

Stream Computing of Lattice-Boltzmann Method on Intel Programmable Accelerator Card

Takaaki Miyajima, Tomohiro Ueno and Kentaro Sano

Processor Research Team,
RIKEN Center for Computational Science,
Kobe, Hyogo, 650-0047, Japan

Agenda

- Motivation and Goal: Provides a common platform for streaming computation with multiple FPGAs
- Our platform
 - ✓ Overview
 - ✓ Intel PAC (Programmable Accelerator Card)
 - Hardware: Arria10 FPGA + I/O + PCIe
 - Software: Open Programmable Acceleration Engine
 - ✓ AFU Shell: DMA transfer API
 - ✓ Application: Lattice Boltzmann Method
- Summary



Motivation and goal of our research

- Motivation

- ✓ Although FPGA gathers increasing attention in HPC area, is still low. Especially for multiple FPGAs cluster.
- ✓ We don't want to reinvent the wheel (Device driver, DMAC,
- ✓ Bad for collaborative research :(

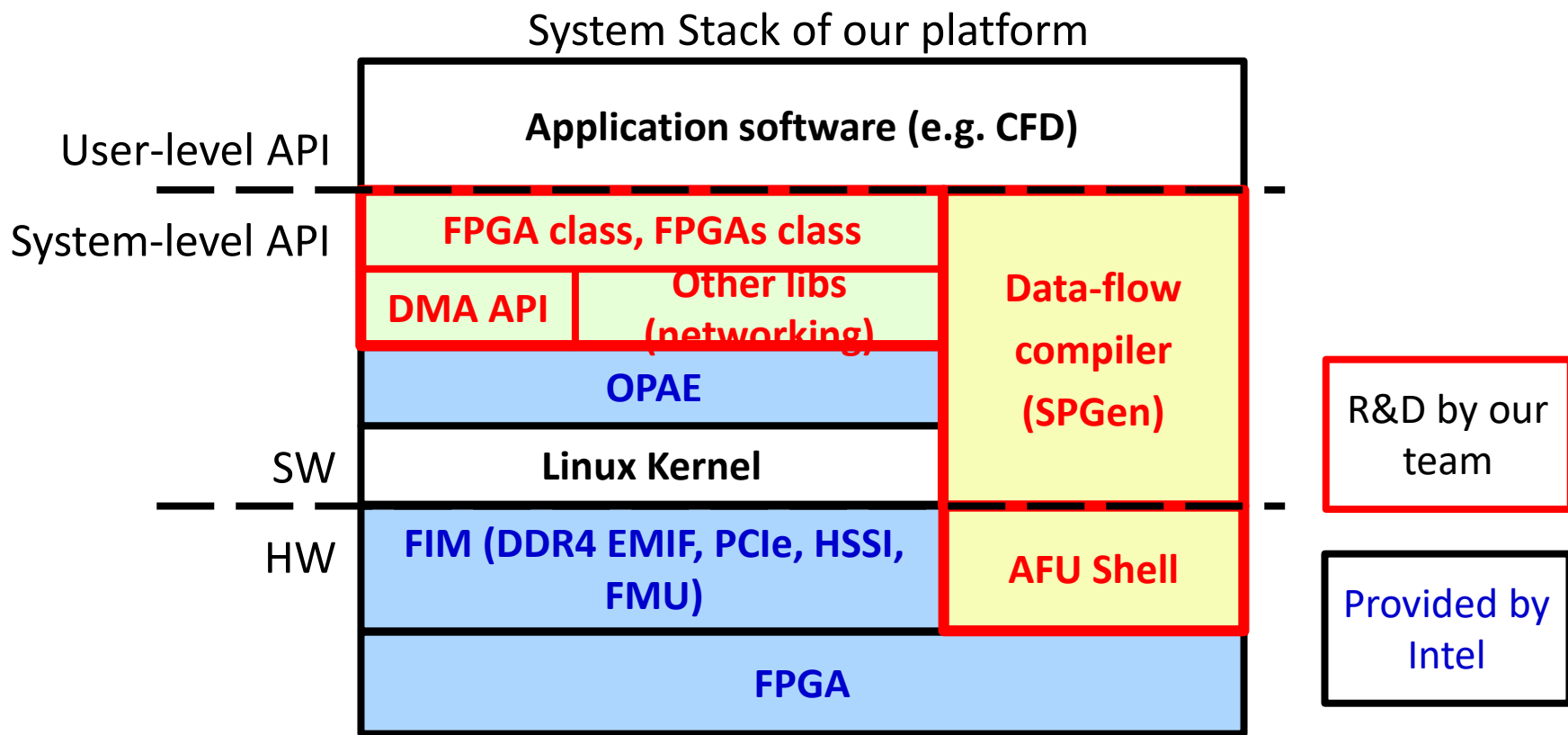
- Goal: **Provides a common platform for streaming computation with multiple FPGAs.**

- ✓ Common software interface
- ✓ Common hardware modules
- ✓ Portable bitstream
- ✓ Good for collaborative research :)

Our platform

Overview of our platform

- We are researching and developing C/C++ common APIs, data-flow compiler and underlying hardware modules.
- Intel FPGA (Intel Acceleration Stack) is a base system currently.

