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

ESSPER: FPGA Cluster for Research on Reconfigurable HPC with Supercomputer Fugaku

Kentaro Sano

Leader, Processor Research Team

Leader, Advanced AI Device Development Unit

Leader, Architecture Research Group in Feasibility Study for FugakuNEXT

RIKEN Center for Computational Science (R-CCS)

Introduce Myself : Kentaro Sano

We're hiring

RIKEN Center for Computational Science

- ✓ Develop and operate Supercomputer Fugaku
- ✓ Facilitate leading edge infrastructures for research based on supercomputers
- ✓ Conduct cutting-edge research on HPC

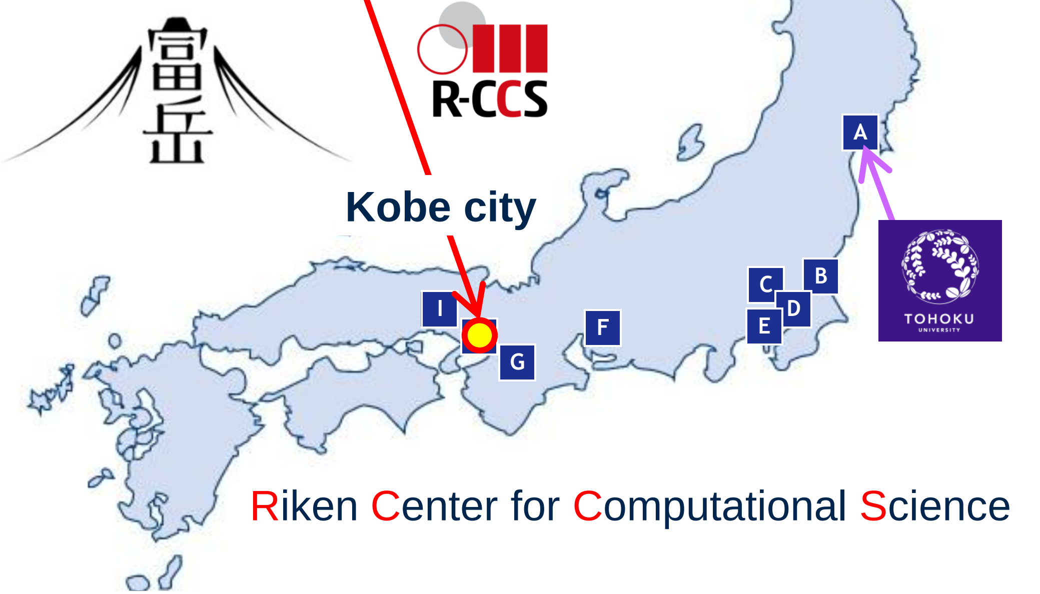
Leader, Processor Research Team
Leader, Next-Gen AI Device R&D Unit

- ✓ Exploration of future HPC&AI architectures

Joint Laboratory at Tohoku University

- ✓ Visiting Professor
  "Advanced Computing Systems Lab"

Supercomputer Fugaku



Riken Center for Computational Science

Processor Research Team, Advanced AI Device Development Unit

PROCT

Goal: Establish HPC & AI architectures suitable in Post-Moore Era

2020
2021
2022
2023
2024
2025
2026
2027

General purpose computing and AI

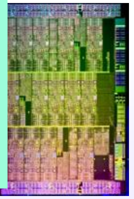
FPGA

1. Advancement of Fugaku

- ✓ Research on **Functional extension with FPGAs** (FPGA cluster development, specialized hardware for HPC)

2. Exploration of new HPC & AI architectures

- ✓ Research on reconfigurable accelerator (e.g. **CGRA**)
- ✓ Research on next-generation **AI chip architecture**

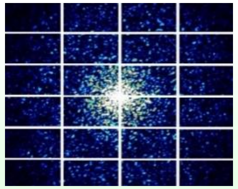


Special purpose computing

GA

3. Near-sensor processing / Scientific edge-computing

- ✓ FPGA-based processing for **X-ray imaging detector** (RIKEN Spring-8)
- ✓ Data-compression hardware for edge-computing (ANL)



FPGA

4. Backend of Fault-Tolerant Quantum Computers

- ✓ Specialized hardware for **quantum error correction** (Hardware algorithms, FPGA demo targeting RIKEN quantum device)

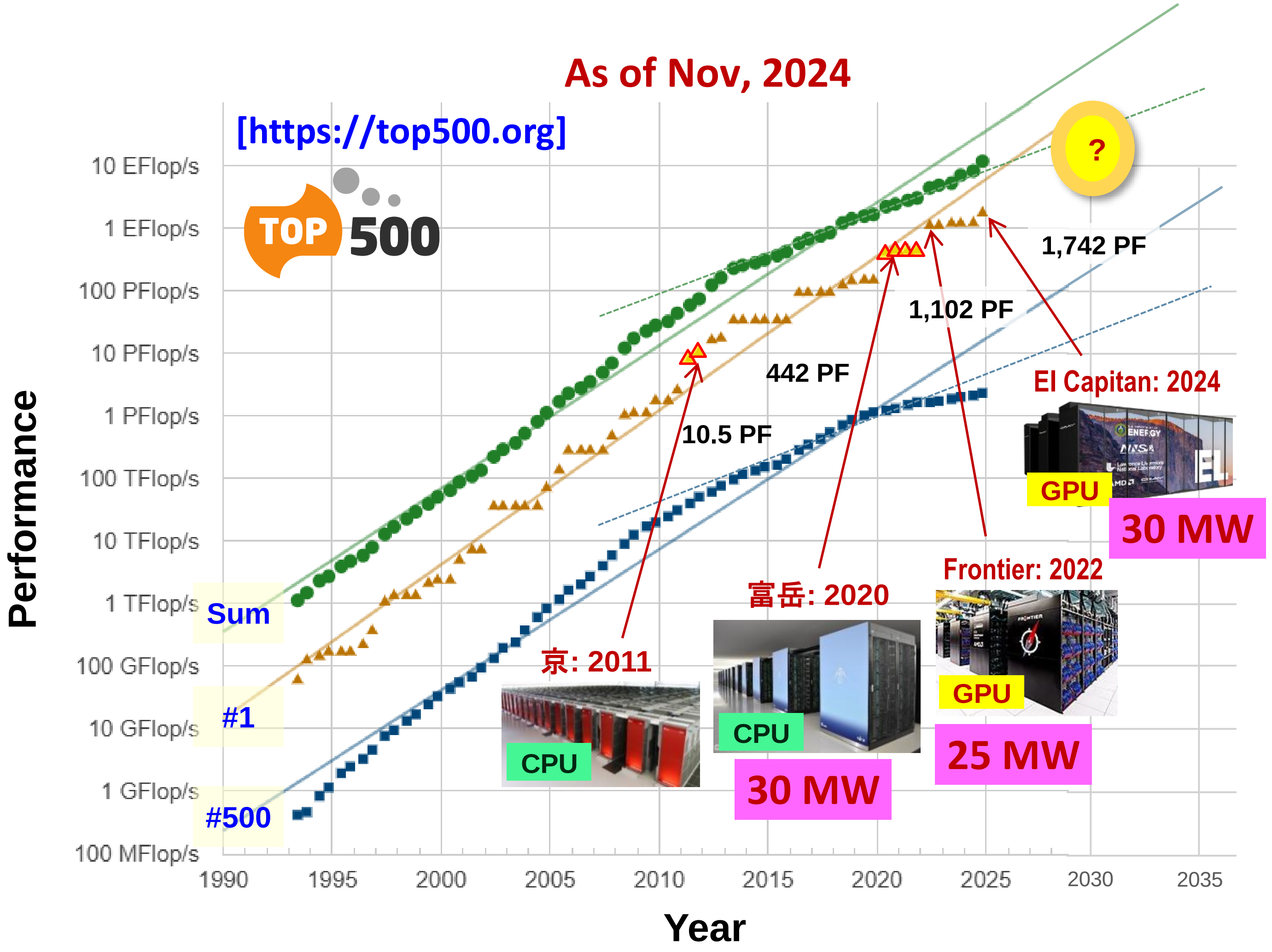
This Talk

- **Introduction**
- **ESSPER: FPGA Cluster for Research on Reconfigurable HPC**
  - ✓ Design concept and system design
  - ✓ Inter-FPGA network
  - ✓ System software
- **Applications**
- **Lessons learned**
- **Plan for ESSPER2**
- **Overview of related topics**
  - ✓ RIKEN CGRA Research
  - ✓ Feasibility Study on FugakuNEXT

Introduction

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

- **System performance is limited by system power.**



Problem : System Power Consumption

Average power consumption

- ✓ in TOP10, TOP50, TOP500

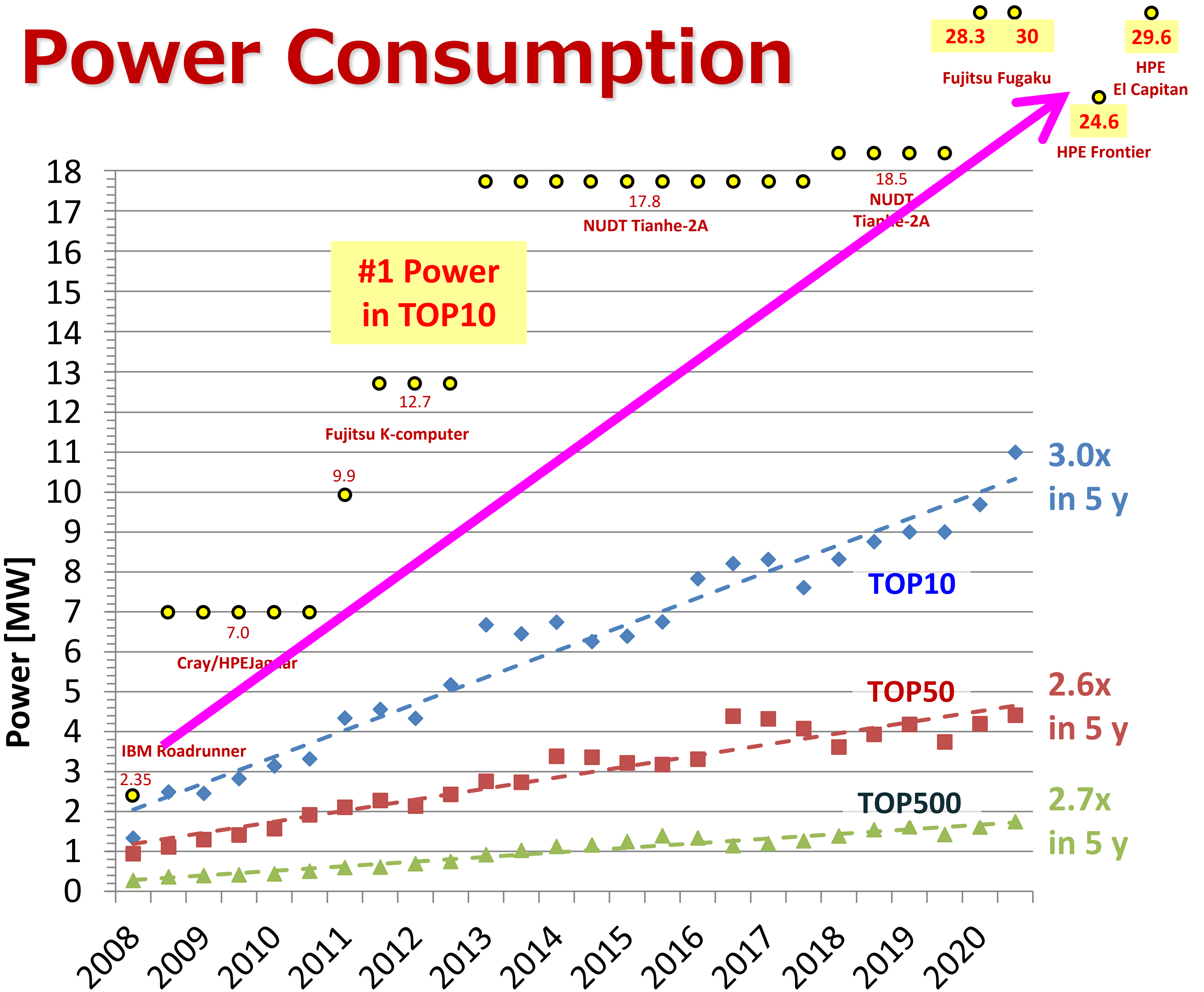
Needed to increase for higher system performance

- ✓ Limited improvement of performance per power

10s of MW for #1 HPL machines

- ✓ **Fugaku 30 MW** for 442 PF
- ✓ **Frontier 24.6MW** for 1353 PF
- ✓ **El Capitan 29.6MW** for 1742 PF

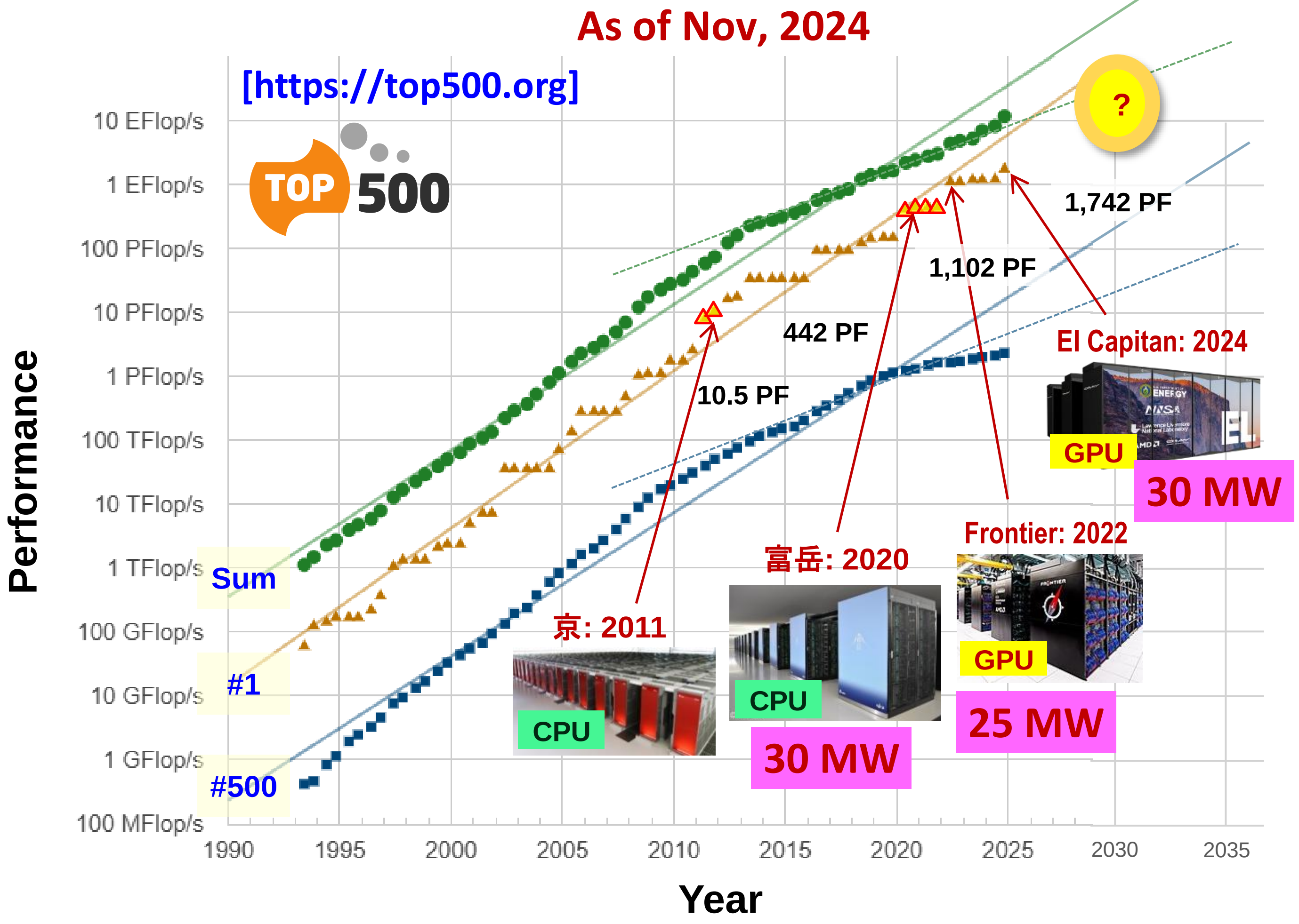
System power budget
= **Critical constraint** of
system performance



Introduction

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

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

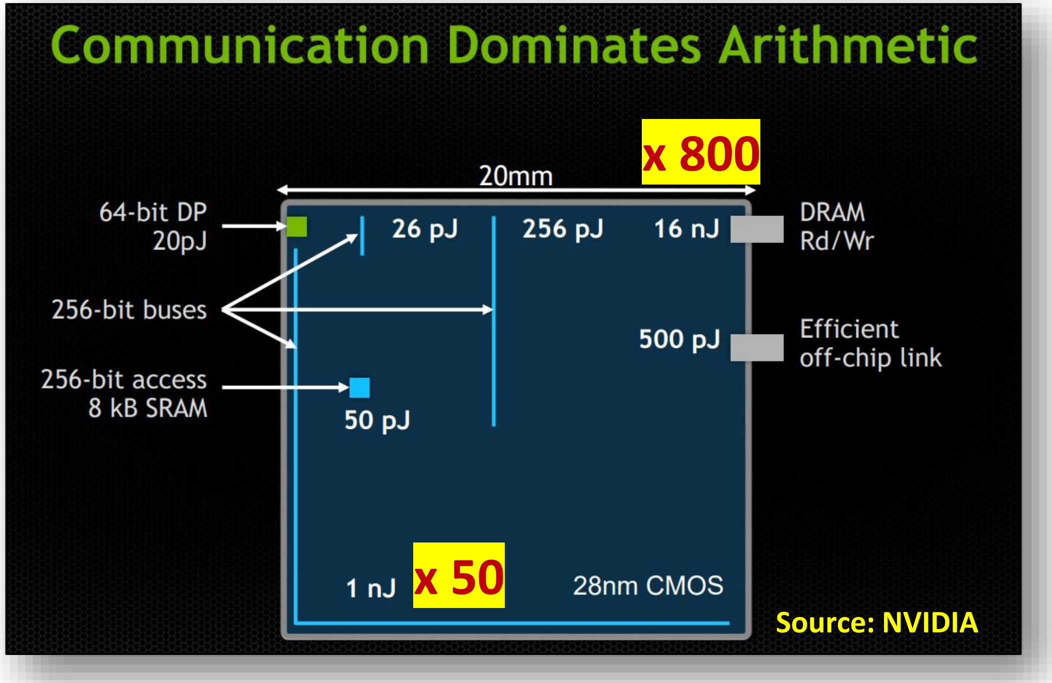

As of Nov, 2024

[https://top500.org]

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

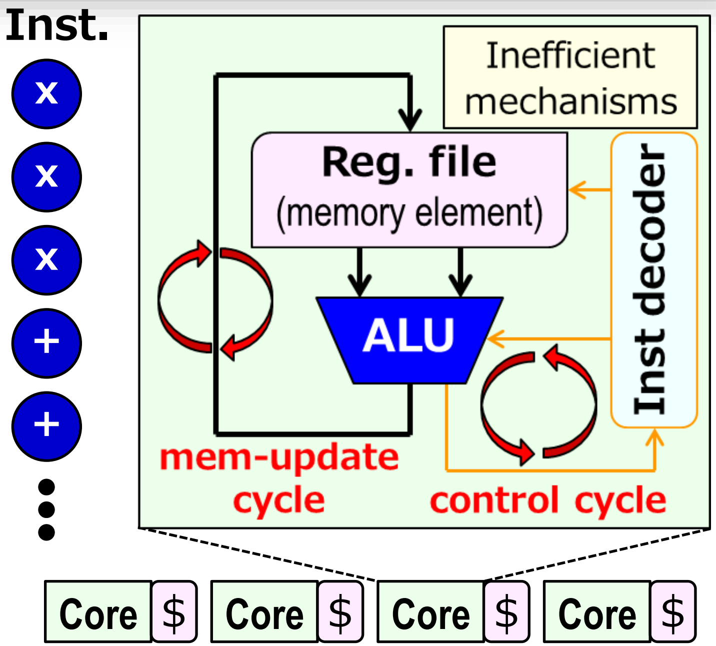
What Eats Power?

- **Data movement rather than computing**
  - ✓ We should remove unnecessary data movement, and make it shorter.
- **Unsuitable architecture**
  resulting in low efficiency and scalability
  - ✓ von-Neumann architectures (CPU & GPU) cannot efficiently scale due to
    - ➢ ***memory-bottlenecked structure;*** such as register files and LLC slices distributed over NoC for multiple cores
    - ➢ ***Extra mechanisms*** consuming power just to increase IPC such as out-of-order, branch predictor, thread scheduler
- **Recent semiconductor scaling cannot save it.**
  - ✓ Power improvement per generation is limited still increasing transistor density for advanced technology nodes like 4, 2, and 1.5nm ...



