



Genomics auf FPGA

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

- Gegenstand am ZIB: Offloading von ausgewählten Anwendungsteilen auf FPGA
  Verwendung von SYCL/oneAPI (ZIB war *oneAPI Center of Excellence*)
- Projektmitarbeiter (M.Sc. Scientific Computing) nach zwei Ausschreibungsrunden gefunden, keine FPGA-Entwicklungs-Erfahrung
- Demonstrator für NAAICE-Konzept mit Vektoraddition
- FP-lastiger Code: *TUG* als Bestandteil von *POET*
  - Erstellen und Lösen von LGS mit Triagonalmatrizen
  - Verwendung von SYCL für HLS-basierte FPGA-Portierung

SeqAn@FPGA

- Gegenstand: Genomics-Schlüsselfunktionen von SeqAn (C++) auf Intel FPGA mit SYCL/oneAPI
- Herausforderung Personal: 3 Projektmitarbeitende in 3 Jahren
  - MA der letzten 2 Jahre mit HDL-Vorkenntnissen aus Industrie …
- Gegenstand ist eigentlich ein „Heim-Spiel“ auf FPGA…
  - flexible Datentypen, z.B. 2 bit für DNA-Sequenz,
  - Durchsatzsszenarium ⇒ Vervielfachung von Pipelines,
  - …

SeqAn-Bibliothek: `https://www.seqan.de`, Reinert-Gruppe, FUB

Literatur

Aus J. Cong et al.: *FPGA HLS Today: Successes, Challenges, and Opportunities*.
ACM Transactions on Reconfigurable Technology and Systems (TRETS), Volume 15, Issue 4 (December 2022):

> *[...] it is not a simple task to achieve high-performance FPGA designs using HLS*
>
> *[...] Unfortunately, many software developers may not have the required knowledge and background.*
>
> *[...] Unfortunately, although SYCL presents a single-source approach, achieving performance still typically requires a device-specific coding style, often leveraging vendor extensions. It remains to be seen whether accelerator compilers, including HLS for FPGAs, advance to the point where most code can be ported from one device to another while still achieving performance.*

NAAICE: Zum Warmwerden… Vektoraddition

- Ergebnis für Demonstrator-App Vektoraddition: funktioniert
  - SYCL-Vektoraddition initial ca. 1000 × langsamer als HDL-Referenz
  - Optimierung mit Unterstützung von Intel/Altera, final ca 2 × langsamer (immerhin), **aber:** Coding like it's 1999 …

Vektoraddition für FPGAs anno 2024

```
q.submit([&](handler& h) {
    h.single_task<parralleladdKernel>([=]() [[intel::kernel_args_restrict]] {
        device_ptr<int> in(in_device);
        device_ptr<int> out(out_device);

        // Init regs to zero
        [[intel::fpga_register]] std::array<int, 16> raw_data;
        #pragma unroll
        for (int i = 0; i < 16; i++) { raw_data[i] = 0; }

        [[intel::fpga_register]] std::array<int, 8> result_data;
        #pragma unroll
        for (int i = 0; i < 8; i++) { result_data[i] = 0; }

        int i_cnt = 0;
```

Vektoraddition für FPGAs anno 2024 (Forts.)

```
// Run over all CLs
while(i_cnt < size / 16)
    // Load complete CL i                    (same for PCIe and DDR4)
    #pragma unroll
    for (uint idx = 0; i
        {
            raw_data[id
        }
    }

    result_data[0] =                          a[8];
    result_data[1] =                          a[9];
    ...
}
```

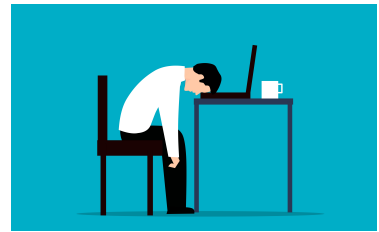
Bildquelle: https://trekkerscrapbook.com/wp-content/uploads/2013/03/eyebrow-07.jpg

NAAICE: Portierung von TUG

- Eigentliches Teilziel im Projekt: Offloading des LGS-Lösers (Thomas-Algorithmus).
- Kein BLAS verfügbar (LINPACK: 1979)
- Hürde: Kenntnis von FPGA-Hardware/Design-Prinzipien.
  Frage für den Projektleiter: Wie viele Personenmonate möchte man einsetzen?
- Glück im Unglück? SYCL-FPGA-Code vom Thomas-Algorithmus für Batches auf Github:
  Spaghetti-Code, anwendungsspezifisch (keine Library), unzuverlässige Ergebnisse, Unterschiede zwischen Emulator- und Hardware-Ausführung, … Eher Pech!

SeqAn@FPGA: Herausforderungen

- Implementierung in SYCL/oneAPI bis 2024 nicht *en par* mit Intel OpenCL
  - SYCL-Performance ca. 90% von OpenCL (auf Stratix 10)
  - größerer Overhead - z.B. Zeiten für Kernel-Startup
- Robustheit und Transparenz der Toolchain
- unklar: Performance **SYCL im Vergleich zu HDL**?



Problem: Herstellersupport durch Intel

Prinzipielle Herausforderungen:

- FPGA SYCL/OpenCL Toolchain erfordert FPGA-Kenntnisse
  - oneAPI FPGA-Compiler kein Black-Box Werkzeug
  - „kryptische“ interne Fehlermeldungen (Compiler & Quartus)
- FPGA-Toolchain proprietär / *closed source*
  - Intel bis 2024: strukturbedingte Schwächen - getrennte Teams für SYCL-Compiler (oneAPI) und Synthese (Quartus)

Besonderheiten durch Intel-interne Entwicklungen:

- Verzögerungen und Abkündigung von FPGA-Produkten:
  - Abkündigung FPGA-Support, OS-Support (CentOS 7), BSP- vs. oneAPI-Versionen
  - Abkündigung des Support von Board-Hardware durch Hersteller (IA-860M, kein HBM-Support durch oneAPI mehr)

Marktsituation

- FPGA-Technologie (HW+SW) weiterhin kein (HPC) Mainstream trotz (theoret.) Potenzial
  (Präsenz in anderen Märkten, Aufschwung durch Edge/Embedded AI?)

- Anbieter haben weiterhin unzureichend (?!) den SW-Entwickler im Blick
  Ist das ein prinzipielles Problem?

- herstellerneutrale Standards fehlen (wieder?)!
  - War SYCL nur eine Episode? Was machen AMD/Xilinx und andere?
  - *es gab mal OpenFPGA in den 2000er Jahren - gleiche Schwierigkeiten…,*
    …und dann kam OpenCL auf Altera-Plattformen

Ein Problem?

Nein… Mehrere:

- **Prinzipiell:** Ist der FPGA überhaupt geeignet für nicht-HW-affine Entwickler?
  Erfahrung: Design-Flow nicht intuitiv für SW-Entwickler ⇒ hohe Einstiegshürde

- **Stand (HPC-) SW auf FPGA:** sehr beschränkt!
  - keine (BLAS-)Libs von Plattform-/FPGA-Anbieter
  - Multi/variable-Precision Optionen
  - überhaupt sinnvoll?

- **Stand Toolchain:** 2× Anbieter = Plattform + FPGA-Hersteller
  ⇒ 2× SW-Produkte = BSP/Infrastruktur + Compiler
  ***closed source* Toolchain und Systemsoftware**

Was nun? Quo Vadis