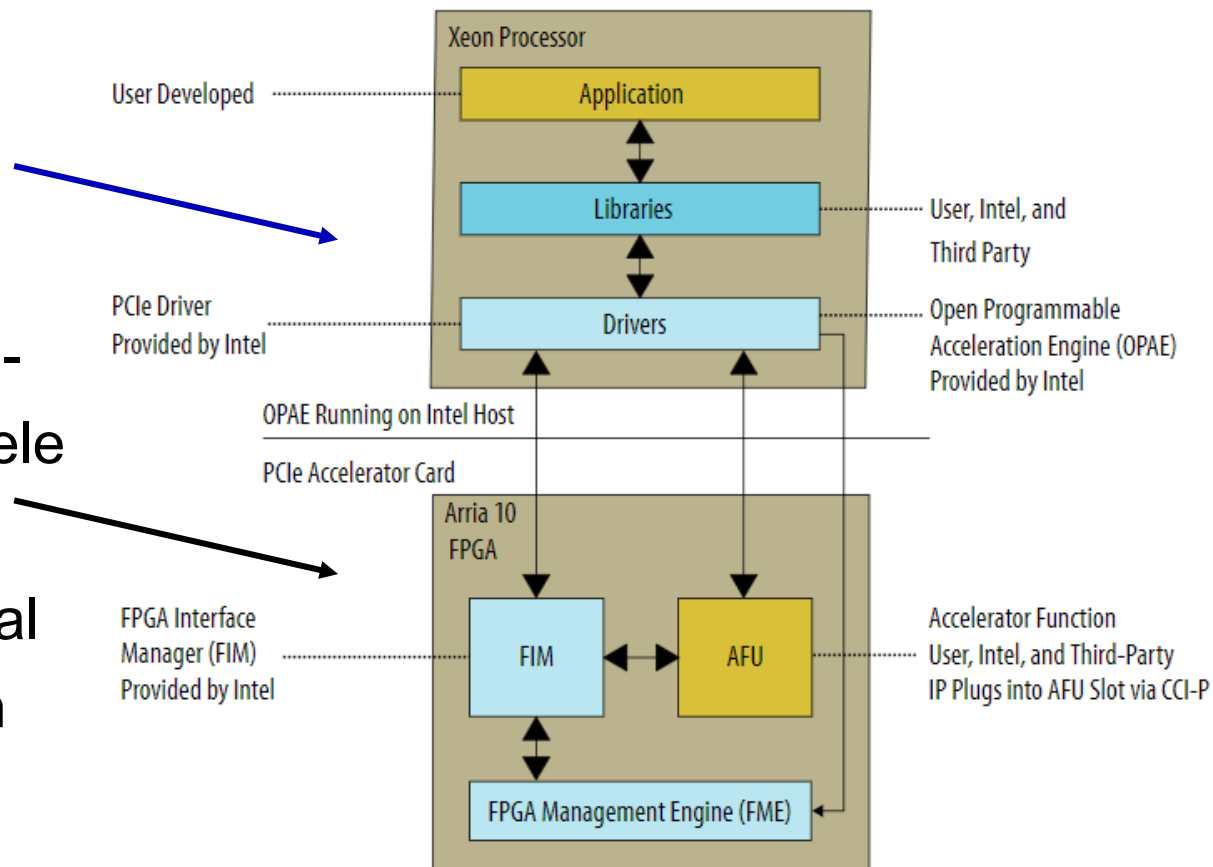


Intel Acceleration Stack (IAS)

- IAS is a robust collection of software, firmware, and tools to make it easier to develop and deploy Intel FPGAs.

- SW: Open Programmable Acceleration Engine(OPAE)

- HW: Intel Programmable FPGA Card. (Accelerator Function unit including CCI-P, partial reconfiguration region and so on.)



Intel Programmable Accelerator Card

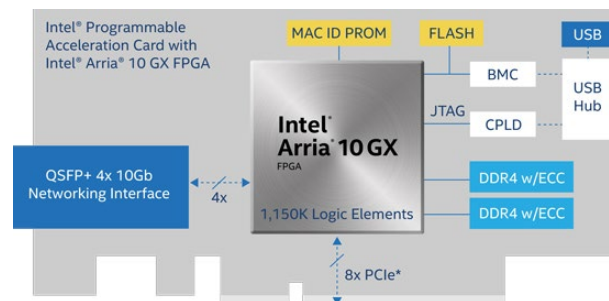
- PAC = PCIe-base FPGA board
- FPGA: Intel Arria 10 GX
 - ✓ 10AX115N2F40E2LG
 - ✓ SERDES transceiver
(15 Gbps per port at maximum)
 - ✓ 1150K Logic elements
(Speed grade: -2L)
 - ✓ 53 Mb Embedded Memory
- Memory
 - ✓ 8 GB DDR4, 2 channels
- External Port
 - ✓ PCIe Gen3 x8
 - ✓ 1X QSFP (4X 10GbE or 40GbE)



Appearance of PAC



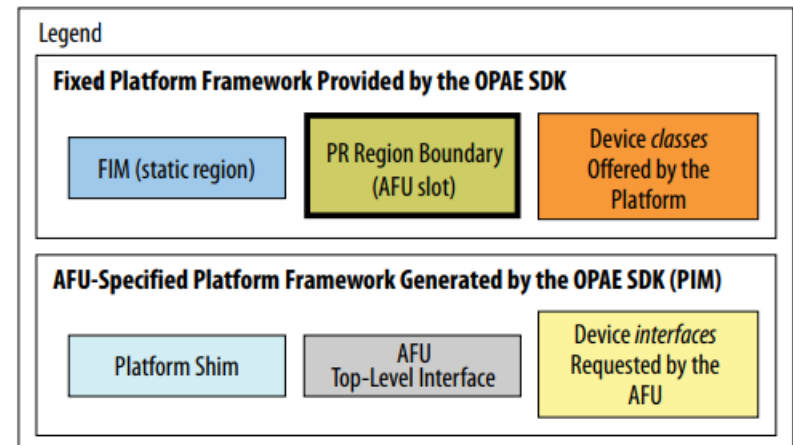
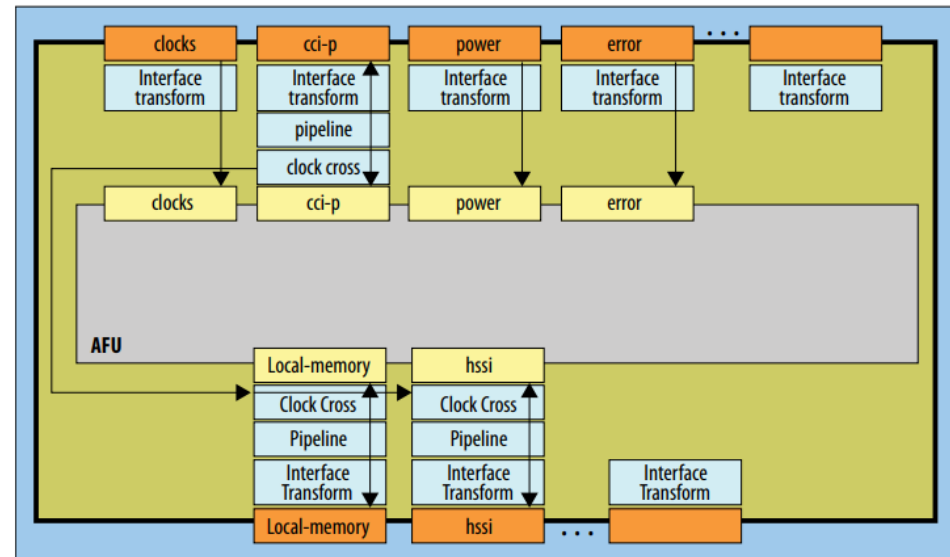
Substrate of PAC



Block diagram

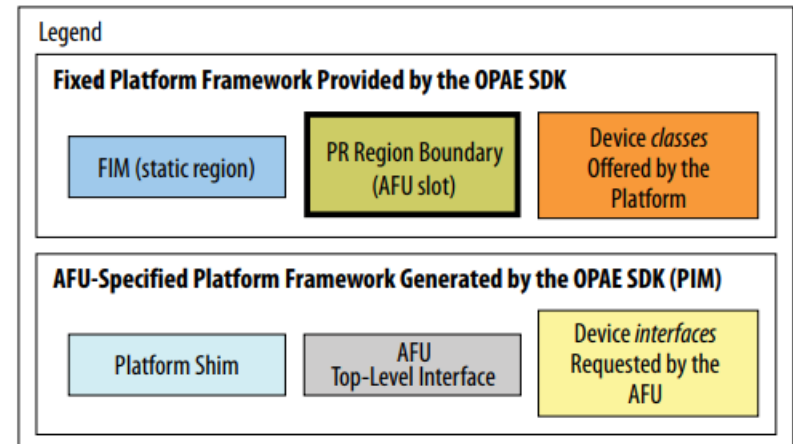
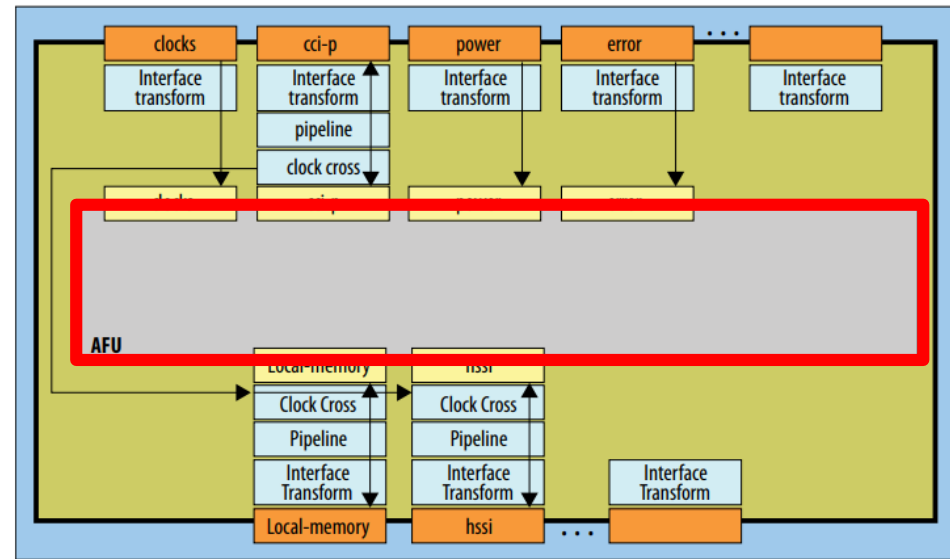
Hardware part (on FPGA)