Source: NVIDIA

von Neumann many-core

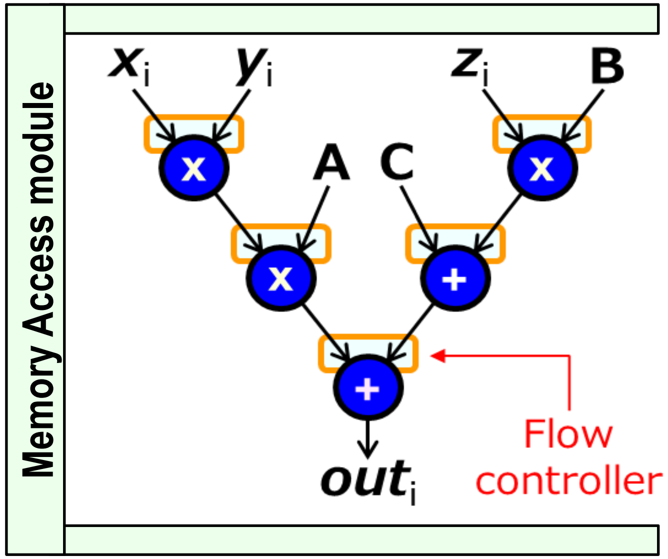


Custom Data-Flow Computing

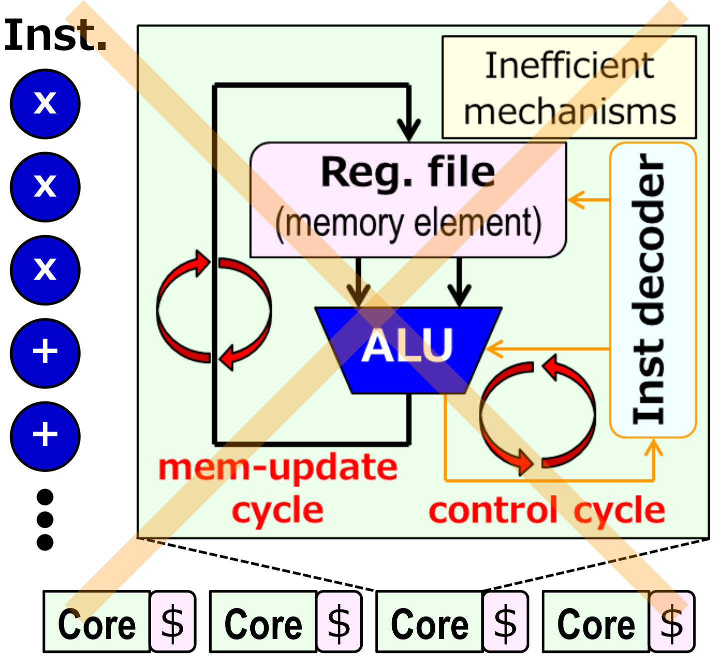
- **Data-flow computing**
  - ✓ Localized data-movement
  - ✓ Lower pressure on memory access with highly pipelined computing by regular data streams
  - ✓ No extra mechanisms for non-computing
- **Customization & reconfiguration**
  - ✓ Higher efficiency by specialization
  - ✓ Programmability for various problems

What technology is suitable for custom data-flow computing? FPGA?

Data-flow computing



von Neumann many-core



1. Advancement of Fugaku

- ✓ Research on **Functional extension with FPGAs** (FPGA cluster development, specialized hardware for HPC)

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

ESSPER: Experimental FPGA Cluster connected with Supercomputer Fugaku

Open-Access paper

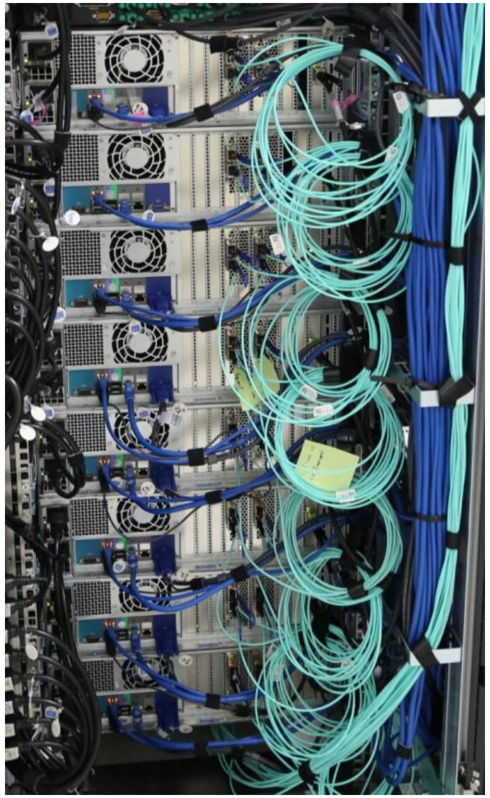
This Work (2020~)

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

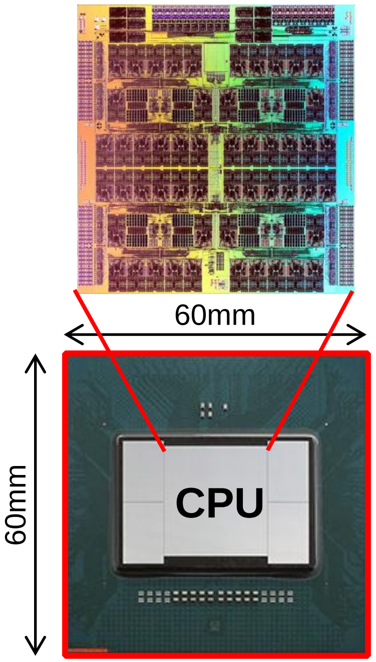
- **ESSPER** : Elastic and scalable FPGA-cluster system
  for high-performance reconfigurable computing

- **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

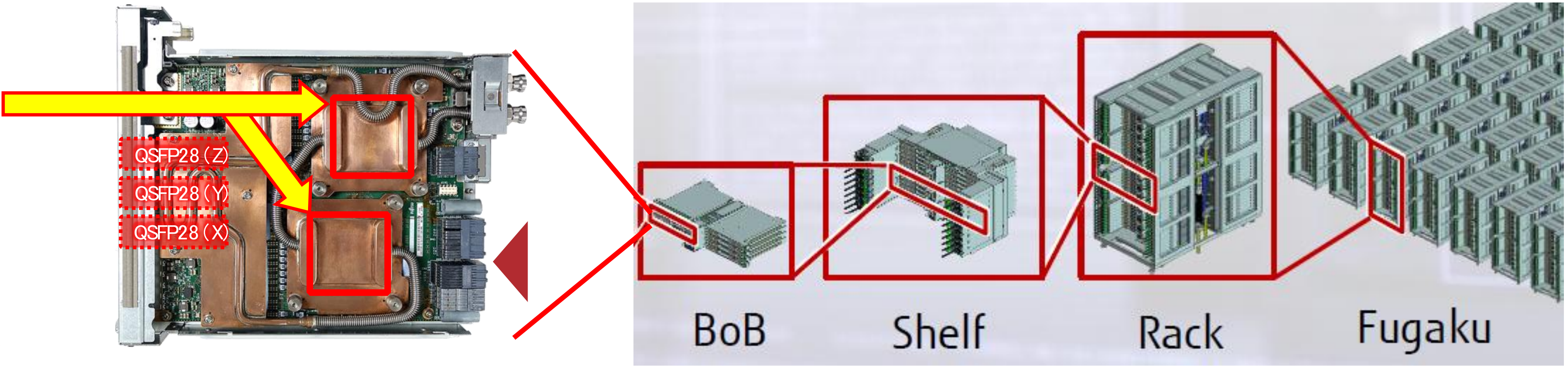
Open-Access paper



Supercomputer Fugaku
(installation in 2020)



48+ cores / 1 node
2.7+ TF



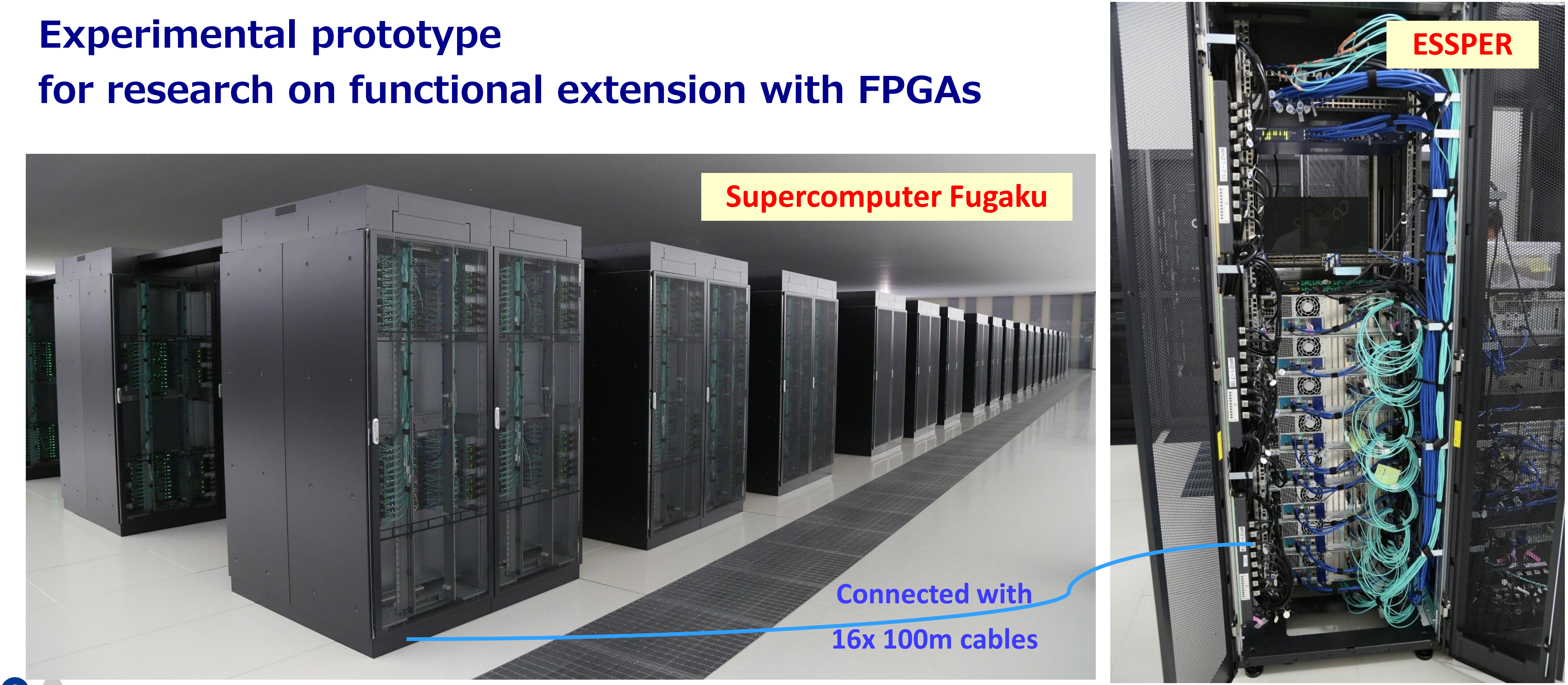
CPU-Memory Unit (CMU)	BoB	Shelf	Rack	Fugaku
2 nodes 5.4+ TF	16 nodes 43+ TF	48 nodes 129+ TF	384 nodes 1+ PF	158,976 nodes 537 PF @ FP64 (414 racks)

Photos & figs by Fujitsu

Elastic and Scalable System for High-PErformance Reconfigurable Computing

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

Experimental prototype for research on functional extension with FPGAs



ESSPER

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

Design concept and related work

Challenges and Approaches for FPGA-based HPC

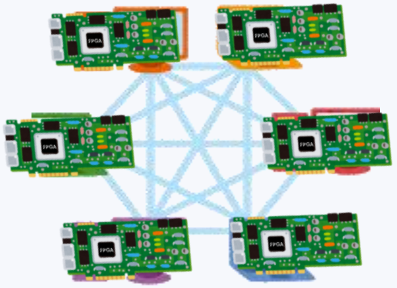
Productive customizability for computing HW

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

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

Performance scalability with multiple FPGAs

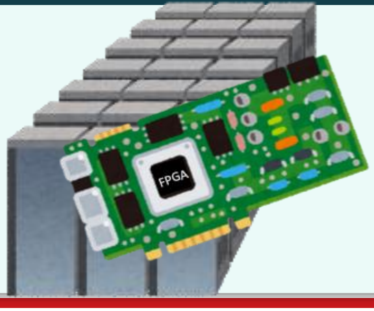
- ✓ Inter-FPGA communication available
- ✓ Allow users to easily try multi-FPGA applications



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

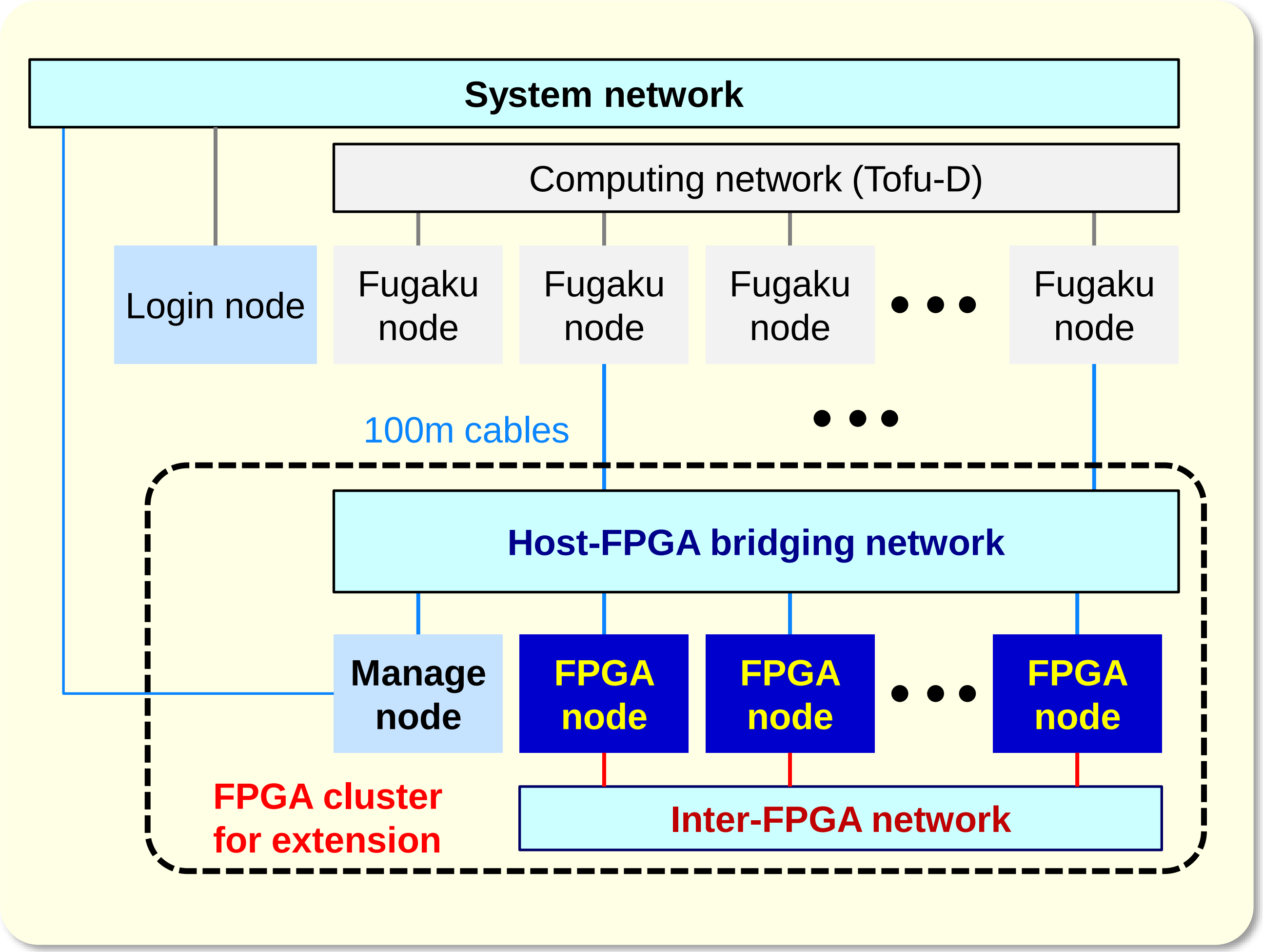
Interoperability with existing HPC systems

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



- ➢ 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

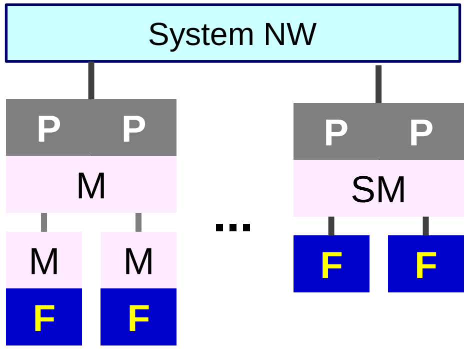
✓ **Interoperability**

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

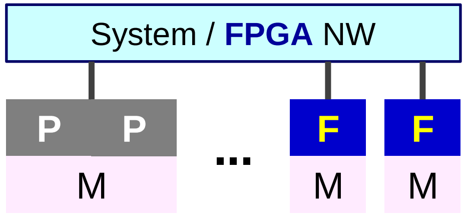
Architecture Classification

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

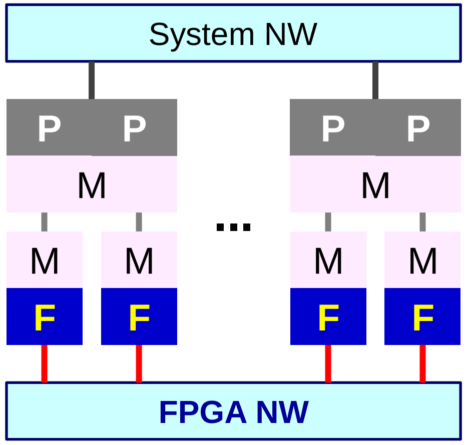
Existing supercomputer w/o FPGAs



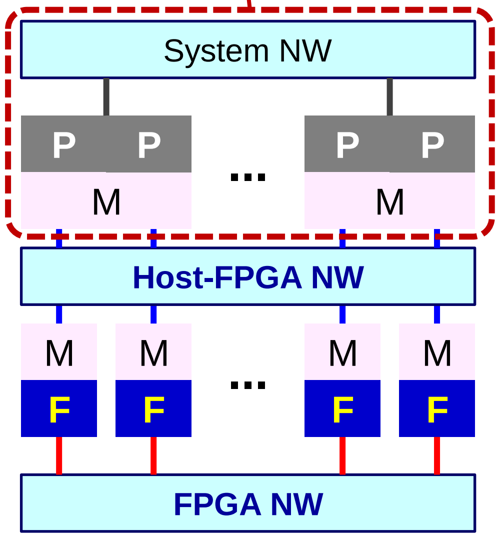
A. Cluster of CPUs with FPGAs (distributed or shared memory)



B. Cluster of CPUs and FPGAs



C. Clusters of CPUs with inter-connected FPGAs



D. Loosely-coupled CPU cluster and FPGA cluster

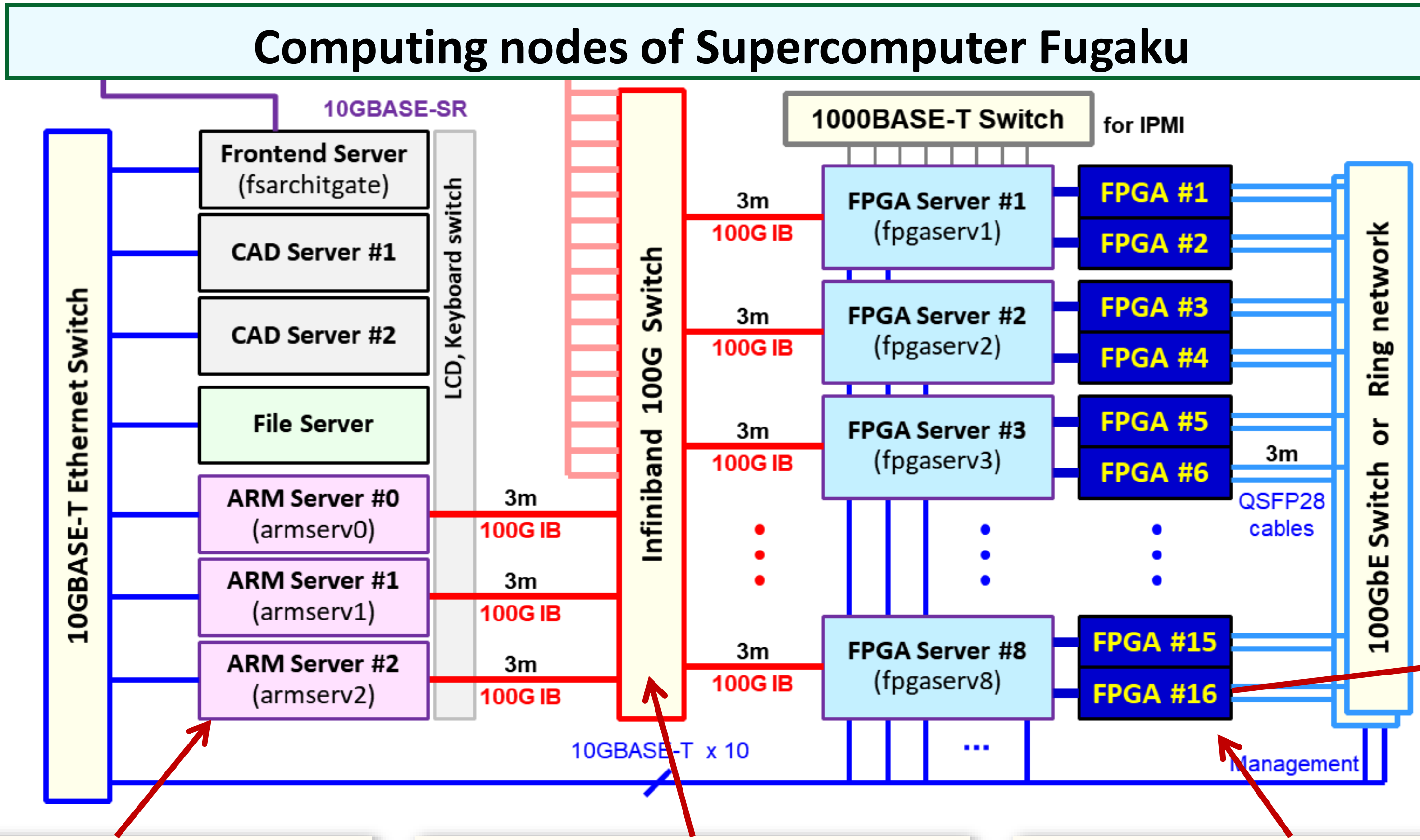
Our architecture

ESSPER

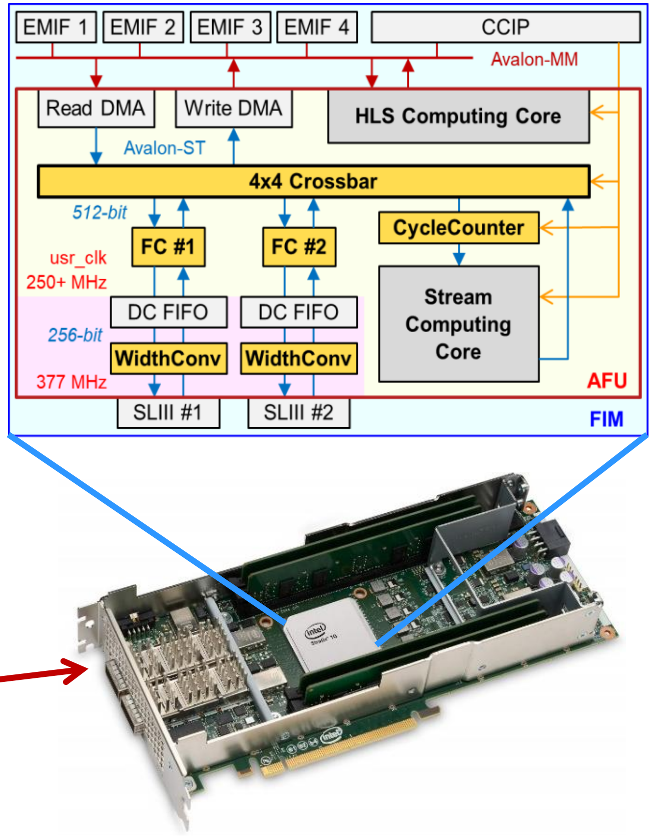
Elastic and Scalable System for High-Performance Re-configurable Computing

