ESSPER

Elastic and Scalable System for High-Performance Re-configurable Computing

FPGA or CGRA?
Reconfigurable Architectures Suitable for High-Performance Computing

Kentaro Sano

RIKEN Center for Computational Science (R-CCS)

Jan 17, 2024

Summary of this Talk

- **Reconfigurable data-flow computing as promising architecture for HPC**

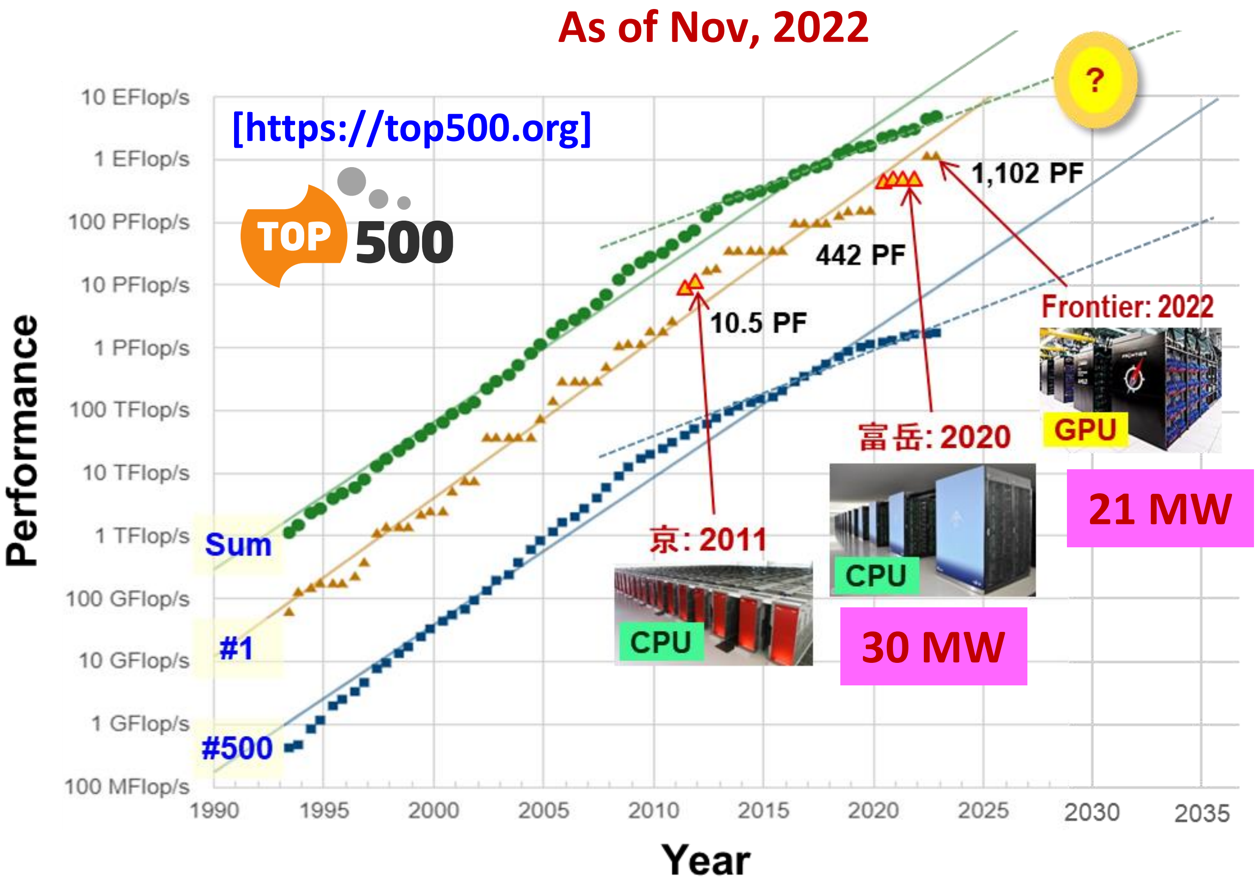
- **FPGA**
  - ✓ **ESSPER** : Elastic and scalable FPGA-cluster system for high-performance reconfigurable computing, as Prototype FPGA cluster for HPC
  - ✓ Lessons we learned

- **CGRA (Coarse-grained reconfigurable array)**
  - ✓ **RIKEN CGRA research**
  - ✓ What we have studied so far.

Introduction

- **World ranking of supercomputers**
  - ✓ TOP500: Ranking of HPL performance
  - ✓ CPU-based vs. GPU/Acc-based
  - ✓ Perf improvement slowed down around 2015.

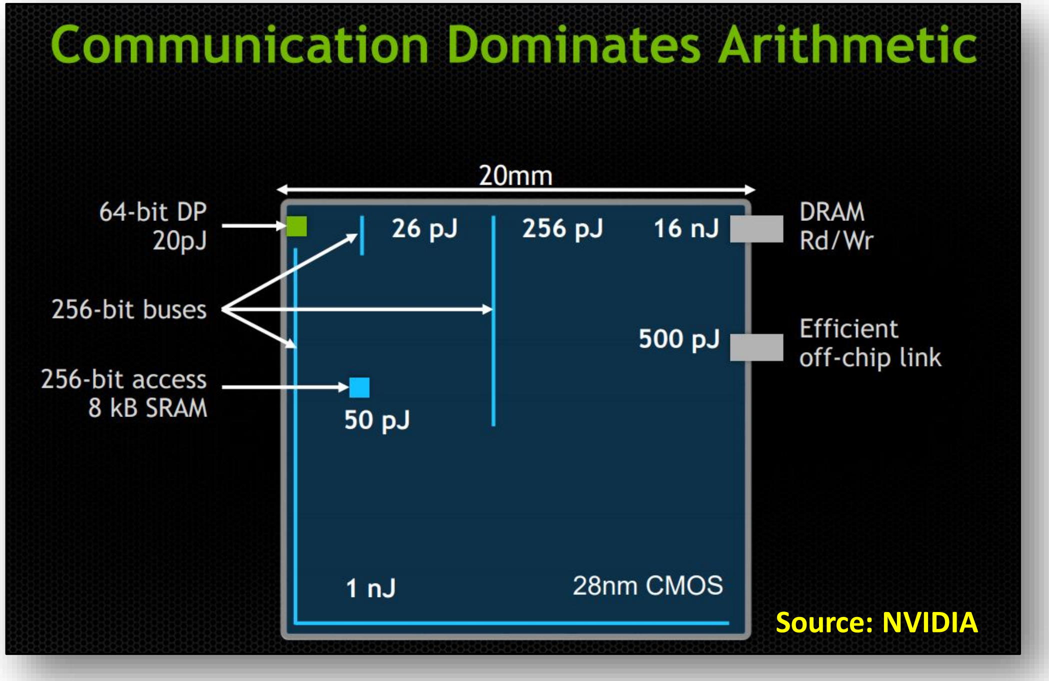
- **System performance is limited by system power.**
  - ✓ Reached tens of MW (Fugaku: 30MW, Frontier: 21MW for HPL)
  - ✓ Not easy to further increase (100MW is not real for SDGs & cost.)



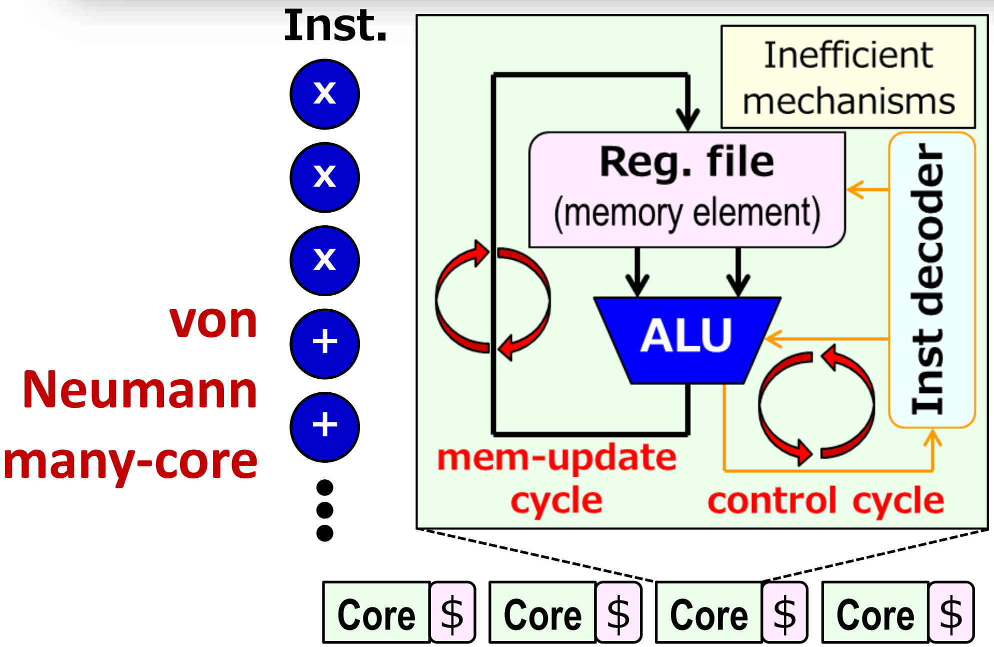
- **With capped power budget, need to increase performance per power**

What Eats Power?

- **Data movement rather than computing**
  - ✓ We should remove unnecessary data movement, and make it shorter.
- **Unsuitable architecture with low efficiency and scalability**
  - ✓ von-Neumann architectures (CPU & GPU) cannot efficiently scale due to
    - ➢ ***memory-bottlenecked structure;*** such as register files and NoC w/ LLC for multiple cores
    - ➢ ***Extra mechanisms*** consuming power just to increase IPC, such as out-of-order and branch predictor.
- **Semiconductor scaling cannot save it.**
  - ✓ Power improvement per generation is limited while can still increase transistors per area for advanced tech nodes like 5, 3, 2, and 1.5nm …



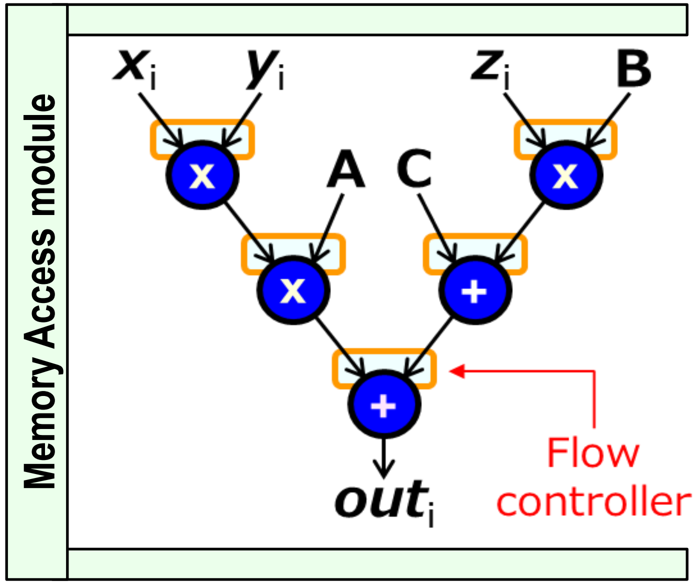
Source: NVIDIA



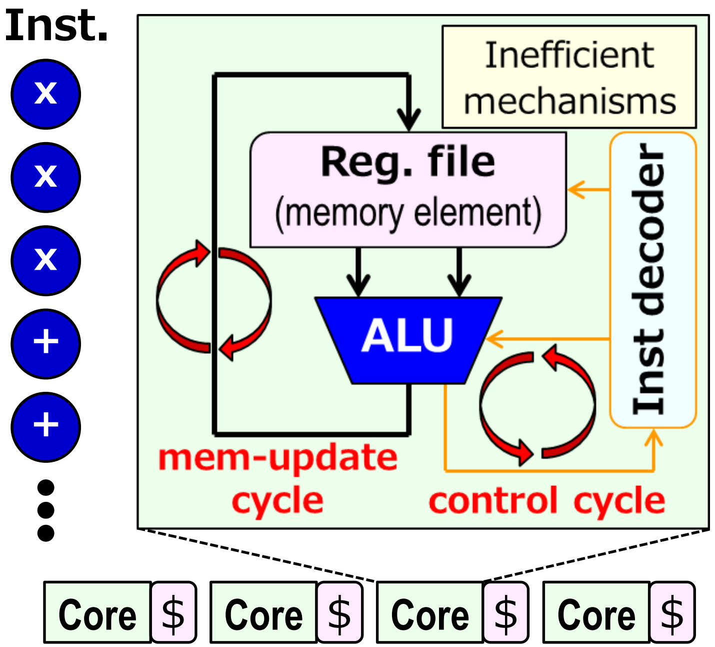
What can Save Us?

- **Data-flow computing architecture**
  - ✓ Localized data-movement
  - ✓ No memory bottleneck;
    distributed and pipelined ALUs with regular/simplified memory access
  - ✓ No extra mechanisms for non-computing
- **Circuit reconfigurability is key.**
  - ✓ Giving programmability
  - ✓ Higher efficiency for target problems
- **What candidate technologies for reconfigurable data-flow?**
  FPGA and CGRA?

Data-flow computing



von Neumann many-core



Goal and Roadmap of Processor Research Team

Goal: Establish HPC architectures suitable in Post-Moore Era

2020 2021 2022 2023 2024 2025 2026 2027

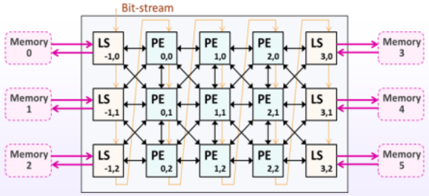
1. Advancement of Fugaku

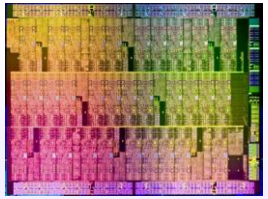
- ✓ **Functional extension with FPGAs** (FPGA cluster, ESSPER)
- ✓ SoC, system software, applications

ESSPER
Elastic and Scalable System for High-Performance Re-configurable Computing

2. Exploration of New HPC Architectures

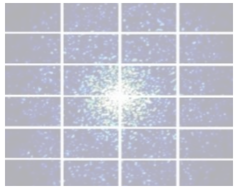
- ✓ Data-flow-based accelerators (**CGRA**)
- ✓ **Next-generation HPC systems**





3. Near-sensor / Near-storage Processing

- ✓ FPGA-based processing for **X-ray imaging detector** (Citius @Spring-8)
- ✓ Planning collaboration with next-generation radiation facility in Tohoku



We are hiring researchers!
Contact me.

