Riding the FPGA Rollercoaster: Ten Years of FPGAs in Production HPC Systems

Christian Plessl
Paderborn University, Germany

Paderborn Center for Parallel Computing &
Department of Computer Science

FPGAs in (High-Performance) Computing ... Quo Vadis? – March 13, 2026 – ZIB Berlin

My Relationship with FPGAs

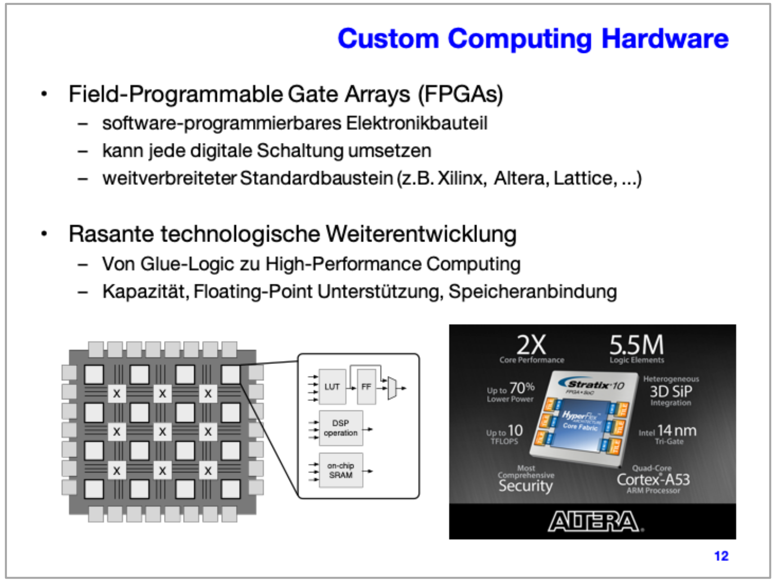
2001	Diploma thesis	• Solving minimum covering problems with FPGA-based custom computing machines	• curiosity driven
2006	PhD thesis	• Hardware virtualization on coarse-grained reconfigurable arrays	• design automation • programming models
2011	Assistant professor for "Custom Computing"	• Performance estimation for hardware accelerators • OS support for FPGAs • On-the-fly compilation for FPGAs • Acceleration of computational nanophotonics with FPGAs	• first steps into scientific computing • system-level codesign (from application to accelerator)
since 2015	Professor for "High-Performance Computing" +	• Domain-specific compilation for FPGAs • Tackling "true" computational science applications (physics, chemistry) • Scalability of FPGA applications (parallelization, networking, system integration)	• exploring/demonstrating the potential and limits of FPGAs in HPC • scalability and usability • from testbed to production systems

It must be true love – or Stockholm syndrome

Once Upon a Time in 2015

- Offer for tenured position in 2015
  - professorship in Computer Science
  - director at Paderborn Center for Parallel Computing
- Seize window of opportunity to bring the long-lasting FPGA experience of my group from testbed to production-grade HPC systems
- The stars aligned perfectly
  - capable FPGAs
  - dedicated FPGA systems for data center
  - usable high-level synthesis tools

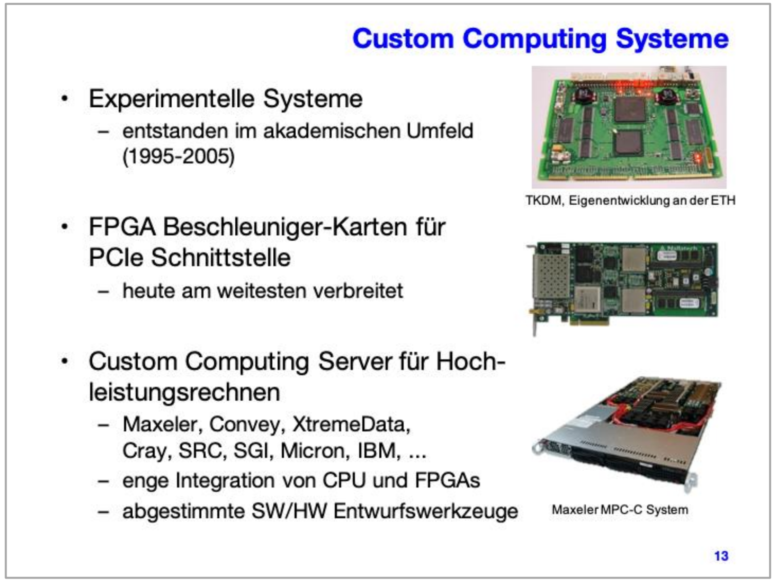
Some Slides from my Inaugural Lecture in 2015