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)

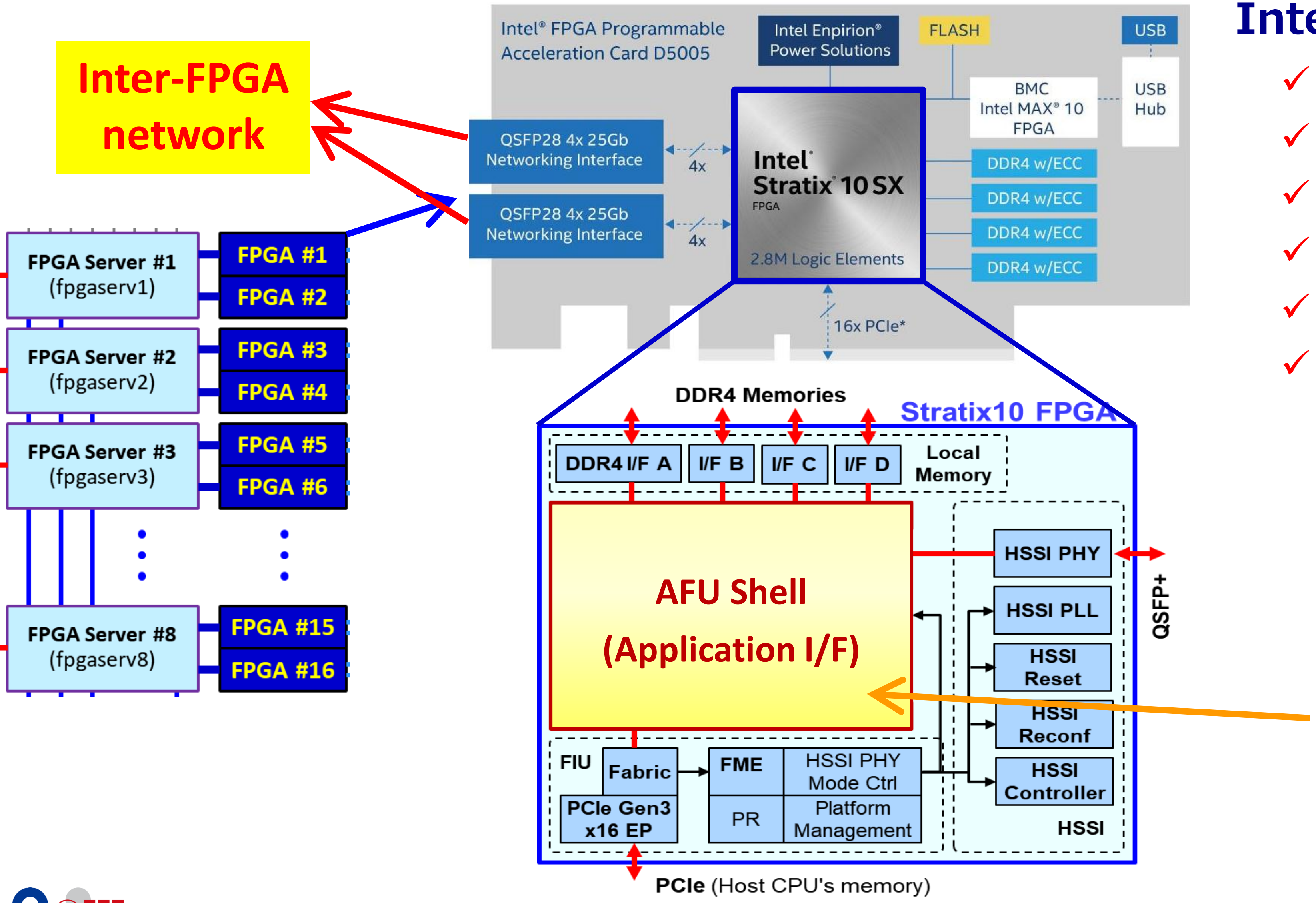
FPGA cluster

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

FPGA Shell (SoC)

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

Design of FPGA System-on-Chip



Intel FPGA PAC D5005

- ✓ *Intel Stratix 10 FPGA (14nm)*
- ✓ *2753K LEs, 229 Mb BRAMs*
- ✓ *5760 FP DSPs (7TF @ 600MHz)*
- ✓ 8GB DDR4 x 4ch
- ✓ PCIe Gen3 x16
- ✓ 2x QSFP28 (100Gb/s)

FIM (FPGA Interface Manager)

- ✓ Fixed region including I/F

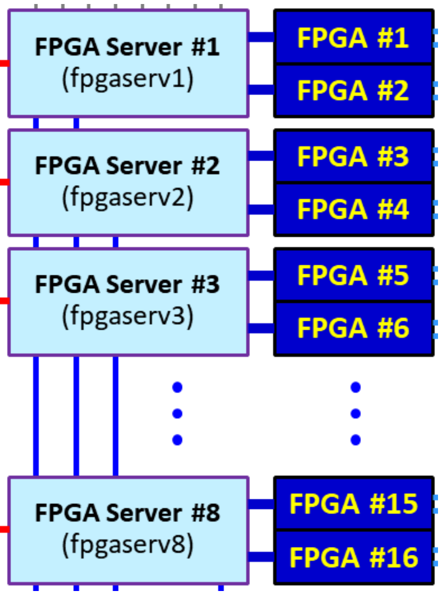
AFU (Acceleration Function Unit)

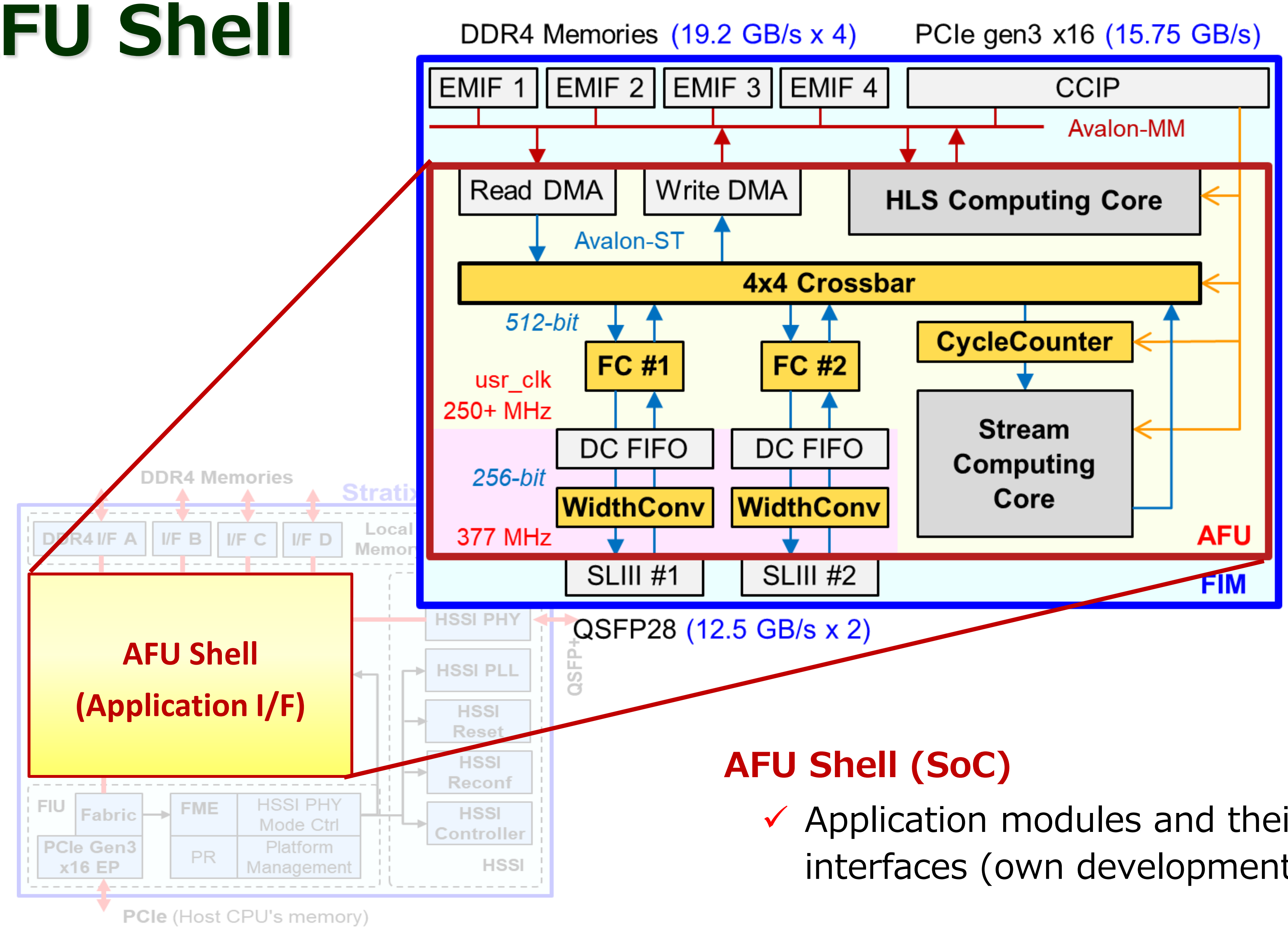
- ✓ Reconfigurable region

AFU Shell (SoC)

- ✓ Application modules and their interfaces (own development)

Design of AFU Shell



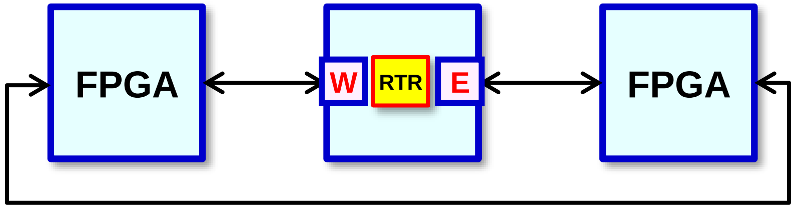
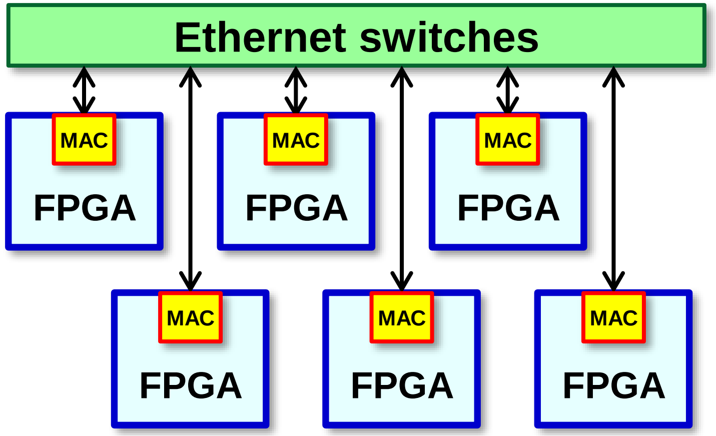


AFU Shell (SoC)

- ✓ Application modules and their interfaces (own development)

Inter-FPGA Network

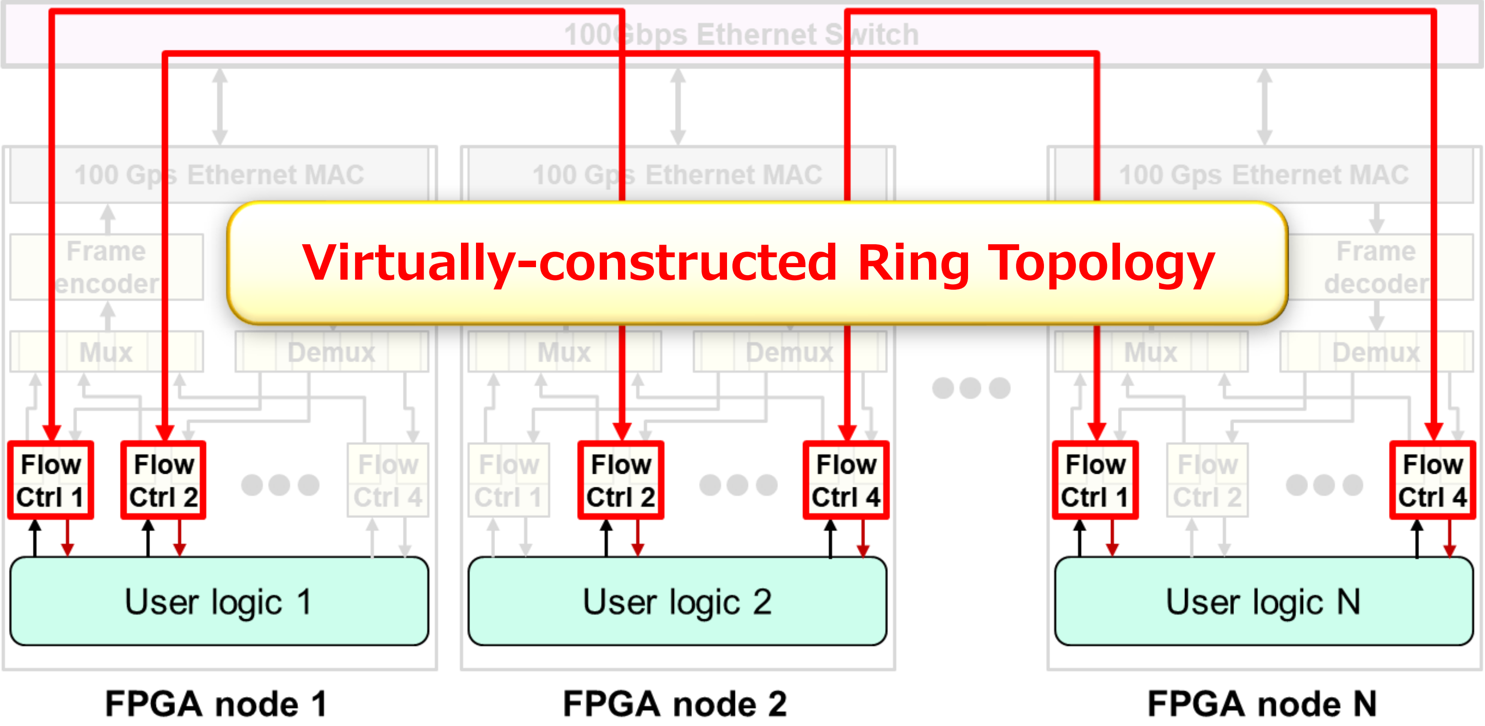
Two Types of Networks

	Direct network	Indirect network
		
Characteristics	**p2p-connection without switches,** typical: torus network	**connection with switches,** typical: Ethernet
Switching	**circuit** or packet (w/ on-chip router)	**packet**
Pros	**low latency,** **easy to use** with simple HW	**flexibility**, small diameter, easy adoption of **cutting-edge**
Cons	large diameter, **inflexibility** in resource allocation	higher latency due to packet processing, **complex and difficult** to use

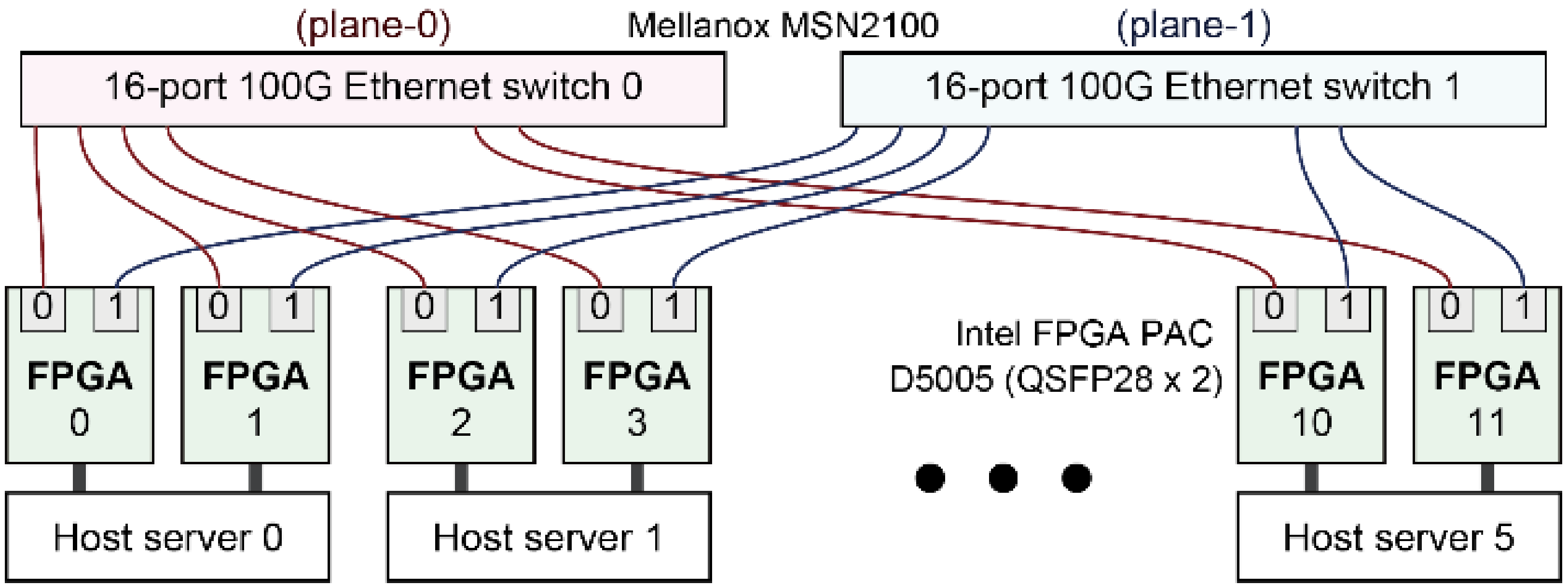
Proposal: Virtual Circuit Switching Network (VCSN)

Provide arbitrary topology with virtual links over Ethernet

✓ Easy to use by simply sending data through a virtual topology.
No complex control required for user logic.



VCSN Setup with 100Gbps Ethernet Switches



Intel PAC D5005 FPGA cluster with VCSN

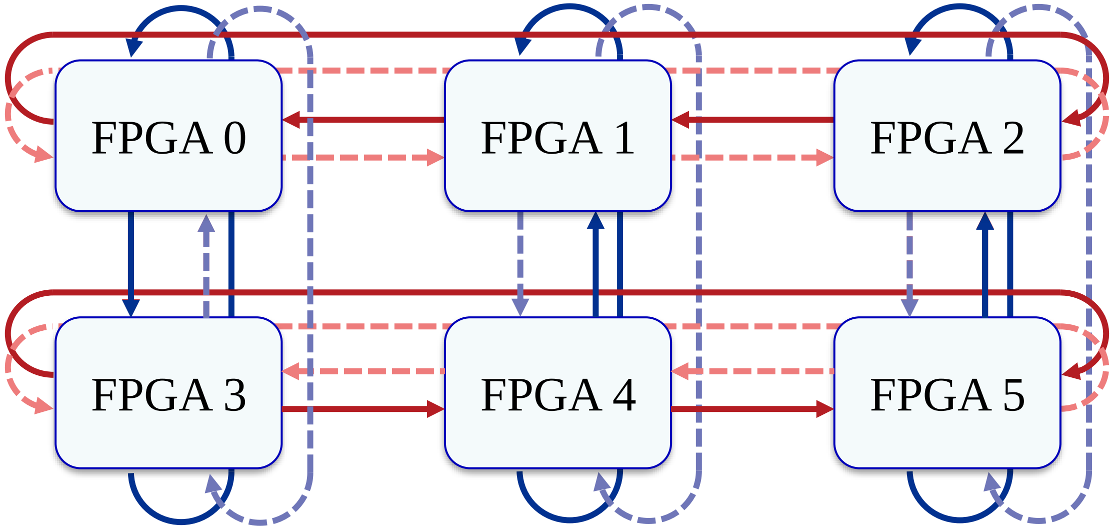
- FPGA's two ports connected to a different switch (Dual Plane)
- Two 16-port 100G Ethernet switches and optical cables

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.