4. Backend of Fault-Tolerant Quantum Computers

- ✓ Specialized hardware in digital circuits for **quantum error correction**
- ✓ FPGA cluster "ESSPER2" with Intel Agilex-M FPGA (7nm)

1. Advancement of Fugaku

- ✓ **Functional extension with FPGAs** (FPGA cluster, ESSPER)
- ✓ SoC, system software, applications

ESSPER

Elastic and Scalable System for High-Performance Re-configurable Computing

Prototype FPGA Cluster for Supercomputer Fugaku

Open-Access paper

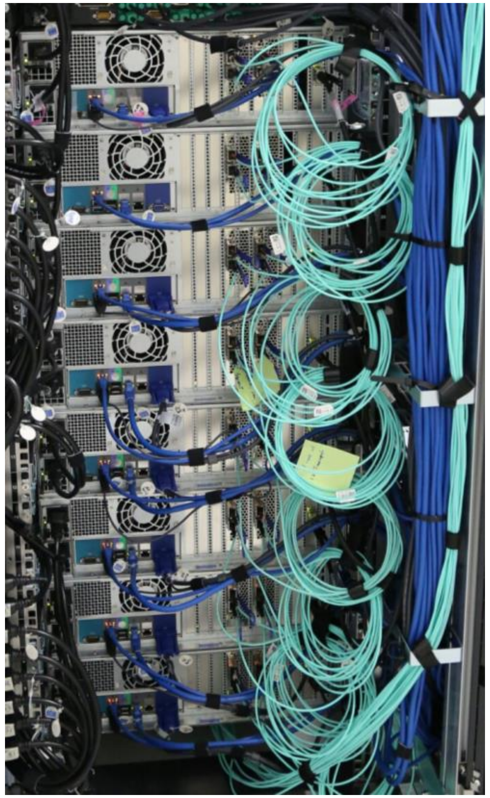
This Work

Goal : Design & demonstrate a proof-of-concept FPGA cluster for HPC research

- **ESSPER** : Elastic and scalable FPGA-cluster system
  for high-performance reconfigurable computing

- **Contributions**
  - ✓ **Design concept** of FPGA cluster for HPC
  - ✓ **Classification** of FPGA cluster architectures
  - ✓ **Proposed system stack** with software-bridged APIs
  - ✓ **Implementation and evaluation** for FPGA-based extension of the world's top-class supercomputer, Fugaku

Open-Access paper



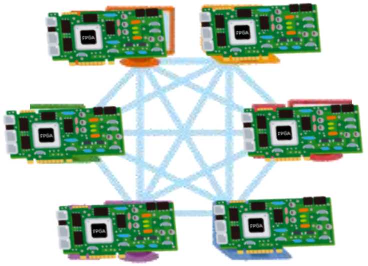
FPGAs have yet to be Mainstream in HPC.

System architecture not matured yet.

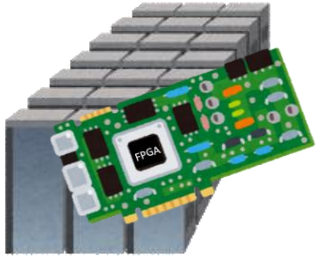
Still have system-level challenges for FPGA-based HPC

Productive customizability for computing HW

Performance scalability with multiple FPGAs



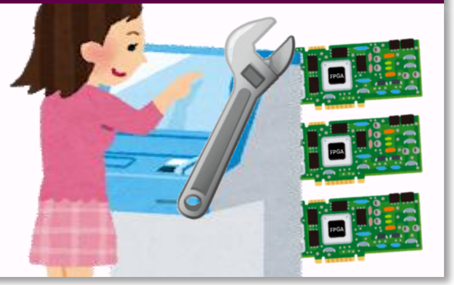
Interoperability with existing HPC systems



Challenges and Approaches for FPGA-based HPC

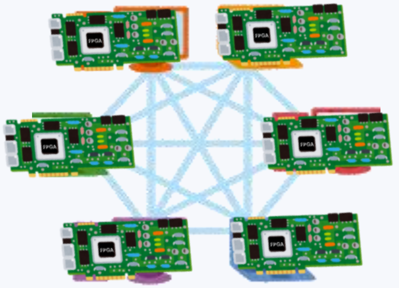
Productive customizability for computing HW

- ✓ Able to implement various hardware (algorithms) on FPGA



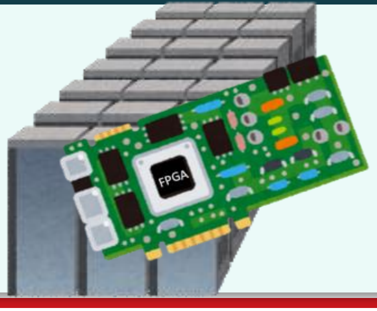
Performance scalability with multiple FPGAs

- ✓ Inter-FPGA communication available
- ✓ Allow users to easily try multi-FPGA applications



Interoperability with existing HPC systems

- ✓ Able to easily extend existing systems with FPGAs
- ✓ Can we extend Supercomputer Fugaku?

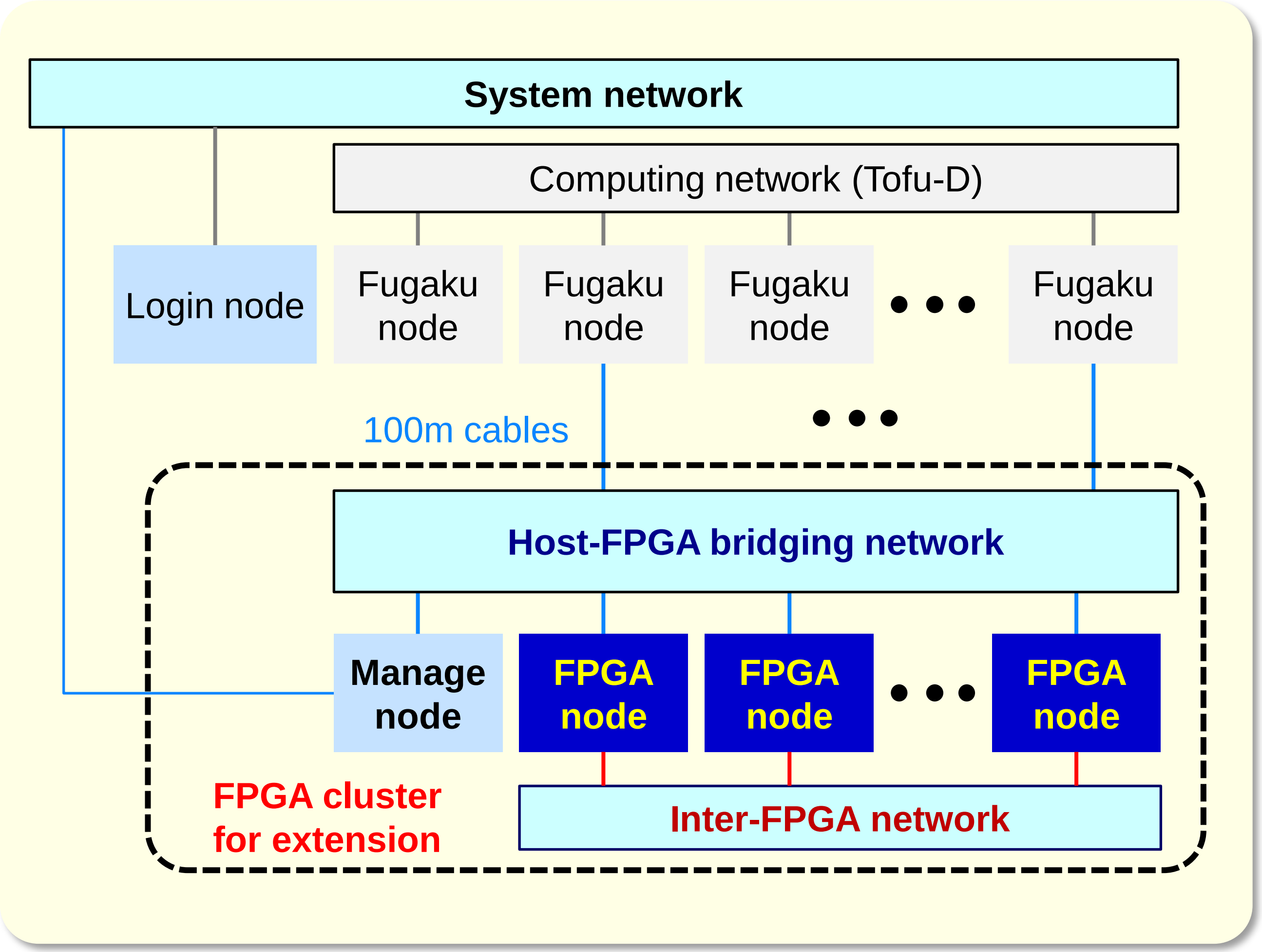


- ➢ No OpenCL (not limit computing models)
- ➢ FPGA Shell & HLS/HDL programming, where any hardware can be easily implemented

- ➢ FPGA Shell supporting high-bandwidth and low-latency network dedicated to FPGAs

- ➢ Software-bridged APIs to access FPGAs remotely through host-FPGA bridging network

Architecture of ESSPER



✓ **Productive customizability**

- ➢ No OpenCL (not limit computing models)
- ➢ FPGA Shell & HLS/HDL programming, where any hardware can be easily implemented

✓ **Performance scalability**

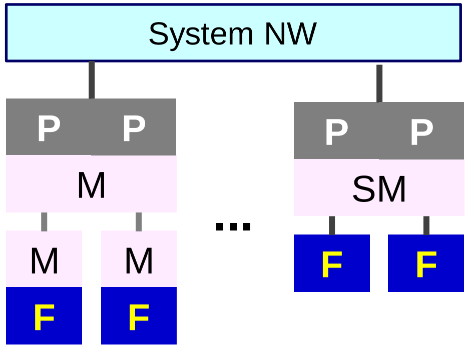
- ➢ FPGA Shell supporting high-bandwidth and low-latency network dedicated to FPGAs

✓ **Interoperability**

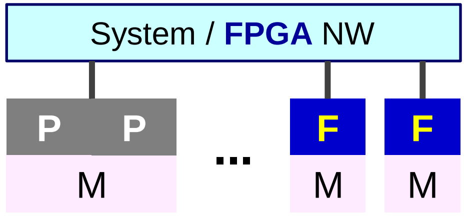
- ➢ Software-bridged APIs to access FPGAs remotely through host-FPGA bridging network

Architecture Classification

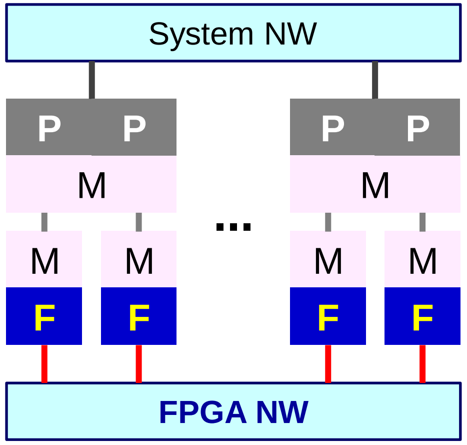
(S)M **(Shared) memory** P **CPU** F **FPGA** NW **Network**



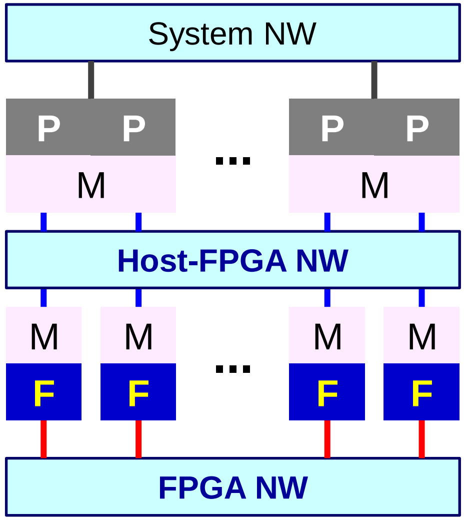
a. Cluster of CPUs with FPGAs (distributed or shared memory)



b. Cluster of CPUs and FPGAs



c. Clusters of CPUs with inter-connected FPGAs



d. Loosely-coupled CPU cluster and FPGA cluster

Our architecture

Related Work : FPGAs Clusters in HPC/DC

FPGA NW Type	Direct network	Indirect network	Indirect circuit-switching nw
Characteristics	p2p-connection without switches, typical: torus	connection with switches, typical: Ethernet	connection with optical switch (MEMS)
Switching	circuit or packet (w/ router)	packet	circuit or packet (w/ router)
Pros	**low latency**	**flexibility, small diameter**	**low latency, flexibility**
Cons	inflexibility, large diameter	higher latency, complex	expensive, signal attenuation