Custom Computing Hardware

- Field-Programmable Gate Arrays (FPGAs)
  - software-programmierbares Elektronikbauteil
  - kann jede digitale Schaltung umsetzen
  - weitverbreiteter Standardbaustein (z.B. Xilinx, Altera, Lattice, ...)
- Rasante technologische Weiterentwicklung
  - Von Glue-Logic zu High-Performance Computing
  - Kapazität, Floating-Point Unterstützung, Speicheranbindung

12



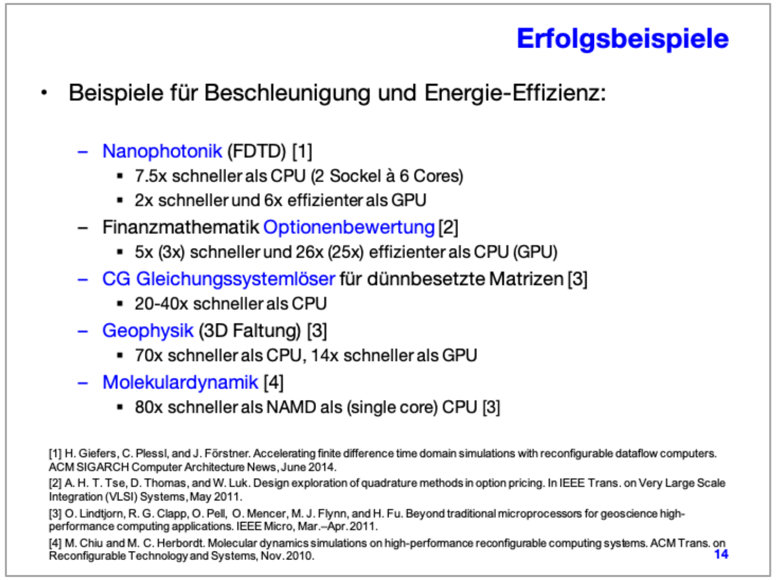
Custom Computing Systeme

- Experimentelle Systeme
  - entstanden im akademischen Umfeld (1995-2005)
- FPGA Beschleuniger-Karten für PCIe Schnittstelle
  - heute am weitesten verbreitet
- Custom Computing Server für Hochleistungsrechnen
  - Maxeler, Convey, XtremeData, Cray, SRC, SGI, Micron, IBM, ...
  - enge Integration von CPU und FPGAs
  - abgestimmte SW/HW Entwurfswerkzeuge

TKDM, Eigenentwicklung an der ETH

Maxeler MPC-C System

13



Erfolgsbeispiele

- Beispiele für Beschleunigung und Energie-Effizienz:
  - Nanophotonik (FDTD) [1]
    - 7.5x schneller als CPU (2 Sockel à 6 Cores)
    - 2x schneller und 6x effizienter als GPU
  - Finanzmathematik Optionenbewertung [2]
    - 5x (3x) schneller und 26x (25x) effizienter als CPU (GPU)
  - CG Gleichungssystemlöser für dünnbesetzte Matrizen [3]
    - 20-40x schneller als CPU
  - Geophysik (3D Faltung) [3]
    - 70x schneller als CPU, 14x schneller als GPU
  - Molekulardynamik [4]
    - 80x schneller als NAMD als (single core) CPU [3]