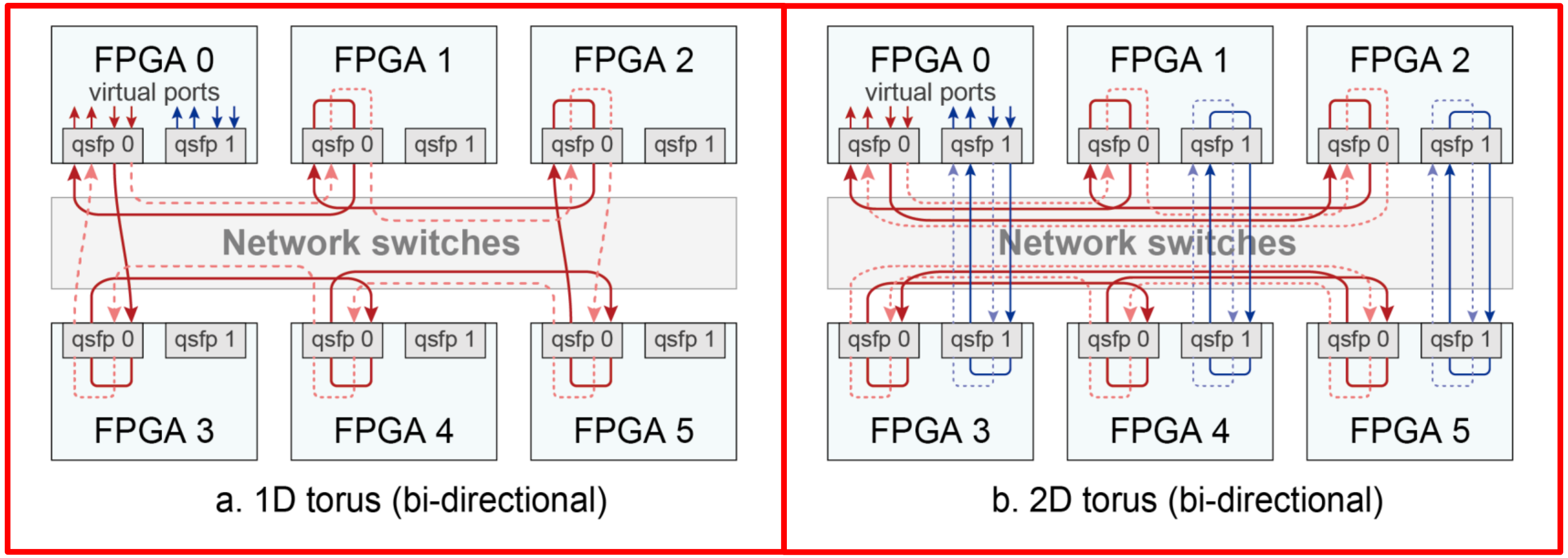
VCSN Configuration Examples

6-FPGA networks

- 1-D torus
- 2-D torus (2x3)



Virtual topology of bi-dir 2D Torus



a. 1D torus (bi-directional)

b. 2D torus (bi-directional)

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.

Throughput (point-to-point)

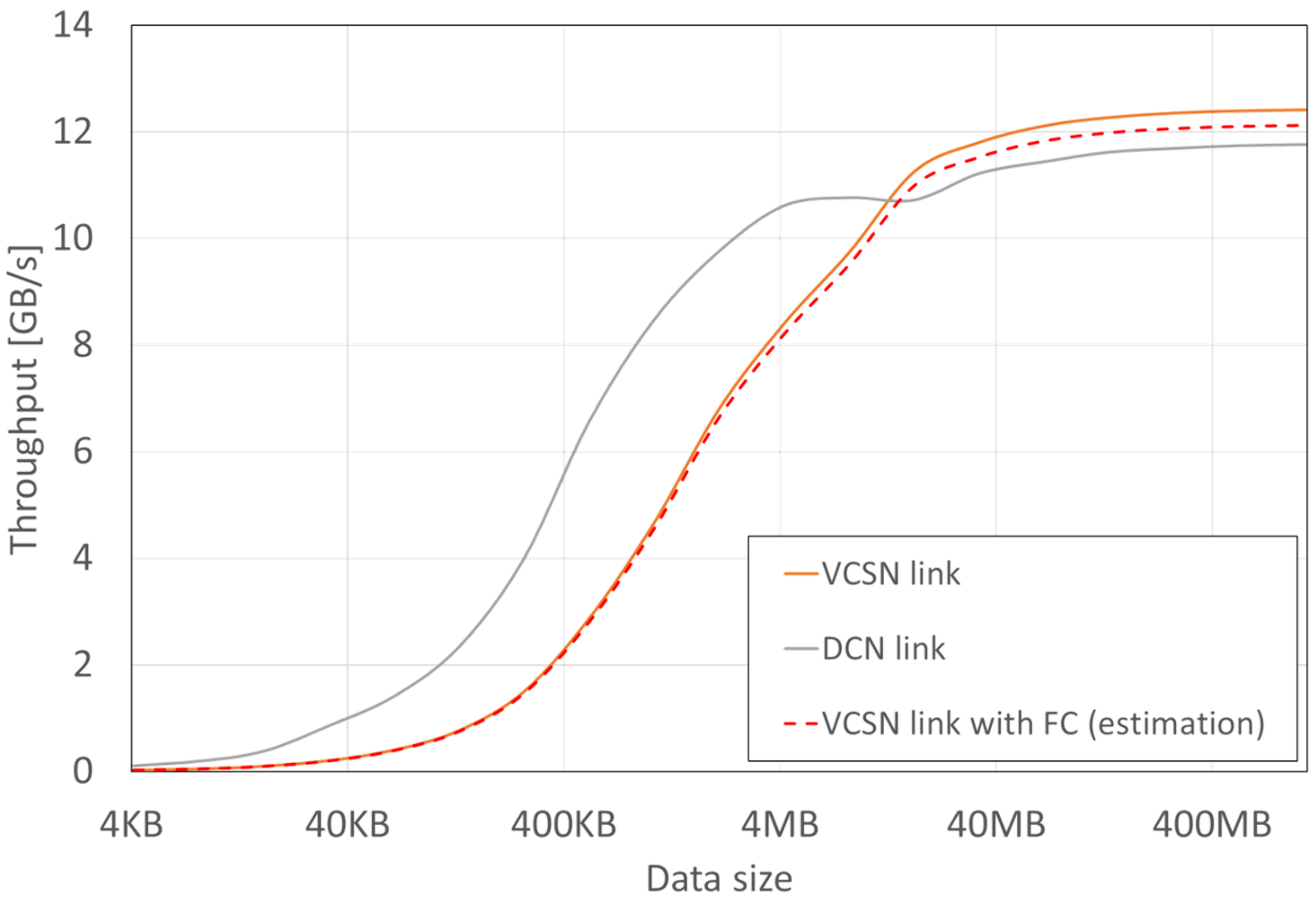
Throughput of VCSN rises slowly due to higher latency.

- ✓ P2P latency of VCSN 851 ns
- ✓ P2P latency of DCN 490 ns

VCSN has higher Max throughput.

- ✓ 100Gbps = 12.5 GB/s
- ✓ Jumbo frame of Ethernet is more efficient : 96% of the peak

Latency-tolerant stream computing can work well.



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.

ESSPER

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

System Software

System Stack of ESSPER

High-level application

- Including both SW & HW

Low-level application

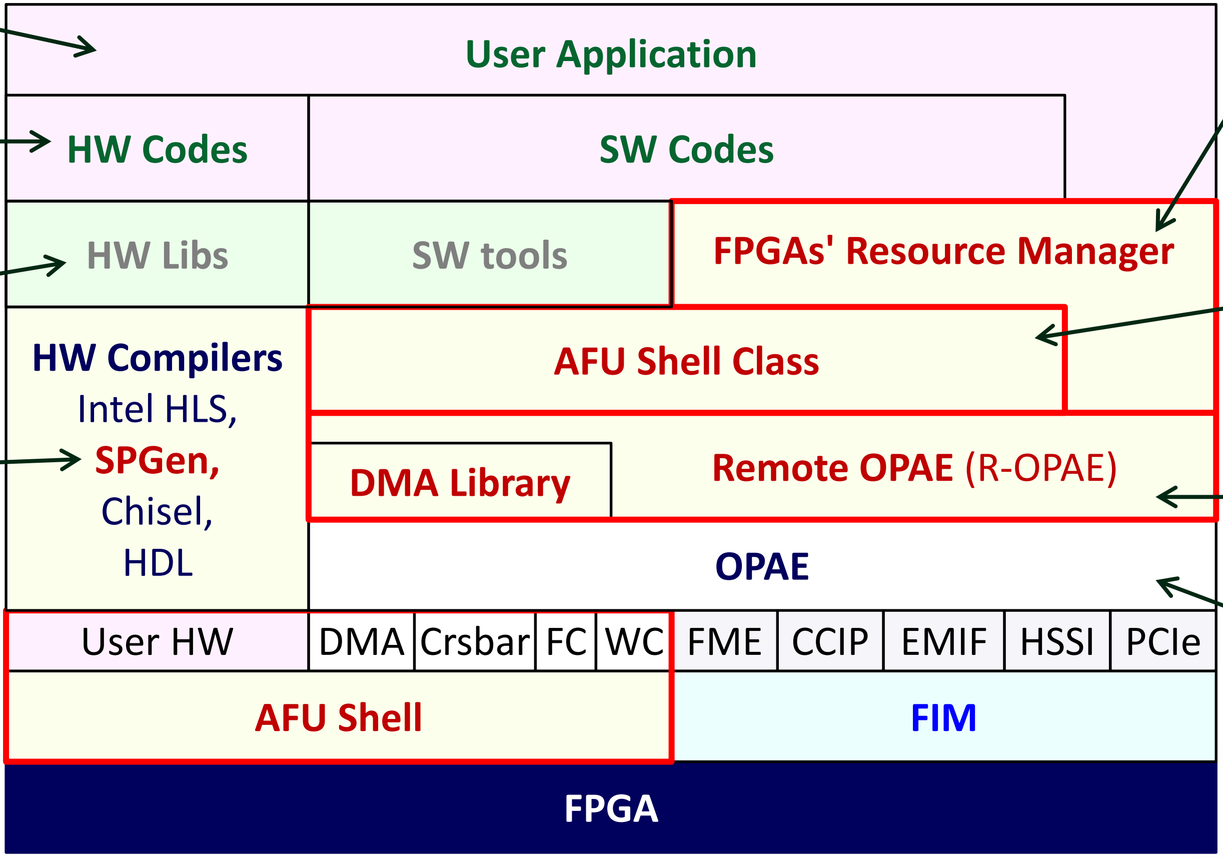
- Separately-written HW & SW

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

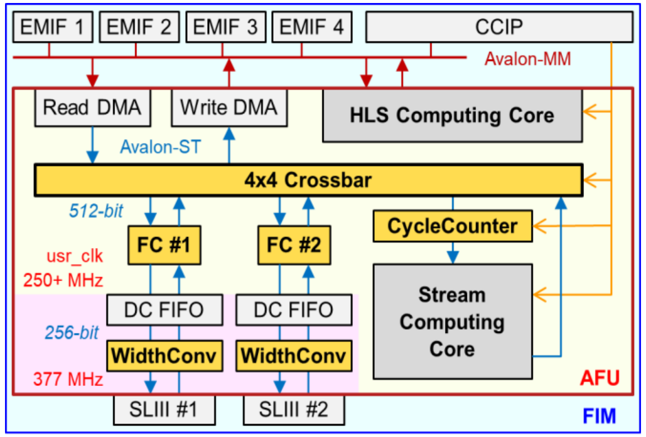
- Object of AFU shell
- Abstraction of HW

Remote OPAE

- Software bridge using Infiniband Verbs

OPAE

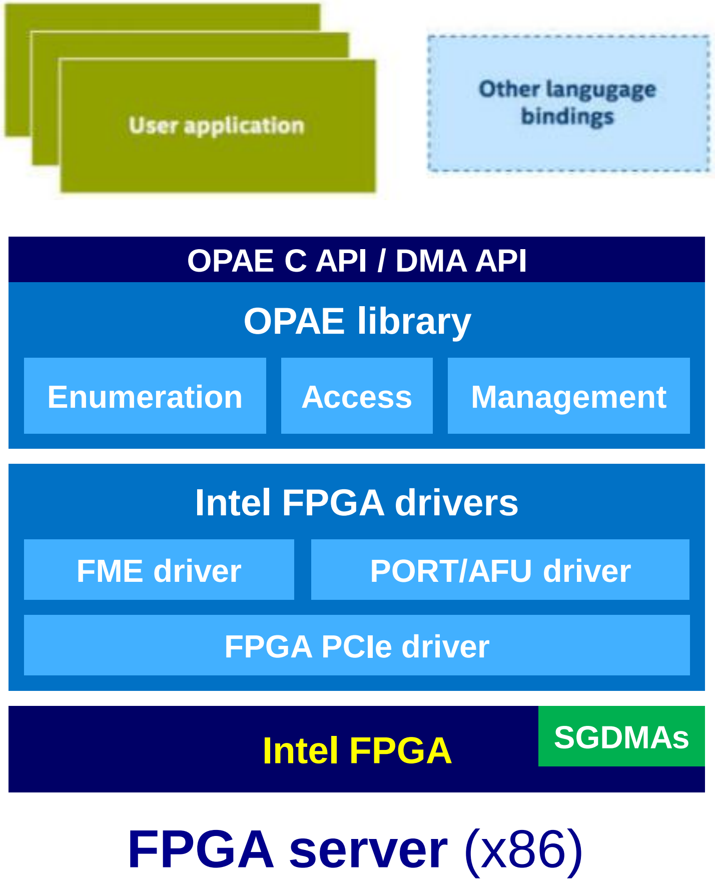
- Low-level driver



Remote-OPAE (for remote FPGA Access)

Software bridge for FPGAs over Infiniband

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

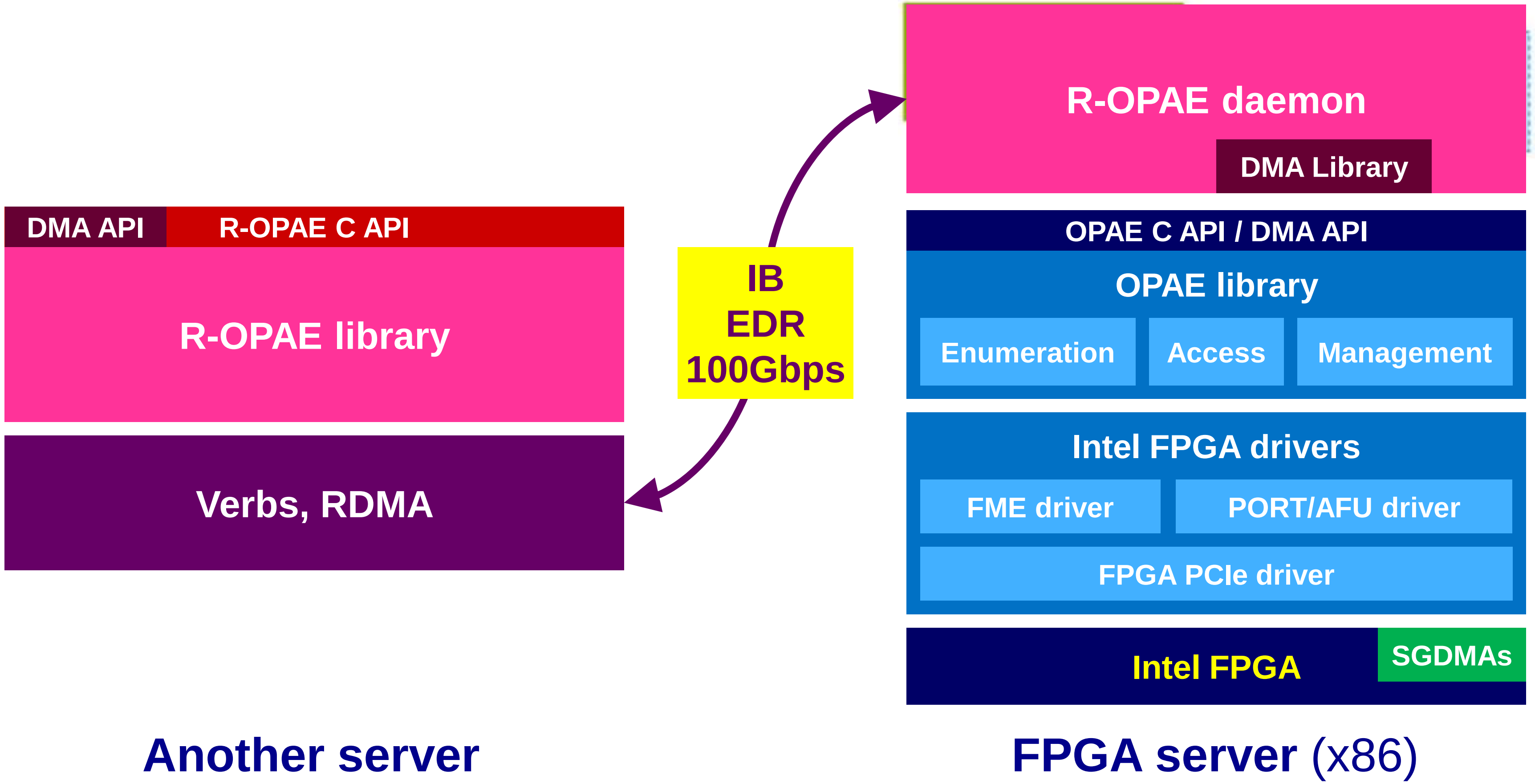


FPGA server (x86)

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

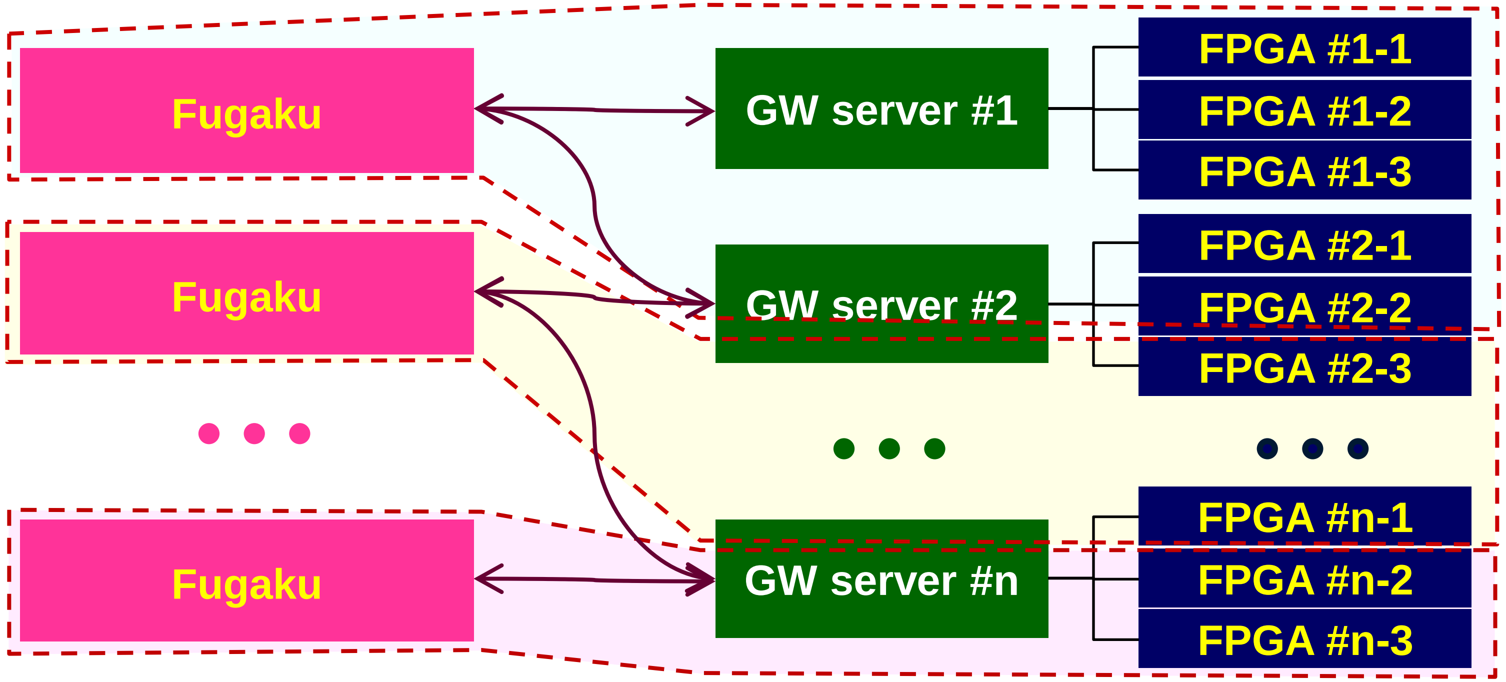
Transparent access to remote FPGAs

Flexible utilization:

- ✓ Can use any available FPGA resources

Inter-operability and extensibility:

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



Other servers **Gateway servers and FPGAs**

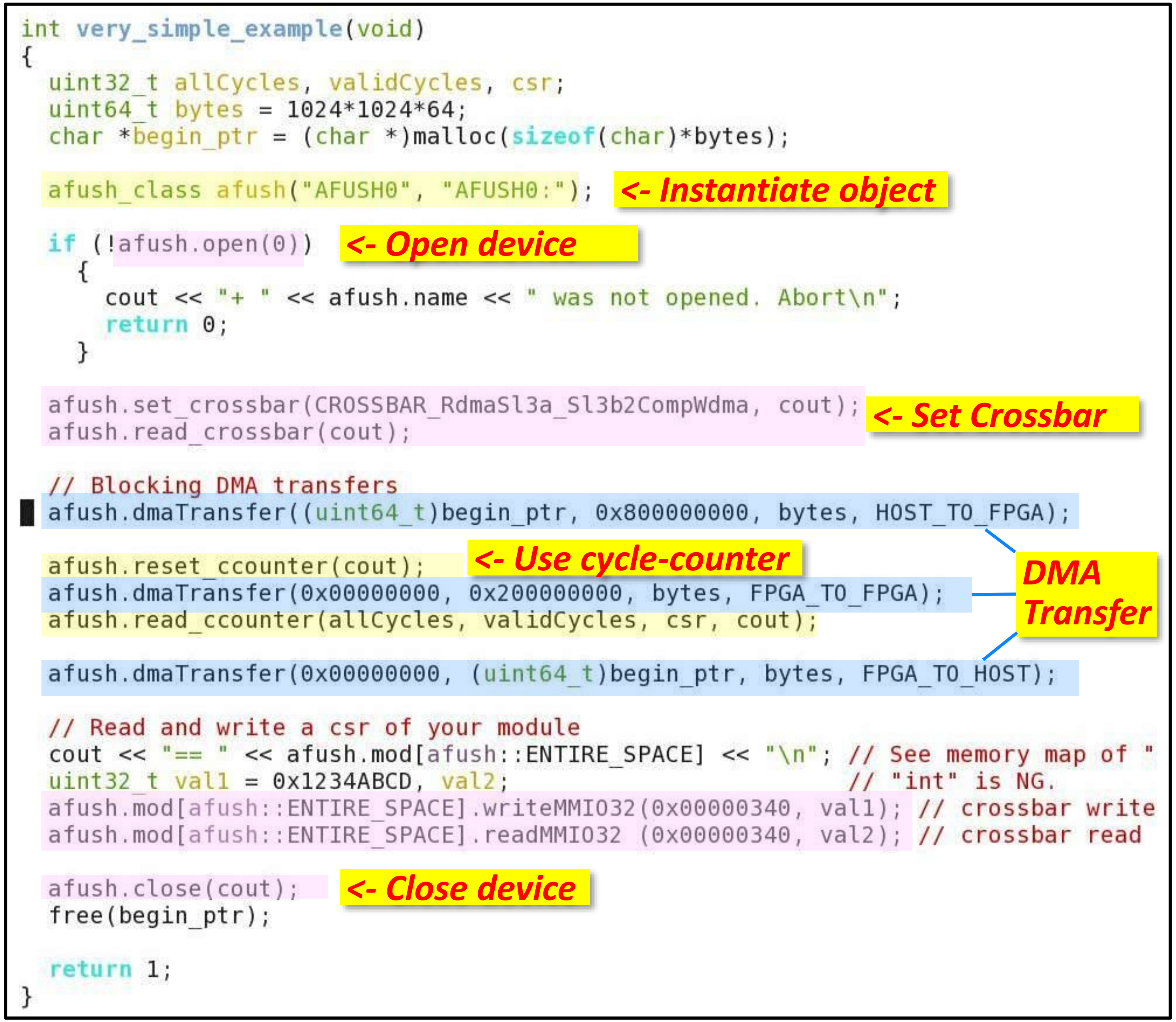
Host Code Programming

Use AFU Shell Class

- ✓ Abstraction of HW (address, etc.)
- ✓ APIs of service functions
- ✓ **Low-level control** still possible
  - ➢ writeMMIO32(), readMMIO32()

App code can be written simply.