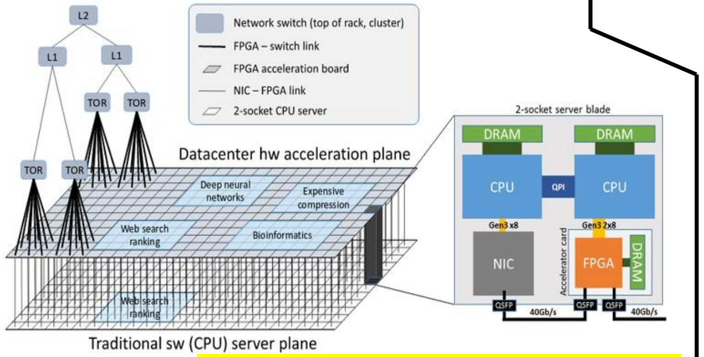
Representative systems

Cygnus @ U of Tsukuba

Archi-C

Novo-G# @ U of Florida

Archi-C



Catapult @ Microsoft

Archi-B

Noctua @ Paderborn U

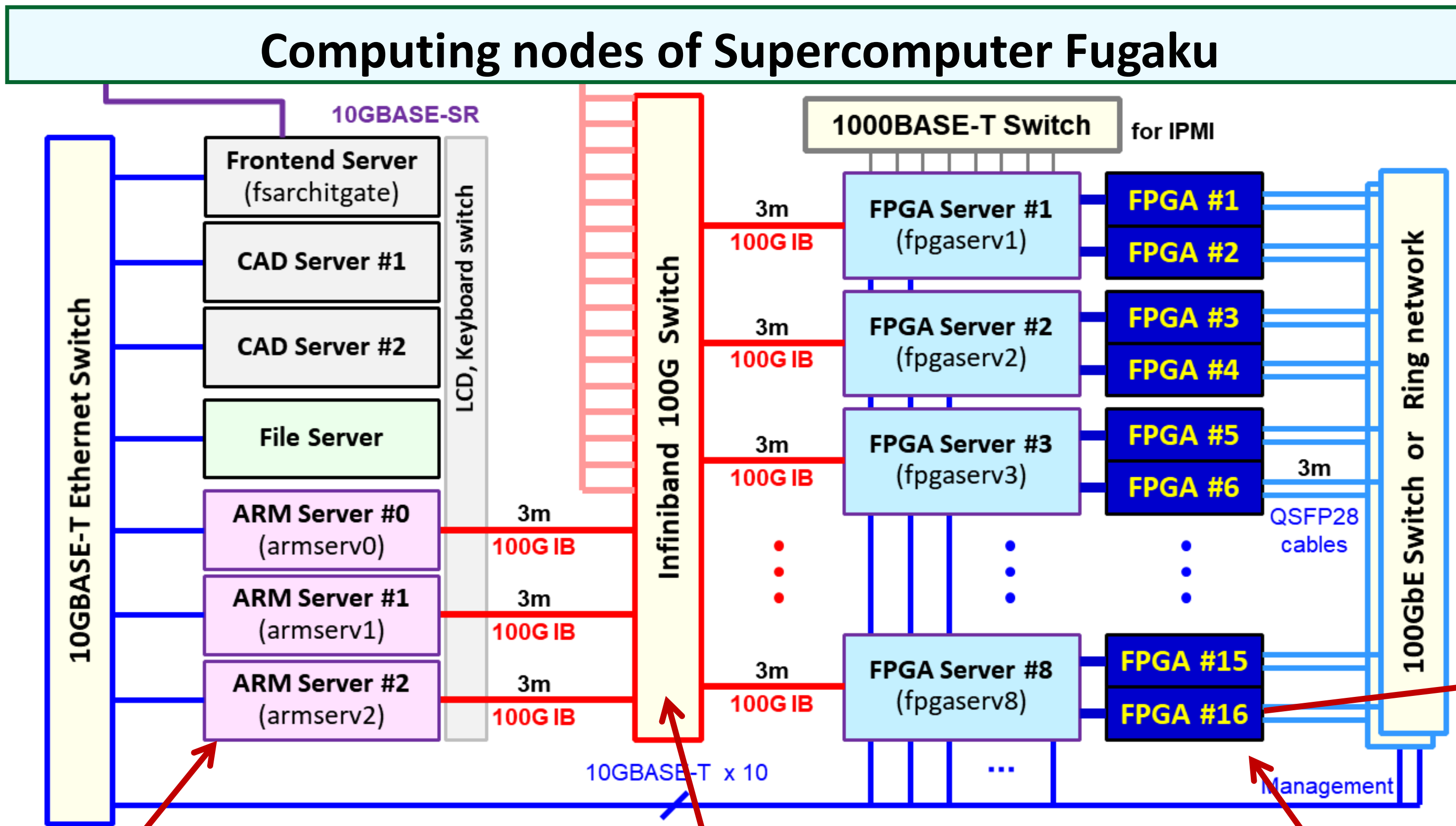
Archi-C

ESSPER

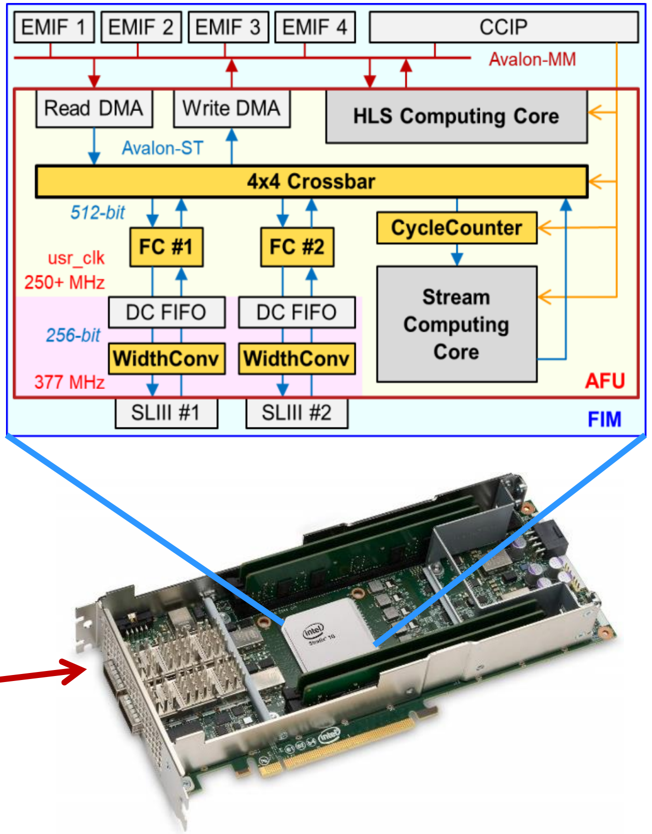
Elastic and Scalable System for High-Performance Re-configurable Computing

System Design

Hardware Organization of ESSPER



FPGA Shell



Service servers

- CAD servers
- Storage server
- ARM servers

CPU - FPGA network

- 100G Infiniband
- Software-bridged driver (R-OPAE)

FPGA cluster

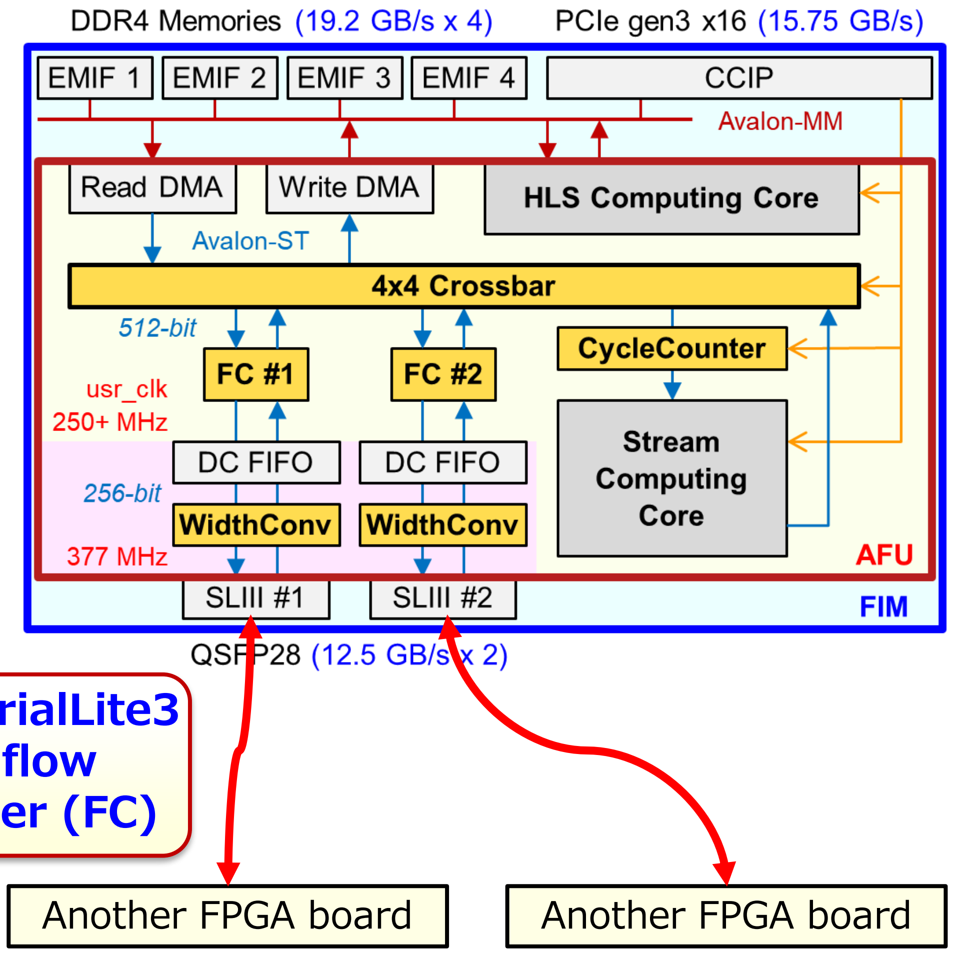
- x86 host servers
- FPGA boards
- Inter-FPGA network

FPGA Shell (SoC)

- AFU Shell design
- User HW modules can be embedded for custom computing.

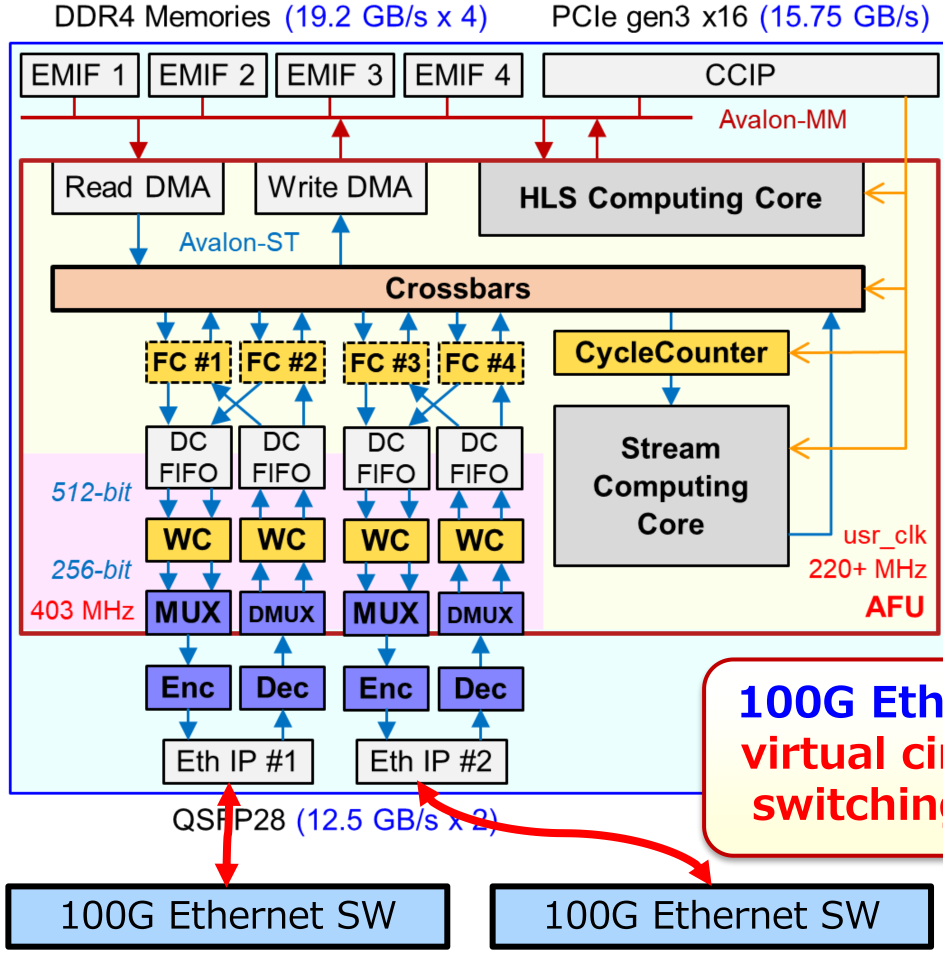
FPGA Shells for Direct and Indirect Networks

Direct connection network (DCN)



100G SerialLite3 with flow controller (FC)

Indirect network (VCSN)



100G Ethernet virtual circuit-switching nw

Tomohiro Ueno, Atsushi Koshiba, Kentaro Sano, "Virtual Circuit-Switching Network with Flexible Topology for High-Performance FPGA Cluster," Procs. of ASAP, pp.41-48, 2021.