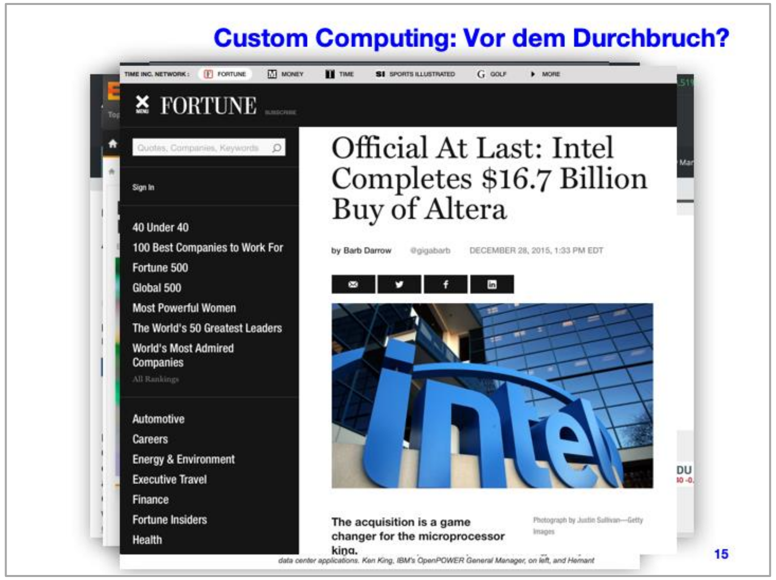
[1] H. Giefers, C. Plessl, and J. Förstner. Accelerating finite difference time domain simulations with reconfigurable dataflow computers. ACM SIGARCH Computer Architecture News, June 2014.

[2] A. H. T. Tse, D. Thomas, and W. Luk. Design exploration of quadrature methods in option pricing. In IEEE Trans. on Very Large Scale Integration (VLSI) Systems, May 2011.

[3] O. Lindtjorn, R. G. Clapp, O. Pell, O. Mencer, M. J. Flynn, and H. Fu. Beyond traditional microprocessors for geoscience high-performance computing applications. IEEE Micro, Mar.–Apr. 2011.

[4] M. Chiu and M. C. Herbordt. Molecular dynamics simulations on high-performance reconfigurable computing systems. ACM Trans. on Reconfigurable Technology and Systems, Nov. 2010.

14



Custom Computing: Vor dem Durchbruch?

Official At Last: Intel Completes $16.7 Billion Buy of Altera

by Barb Darrow @gigabarb DECEMBER 28, 2015, 1:33 PM EDT

The acquisition is a game changer for the microprocessor king.

Photograph by Justin Sullivan—Getty Images

data center applications. Ken King, IBM's OpenPOWER General Manager, on left, and Hemant

15

High Hopes and Expectations ...

- Microsoft Catapult / Brainwave (2015-19)
  - proof of benefit and practicality for search acceleration
  - FPGAs installed in almost all servers in Bing datacenters
  - later converted to low-latency AI inference (Brainwave)

- Intel purchases Altera (2015), AMD buys Xilinx (2020)
  - Intel: 16.7 G$, AMD: 20 G$
  - FPGAs are finally on the doorsteps of the data center

- Amazon AWS F1 instances available (2017)
  - fully programmable FPGA become part of public cloud offerings
  - Accelerator-as-a-Service vision

- Numerous SmartNIC projects
  - FPGAs find relevant and sustainable niche in in-network processing

- HLS/OpenCL Programming flows
  - any software developer can be a hardware developer

... and Major Blows and Setbacks

- Intel cancelling hybrid Xeon+FPGA device (2020)
  - too complex, different processes/timelines for CPU and FPGA development
  - thermal problems
  - programming too challenging

- Microsoft deprioritizes/cancels Brainwave (~2022)
  - role of FPGAs relegated to in-network processing

- GPUs dominate AI and HPC workloads
  - rapid improvement of compute and memory bandwidth/size
  - vibrant NVidia software ecosystem
  - massive profits fuel development

- Intel and AMD neglect FPGAs
  - development tools remain cumbersome
  - hardware suffers from stagnation and delay
  - depriorization over GPU and CPU development

