Accelerator-in-Switch: a novel cooperation framework for FPGAs and GPUs
– Building a Virtual Heterogeneous Computing System –

2018.12

Hideharu Amano

Keio University

This work is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO)
Special Thanks to T.Kudoh, R.Takano, T.Ikegami, M.Koibuchi, A.B. Ahmed, K.Hironaka, K.Musha, K.Azegami, K.Iizuka, Y.Yamauchi, M.Yamakura and other members of the project.

FPGA computing in cloud

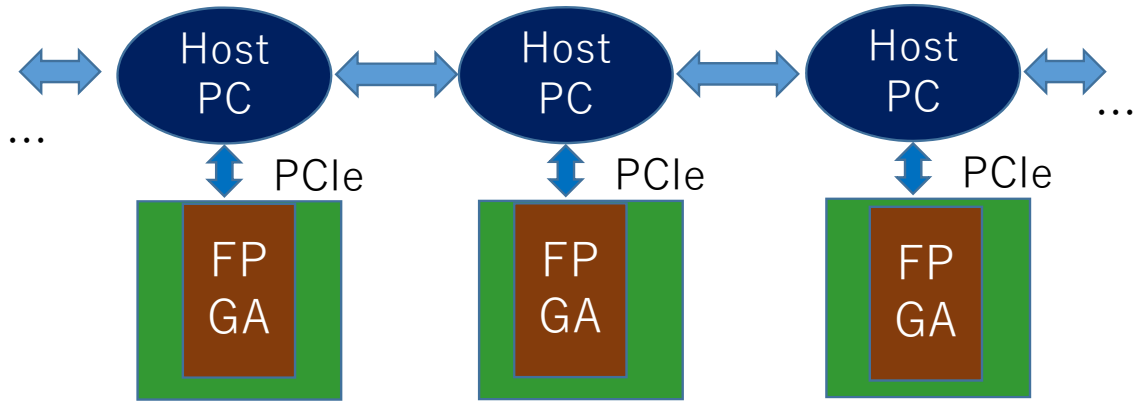
- High Performance FPGAs are available:
  - High computational power: Intel Stratix-10
  - With a large memory: Xilinx UltraScale+ with UltraRAM
  - But they are expensive for most users to keep themselves.
- Programming environment is improved:
  - Open-CL is widespread for computational usage.
  - Vivado-HLS is popularly used for general usage.

→ FPGAs in cloud:

More flexible and power efficient than using GPU.

- FPGA in the Cloud: Booting Virtualized Hardware Accelerators with OpenStack [FCCM2014]
- Microsoft Catapult [ISCA2014][HEART2017]
- FPGA Supervessel Cloud by IBM[ICFPT2016]
- Amazon EC2 F1 Instance [https://aws.amazon.com/ec2/instance-types/f1/]