- ✓ Instantiate **AFUShell object**
- ✓ Open object
- ✓ Use modules / services
  - ➢ *Crossbar*
  - ➢ *Hardware cycle-counter*
  - ➢ *DMA (Host-FPGA, FPGA-FPGA)*
  - ➢ *Computing core*
- ✓ Close object



```
int very_simple_example(void)
{
  uint32_t allCycles, validCycles, csr;
  uint64_t bytes = 1024*1024*64;
  char *begin_ptr = (char *)malloc(sizeof(char)*bytes);

  afush_class afush("AFUSH0", "AFUSH0:");

  if (!afush.open(0))
    {
      cout << "+ " << afush.name << " was not opened. Abort\n";
      return 0;
    }

  afush.set_crossbar(CROSSBAR_RdmaSl3a_Sl3b2CompWdma, cout);
  afush.read_crossbar(cout);

  // Blocking DMA transfers
█ afush.dmaTransfer((uint64_t)begin_ptr, 0x800000000, bytes, HOST_TO_FPGA);

  afush.reset_ccounter(cout);
  afush.dmaTransfer(0x00000000, 0x200000000, bytes, FPGA_TO_FPGA);
  afush.read_ccounter(allCycles, validCycles, csr, cout);

  afush.dmaTransfer(0x00000000, (uint64_t)begin_ptr, bytes, FPGA_TO_HOST);

  // Read and write a csr of your module
  cout << "== " << afush.mod[afush::ENTIRE_SPACE] << "\n"; // See memory map of "
  uint32_t val1 = 0x1234ABCD, val2;                         // "int" is NG.
  afush.mod[afush::ENTIRE_SPACE].writeMMIO32(0x00000340, val1); // crossbar write
  afush.mod[afush::ENTIRE_SPACE].readMMIO32 (0x00000340, val2); // crossbar read

  afush.close(cout);
  free(begin_ptr);

  return 1;
}
```

<- Instantiate object

<- Open device

<- Set Crossbar

<- Use cycle-counter

DMA Transfer

<- Close device

Open-Access paper

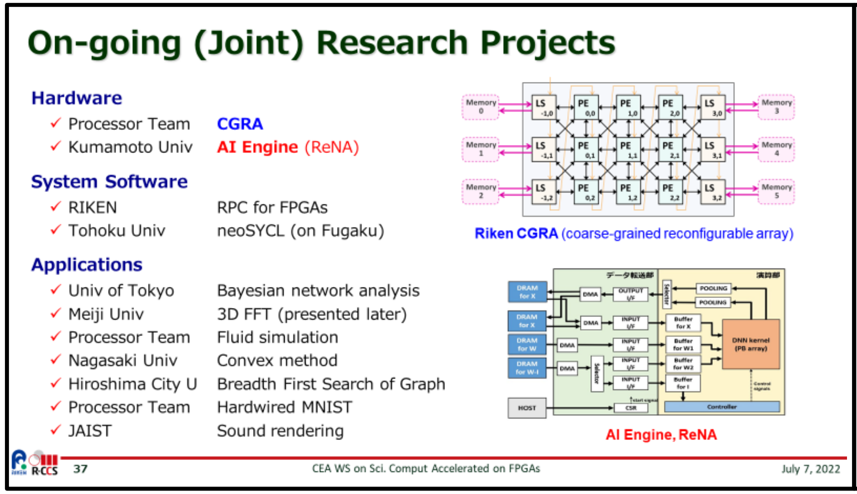
ESSPER

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

Applications, Joint Research Projects

Projects with ESSPER (Selected)

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



On-going (Joint) Research Projects

Hardware
- ✓ Processor Team — CGRA
- ✓ Kumamoto Univ — AI Engine (ReNA)

System Software
- ✓ RIKEN — RPC for FPGAs
- ✓ Tohoku Univ — neoSYCL (on Fugaku)

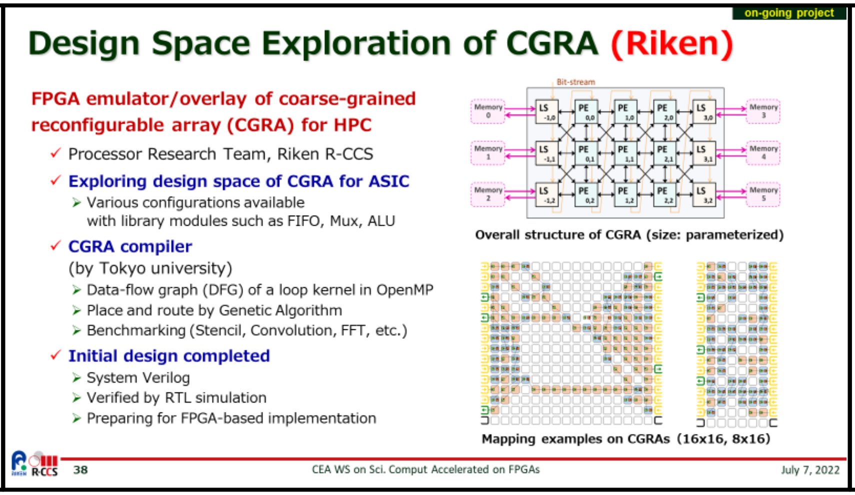
Applications

✓ Univ of Tokyo	Bayesian network analysis
✓ Meiji Univ	3D FFT (presented later)
✓ Processor Team	Fluid simulation
✓ Nagasaki Univ	Convex method
✓ Hiroshima City U	Breadth First Search of Graph
✓ Processor Team	Hardwired MNIST
✓ JAIST	Sound rendering

Riken CGRA (coarse-grained reconfigurable array)

AI Engine, ReNA

37 CEA WS on Sci. Comput Accelerated on FPGAs July 7, 2022



on-going project

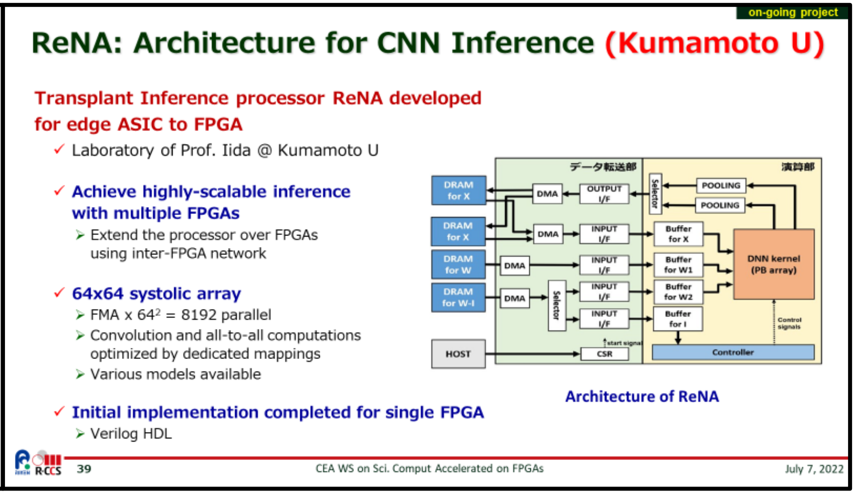
Design Space Exploration of CGRA (Riken)

FPGA emulator/overlay of coarse-grained reconfigurable array (CGRA) for HPC
- ✓ Processor Research Team, Riken R-CCS
- ✓ Exploring design space of CGRA for ASIC
  - Various configurations available with library modules such as FIFO, Mux, ALU
- ✓ CGRA compiler (by Tokyo university)
  - Data-flow graph (DFG) of a loop kernel in OpenMP
  - Place and route by Genetic Algorithm
  - Benchmarking (Stencil, Convolution, FFT, etc.)
- ✓ Initial design completed
  - System Verilog
  - Verified by RTL simulation
  - Preparing for FPGA-based implementation

Overall structure of CGRA (size: parameterized)

Mapping examples on CGRAs (16x16, 8x16)

38 CEA WS on Sci. Comput Accelerated on FPGAs July 7, 2022



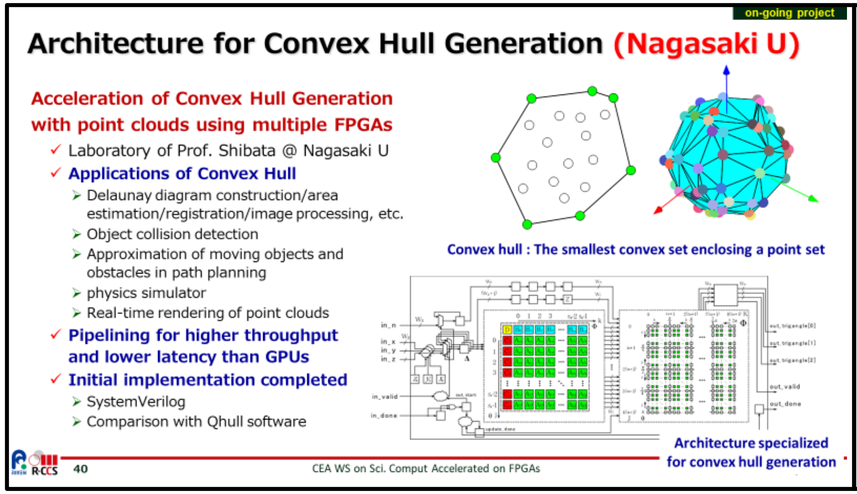
on-going project

ReNA: Architecture for CNN Inference (Kumamoto U)

Transplant Inference processor ReNA developed for edge ASIC to FPGA
- ✓ Laboratory of Prof. Iida @ Kumamoto U
- ✓ Achieve highly-scalable inference with multiple FPGAs
  - Extend the processor over FPGAs using inter-FPGA network
- ✓ 64x64 systolic array
  - FMA x 64^2 = 8192 parallel
  - Convolution and all-to-all computations optimized by dedicated mappings
  - Various models available
- ✓ Initial implementation completed for single FPGA
  - Verilog HDL

Architecture of ReNA

39 CEA WS on Sci. Comput Accelerated on FPGAs July 7, 2022



on-going project

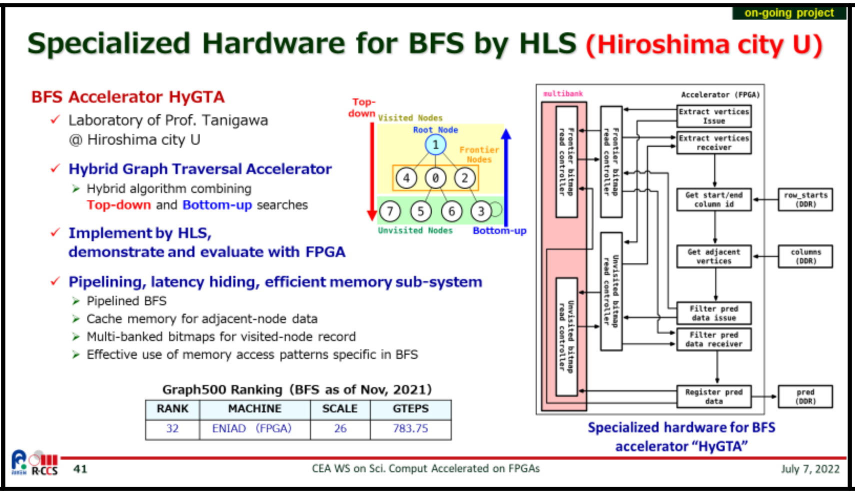
Architecture for Convex Hull Generation (Nagasaki U)

Acceleration of Convex Hull Generation with point clouds using multiple FPGAs
- ✓ Laboratory of Prof. Shibata @ Nagasaki U
- ✓ Applications of Convex Hull
  - Delaunay diagram construction/area estimation/registration/image processing, etc.
  - Object collision detection
  - Approximation of moving objects and obstacles in path planning
  - physics simulator
  - Real-time rendering of point clouds
- ✓ Pipelining for higher throughput and lower latency than GPUs
- ✓ Initial implementation completed
  - SystemVerilog
  - Comparison with Qhull software

Convex hull : The smallest convex set enclosing a point set

Architecture specialized for convex hull generation

40 CEA WS on Sci. Comput Accelerated on FPGAs July 7, 2022



on-going project

Specialized Hardware for BFS by HLS (Hiroshima city U)

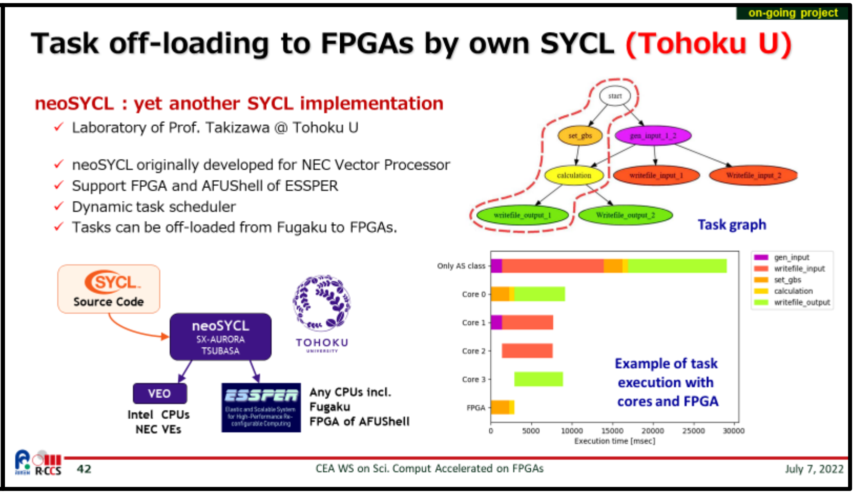
BFS Accelerator HyGTA
- ✓ Laboratory of Prof. Tanigawa @ Hiroshima city U
- ✓ Hybrid Graph Traversal Accelerator
  - Hybrid algorithm combining Top-down and Bottom-up searches
- ✓ Implement by HLS, demonstrate and evaluate with FPGA
- ✓ Pipelining, latency hiding, efficient memory sub-system
  - Pipelined BFS
  - Cache memory for adjacent-node data
  - Multi-banked bitmaps for visited-node record
  - Effective use of memory access patterns specific in BFS

Graph500 Ranking (BFS as of Nov, 2021)

RANK	MACHINE	SCALE	GTEPS
32	ENIAD (FPGA)	26	783.75

Specialized hardware for BFS accelerator "HyGTA"

41 CEA WS on Sci. Comput Accelerated on FPGAs July 7, 2022



on-going project

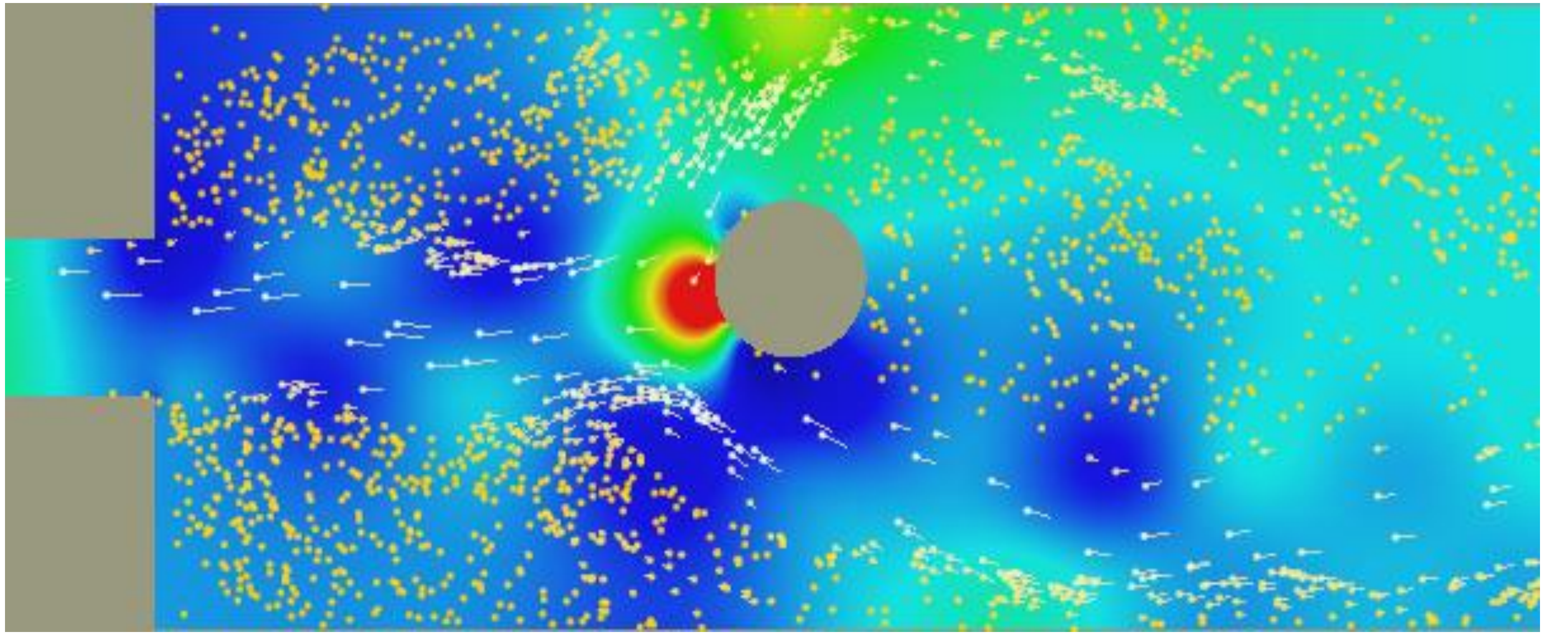
Task off-loading to FPGAs by own SYCL (Tohoku U)

neoSYCL : yet another SYCL implementation
- ✓ Laboratory of Prof. Takizawa @ Tohoku U
- ✓ neoSYCL originally developed for NEC Vector Processor
- ✓ Support FPGA and AFUShell of ESSPER
- ✓ Dynamic task scheduler
- ✓ Tasks can be off-loaded from Fugaku to FPGAs.