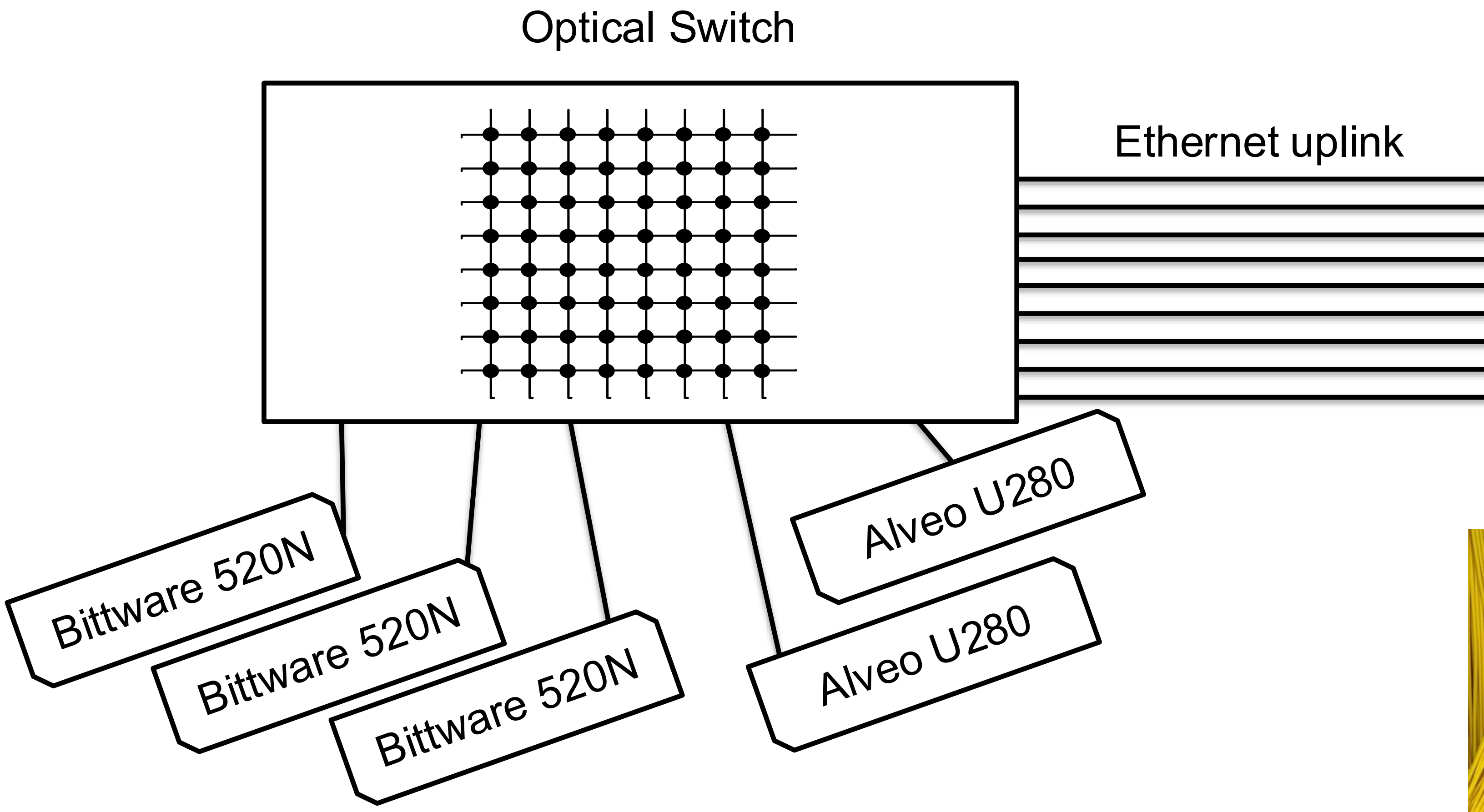
Meanwhile at Paderborn Center for Parallel Computing

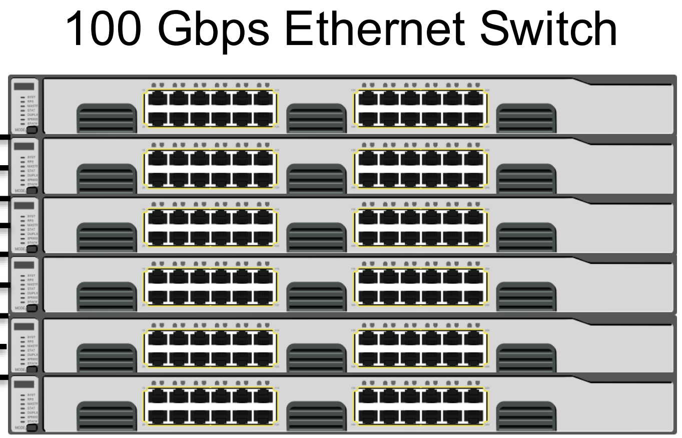
Completed Transition from Testbed to HPC Production Systems