- Hardware part includes
 - ✓ Accelerator Function Unit
 - ✓ FPGA Interface Manager
 - ✓ FPGA Management Engine
 - ✓ FPGA Interface Unit
- Common I/O and sample design are prepared :)
 - ✓ 10, 40GbE
 - ✓ DDR4, DMA controller
- AFU is a computation logic preconfigured on FPGA designed in RTL.



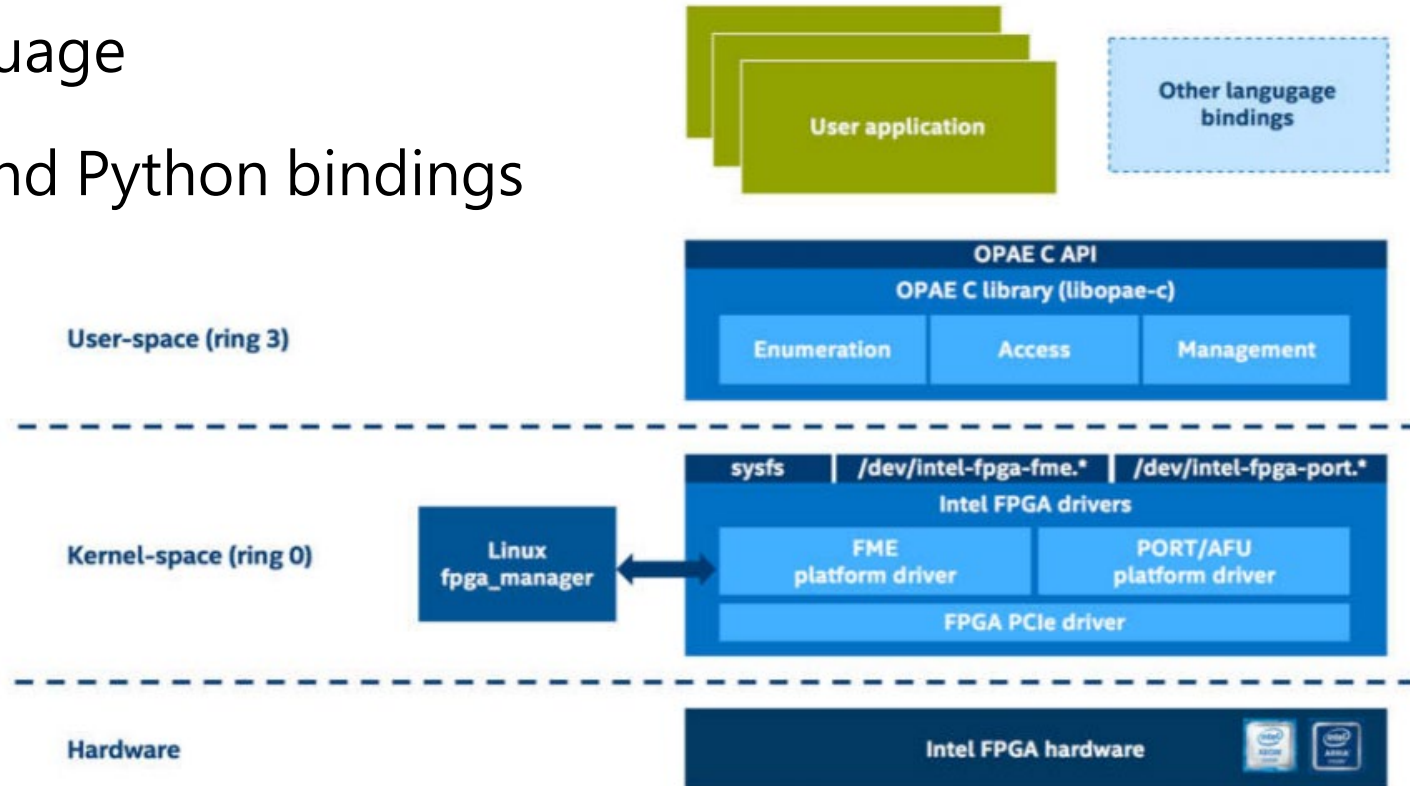
Accelerator Functional Unit (AFU)

- AFU is a computation logic preconfigured on FPGA. The logic is designed in RTL and synthesized into a bitstream.
- It contains AF and control and status registers. It represents a resource discoverable and usable by your applications. *fpgaconf* is provided to reconfigure an FPGA using a



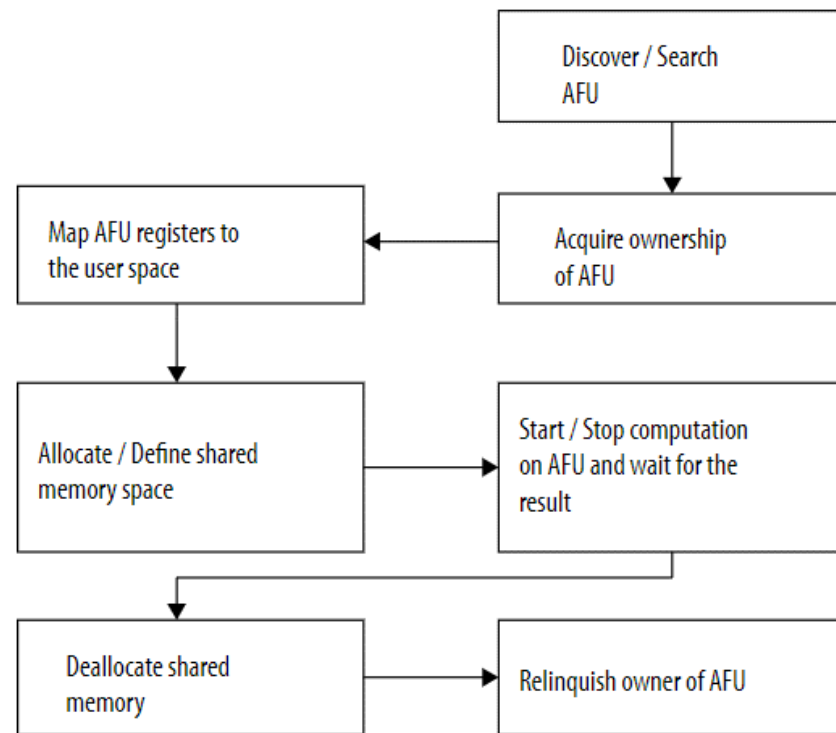
Software part (on Host computer) = Open Programmable Acceleration Engine