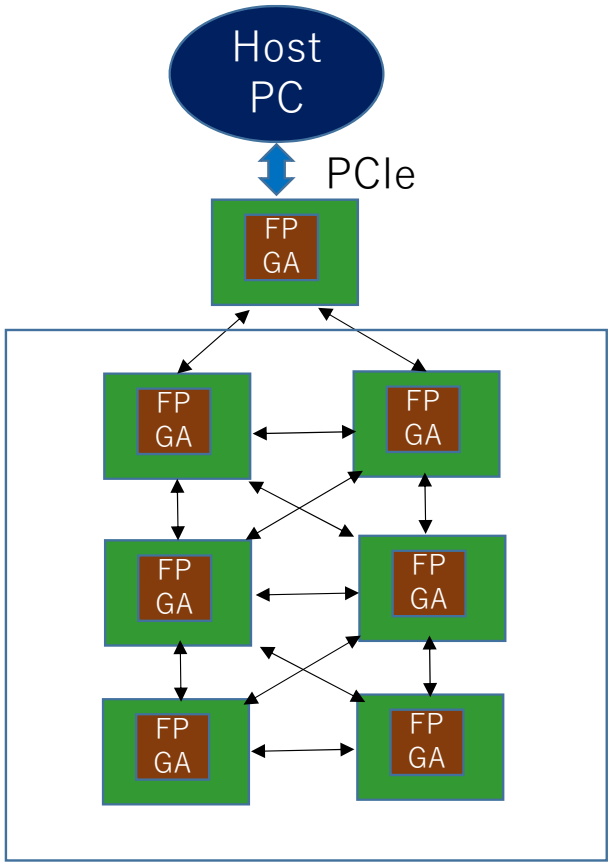
Conventional FPGA-in-Cloud



High-Performance FPGAs are attached into each Host.

- The total cost becomes large.
  - High performance FPGAs are still expensive.
- The size of FPGA is limited.
  - Multiple FPGAs cannot be used together.

Our Proposal : Virtual Large FPGA



A lot of cost-efficient middle-scale FPGAs are tightly connected.
They can be treated as if they were a single FPGA in HLS description level.

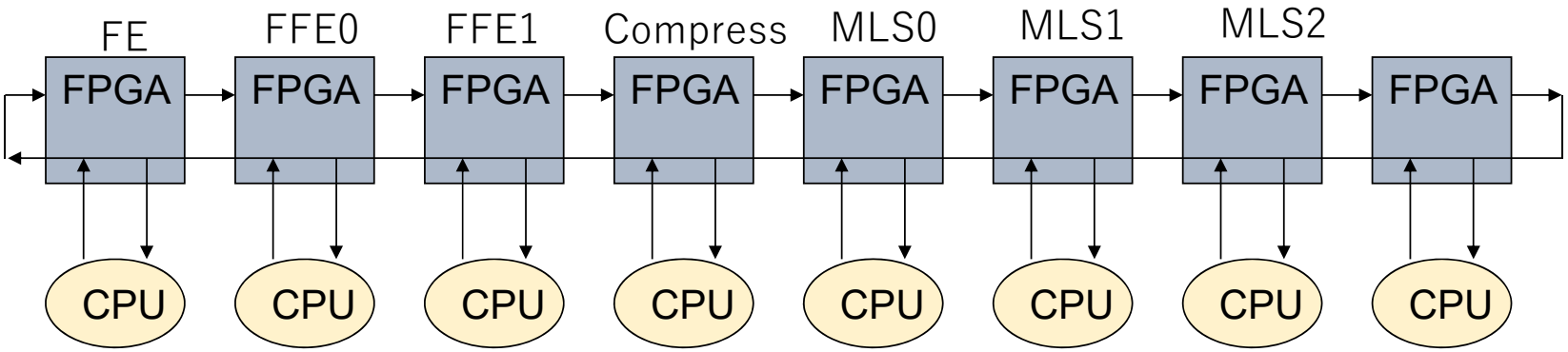
Higher performance per cost than conventional FPGA in cloud.
Practically infinite resource is used.
Separated into a number of virtual FPGAs and shared by the multiple users.

Flow-in-Cloud (FiC) is the first prototype.

Microsoft's Catapult V1/V2

[Putnum：ISCA-2014][Caulfield：Micro-2016]