System	Inst	CPU	FPGA	Toolflow	Properties
Maxeler MPC-C	2012	Xeon X5660	4x Xilinx Virtex-6 SX475T	MaxCompiler	MAX3 FPGA card, MaxRing intercon.
Nallatech 385A	2016	Xeon E5-1260v2	Intel Arria 10 GX1150	Intel OpenCL	Nallatech 385A FPGA card
IBM S812L	2016	POWER8	Xilinx Virtex-7 VX690T	IBM CAPI + HDL/Xilinx HLS	AlphaData 7V3 FPGA board
Micron Workstation	2016	Intel i7-5930K	Xilinx Kintex Ultrascale KU060	Xilinx OpenCL	Pico AC-510 FPGA card with Hybrid-memory cube
XCL cluster	2017	Intel Xeon E5-1630v4	Xilinx Virtex-7 VX690T Xilinx Kintex Ultrascale KU115	Xilinx OpenCL	8 AlphaData 7V3 and 8 8K5 FPGA cards
HARP cluster	2017	Intel Xeon E5-v4	Intel Xeon Broadwell + FPGA hybrid CPU/FPGA	Intel OpenCL, HDL	10 node cluster with 1 BDW+FPGA processor per node
Noctua 1 cluster	2018	Intel Xeon 6148	Intel Stratix 10 GX2800	Intel OpenCL/oneAPI	32 BittWare 520N, optical switch
Noctua 2 cluster	2022	AMD 7713 (Milan)	Xilinx UltraScale+ XCU280 Intel Stratix 10 GX2800	Xilinx Vitis Intel OpenCL/oneAPI	48 Xilinx Alveo U280 32 BittWare 520N optical switch + Ethernet switch
HACC cluster	2023	AMD 7V13 (Milan)	Xilinx UltraScale+ Xilinx Versal	Xilinx Vitis, Vitis AI	3 nodes, each with 2 Alveo U55C, 2 VCK5000, 4x AMD Instinct MI210 GPU
Otus cluster	2025	AMD 9655 (Turin)	AMD Versal HBM Altera Agilex-M	AMD Vitis, Vitis AI Intel OneAPI	3 AMD V80 FPGA cards (pilot phase) Ibex Ipac-1000 (planned)

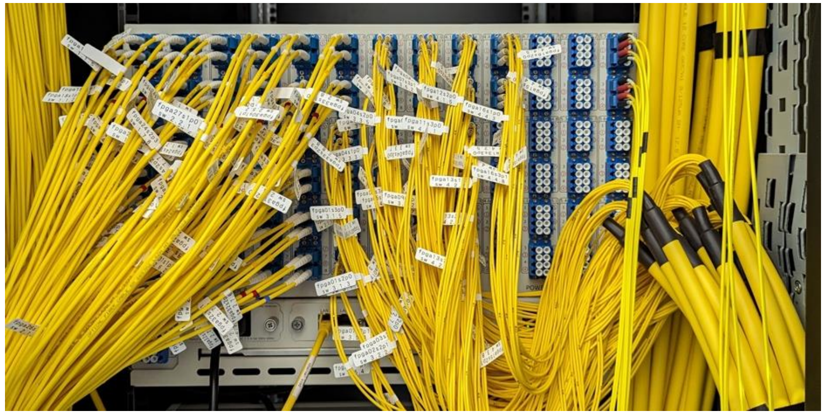
selected FPGA systems at PC2 (in grey: decommissioned) 🔵 lab 🟠 testbed clusters 🟢 production

Developed Network Infrastructures for Networking of FPGAs: Optical P2P / Ethernet / PCIe Switching





Next generation based on [Dolphin PCIe switch] currently in pilot phase



Calient S320 optical crossbar switch

photo Paderborn University

Improved Flexible FPGA Development and Runtime Environments

- Objective: Infrastructure-as-Code
- Reproduceable and configurable environment with Lmod/Slrum
  - FPGA development and backend tools
  - runtime systems and device drivers
  - board support packages
  - network/switch configurations
  - modules verify compatibility among each other, users can rely on defaults

```
module load fpga devel compiler
module load intel/oneapi/25.0.0

module load bittware/520n/20.4.0_hpc

srun -N 1 \
  --partition=fpga \
  --constraint=bittware_520n_20.4.0_max \
  --fpgalink="n00:acl0:ch0-n00:acl1:ch0" --fpgalink="n00:acl0:ch1-n00:acl1:ch1" \
  --fpgalink="n00:acl0:ch2-n00:acl1:ch2" --fpgalink="n00:acl0:ch3-n00:acl1:ch3" \
  --t 2:00:00 ./my_fpga_app
```