- The OPAE library is a lightweight user-space library that provides abstraction for FPGA resources in a compute environment.
- C language
- C++ and Python bindings



Controlling hardware with OPAE

1. Discover/Search AFU
2. Acquire ownership of AFU
3. Map AFU registers to the user space
4. Allocate / Define shared memory space
5. Start / Stop computation on AFU and wait for the result
6. Deallocate shared memory



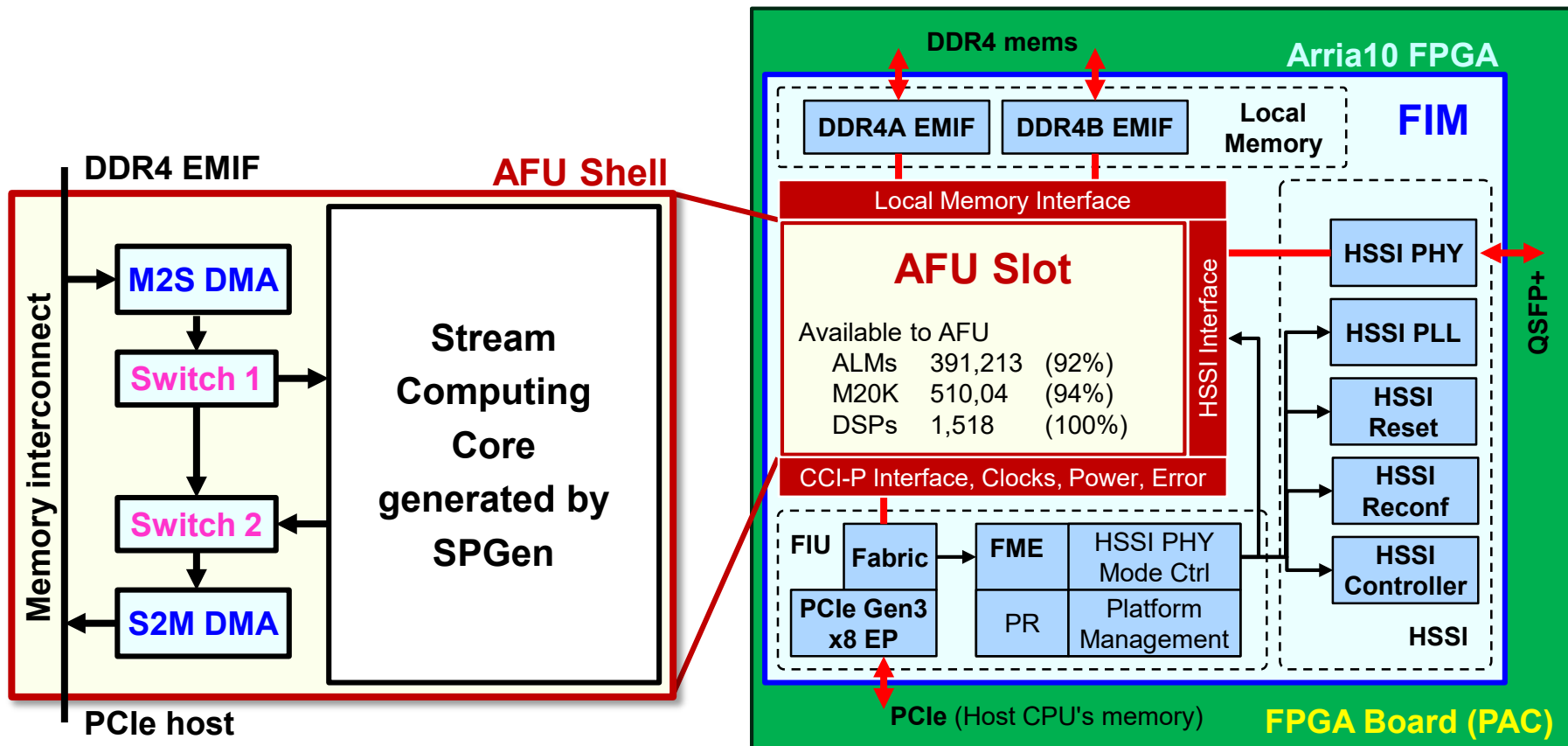
Difference between BSP

- Board Support Package (BSP) is provided by FPGA vendor to use OpenCL on their boards.
- IAS is more flexible than BSP and gives the users more responsibility.
 - ✓ No limitations of BSP.
 - ✓ No need to write OpenCL runtime.
 - ✓ Need to write almost everything...
- Requires more comprehensive system-wide knowledge
 - ✓ e.g. DMA controller and drivers.

AFU Shell and DMA Transfer API

AFU Shell for Intel PAC w/ Arria10

- AFU Shell is a base hardware of our platform.
 - ✓ includes two DMA controller and computing core.
 - ✓ Semi-automated design flow for FIM & OPAE 1.1beta
 - ✓ "make (for spgen); make embed generate makeegbs (for



DMA hardware and API for AFU Shell

- **DMA Controller is based on mSGDMA**

- ✓ TX DMAC and RX DMAC are separated for flexibility

FPGA/module	ALMs	Registers	BRAM Kbits	DSPs
afuShell (2 DMACs)	2905	3055	135040	27