System Stack of ESSPER

High-level application

- Including both SW & HW

Low-level application

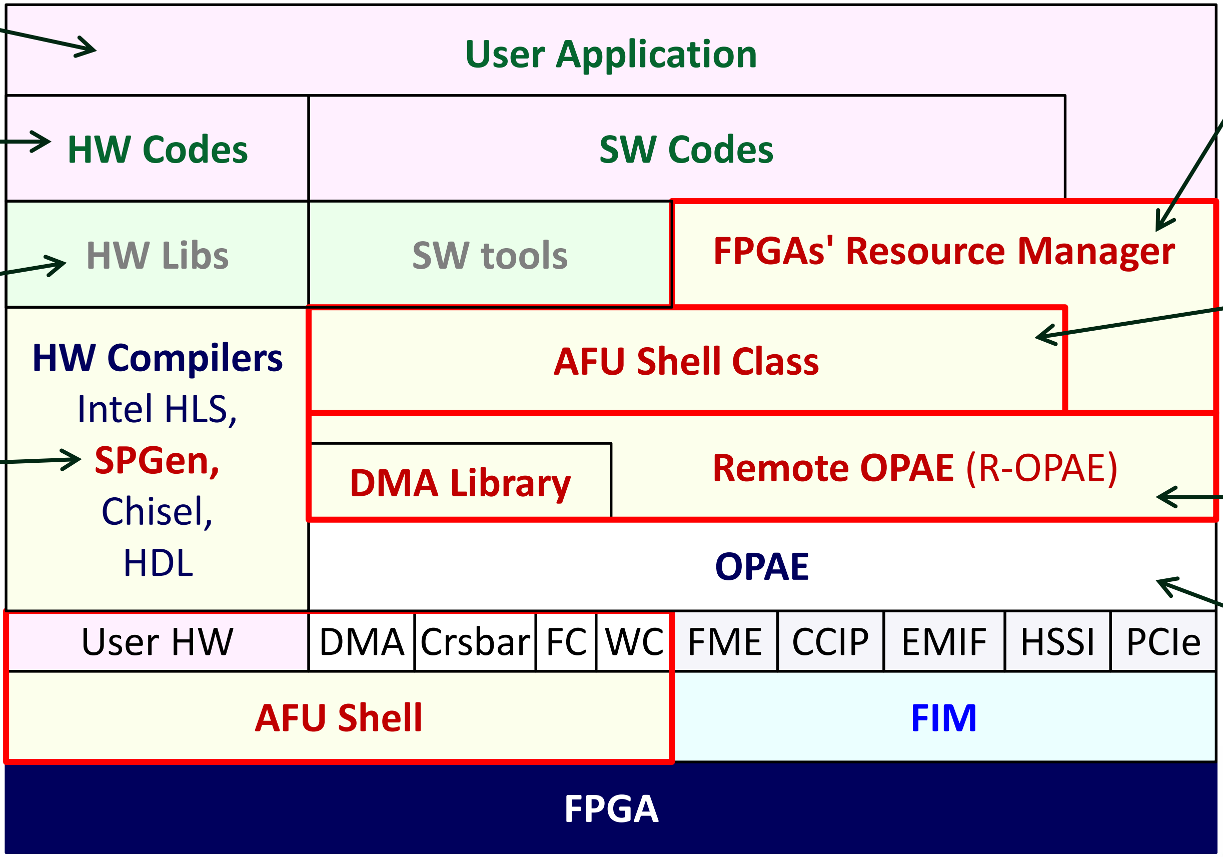
- Separately-written HW & SW

Hardware, software libs/tools

- Pre-implemented functions

Programming tools

- SPGen : DSL for stream/ systolic computing



User Application

HW Codes | SW Codes

HW Libs | SW tools | FPGAs' Resource Manager

HW Compilers Intel HLS, SPGen, Chisel, HDL

AFU Shell Class

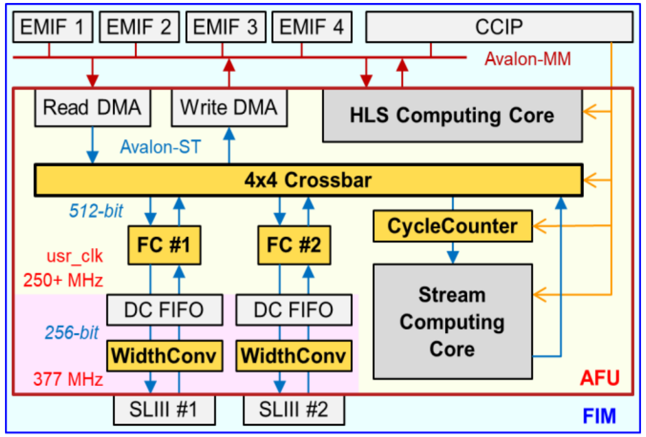
DMA Library | Remote OPAE (R-OPAE)

OPAE

User HW | DMA | Crsbar | FC | WC | FME | CCIP | EMIF | HSSI | PCIe

AFU Shell | FIM

FPGA



Resource manager

- Search and allocate resources of multiple FPGAs
- FPGA network management / control

AFU Shell class

- Object of AFU shell
- Abstraction of HW

Remote OPAE

- Software bridge using Infiniband Verbs

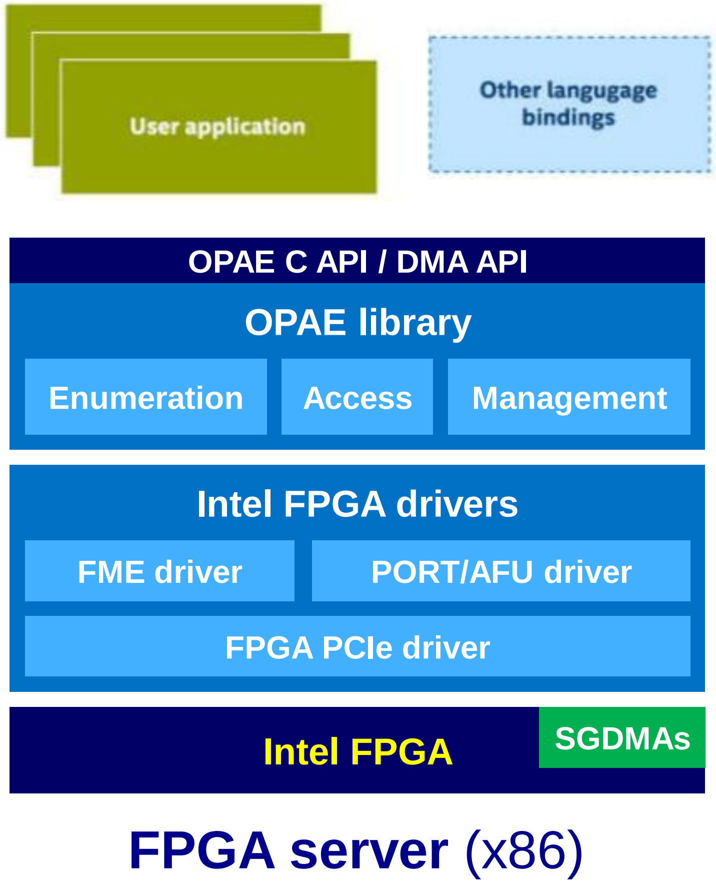
OPAE

- Low-level driver

Remote-OPAE (for remote FPGA Access)

Software bridge for FPGAs over Infiniband

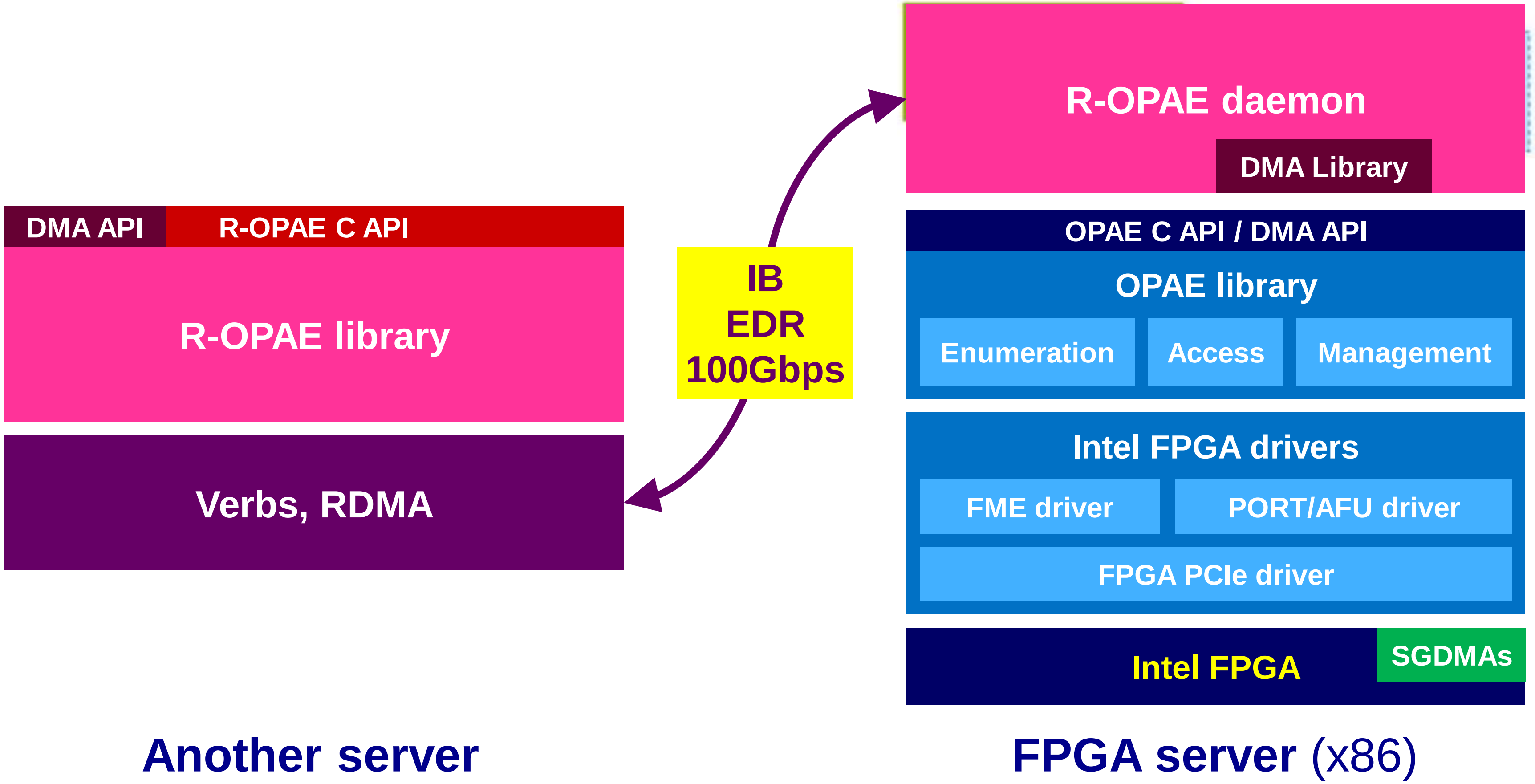
- ✓ **OPAE**: Open Programmable Acceleration Engine (PCIe FPGA driver)



Remote-OPAE (for remote FPGA Access)

Software bridge for FPGAs over Infiniband

- ✓ **OPAE**: Open Programmable Acceleration Engine (PCIe FPGA driver)
- ✓ 99% of OPAE APIs are supported.
- ✓ We can use any FPGAs in a system via IB as if they were locally installed.



R-OPAE as Software-based Resource Disaggregation

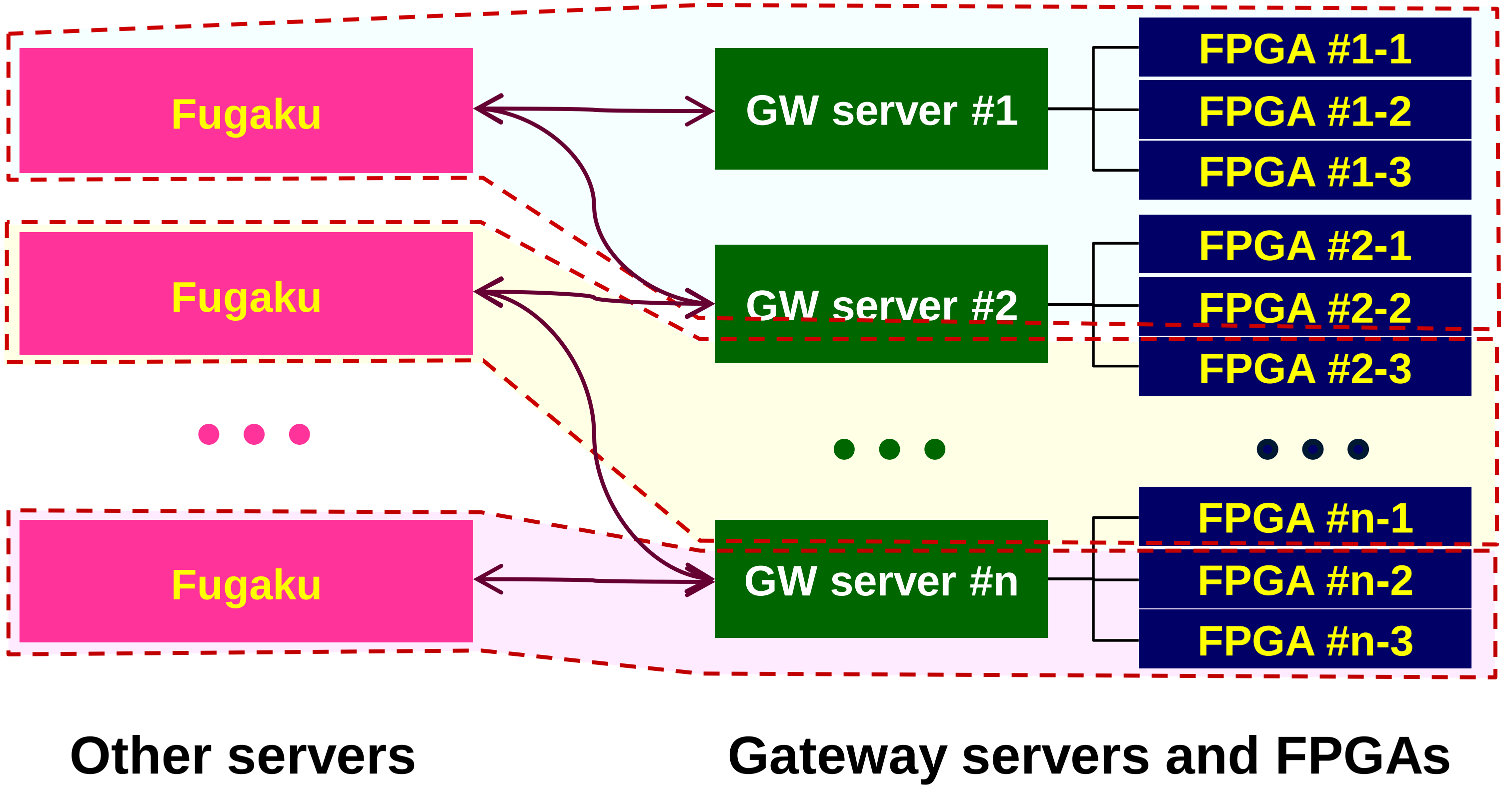
Transparent access to remote FPGAs

Flexible utilization:

- ✓ Can use any available FPGA resources

Inter-operability and extensibility:

- ✓ Vendor/ISA-independent
- ✓ Operable with various architectures such as Fugaku (ARM)



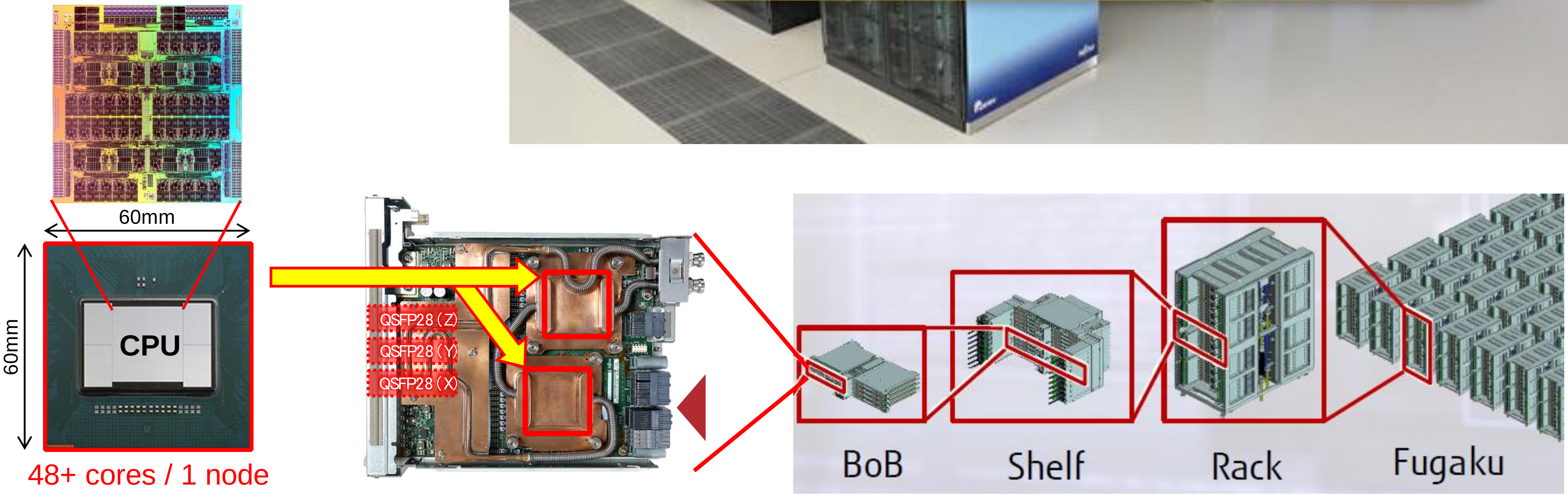
ESSPER

Elastic and Scalable System for High-Performance Re-configurable Computing

Proof-of-concept and evaluation

Supercomputer Fugaku

Modern Supercomputers are based on Many-core CPUs (& GPUs).



48+ cores / 1 node
2.7+ TF

CPU-Memory Unit (CMU)
2 nodes
5.4+ TF

16 nodes
43+ TF

48 nodes
129+ TF

384 nodes
1+ PF

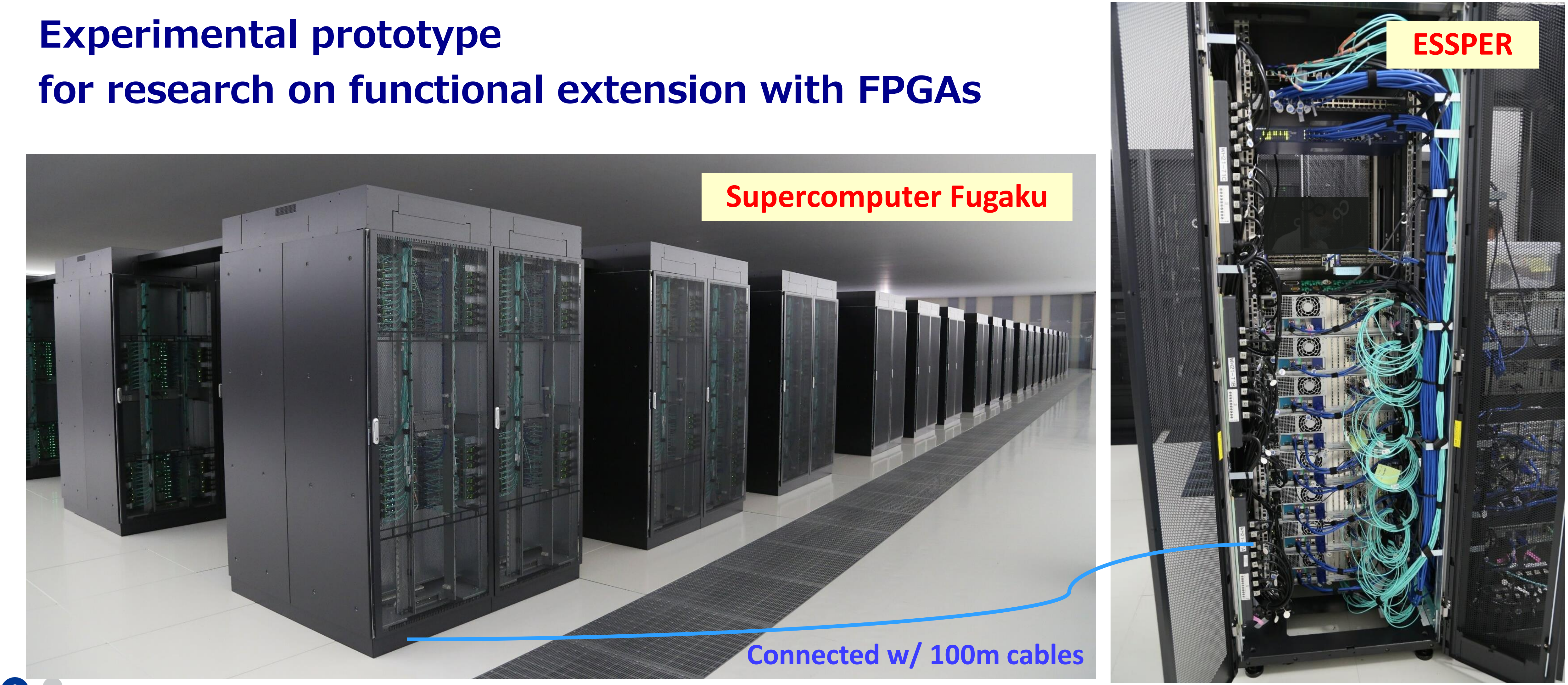
158,976 nodes
537 PF @ FP64
(414 racks)

Photos & figs by Fujitsu

Elastic and Scalable System for High-Performance Reconfigurable Computing

Elastic and Scalable System for High-PErformance Reconfigurable Computing

Experimental prototype for research on functional extension with FPGAs



Supercomputer Fugaku

ESSPER

Connected w/ 100m cables

ESSPER

Elastic and Scalable System for High-Performance Re-configurable Computing

Elastic and Scalable System for High-PErformance Reconfigurable Computing

Open-Access paper

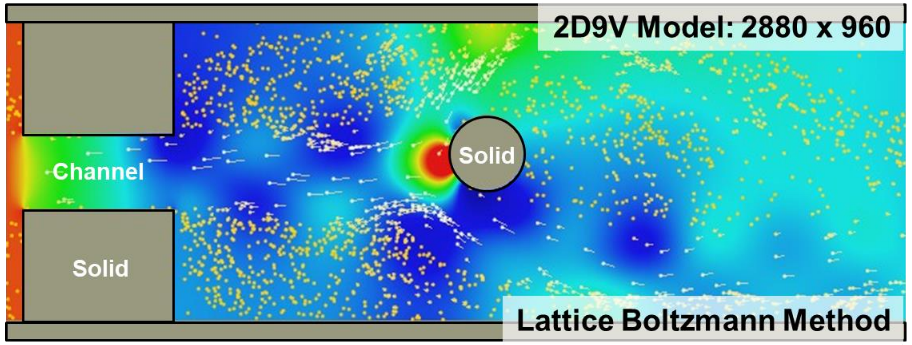
ESSPER

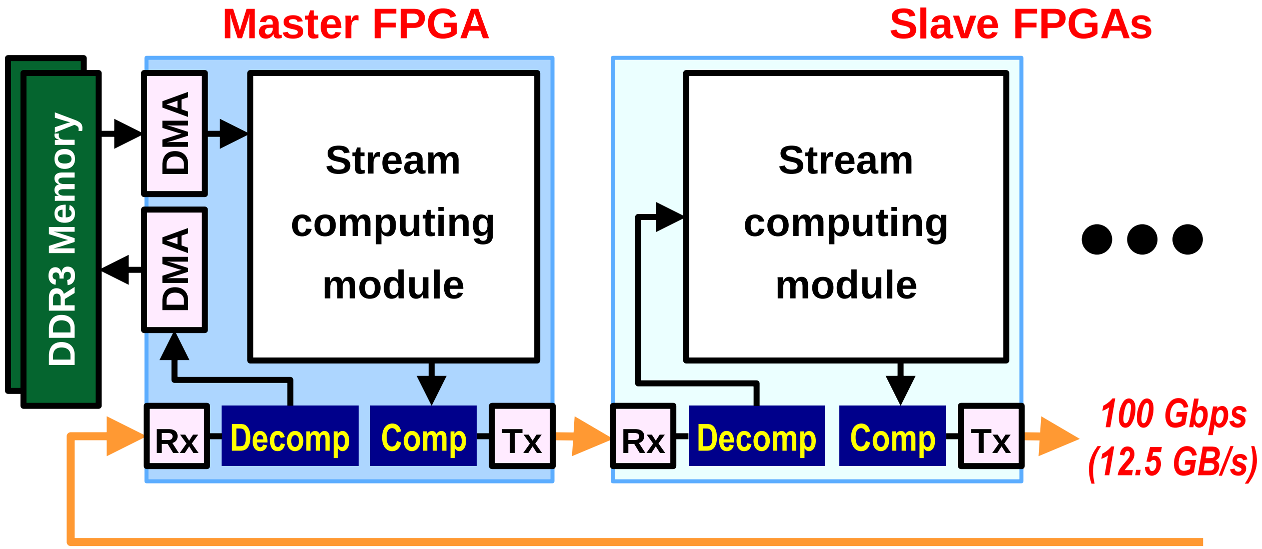
Elastic and Scalable System for High-Performance Re-configurable Computing

Applications, Joint Research Projects

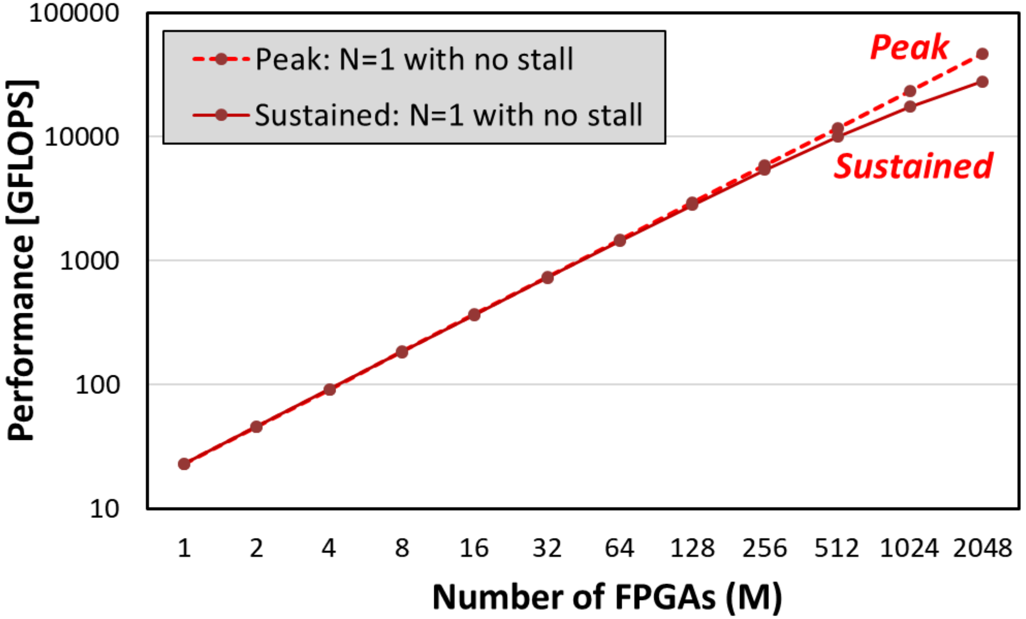
Ringed FPGAs for Deeper Pipelining

- **Deeply-pipelined FPGAs with 1D ring**
  - ✓ Linear array of Stratix10 FPGAs
  - ✓ Pipelining works well for almost linear speedup if data stream is sufficiently large.