Task graph

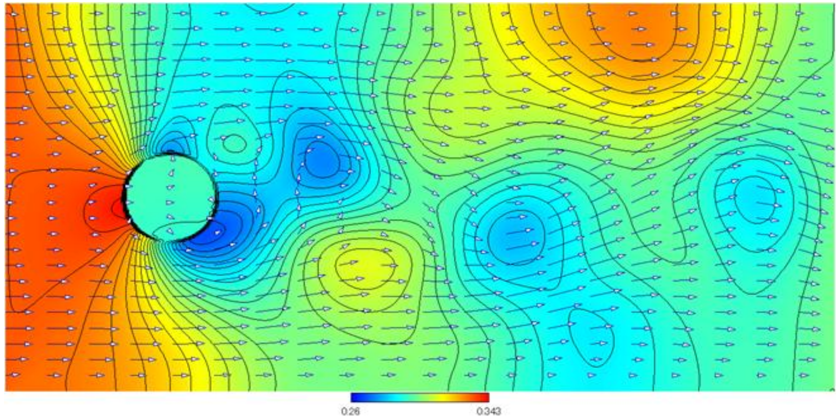
Example of task execution with cores and FPGA

42 CEA WS on Sci. Comput Accelerated on FPGAs July 7, 2022



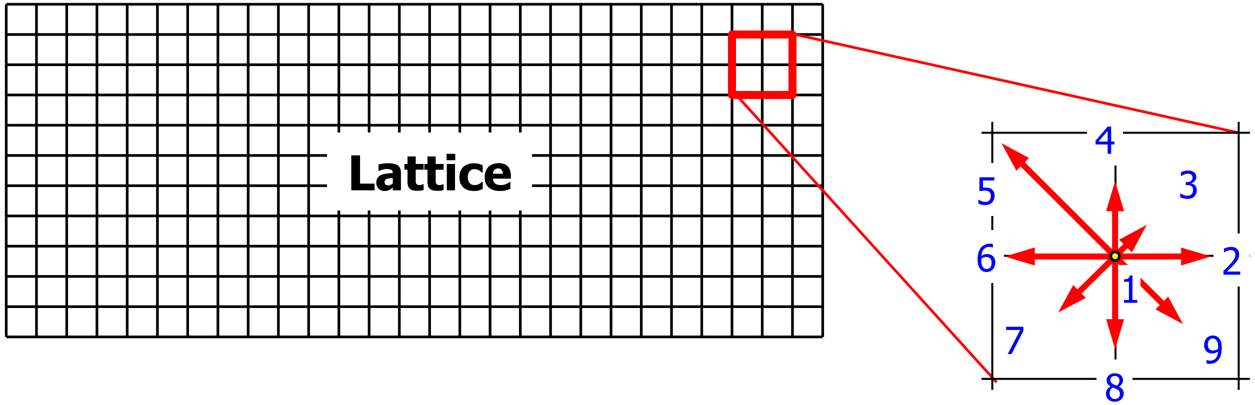
Fluid dynamics simulator based on stream computing with FPGAs

Lattice Boltzmann Method (LBM)



Compute fluids with particles moving and colliding over a lattice mesh

Particles are expressed with densities for nine discrete speeds

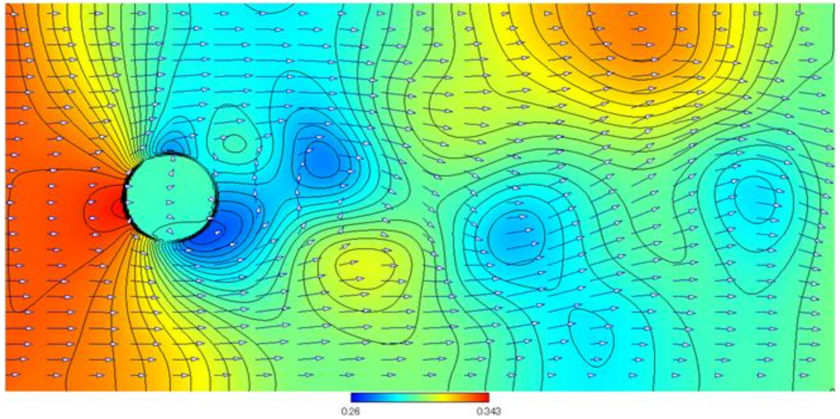


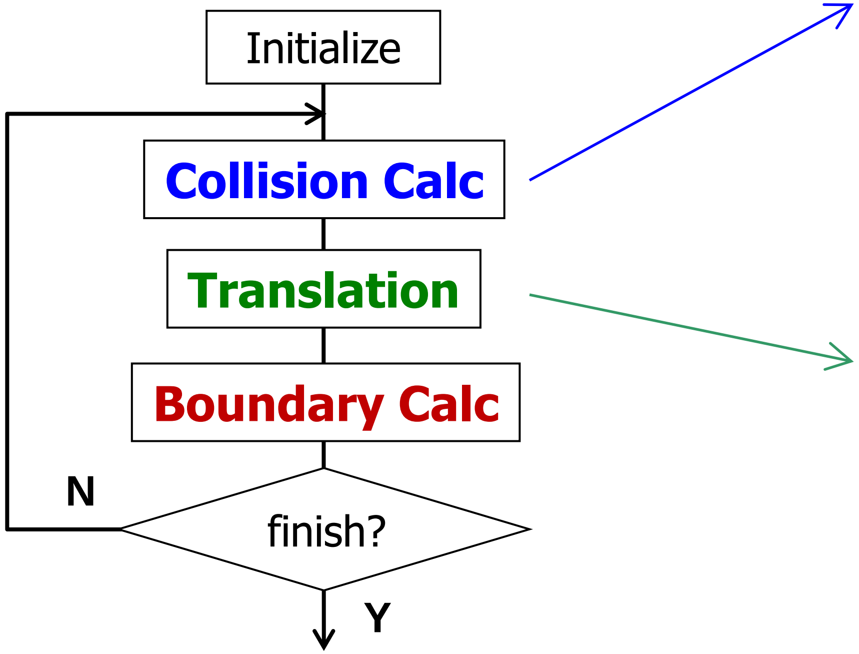
Lattice cell

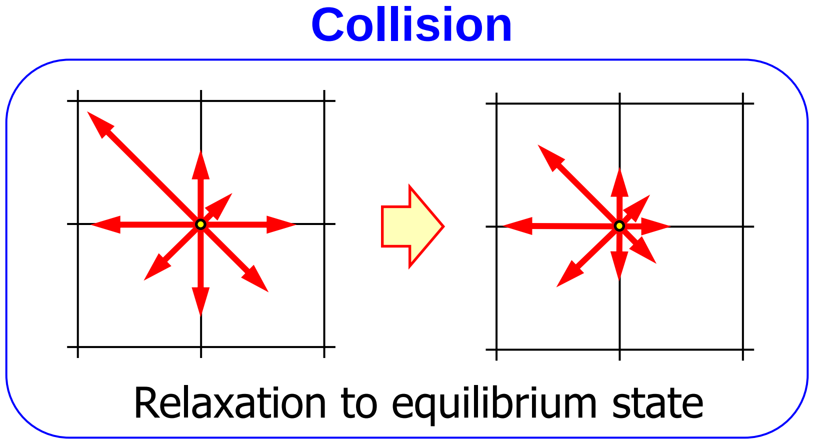
Particle densities for 9 directions

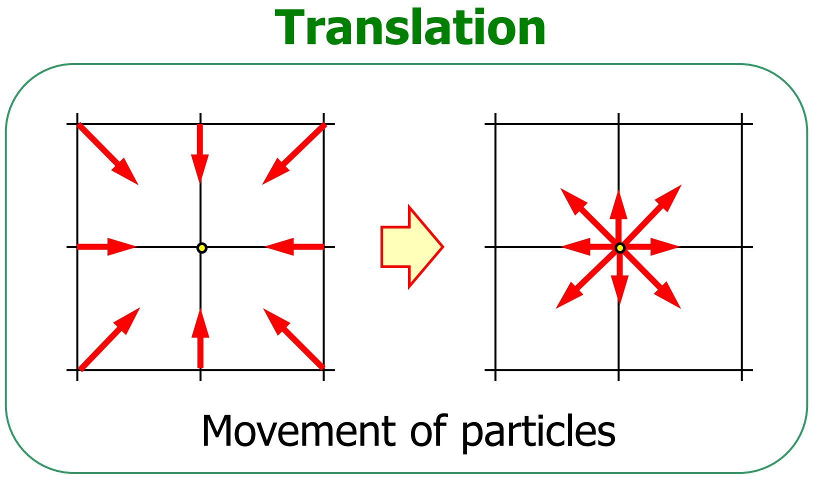
(0.0 to 1.0 in each dir)

Algorithm of LBM









How to Stream it? – Case for Stencil Computation

(Computing dependency of LBM is similar to it.)

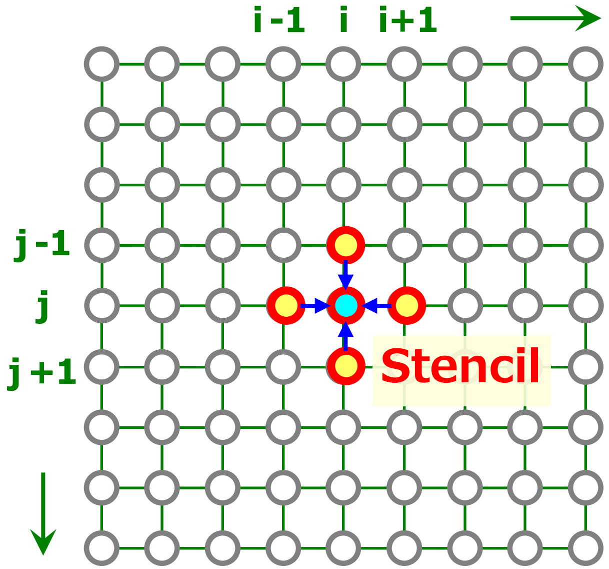
```
for (n=0; n< $N_{max}$ ; n++)
  for (j=0; j< $J_{max}$ ; j++)
    for (i=0; i< $I_{max}$ ; i++)
      $v'_{i,j} = f\ (v_{i,j}, v_{i+1,j}, v_{i-1,j}, v_{i,j+1}, v_{i,j-1})$
```

← Iteration (time-integral)

Sweep of grid

Stencil reference & computation

Pseudo code of 2D stencil computation

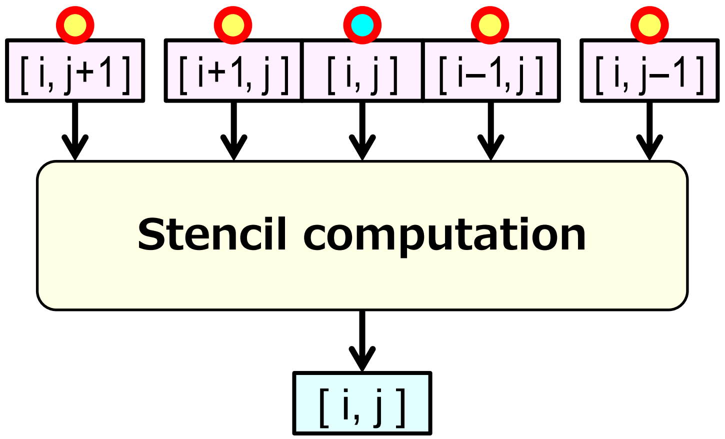


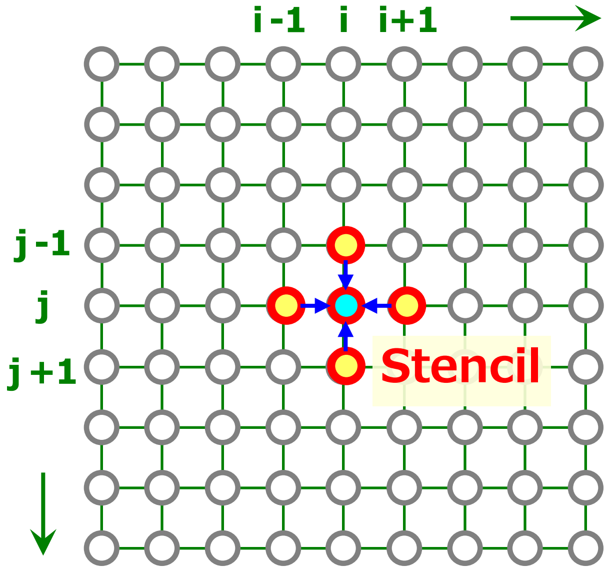
Computational grid

- ✓ Read local grid points (stencil), and compute
- ✓ Sweep and update the entire grid
- ✓ Iterate the grid update for time-integral

How to Stream it? – Case for Stencil Computation

(Computing dependency of LBM is similar to it.)

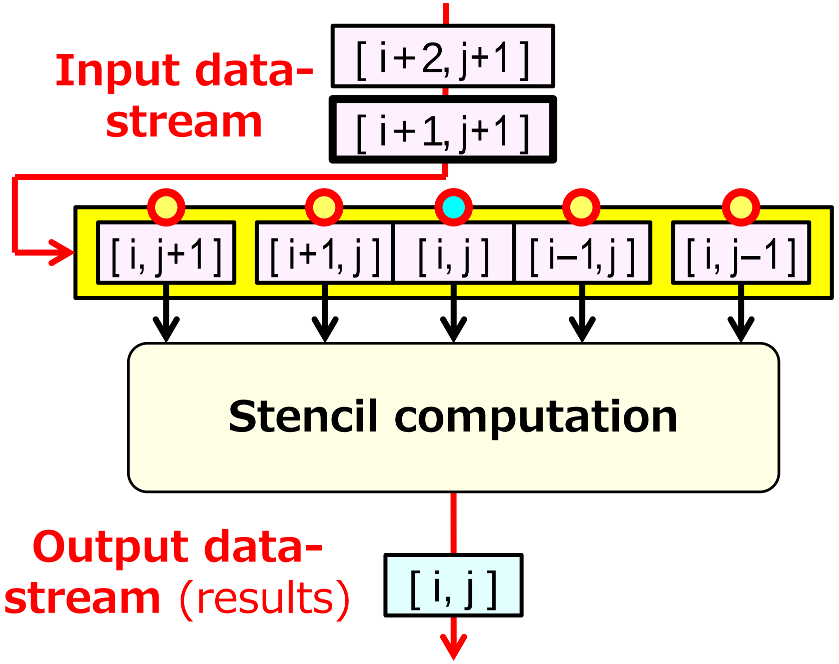


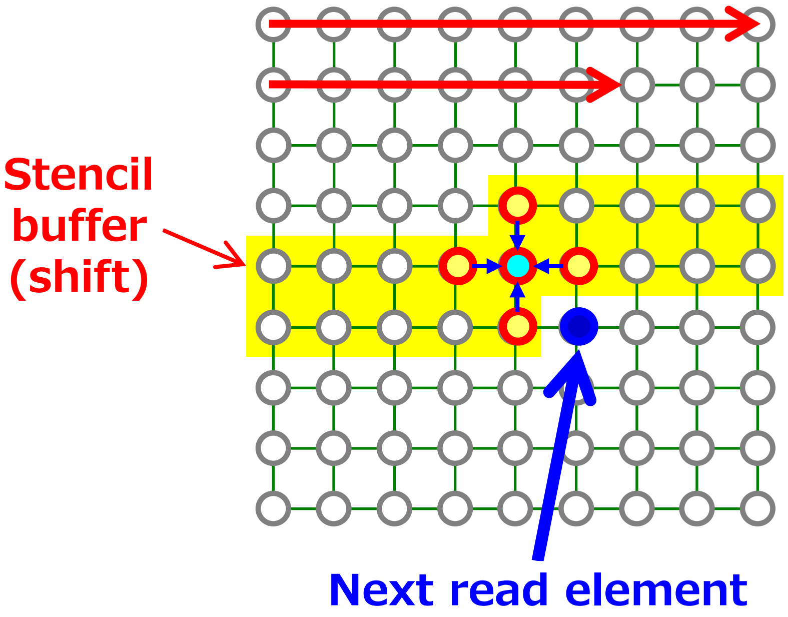


Computational grid

- ✓ Read local grid points (stencil), and compute
- ✓ Sweep and update the entire grid
- ✓ Iterate the grid update for time-integral

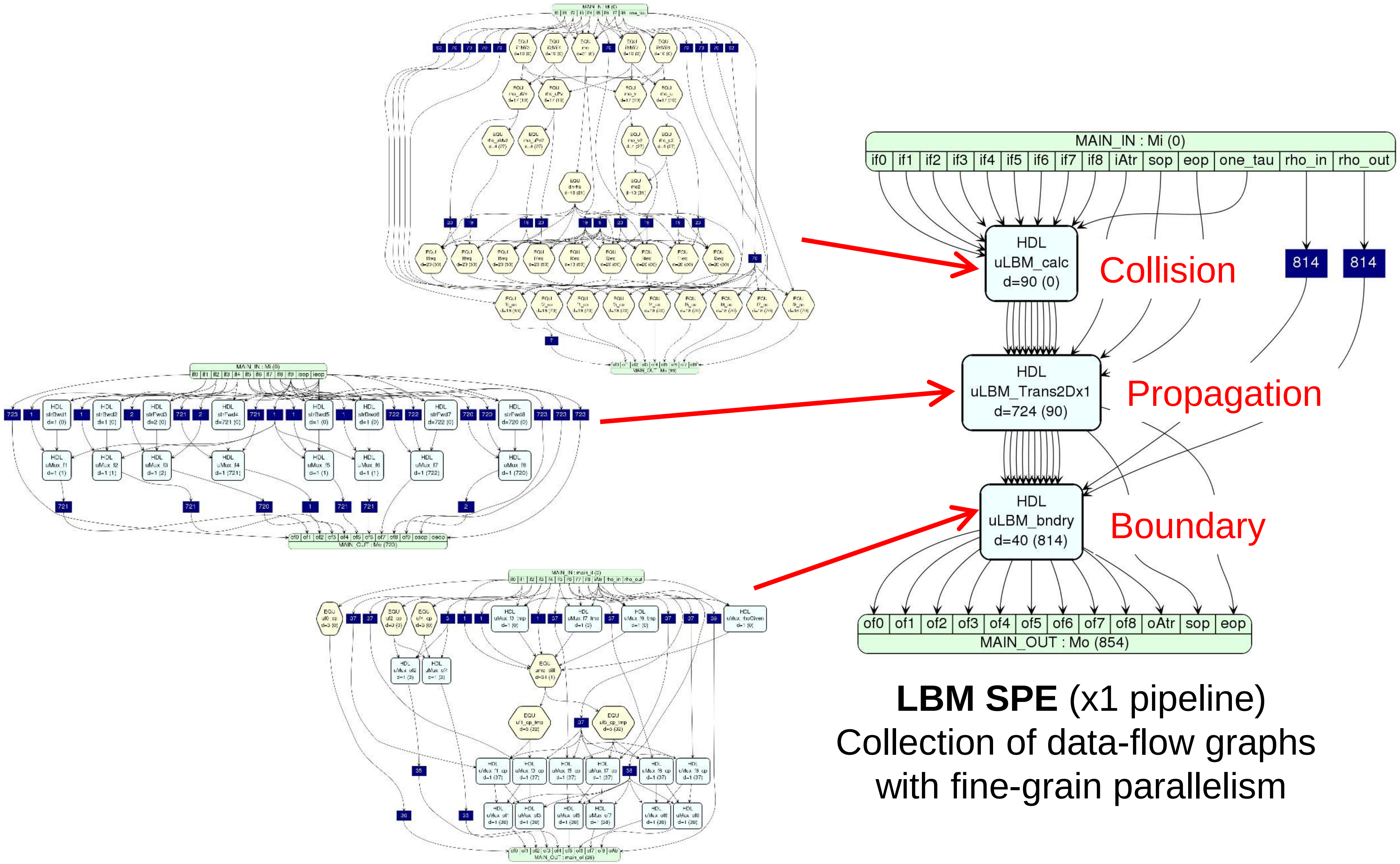
Hardware Algorithm for Streaming (=Pipelining)





- ✓ Stencil buffer with the size of two rows
- ✓ Push read elements into stencil buffer, and shift it
- ✓ Reference stored elements in parallel, and compute
- ✓ Pipelined read and computation (= high throughput)

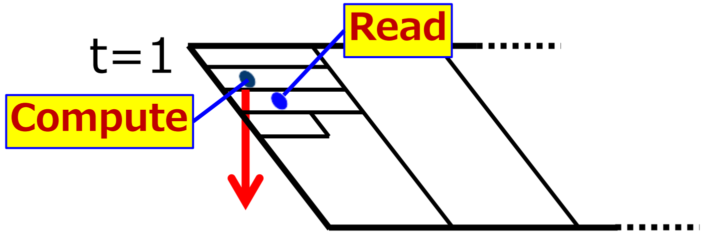
Stream Processing Element (SPE)


MAIN_IN : Mi (0)
if0 | if1 | if2 | if3 | if4 | if5 | if6 | if7 | if8 | iAtr | sop | eop | one_tau | rho_in | rho_out
HDL
uLBM_calc
d=90 (0)
Collision
814
814
HDL
uLBM_Trans2Dx1
d=724 (90)
Propagation
HDL
uLBM_bndry
d=40 (814)
Boundary
of0 | of1 | of2 | of3 | of4 | of5 | of6 | of7 | of8 | oAtr | sop | eop
MAIN_OUT : Mo (854)

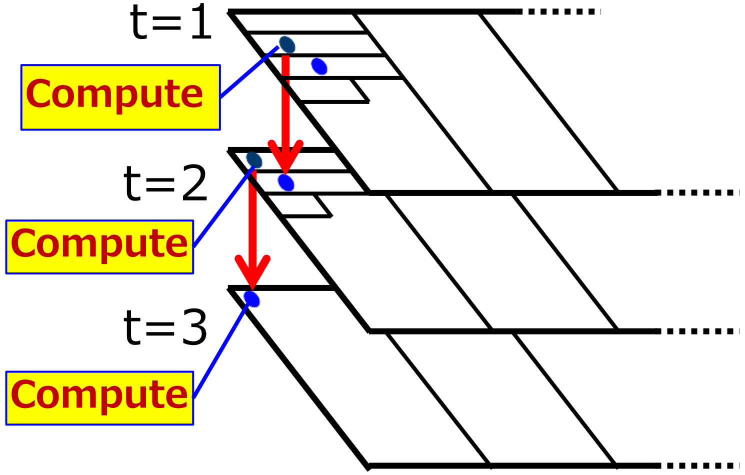
LBM SPE (x1 pipeline)
Collection of data-flow graphs
with fine-grain parallelism

More Pipelining by Loop Unrolling (cascading SPEs)