Lmod

slurm
workload manager

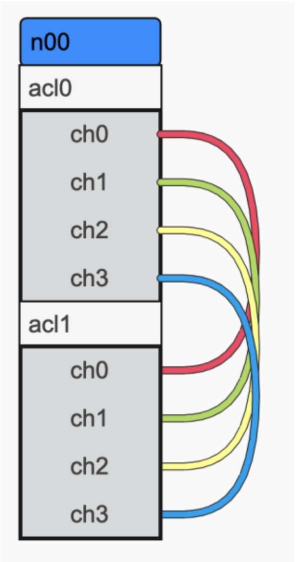


photo credits: https://github.com/TACC/Lmod, SchedMD.com

Studied Numerous HPC Applications with FPGA Acceleration

Application	Method Details	FPGA Results	FPGA Advantages
Shallow Water Equation	Discontinuous Galerkin on Unstructured 2D Meshes	up to 144x speedup vs. 1 CPU core, and scaling challenge for CPU	dataflow between tasks, custom local memory, custom parallelism, scaling with direct communication
Photonics with Maxwell's Equations	Discontinuous Galerkin on Unstructured 3D Meshes	1 FPGA > 2 socket CPU, much lower energy	exact data pre-fetching, separation of regular and irregular memory access, dataflow
Electron Repulsion Integrals (AIMD)	Rys Quadrature, Data Compression (ongoing work)	1 FPGA > 2 socket CPU, bandwidth bound, extra potential via compression	custom parallelism, custom local memory layout, bit operations for compression
Linear Algebra	Dense Matrix Multiplication	3 TFLOPs SP @~120W	dataflow with other functions possible
Spectral Methods in MD	3D Fast Fourier Transformation	deterministic 8 taps per cycle, multiple lanes	dataflow with other functions possible
N-Body Methods (MD, Astrophysics)	Direct Forces, Ring Communication, Pairwise Summation	DP peak flops < 1 CPU, SP peak flops > 1 CPU, better strong scaling	deterministic on-chip and off-chip data movement also for small data sizes

Studied Numerous HPC Applications with FPGA Acceleration

Application	Method Details	FPGA Results	FPGA Advantages
Shallow Water Equation	Discontinuous Galerkin on Unstructured 2D Meshes	up to 144x speedup vs. 1 CPU core, and scaling challenge for CPU	dataflow between tasks, custom local memory, custom parallelism, scaling with direct communication
Photonics with Maxwell's Equations	Discontinuous Galerkin on Unstructured 3D Meshes	1 FPGA > 2 socket CPU, much lower energy	exact data pre-fetching, separation of regular and irregular memory access, dataflow
Electron Repulsion Integrals (AIMD)	Rys Quadrature, Data Compression (ongoing work)	1 FPGA > 2 socket CPU, bandwidth bound, extra potential via compression	custom parallelism, custom local memory layout, bit operations for compression
Linear Algebra	Dense Matrix Mult		dataflow with other functions possible
Spectral Methods in MD	3D Fast Fourier Transformation	deterministic 8 taps per cycle, multiple lanes	dataflow with other functions possible
N-Body Methods (MD, Astrophysics)	Direct Forces, Ring Communication, Pairwise Summation	DP peak flops < 1 CPU, SP peak flops > 1 CPU, better strong scaling	deterministic on-chip and off-chip data movement also for small data sizes

1 FPGA outperforms 2 CPUs
inherent advantages of FPGA

Studied Numerous HPC Applications with FPGA Acceleration

Application	**Method Details**	**FPGA Results**	**FPGA Advantages**
Shallow Water Equation	Discontinuous Galerkin on Unstructured 2D Meshes	up to 144x speedup vs. 1 CPU core, and scaling challenge for CPU	dataflow between tasks, custom local memory, custom parallelism, scaling with direct communication
Photonics with Maxwell's Equations	Discontinuous Galerkin on Unstructured 3D Meshes	1 FPGA > 2 socket CPU, much lower energy	exact data pre-fetching, separation of regular and irregular memory access,
Electron Repulsion Integrals (AIMD)	Rys Com work		llelism, custom local put, bit operations for n
Linear Algebra	Dense Matrix Multiplication	3 TFLOPs SP @~120W	dataflow with other functions possible
Spectral Methods in MD	3D Fast Fourier Transformation	deterministic 8 taps per cycle, multiple lanes	dataflow with other functions possible
N-Body Methods (MD, Astrophysics)	Direct Forces, Ring Communication, Pairwise Summation	DP peak flops < 1 CPU, SP peak flops > 1 CPU, better strong scaling	deterministic on-chip and off-chip data movement also for small data sizes