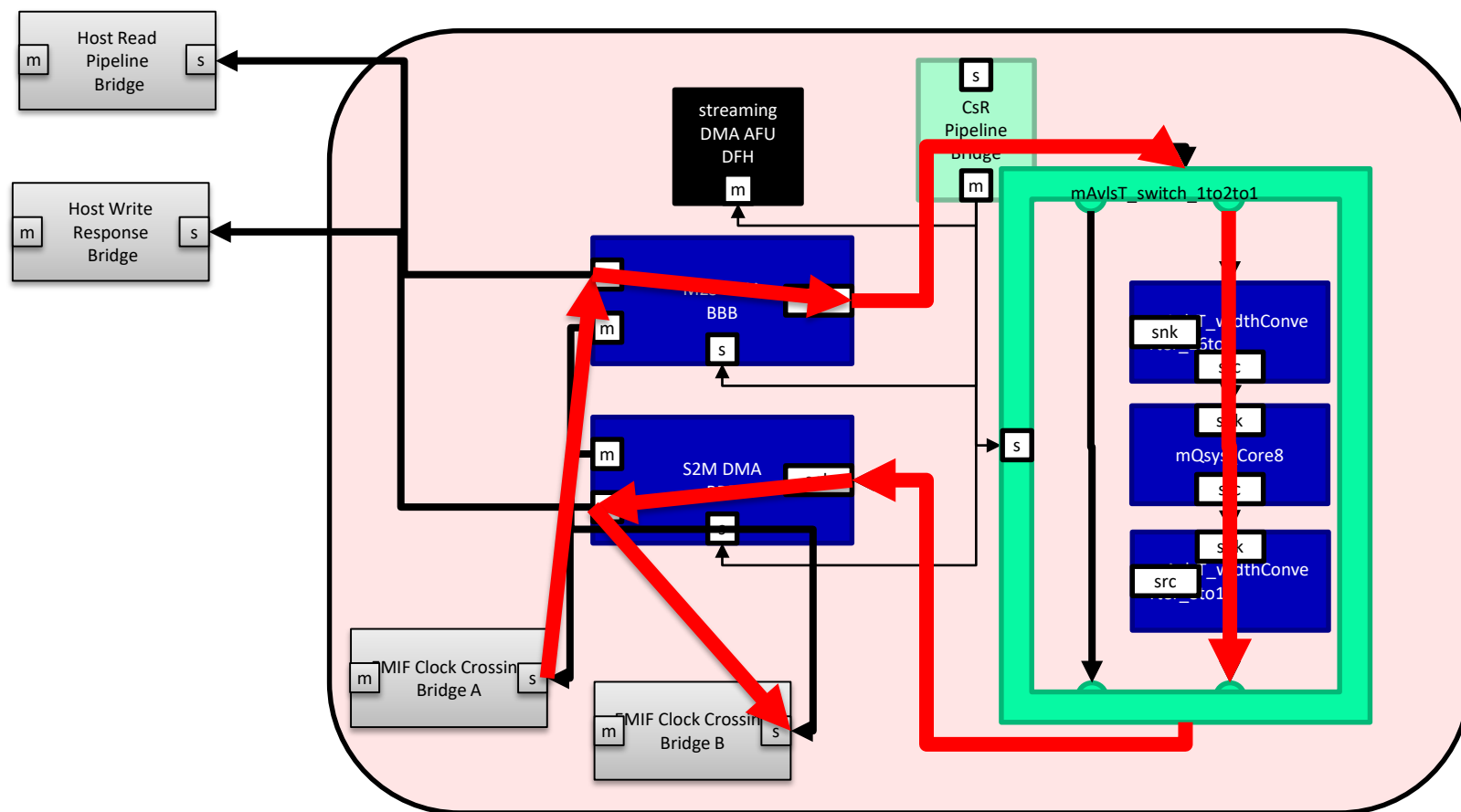
- **DMA Transfer API (Synchronous)**

- ✓ *fpga_result afuShellDMATransfer(
void* dst,
const void* src,
size_t count,
dma_transfer_type_t type)*

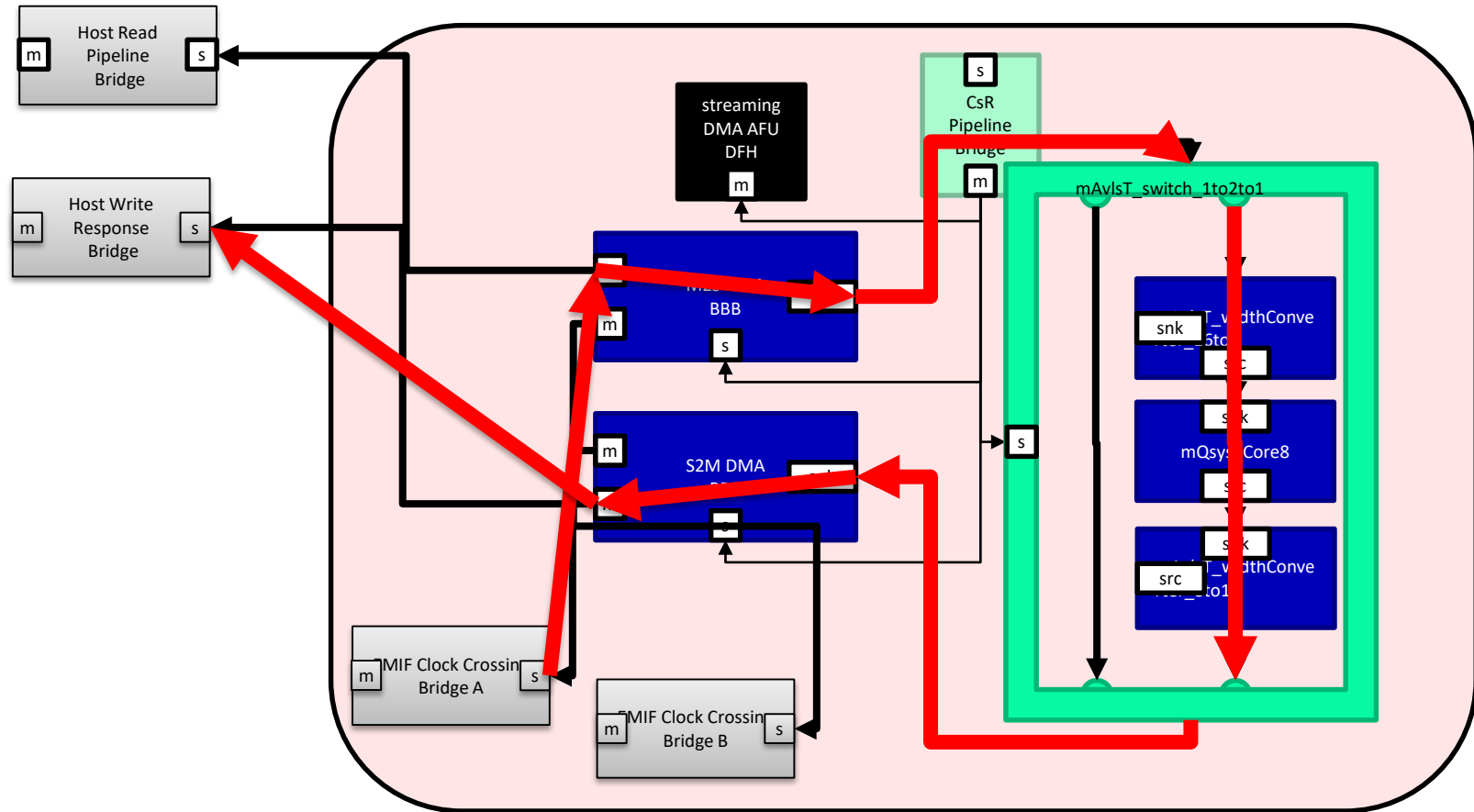
- **Four types of data transfer are supported**