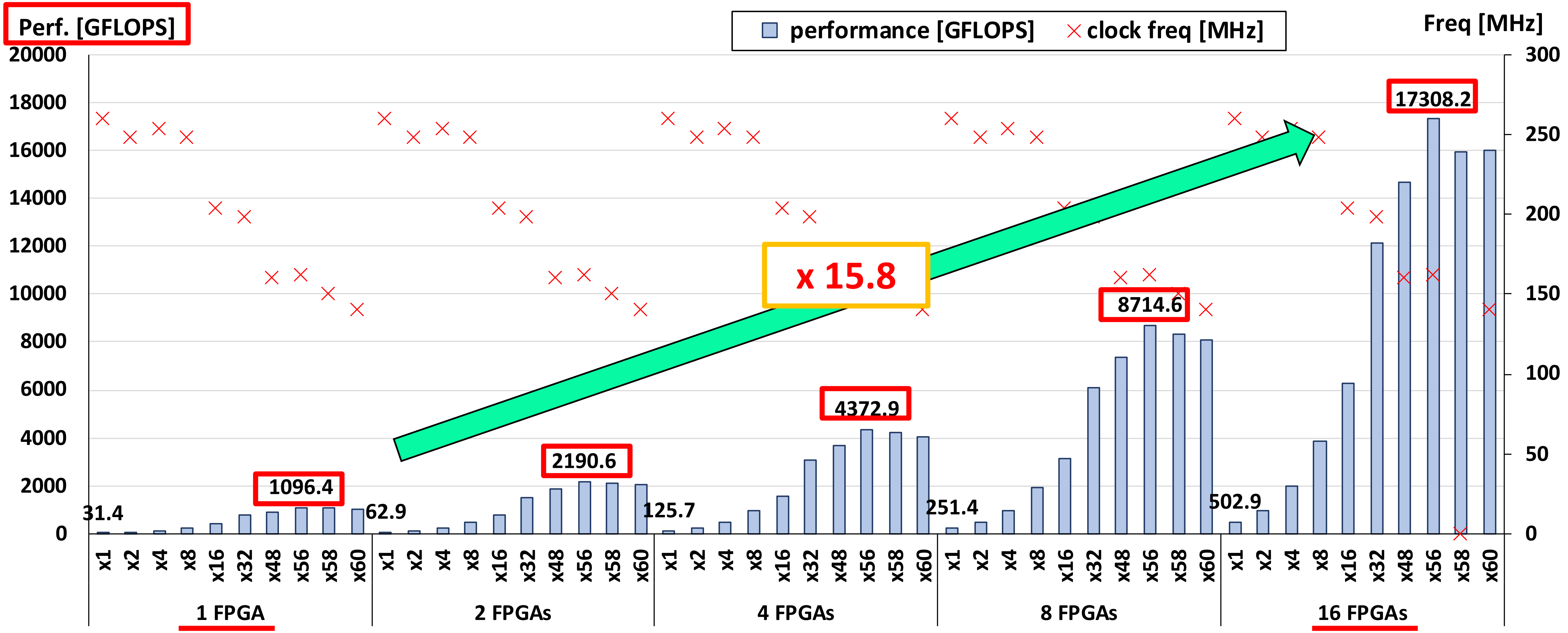
Block diagram of FPGAs in a ring



Performance model for Arria10 FPGAs

Performance of 2D LBM with 100Gbps Ring NW

Computational performance (FLOPS) when processing about 2GB data



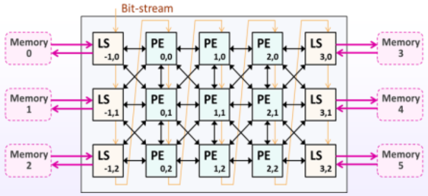
Lessons Learned with ESSPER

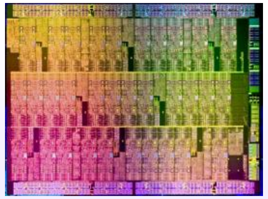
Open-Access paper

- **FPGA-based reconfigurable computing works.**
- **Productivity is not high, especially for multiple FPGAs.**
  - ✓ Even HLS requires know-how on optimizing computation and memory access.
  - ✓ Lack of debugging tool, and simulation environment.
- **Can obtain scalability, but absolute performance in FP is lower than competitors (GPUs)**
  - ✓ FPGA-bases system development takes time while GPUs are being further advanced.
  - ✓ For fixed domain of computing *(such as HPC in FP and AI workloads)*, FPGAs are redundant with more area, more power, and lower frequency with low memory bw.
- **Concept should be Okay for reconfigurable data-flow computing, but implementation approach could be improved : CGRA instead of FPGA?**

2. Exploration of New HPC Architectures

- ✓ Data-flow-based accelerators (**CGRA**)
- ✓ **Next-generation HPC systems**



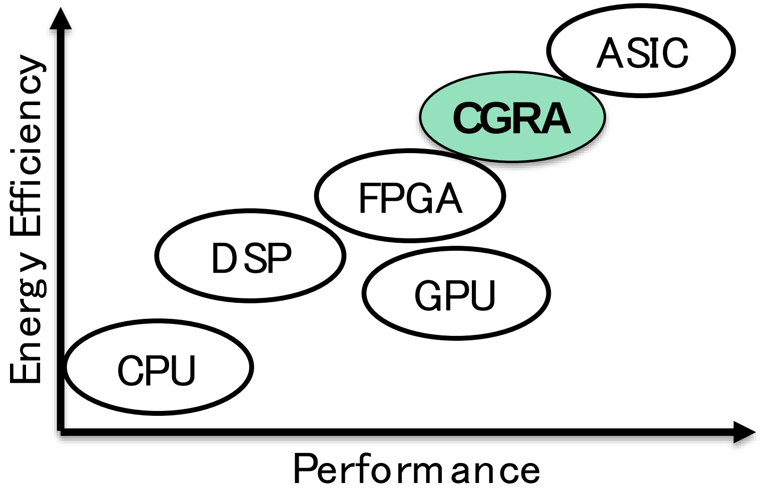


Exploration of New HPC Architectures

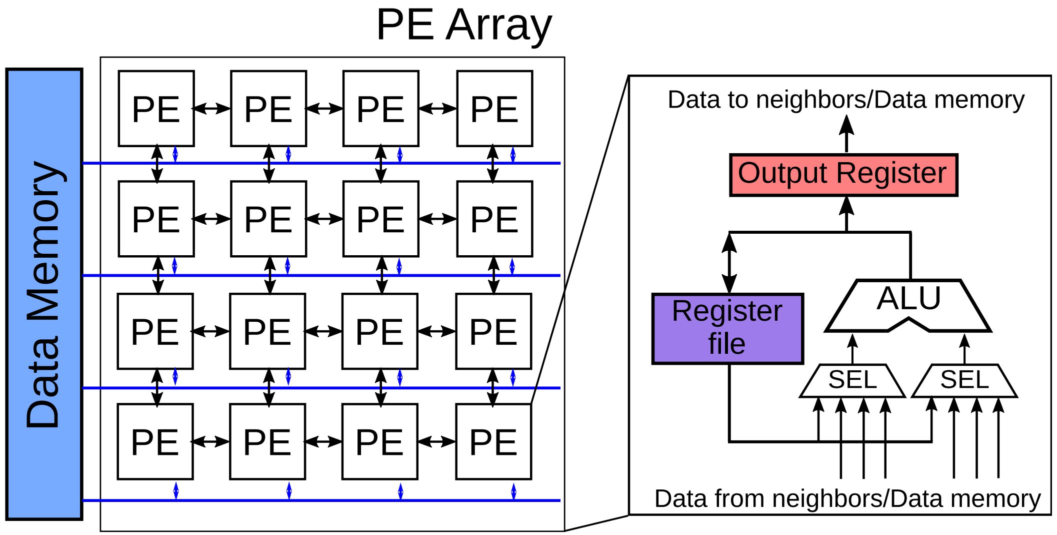
Data-flow-based accelerators (CGRA)

Coarse-Grained Reconfigurable Array (CGRA)

- **Architecture for reconf. data-flow computing**
  - ✓ Composed of an array of processing elements (PEs), where we can map DFGs for computing
  - ✓ Provide a word-wise reconfigurability (e.g., 32-bit)
  - ✓ Higher energy efficiency than FPGAs (of bit-level)
  - ✓ Performance close to ASIC-based accelerators
- **Application area of CGRAs**
  - ✓ Traditionally, targeted for lower-power embedded apps, e.g., image processing
  - ✓ Recently, expected for hi-performance AI
- **Questions**
  - ✓ CGRAs also promising for HPC?
  - ✓ What architecture/design decision required HPC?



Comparison with other architectures [1]


PE Array
Data Memory
PE
Data to neighbors/Data memory
Output Register
Register file
ALU
SEL
SEL
Data from neighbors/Data memory

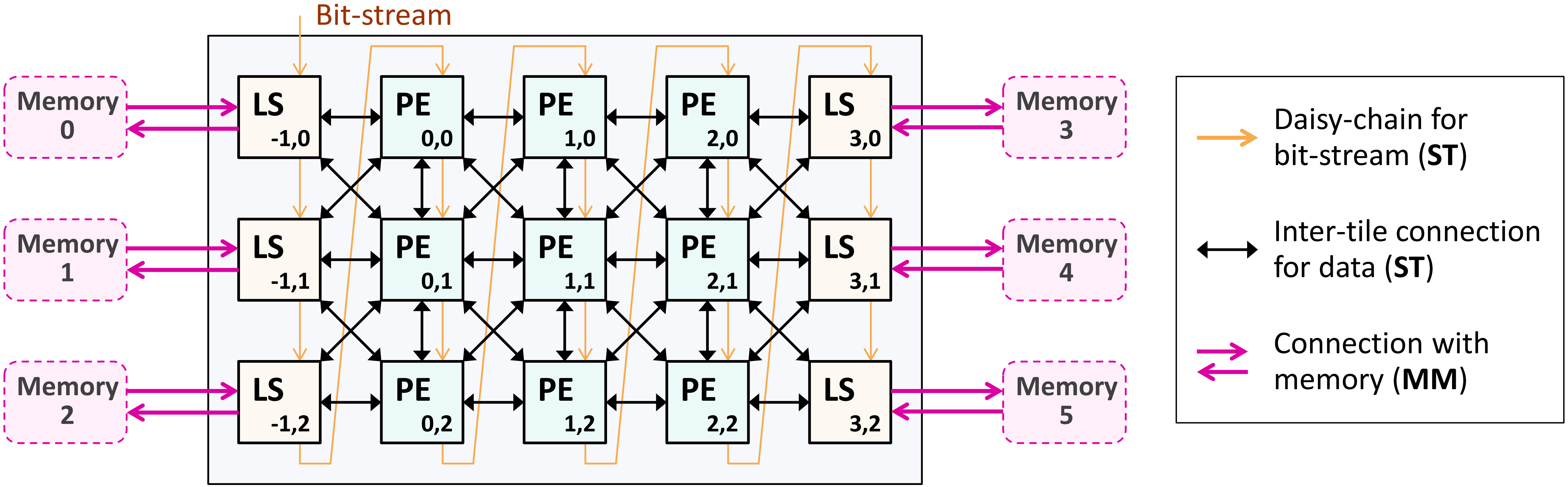
General structure of the CGRAs [2]

[1] Liu, Leibo, et al. "A survey of coarse-grained reconfigurable architecture and design: Taxonomy, challenges, and applications." *ACM Computing Surveys (CSUR)* 52.6 (2019): 1-39.

[2] Takuya Kojima, et al., "Exploration Framework for Synthesizable CGRAs Targeting HPC: Initial Design and Evaluation," Procs. CGRA4HPC, May 30-June 3, 2022.

RIKEN CGRA Architecture (baseline)

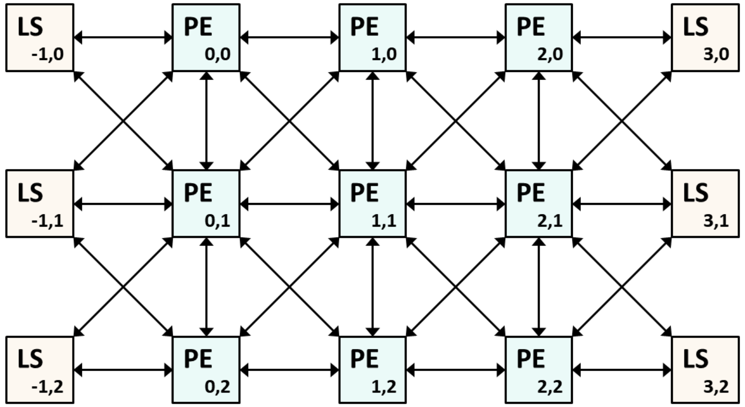
- **HPC-oriented CGRA with the following design philosophy**
  - ✓ Modular design for design space exploration with various architecture configuration and sizes
  - ✓ Isolation between computation in a PE array and memory access with load-and-store (LS) tiles
  - ✓ Capability of floating-point operations for HPC apps



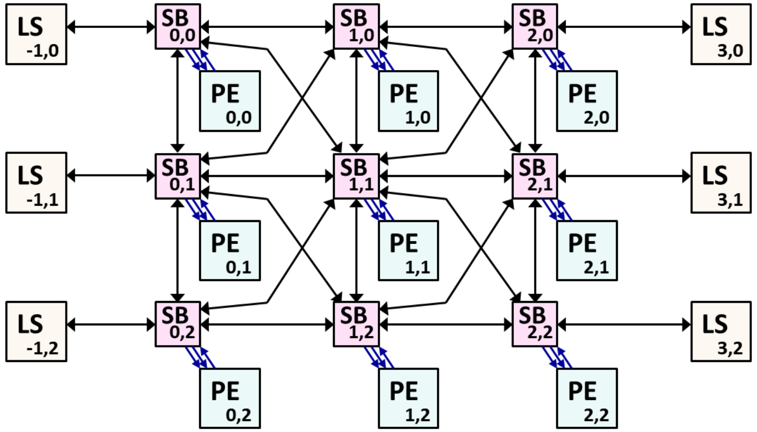
Past work:
CGRA Designs with Embedded routers (ER) or Discrete routers (DR)

Design Decision on Intra-CGRA Interconnects

Embedded router in PE



Discrete router as switch block (SB)