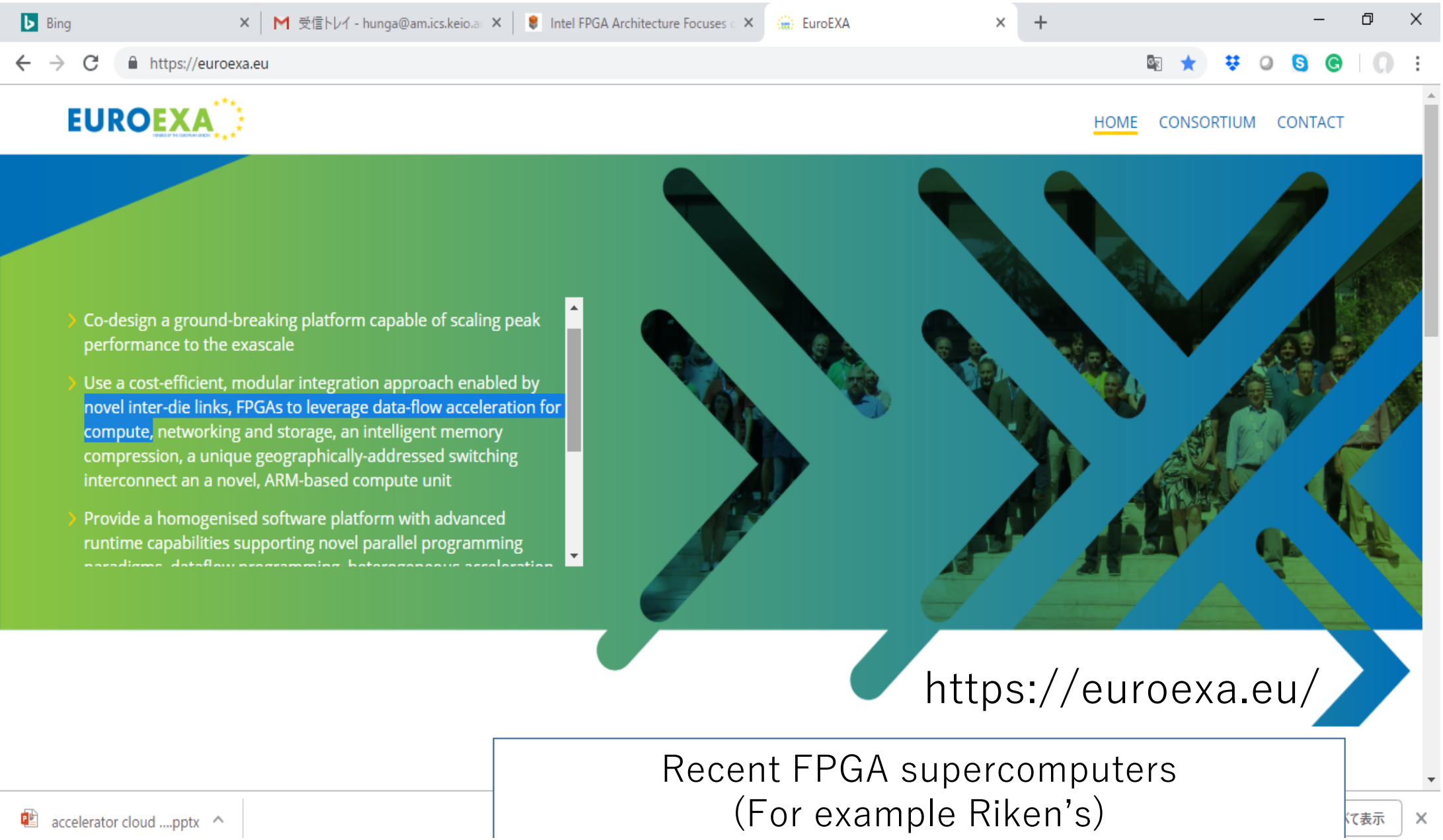
Rank computation for Web search on Bing.
Task Level Macro-Pipelining (MISD)
FE: Feature Extraction
FFE: Free Form Expression: Synthesis of feature values
MLS: Machine Learning Scoring



FPGA: Intel Stratix V

2-Dimensional Mesh is formed (8x6) for 1 cluster.