- ✓ HOST→FPGA, FPGA→HOST, FPGA→FPGA, HOST→HOST

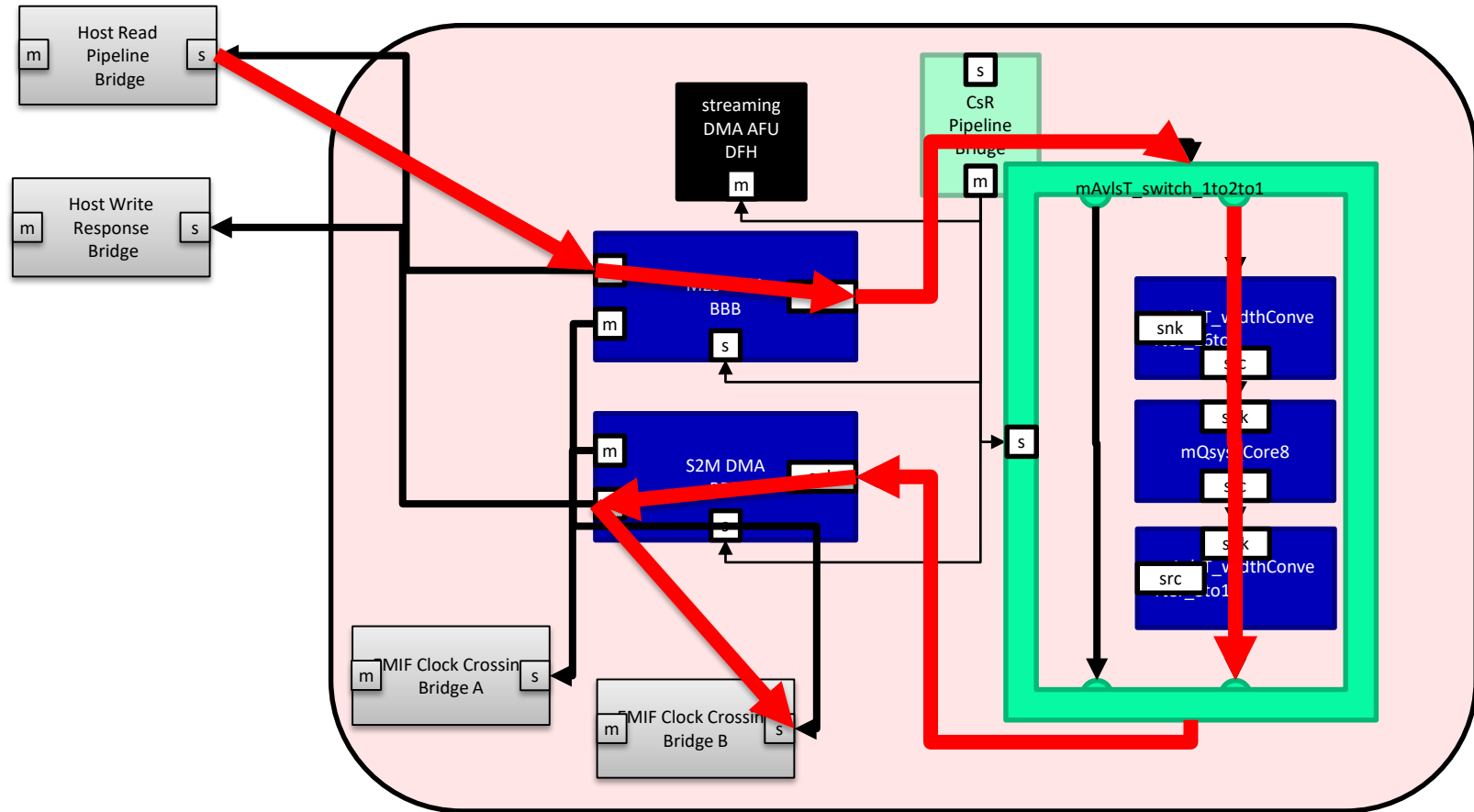
afuShellDMATransfer(0x0, 0x100, 100, FPGA_TO_FPGA)



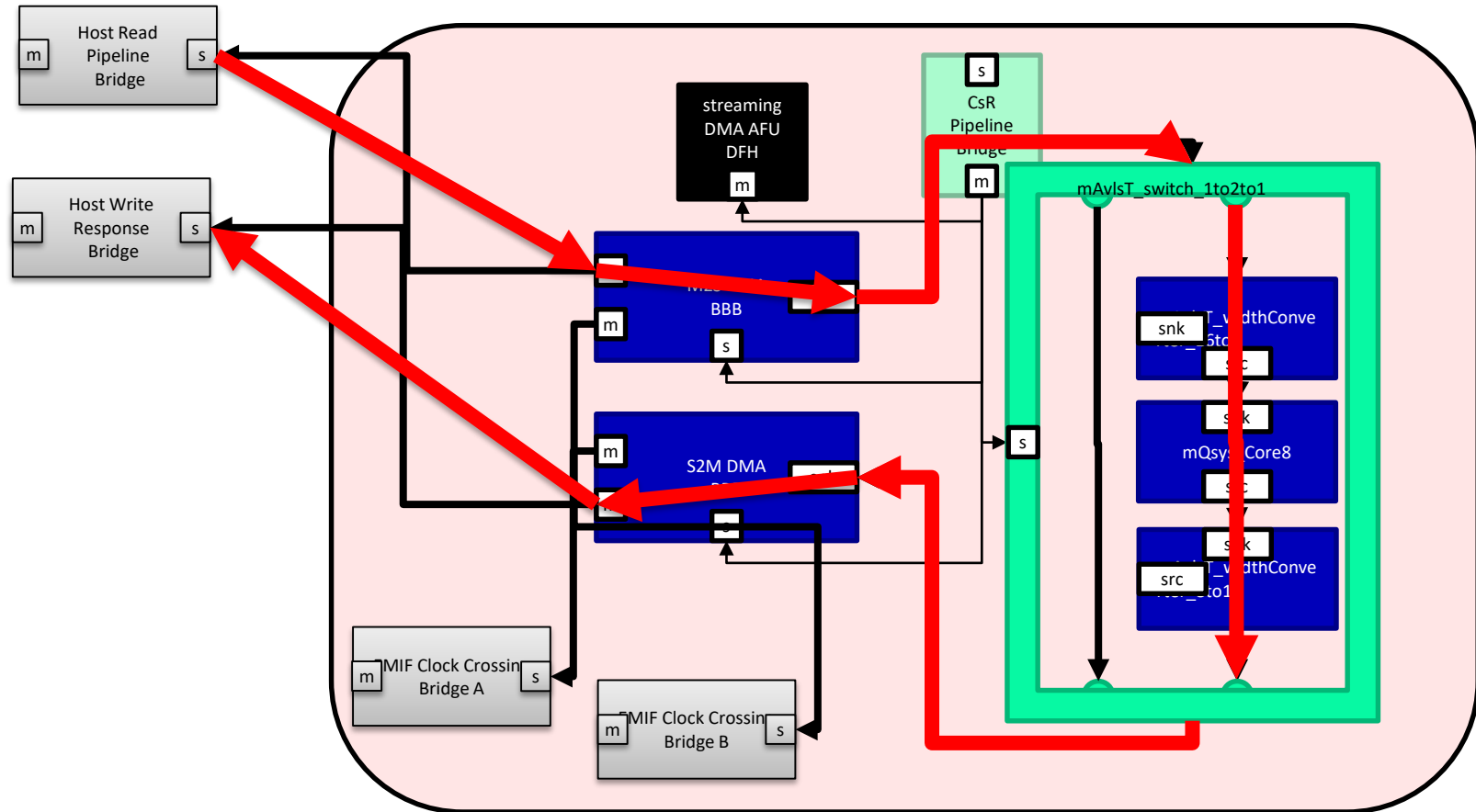
afuShellDMATransfer(0x0, host_dst, 100, FPGA_TO_HOST)