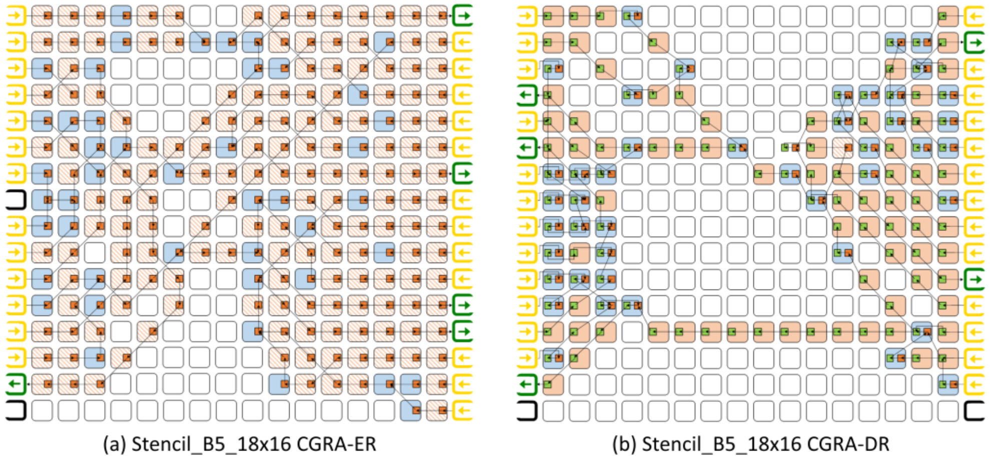
CGRA with Embedded Router (CGRA-ER)	CGRA with Discrete Router (CGRA-DR)
Routers in each PE mediate communication between PEs	Discrete switch blocks for communication between PEs
Simpler in design with smaller area and higher frequency	More complex design with large area and lower frequency
May lead to wastage of computing resources: When mapping complex graphs, some PEs must be configured to bypass data without computation	More efficient: allow more PEs to be used for computing: By relieving PEs from compulsory data bypass with better routability with switch blocks
Ex) ADRES, CGRA-ME, HiPreP, and MorphoSys	Ex) HyCube, RAW (& the Tilera Processor), and Plasticine

Which one is suitable for HPC considering computing efficiency and HW area?

Objective and Contributions

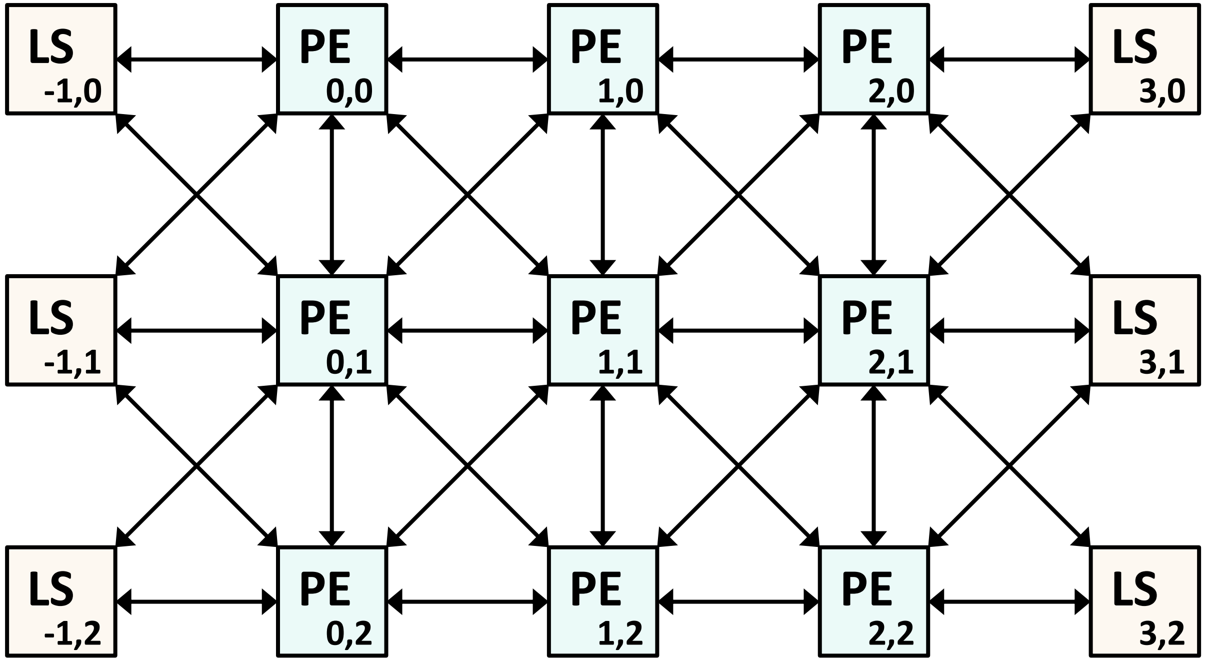
- **Objective** **Evaluate the positive and negative aspects of the different routing architectures: CGRA-ER and CGRA-DR for HPC apps**

- **Contributions**
  - ✓ Parameterized implementation of CGRA-ER and CGRA-DR using our baseline architecture
  - ✓ Evaluation and verification of benchmarks (DFGs) by RTL simulation with CGRA-evaluation framework (our previous work)
  - ✓ Comparison between CGRA-ER and CGRA-DR
    - ➢ Difficulty in place-and-route of DFGs
    - ➢ PE utilization for computation
    - ➢ Hardware resource consumption

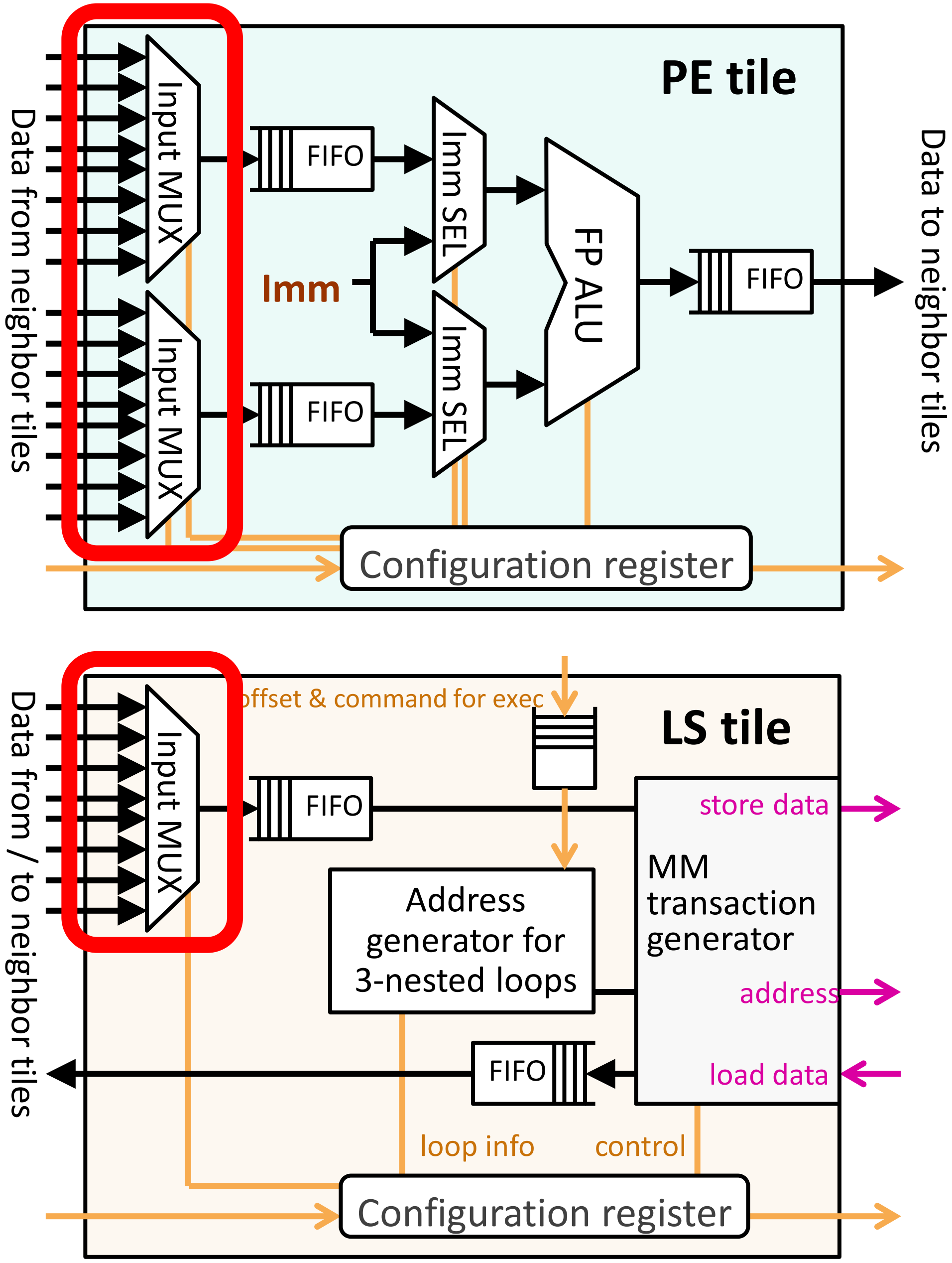

(a) Stencil_B5_18x16 CGRA-ER

(b) Stencil_B5_18x16 CGRA-DR

CGRA with Embedded Router (CGRA-ER)


LS -1,0
PE 0,0
PE 1,0
PE 2,0
LS 3,0
LS -1,1
PE 0,1
PE 1,1
PE 2,1
LS 3,1
LS -1,2
PE 0,2
PE 1,2
PE 2,2
LS 3,2

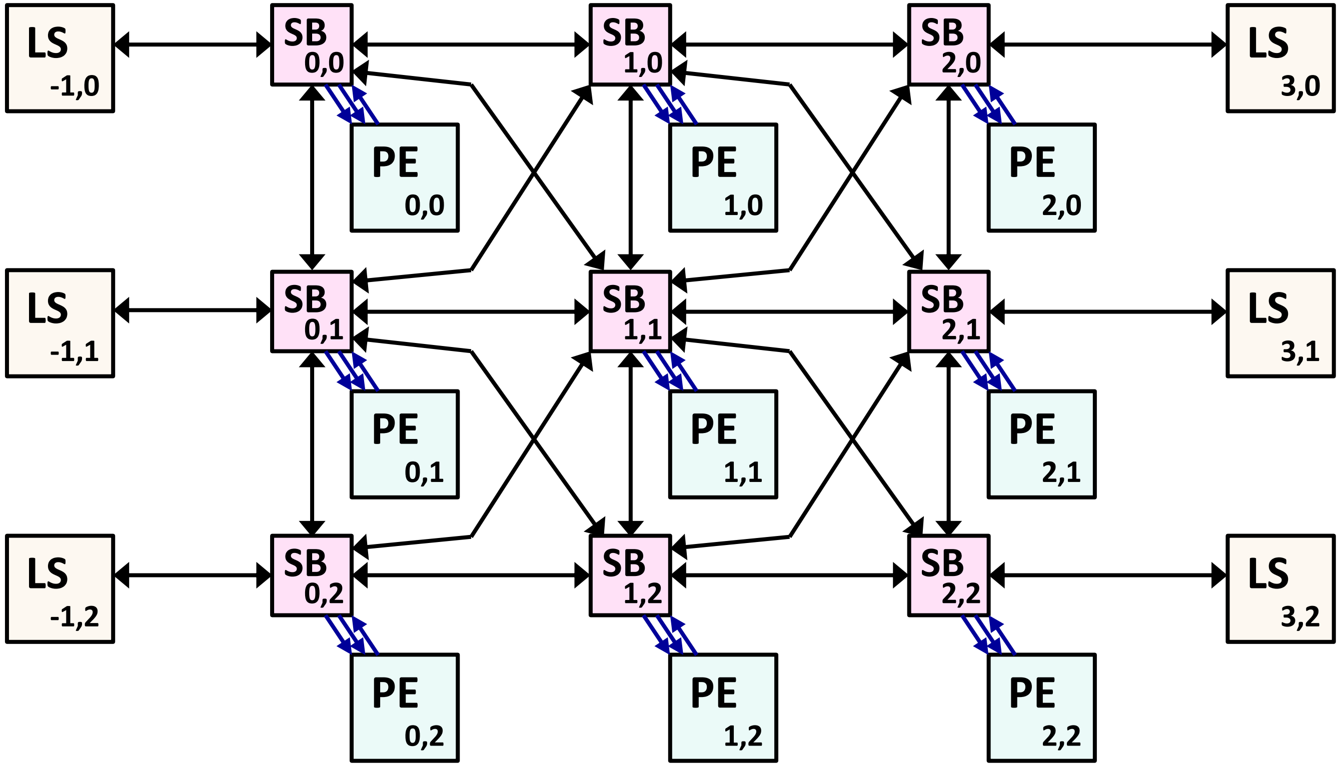
- ✓ PEs and LSs are directly connected to each other.
- ✓ Limited routing capability causes inefficient PE utilization.
- ✓ Wasted PEs for routing with NOP to bypass data through (No computing)