1 FPGA performance similar to 1-2 CPUs
no inherent advantage of FPGA for single kernel, but
useful building block in more complex dataflow design

Studied Numerous HPC Applications with FPGA Acceleration

Application	Method Details	FPGA Results	FPGA Advantages
Shallow Water Equation	Discontinuous Galerkin on Unstructured 2D Meshes	up to 144x speedup vs. 1 CPU core, and scaling challenge for CPU	dataflow between tasks, custom local memory, custom parallelism, scaling with direct communication
Photonics with Maxwell's Equations	Discontinuous Galerkin on Unstructured 3D Meshes	1 FPGA > 2 socket CPU, much lower energy	exact data pre-fetching, separation of regular and irregular memory access, dataflow
Electron Repulsion Integrals (AIMD)	Rys Quadrature, Data Compression (ongoing work)	1 FPGA > 2 socket CPU, bandwidth bound, extra potential via compression	custom parallelism, custom local memory layout, bit operations for compression
Linear Algebra	Dense Matrix Multiplication	3 TFLOPs SP @~120W	dataflow with other functions possible
Spectral Methods in MD	3D F Tran		n other functions possible
N-Body Methods (MD, Astrophysics)	**Direct Forces, Ring Communication, Pairwise Summation**	**DP peak flops < 1 CPU, SP peak flops > 1 CPU, better strong scaling**	**deterministic on-chip and off-chip data movement also for small data sizes**

1 FPGA performance similar to 1-2 CPUs
but better scalability

State of the FPGA Union

FPGA Paradox: Technically Strong – Yet Not Mainstream

Why FPGAs should be attractive

- custom processing engines
- flexible parallelism (task/data/pipeline/bit-level)
- tailored memory system
- energy efficiency
- low-latency communication

Reality in HPC

- niche accelerator
- rarely a default choice
- small ecosystem
- limited applications

If FPGAs are so good, why are the not everywhere in HPC?

4 missing pieces

1. Programming Model and Hardware Abstraction
2. Documentation & Reference Designs
3. Library and Application Support
4. System Integration

1. Programming Model: Still Too Close to Hardware

- We have come a long way from HDL to HLS, OpenCL and oneAPI
  - familiar and expressive programming languages
  - no full system design necessary
  - we are (almost) exclusively using HLS since a decade

- High-level tools still requires understanding of the architecture
  - hw-specific optimization techniques: custom pipelining, unrolling in space, memory banking
  - strategic mistake to focus on turning SW developers into HW developers

- Domain-specific tools could be the solution – but we lack LLVM for FPGA design
  - open source has had little impact on high-end FPGA design

2. Knowledge Exists but is Fragmented

- Extensive collection example for FPGA exists
  - vendor documentation and examples
  - GitHub projects
  - discussion forums
  - research papers

- But knowledge about competitive FPGA for HPC is still tribal knowledge

- Could LLMs fill this gap?
  - how to capture important non-functional metrics (latency, throughput, resource demand)