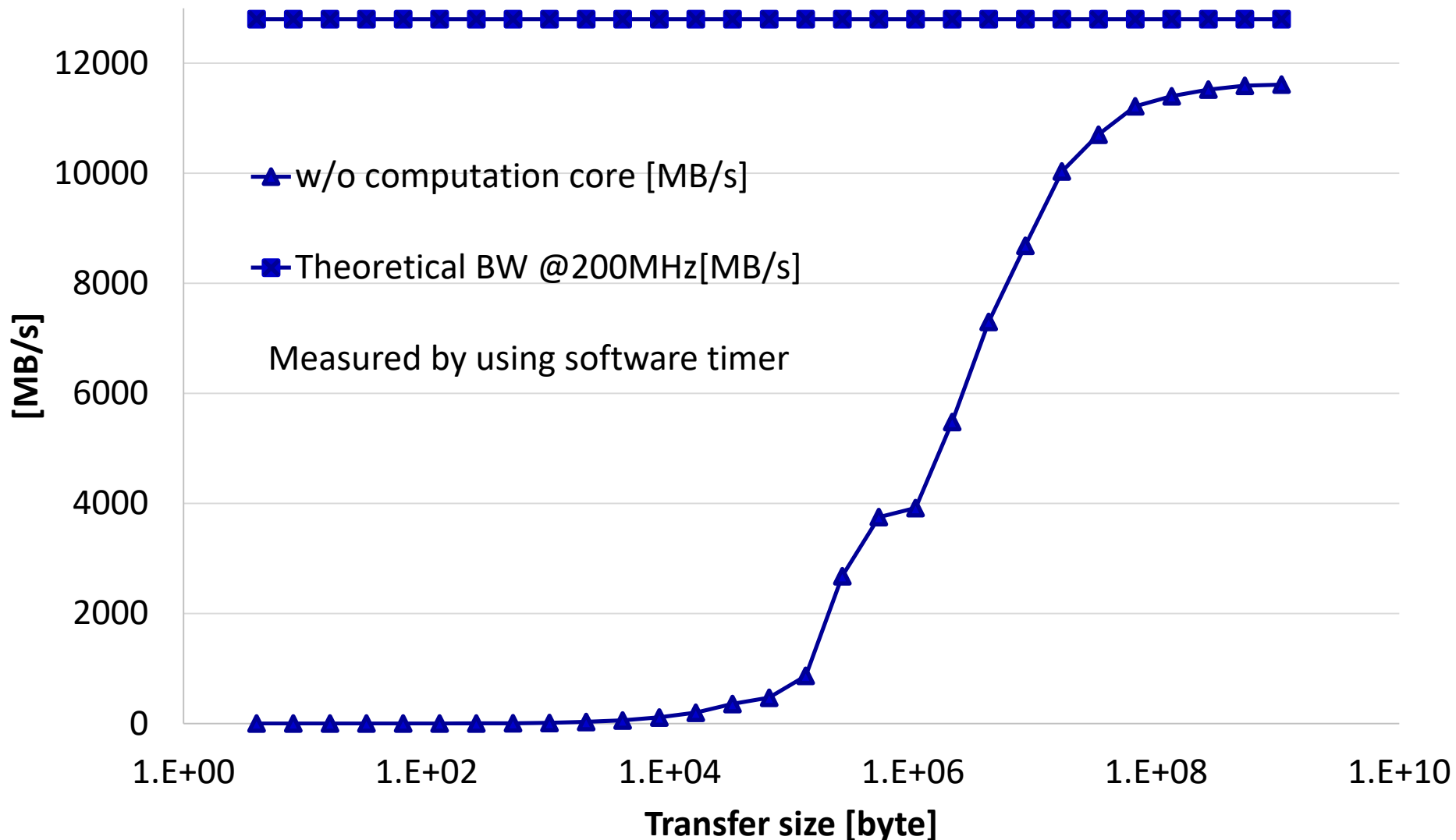
afuShellDMATransfer(host_src, 0x100, 100, HOST_TO_FPGA)



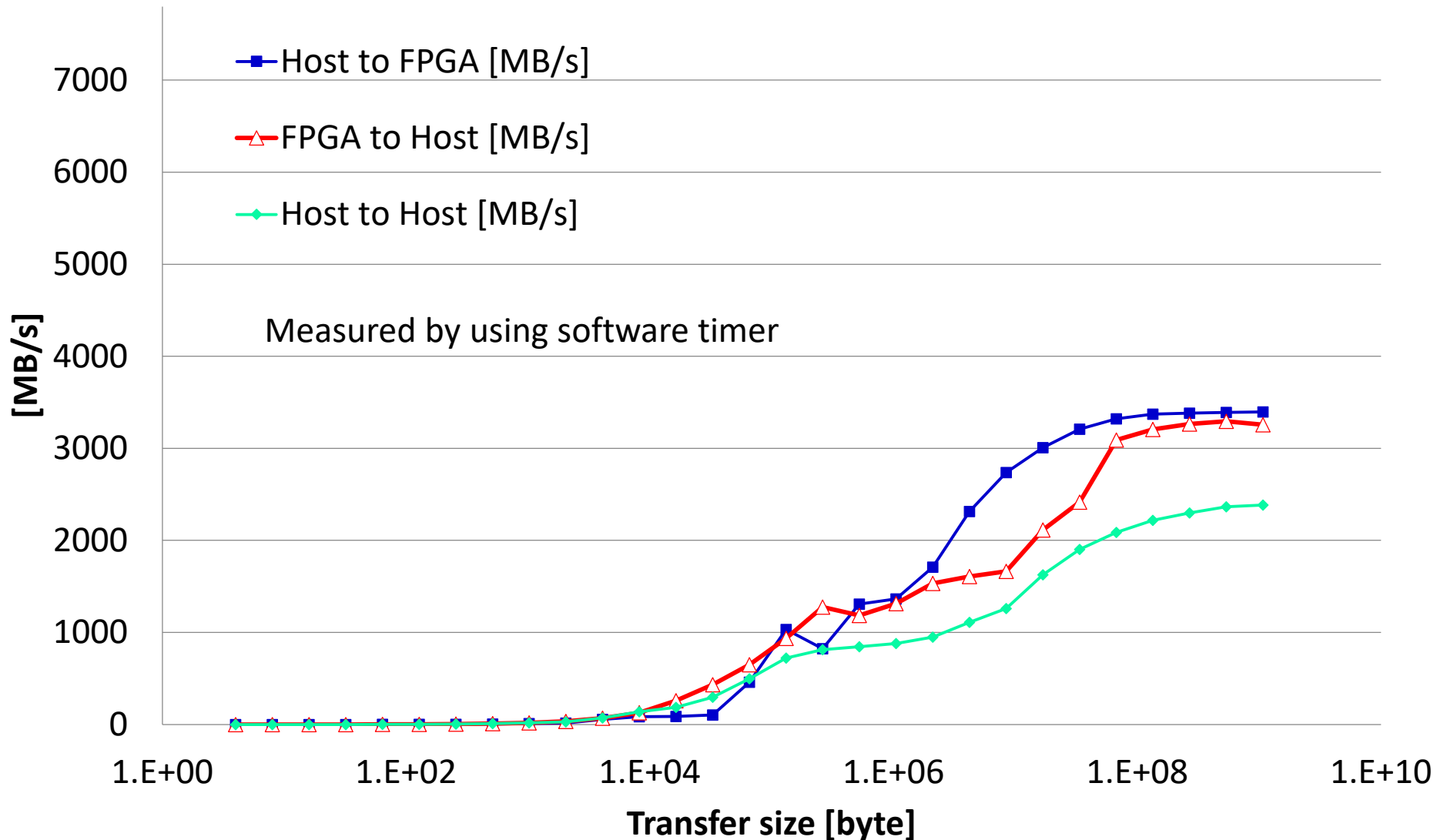
afuShellDMATransfer(host_src, host_dst, 100, HOST_TO_HOST)



Bandwidth: FPGA and FPGA

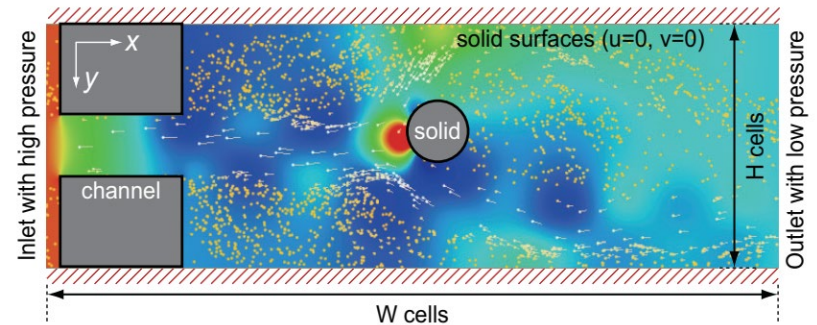


Bandwidth: Host and FPGA



Sample app: Lattice Boltzmann Method

- We put LBM computing core [1] into the AFU Shell
 - ✓ 131 Single precision floating-point / LBM core
 - ✓ Working frequency: 200MHz
 - ✓ Input width: 40byte
 - ✓ Required bandwidth: $200\text{MHz} * 40\text{byte} = 8000 \text{ MB/s}$
 - ✓ Theoretical perf. : $200\text{MHz} * 131 \text{ FP} = 26.2 \text{ GFlops / LBM}$
- Our LBM core can improve performance by cascading the core without increasing input bandwidth.