Embedded multiplexors

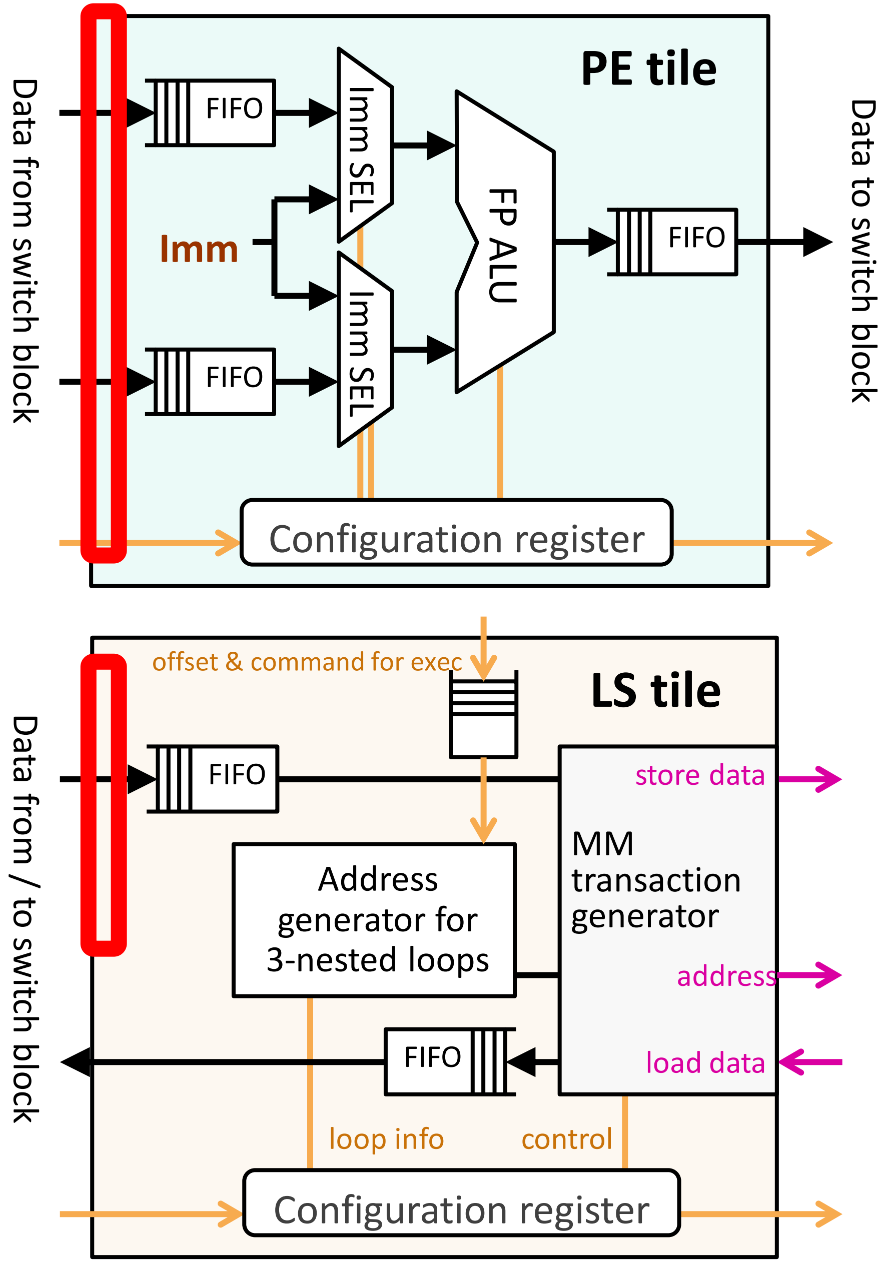
PE tile
Data from neighbor tiles
Input MUX
Input MUX
FIFO
FIFO
Imm
Imm SEL
Imm SEL
FP ALU
FIFO
Data to neighbor tiles
Configuration register

LS tile
Data from / to neighbor tiles
Input MUX
offset & command for exec
FIFO
store data
MM transaction generator
Address generator for 3-nested loops
address
FIFO
load data
loop info
control
Configuration register

CGRA with Discrete Router (CGRA-DR)



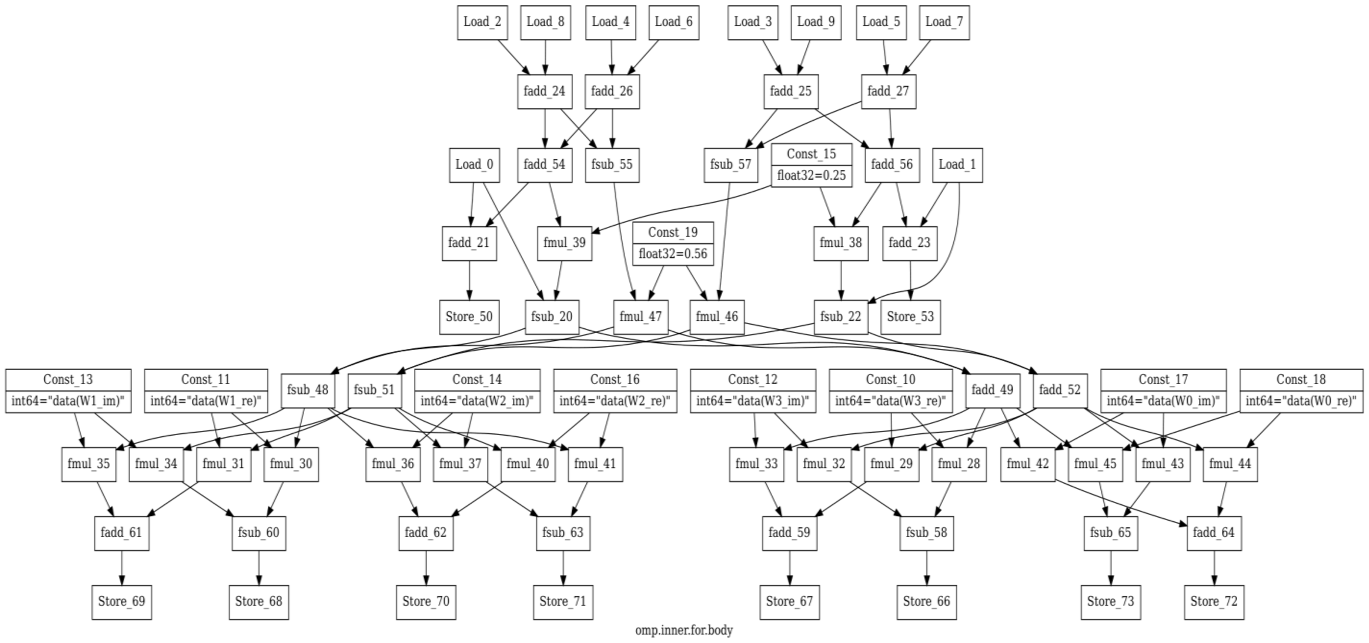
No embedded multiplexor



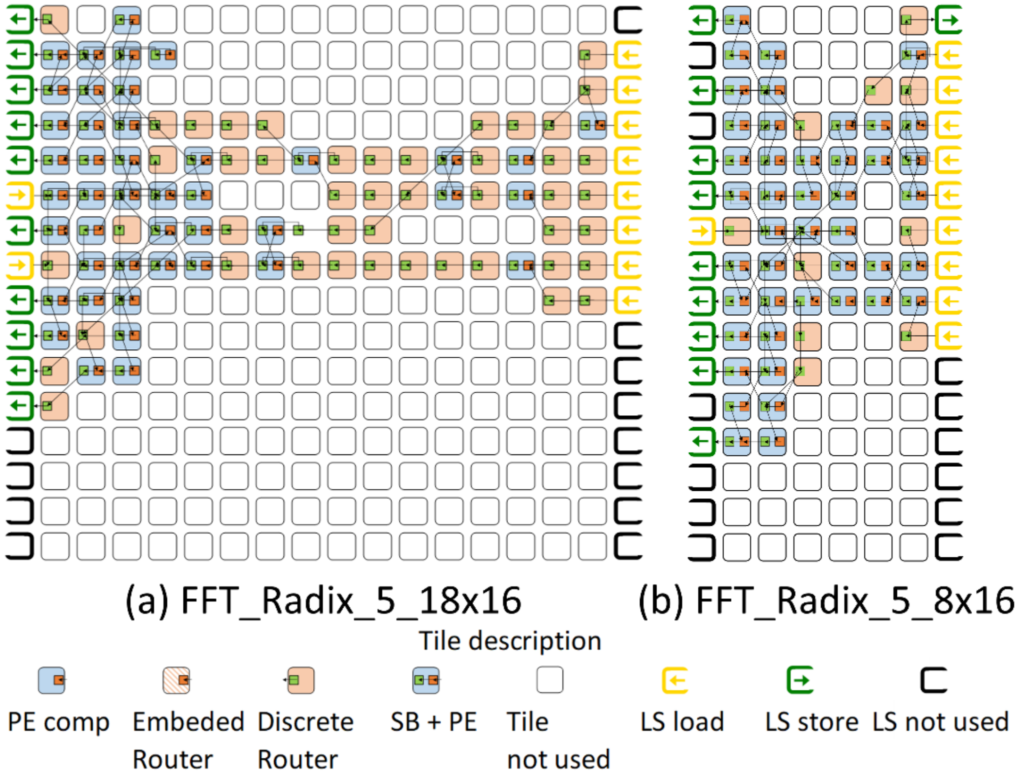
- ✓ Switch blocks as discrete router
- ✓ Simpler PE and LS tile with no multiplexor
- ✓ Higher hardware resources required for SBs

Routing Flexibility vs. Complex Kernel

- **Stockham Radix-5 FFT Kernel DFG has more edges connecting among nodes.**
  - ✓ **Could not map onto CGRA-ER** ; routing with PEs is not enough.
  - ✓ CGRA-DR allows it to be mapped onto 18×16 CGRA-DR and even smaller one (8×16).



DFG of the Innermost Loop of Stockham Radix 5 FFT kernel

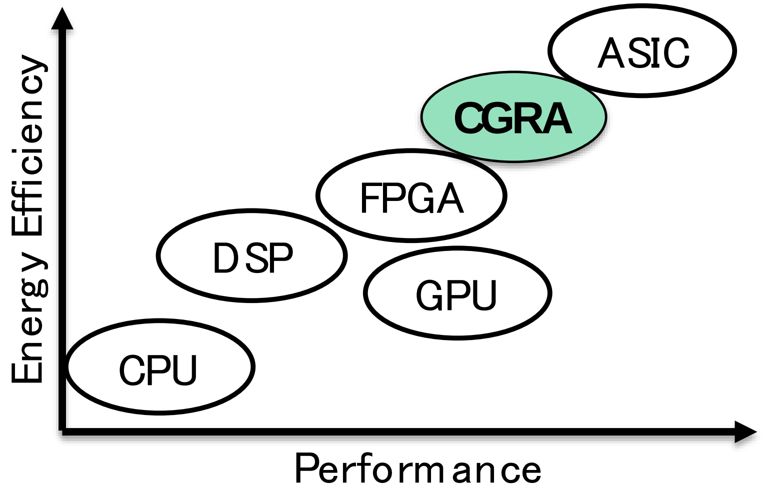

(a) FFT_Radix_5_18x16

(b) FFT_Radix_5_8x16

Stockham Radix 5 FFT kernel mapped on CGRA-DR

Future Work on CGRA for HPC

- **Have not obtained conclusions yet.**
- **Extension of the baseline architecture for practical kernels**
  - ✓ Predication for conditional execution
  - ✓ Heterogeneous architecture with FP div, sqrt, log, and transcendental functions to cover wider range of applications
  - ✓ Programmable buffer for data reuse (such like line/stencil buffer)



Comparison with other architectures [1]

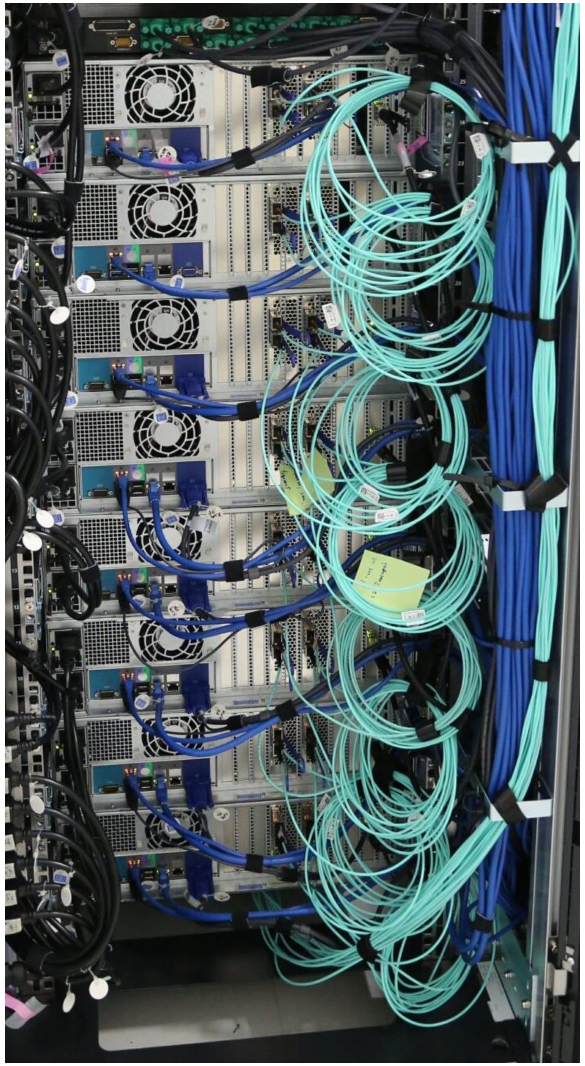
- **Evaluation of operation frequency and hardware resource consumption with FPGA-based and/or ASIC implementation**
  - ✓ Initial rough evaluation results for ASIC
  - ✓ FPGA implementation is also on-going.
    - ➢ Supercomputer Fugaku – CGRA emulation on ESSPER.

Summary

Hiring researchers,
Contact me!

Reconfigurable data-flow computing should be promising for power-efficient HPC.

- ✓ ***FPGA-based HPC testbed; ESSPER*** *(prototype FPGA cluster)*
  - ➢ *Stratix 10 FPGAs*
  - ➢ *FPGA Shell with inter-FPGA network*
- ✓ ***RIKEN CGRA for HPC***
  - ➢ *Baseline architecture*
  - ➢ *Design space exploration for inter-tile connection*



Future work

- ✓ ESSPER2 with Intel Agilex-M FPGA
- ✓ CGRA for HPC and AI, (design for ASIC and compiler)
- ✓ Feasibility study for next-gen supercomputers (conducted)