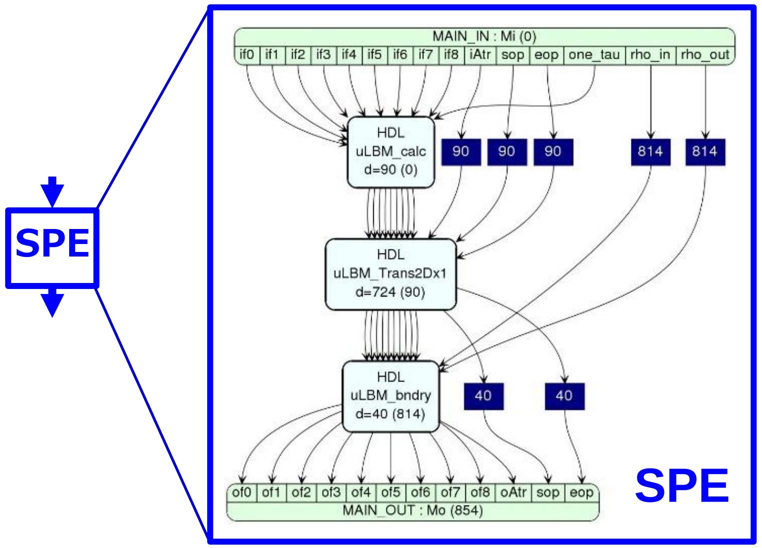
No temporal parallelism

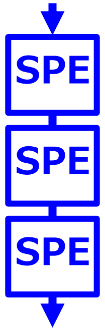


With temporal parallelism



Unroll iteration loop to compute multiple step-steps at a time



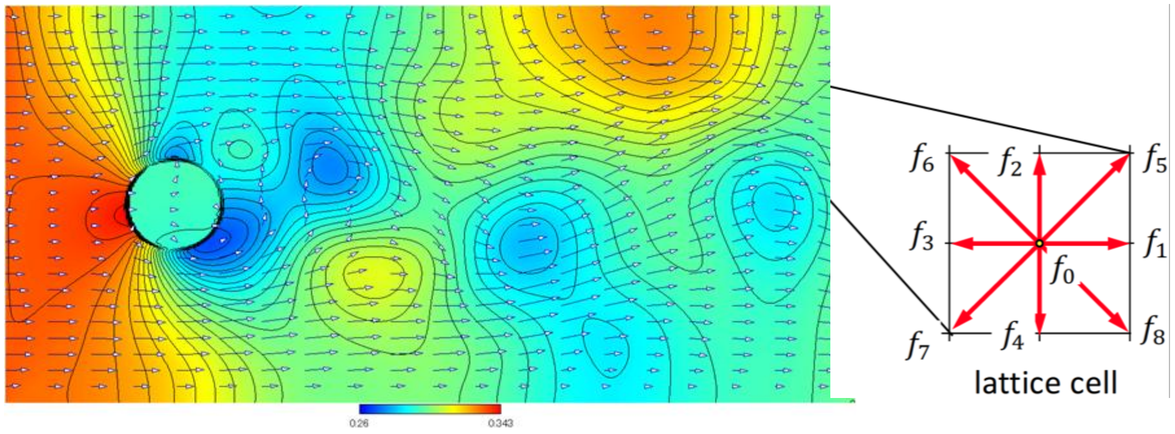


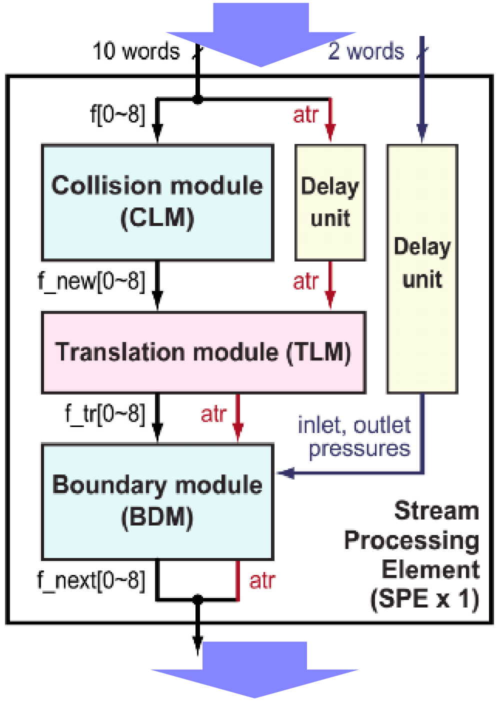
Scaling Performance by increasing pipeline-stages at a constant bandwidth

~ 18 SPEs

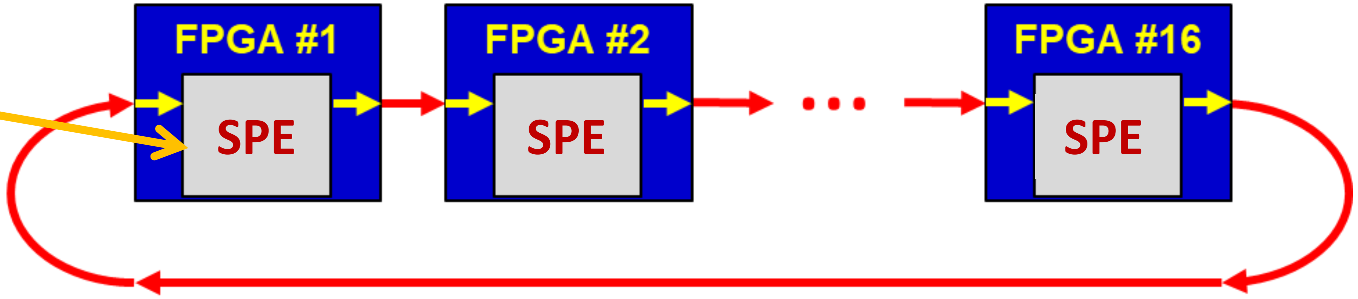
Further More Pipelining with Multiple FPGAs in a Ring

- **Steam computing of Fluid simulation with multiple FPGAs**
  - ✓ Lattice Boltzmann method (**LBM**)
  - ✓ Extended pipeline with ringed FPGAs





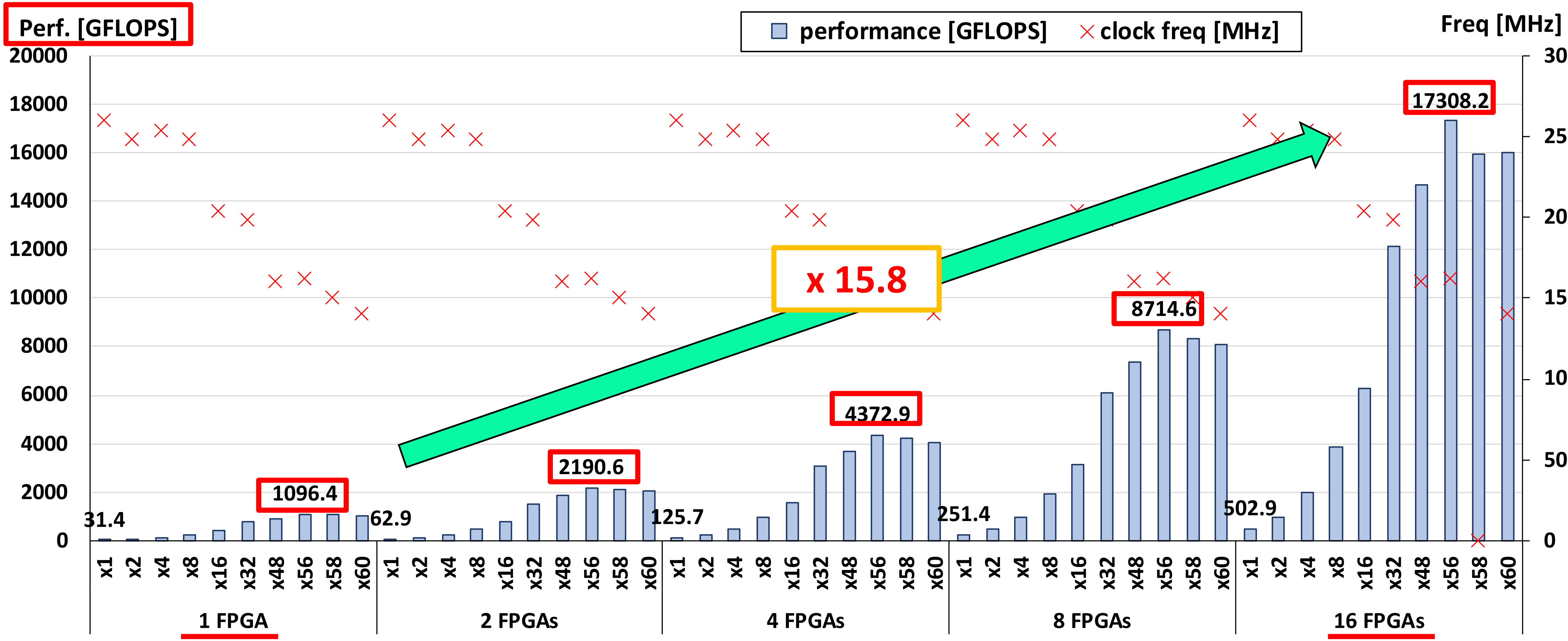
Stream Processing element (SPE)



FPGAs in a ring

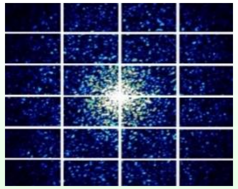
Performance of 2D LBM with 100Gbps Ring NW

Computational performance (FLOPS) when processing about 2GB data



3. Near-sensor processing / Scientific edge-computing

- ✓ FPGA-based processing for **X-ray imaging detector** (RIKEN Spring-8)
- ✓ Data-compression hardware for edge-computing (ANL)



4. Backend of Fault-Tolerant Quantum Computers

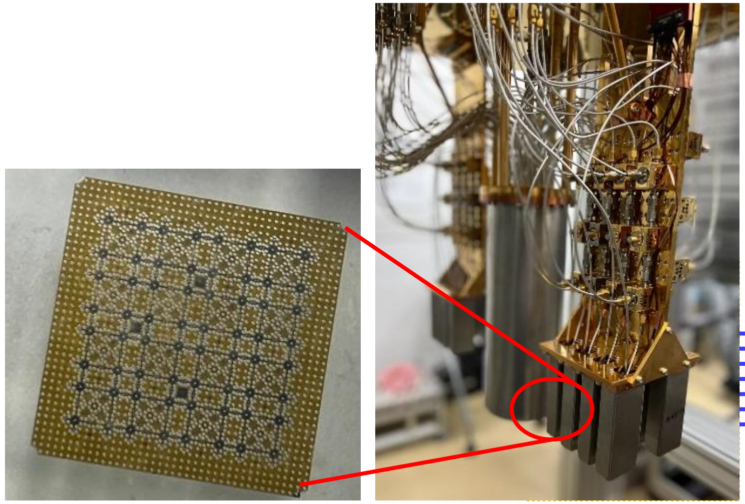
- ✓ Specialized hardware for **quantum error correction**
  (Hardware algorithms, FPGA demo targeting RIKEN quantum device)

Domain-specific computing for quantum error correction in FTQC

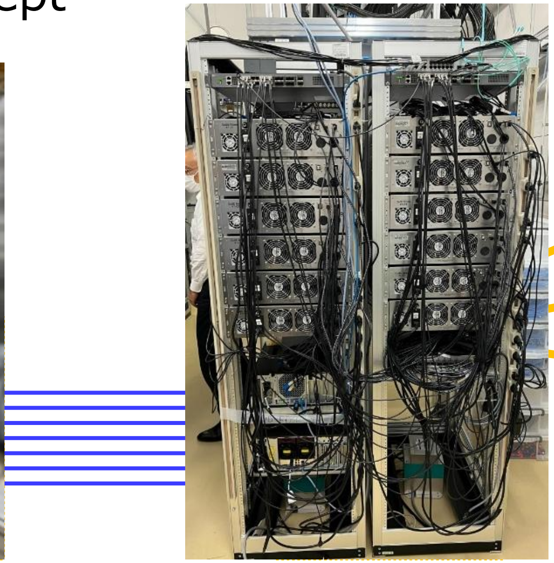
Quantum Error Correction with FPGAs

MOONSHOT RESEARCH & DEVELOPMENT PROGRAM

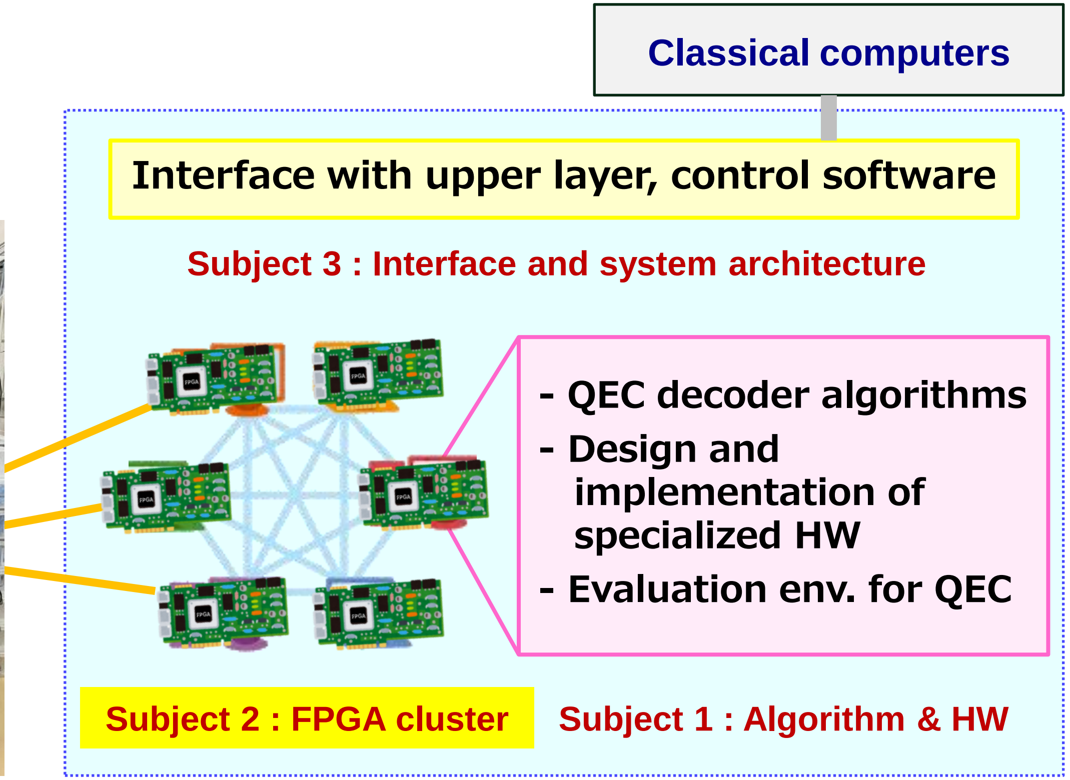
- **Fault-tolerant quantum computers (FTQC) using quantum error correction (QEC)**
  - ✓ Need to solve minimum-weight perfect matching (MWPM) problem
  - ✓ Need to encode 1000 logical qubits using 1M physical qubits finally
  - ✓ Scalability and low-latency (< 10us) are required.
- **Goal**
  - ✓ Explore scalable QEC hardware algorithm and system
  - ✓ Demonstrate for proof-of-concept



RIKEN's superconducting qubits

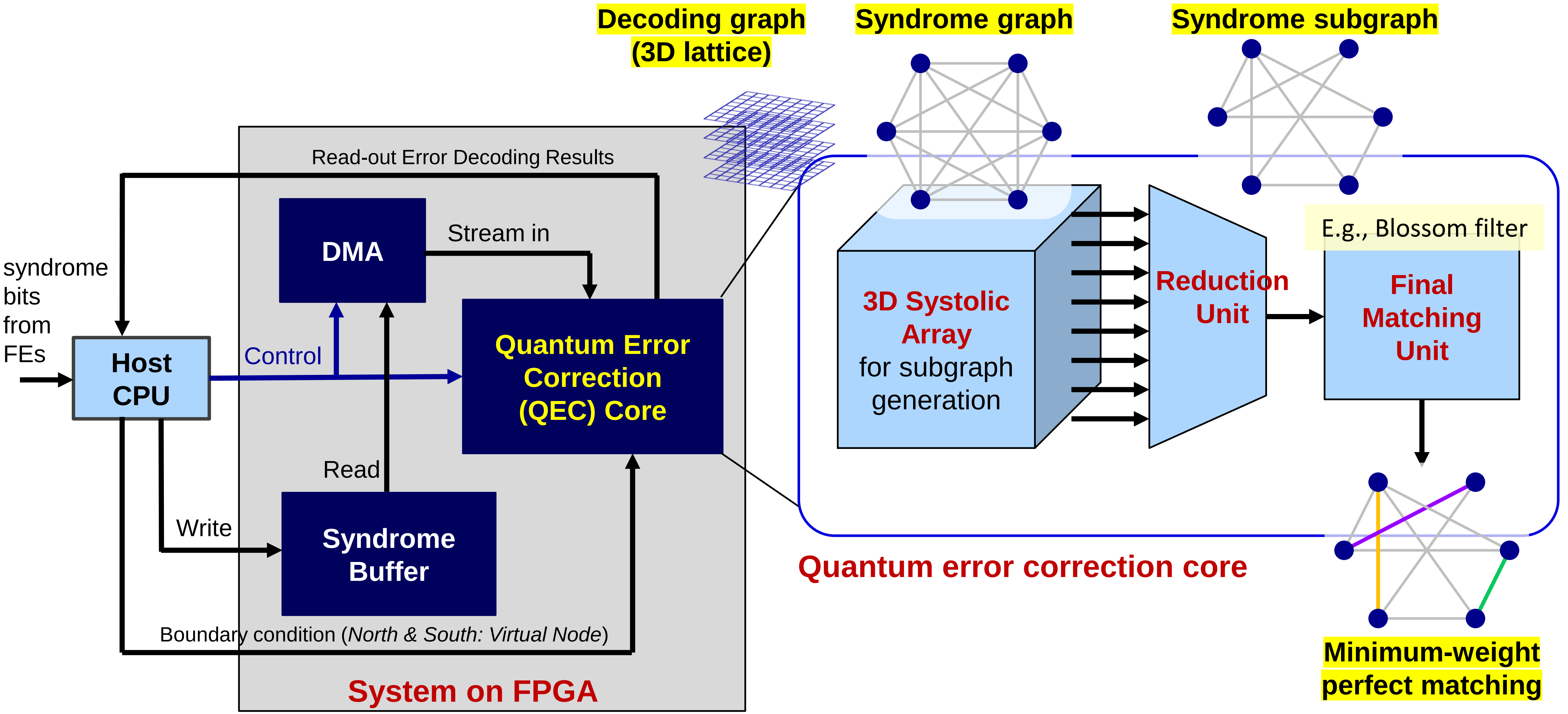


Quantum-Classical Frontend


Classical computers
Interface with upper layer, control software
Subject 3 : Interface and system architecture
FPGA
- QEC decoder algorithms
- Design and implementation of specialized HW
- Evaluation env. for QEC
Subject 2 : FPGA cluster
Subject 1 : Algorithm & HW

Backend system for QEC

Hardware Design for Syndrome Subgraph Decoder


Decoding graph (3D lattice)
Syndrome graph
Syndrome subgraph
Read-out Error Decoding Results
DMA
Stream in
syndrome bits from FEs
Host CPU
Control
Quantum Error Correction (QEC) Core
Read
Write
Syndrome Buffer
Boundary condition (North & South: Virtual Node)
System on FPGA
3D Systolic Array for subgraph generation
Reduction Unit
E.g., Blossom filter
Final Matching Unit
Quantum error correction core
Minimum-weight perfect matching

ESSPER2

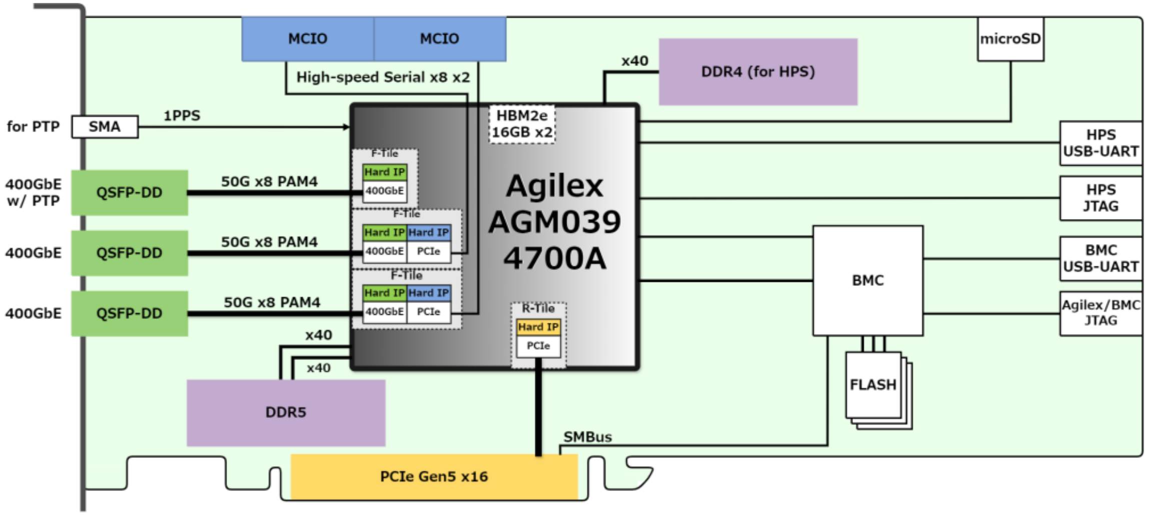
Next-Generation FPGA Cluster

Concept

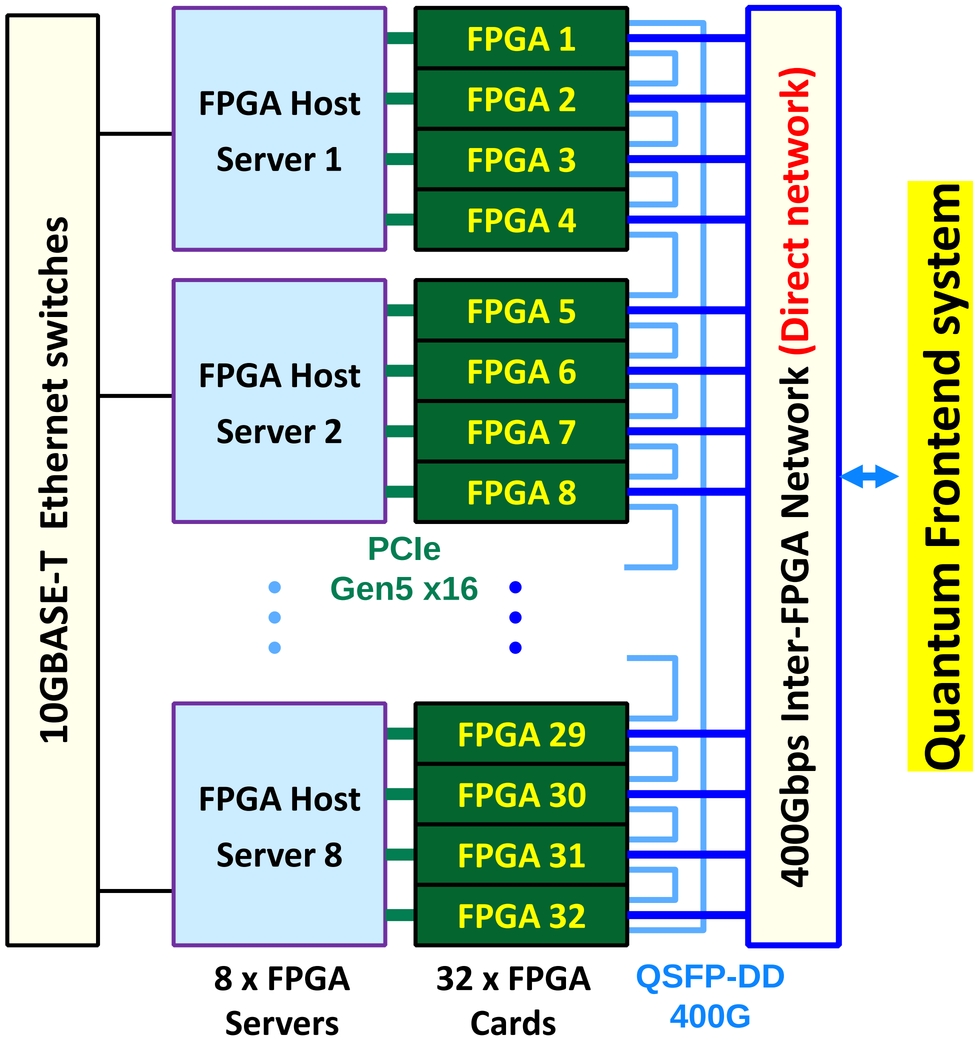
- **Target area/applications**
  - ✓ High-performance computing
  - ✓ Quantum error correction / Quantum-Classical interface
- **System stack**
  - ✓ OPAE-based : FIM/AFU/AFU Shell
  - ✓ oneAPI-based : FIM/AFU(BSP)/oneAPI
  - ✓ Custom inter-FPGA network implemented in FIM/AFU
- **Concept**
  - ✓ Common platform for HPC researchers and community
  - ✓ Standard platform for data-center
  - ✓ Encourage open-source activity on system stack development in the community

ESSPER2 for Scalable QEC

- **Specification**
  - ✓ Altera Agilex 7 M-Series FPGA
  - ✓ Memory
    - ➢ HBM2e 32GB (in FPGA package)
    - ➢ DDR5-4800 RDIMM 32GB (on board)
  - ✓ Interfaces
    - ➢ PCIe Gen5x16
    - ➢ 400G QSFP-DD x3, 72pin MCIO x2



IBEX IPAC-1000 FPGA Card



Organization of ESSPER2

Lessons Learned with ESSPER

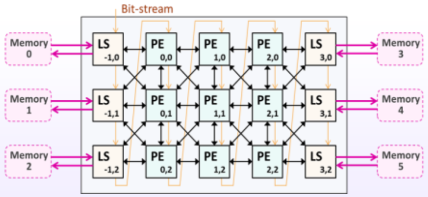
Open-Access paper

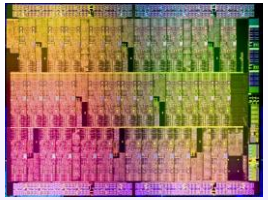
- **FPGA-based reconfigurable computing works.**
- **Productivity is not high, especially for multiple FPGAs.**
  - ✓ Even HLS requires know-how on optimization.
  - ✓ Lack of debugging tool, and simulation environment.
- **High scalability, but FP performance is lower than GPUs for major domain.**
  - ✓ FPGA has higher overhead (area, power, freq) and lower memory bandwidth.
    - ➢ Sometimes, FPGA's resource balance doesn't fit requirement (e.g., insufficient on-chip RAM).
  - ✓ Customization with FPGA can give better solution for some specific requirement:
    - ➢ E.g., non-numerical & low-latency for quantum error correction
- **Reconfigurable data-flow itself should be Okay, but**

  How can we make it a first-class citizen in HPC?