Example result of our LBM

[1] K. Sano and S. Yamamoto, "Fpga-based scalable and power-efficient fluid simulation using floating-point dsp blocks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2823–2837, Oct 2017.

Performance of LBM

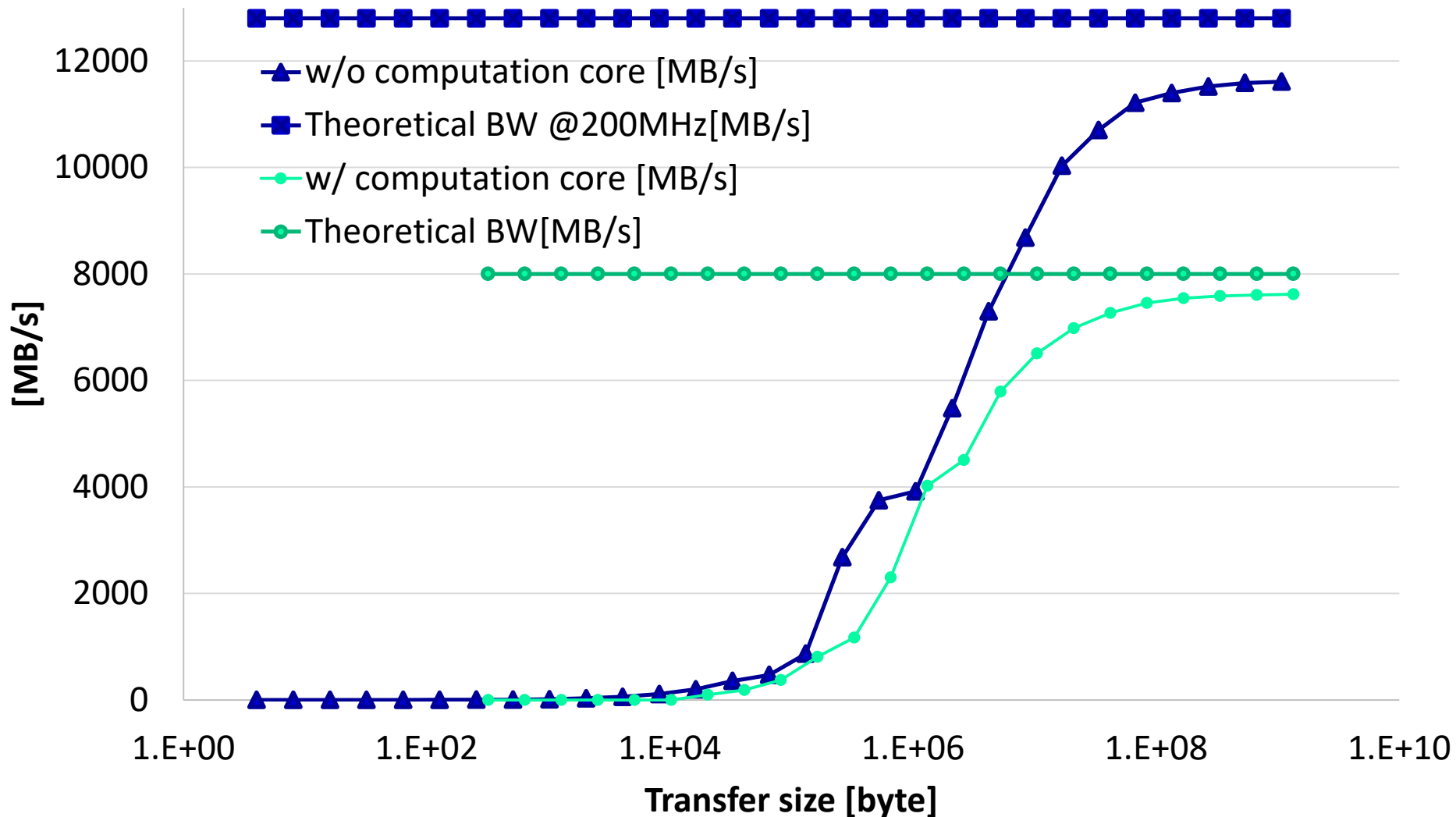
- Eight LBM can be cascaded, currently

# of cascaded core	1	2	4	8
ALMs	2.85	1.94	1.06	5.53
Registers	5.62	3.85	2.3	11.06
BRAM Kbits	11.1	7.66	4.78	22.13
DSPs	22.1	15.28	9.74	44.26
Theoretical performance [Gflops]	26.2	52.4	104.8	209.6

- **Sustained performance**

- ✓ Initial data on a FPGA DRAM channel go through LBM core, back to the other FPGA DRAM channel.
- ✓ Ratio of stall cycles to total cycles is measured by HW counter it is 1.04×10^{-5} when the input data size is 1.92MB
- ✓ Thus, sustained performance for each implementation is the as theoretical peak.

LBM core bandwidth (FPGA and FPGA)



Summary

- We are researching and developing a common platform for streaming computation with multiple FPGAs based on Intel PAC
- Intel PAC consist of Arria10 FPGA (HW) and OPAE (SW)
- AFU Shell is a base hardware of our platform including DMA Controller and API.
 - ✓ afuShellDMATransfer ()
- Sample application: Lattice Boltzmann Method
 - ✓ Sustained performance is equal to theoretical performance.
- Need collaboration :)

Call For Participation

FPT'18

International Conference on Field-programmable Technology
Dec 10-14, 2018, Naha, Okinawa, Japan

Online Registration Due: Nov 25

<http://www.fpt18.sakura.ne.jp>



Program: Technical Sessions, Keynote/Invited Talks,
Workshops, and Design Contest



Dec 12

Keynote Speaker

Hideharu Amano

Keio University, Professor

"Accelerator-in-Switch:
a novel cooperation
framework for FPGAs and
GPUs"



Dec 13

Keynote Speaker

Mike Strickland

Intel PSG, Director

"FPGA Accelerated HPC
and Data Analytics"



Dec 13

Invited Speaker

Hiroshi Miyata

Fujitsu Laboratories Ltd., Fellow

"Digital Transformation of
Automobile and Mobility
Service"



Dec 14

Keynote Speaker

Kees Visser

Xilinx, Fellow

"Novel Neural Network
Applications on New Python
Enabled Platforms"

Sponsor



Technical co-sponsors



In participation with



Supporters

Platinum



Gold



Silver



Bronze