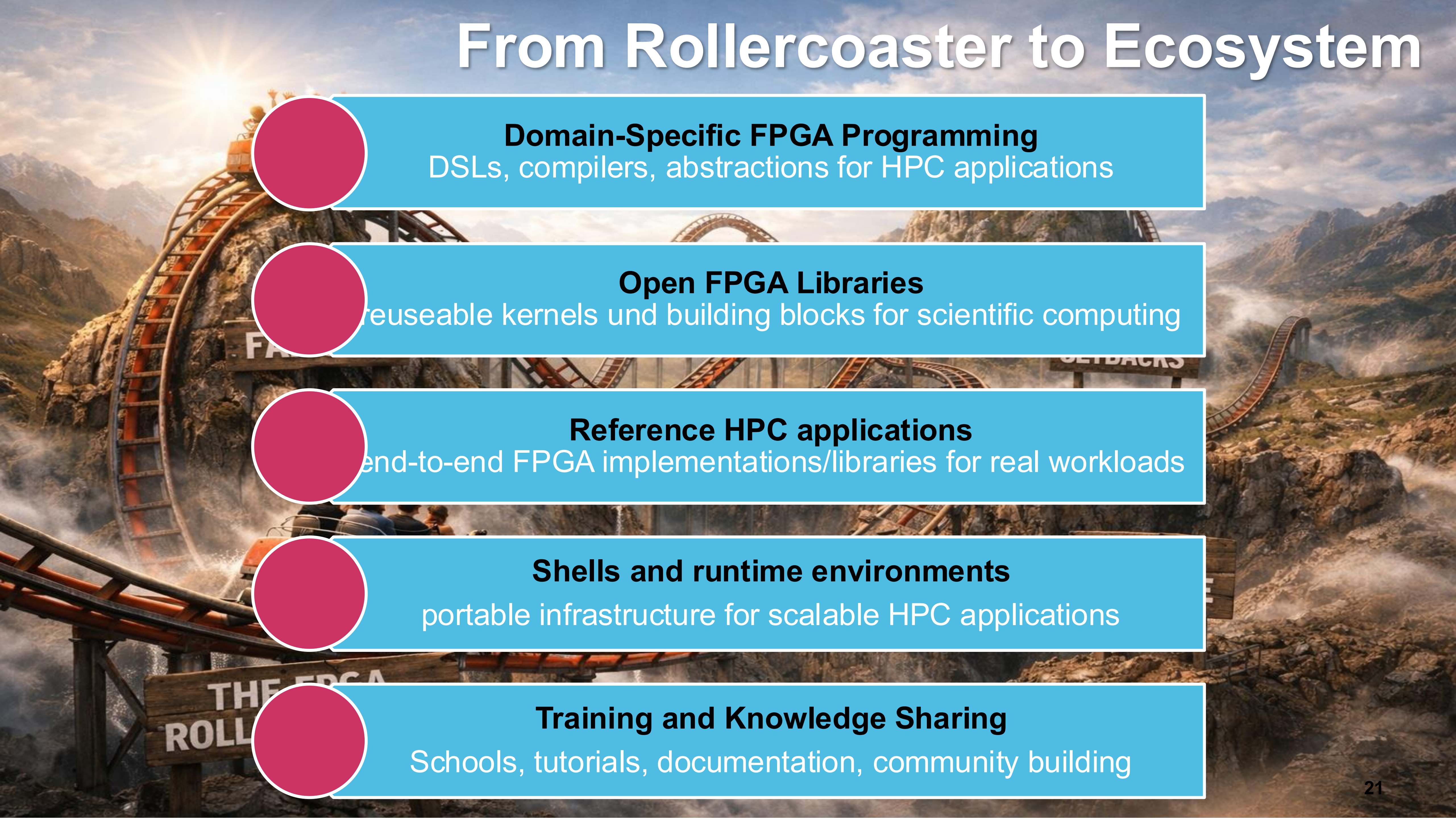
3. Missing FPGA Software Stacks

- Using FPGAs typically means building your own design
- Developing good libraries for FPGAs is difficult
  - suitable granularity (from building blocks to complete designs)?
  - how to optimize for the specific FPGA?
  - hardware size restrictions require design time tradeoffs ...
    ... software can be arbitrarily large without penalty
- Lack of ready-to-use libraries and applications with built-in FPGA support
- Usability gap
  - using GPUs means linking a library
  - using an FPGA means designing hardware top to bottom

4. Operating FPGAs is Still a Craft

- Operation in production HPC clusters possible
- Setup and maintenance still take some effort
  - handling errors caused by user designs
  - providing backwards compatibility
  - keeping software up to date
- High dependence on vendor support

From Rollercoaster to Ecosystem



Domain-Specific FPGA Programming
DSLs, compilers, abstractions for HPC applications

Open FPGA Libraries
reuseable kernels und building blocks for scientific computing

Reference HPC applications
end-to-end FPGA implementations/libraries for real workloads

Shells and runtime environments
portable infrastructure for scalable HPC applications

Training and Knowledge Sharing
Schools, tutorials, documentation, community building