2. Exploration of new HPC & AI architectures

- ✓ Research on reconfigurable accelerator (e.g. **CGRA**)
- ✓ Research on next-generation **AI chip architecture**

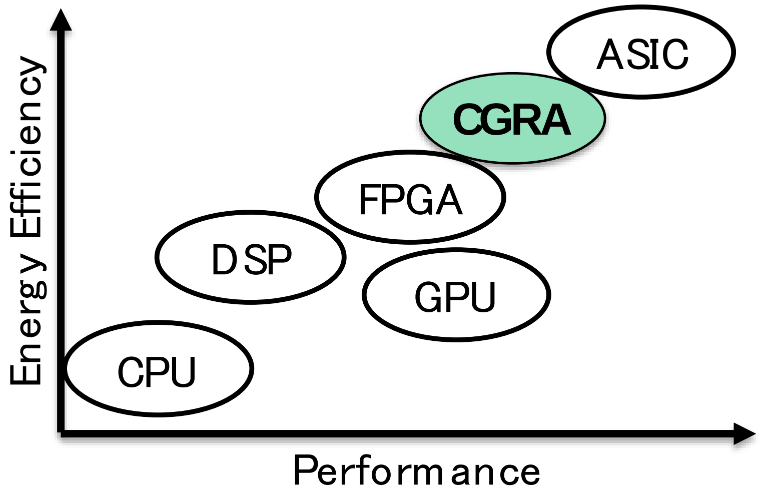




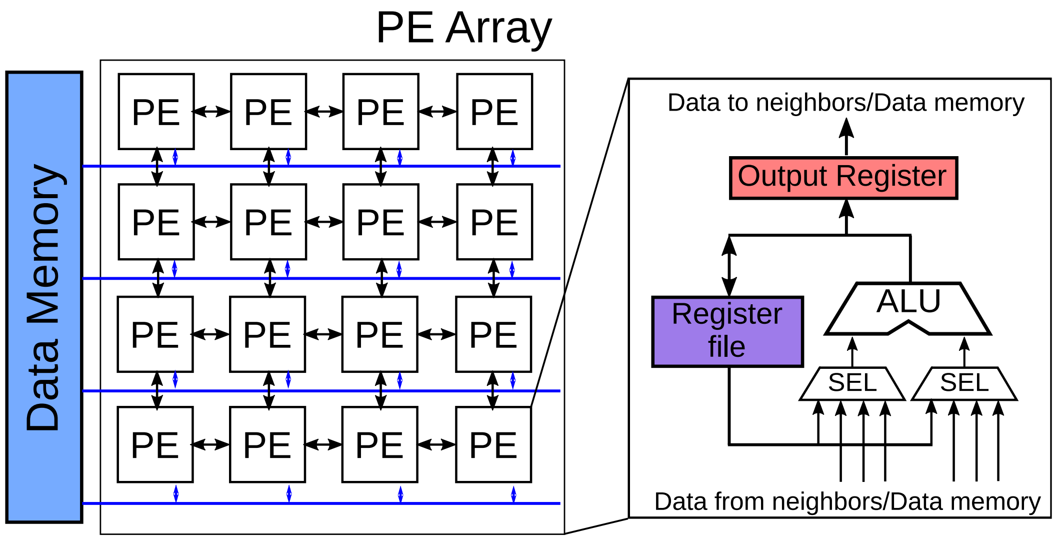
Coarse-Grained Reconfigurable Array for HPC (and AI)

Coarse-Grained Reconfigurable Array (CGRA)

- **Architecture for data-driven computing**
  - ✓ Composed of an array of processing elements (PEs), where we map DFGs for computing
  - ✓ Provide a word-level 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 deep-learning
- **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.

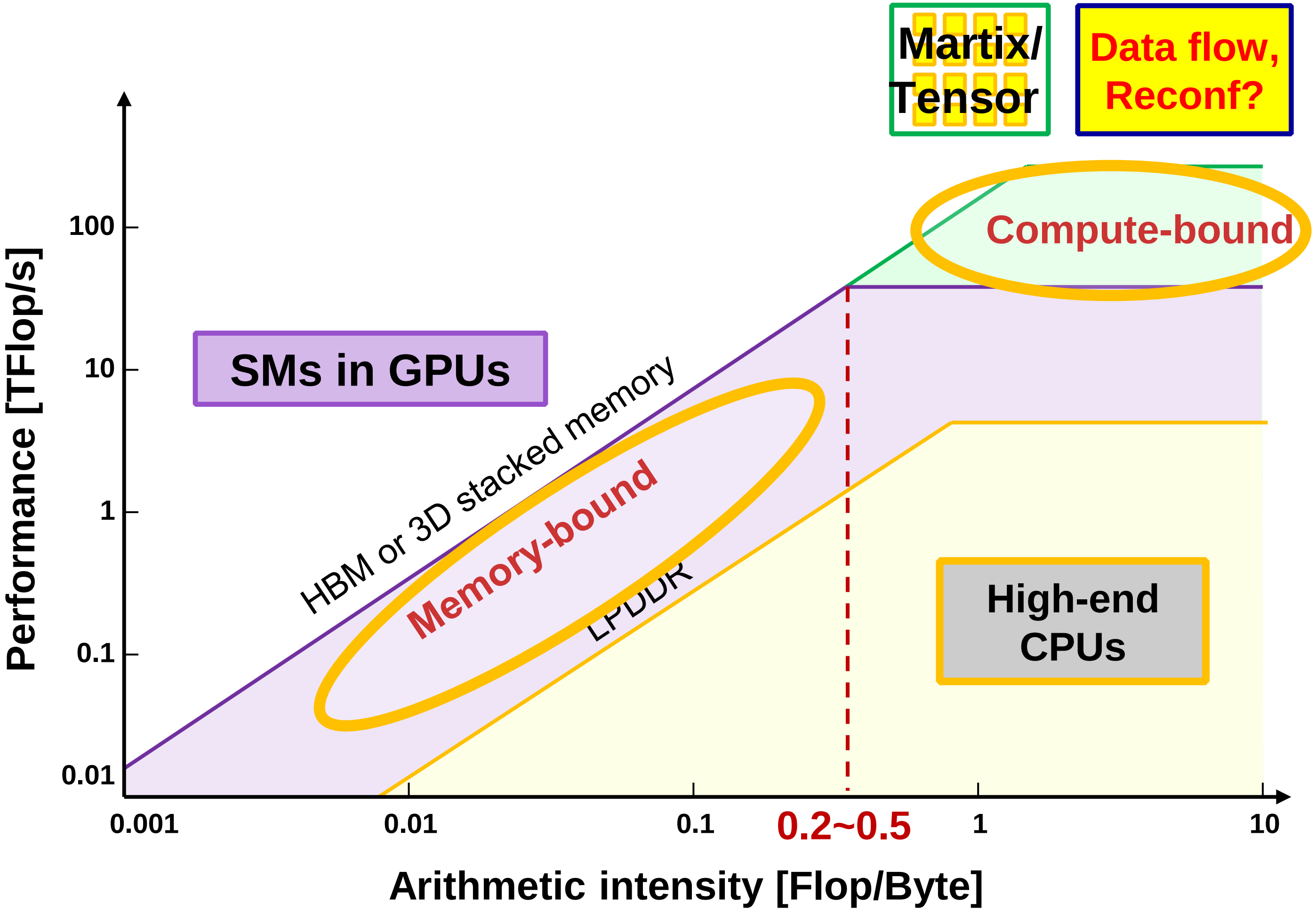
[2] Takuya Kojima, et al., "Exploration Framework for Synthesizable CGRAs Targeting HPC: Initial Design and Evaluation," Procs. CGRA4HPC, May 30-June 3, 2022.

HPC Performance Requirement in Roofline Model

- **Roofline model**
  - ✓ Peak performance available according to *arithmetic intensity*
  - ✓ *Memory-bound or Compute-bound*

- **Steaming processor can cover memory-bound applications.**

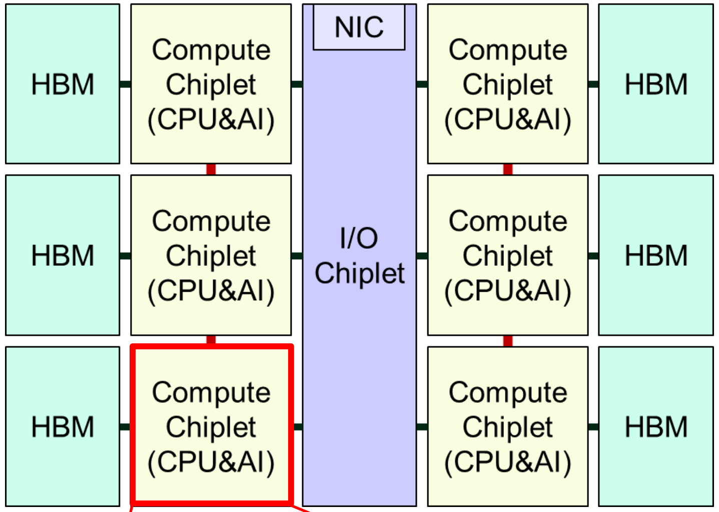
- **What architecture should be applied to compute-bound?**
  - ✓ **Higher compute density**
  - ✓ **Higher performance per power**



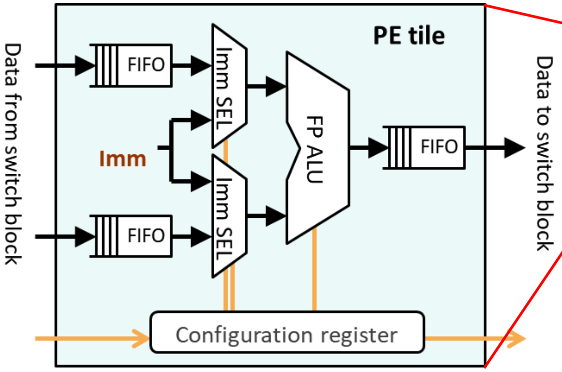
Roofline model for different performance characteristics

Baseline Reconfigurable Data-flow Architecture

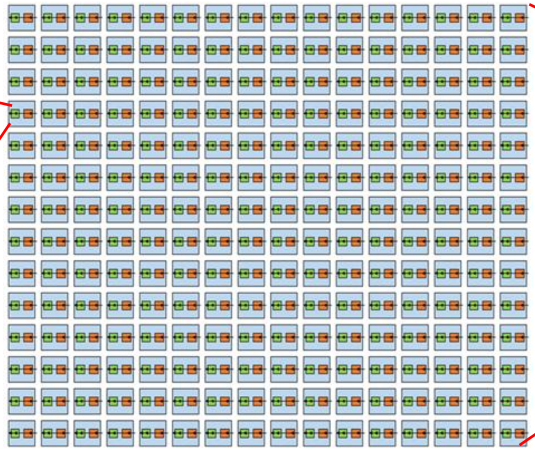
- **Coarse-grained reconfigurable array (CGRA)**
  - ✓ Data-flow computing for HPC and non-linear functions
  - ✓ Systolic computing for GEMM in AI
  - ✓ SiP/chiplet design with memory subsystem to evaluate CGRA
- **Expected outcomes**
  - ✓ Very regular memory access for high utilization of bandwidth
  - ✓ Specialized mechanism for more sparsity and mixed precision
  - ✓ Testbed for AI accelerator research



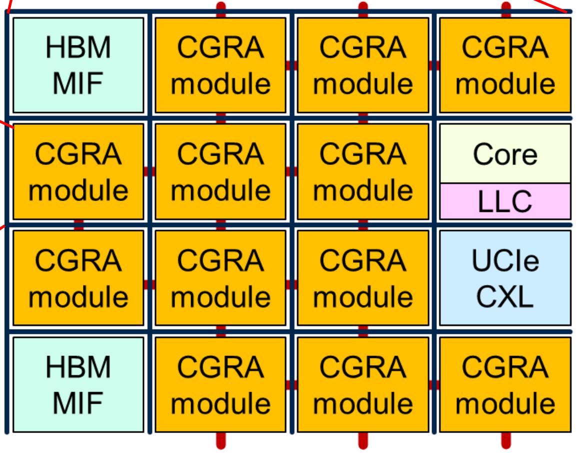
Baseline System-in-Package (SiP) design



Processing element (PE)



CGRA



Example of compute chiplet

Feasibility Study on Fugaku NEXT

Feasibility Studies on Next-Gen Supercomputing Infrastructures

Project Overview

The next-generation computational infrastructure is expected to become a platform for realizing SDGs and Society 5.0 by **providing advanced digital twins** that will bring "**Research DX**" in the science. Aiming to realize a versatile computing infrastructure that can execute entire workflow by making full use of wide range of computational methods, simulation techniques, and BigData at scale, we conduct a holistic investigation on architecture, system software and library technologies through co-design with applications.

As a basic principle of system design, we **practice the "FLOPS to Byte" concept** from architecture development to algorithm or application design to streamline data transfer and computation under power constraints, while taking necessary computing accuracy into consideration. Under the ALL JAPAN team composition, we will investigate system configurations and elementary technologies which improve effective performance of the next-generation computing infrastructure.

Subject of Investigation

Research on Architecture

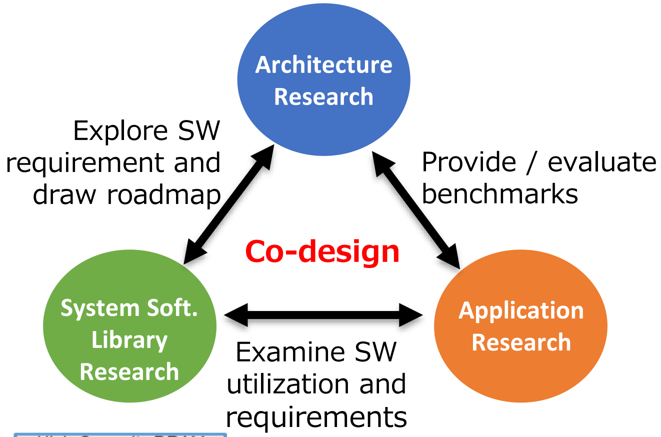
- Investigating technological possibilities (such as 3D stacked mem, accelerators, chip-to-chip direct optical link) and performance of the entire system or its components based on trends in semiconductor and packaging technologies
- Predicting future system performance based on performance analysis of benchmark sets provided by Application Research Group, and feeding back to next-generation application development

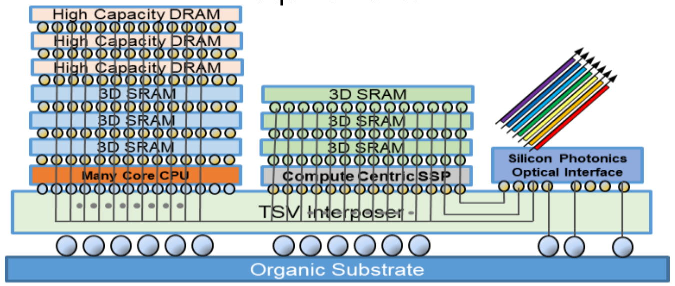
Research on System Software and Library

- Drawing roadmap for future system software development in Japan, specially considering data utilization enhancement, integration of AI technology with first-principles simulation, real-time data processing, and assurance of high security

Research on Applications

- Building a broad benchmark set to evaluate multiple architecture choices while considering improvements in algorithms and parameters of application based on the results of architectural evaluations and exploratory "what-if" performance analysis
- Investigating what classes of algorithms are expected to evolve significantly for future systems


Architecture Research
Explore SW requirement and draw roadmap
Provide / evaluate benchmarks
Co-design
System Soft. Library Research
Application Research
Examine SW utilization and requirements



Strawman processing element architecture

Investigation Schedule

	2022 Q3	2022 Q4	2023 Q1	2023 Q2	2023 Q3	2023 Q4	2024 Q1
Architecture	Explore device/architecture technology			Performance estimation with benchmarks		Architecture study	
System Software	Examine existing SW and its utilization			Identify requirement of SW development		Draw roadmap	
Application	Examine existing apps and benchmark design			Perf. analysis by benchmark evaluation		Study algorithm improvement	

Summary

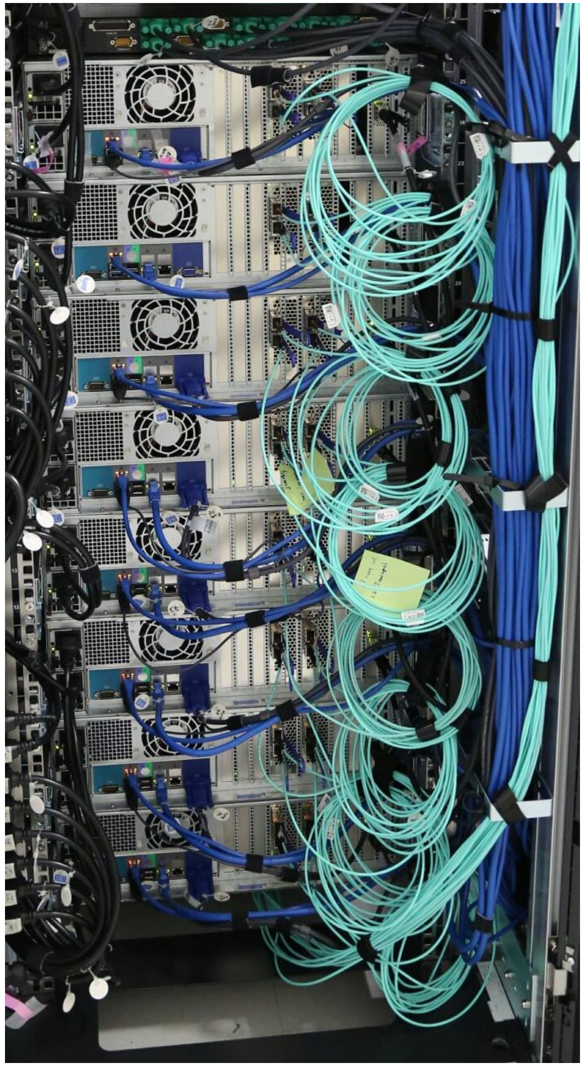
Reconfigurable data-flow computing should be promising for power-efficient HPC.

- **FPGAs are suitable for domain-specific computing.**
  - ✓ *ESSPER: FPGA cluster testbed*
  - ✓ *Quantum error correction*
- **Researching CGRA for general HPC.**
  - ✓ *RIKEN CGRA for HPC and AI*
  - ✓ *Need engineering work and compiler development*

Future work

- ✓ **ESSPER2 with Altera Agilex-M FPGA** (mainly for Quantum research)
- ✓ SoC design of CGRA for HPC and AI (preparation for future ASIC)
- ✓ Have more collaboration!

Hiring researchers,
Contact me!



PRELIMINARY

HEART 2025

International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies

May. 26-28, 2025 Kumamoto Japan

Submission Due: Feb 25, 2024

Overview

The International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies (HEART) is a forum to present and discuss new research on computing systems utilizing acceleration technology. The main theme of HEART is achieving high efficiency with accelerators. Today, performance acceleration with high efficiency is highly demanded in various computing domains such as high-performance computing and data centers.

In HEART 2025, we focus on power, energy and algorithmic efficiency on the AI technology such as **LLM** (large-language model) **and beyond.**

Thank you !