10Gbps network is upgraded to 40Gbps network in V2

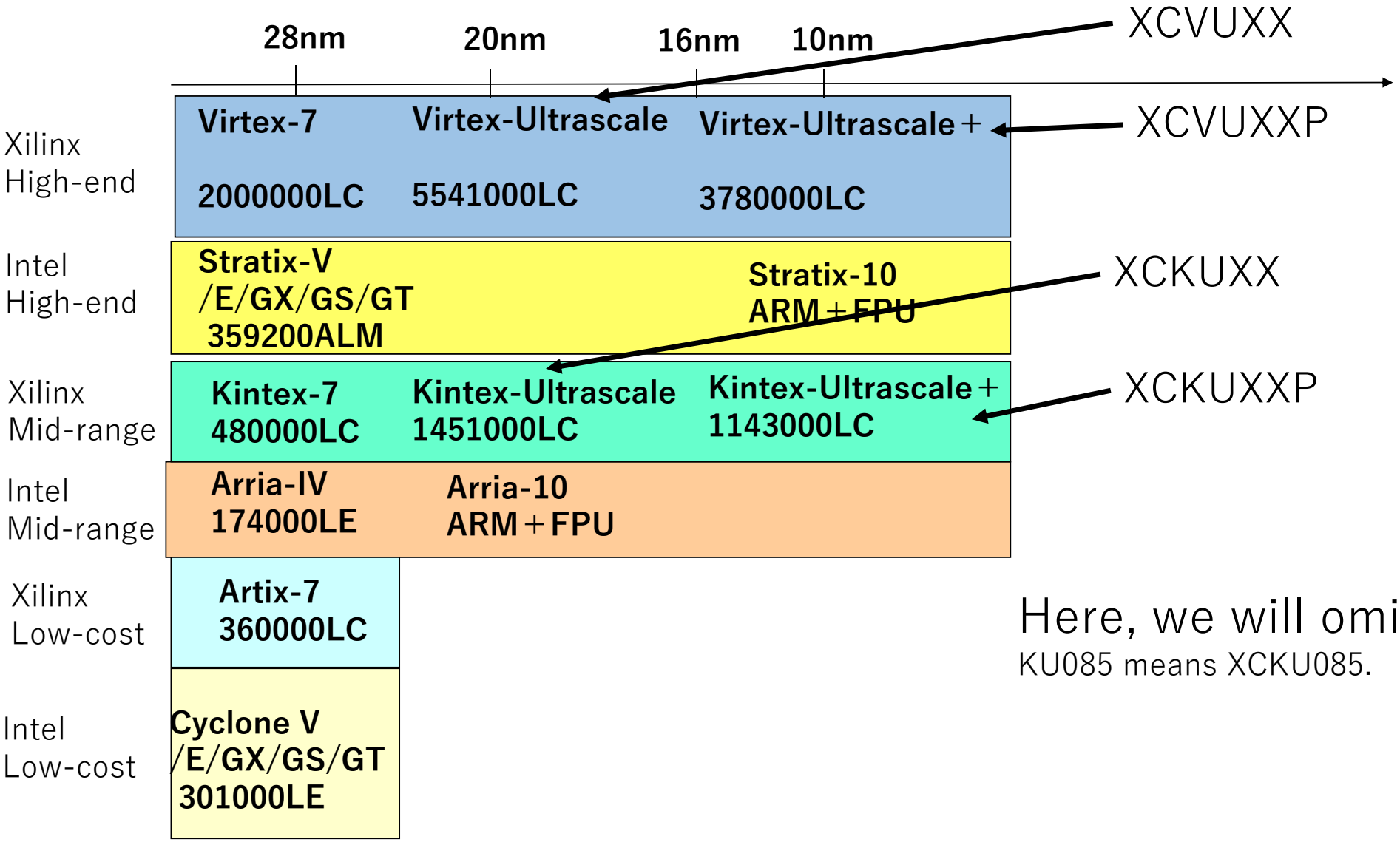


Recent FPGA supercomputers
(For example Riken's)

Today's talk

- Building a virtual large FPGA
  - Concept 1: Use middle-range FPGAs and common serial links
  - Concept 2: Virtualize at the level of HLS description
  - Concept 3: Couple accelerators and a switch tightly in an FPGA
    → Accelerator-in-Switch
- Our prototype: FiC (Flow-in-Cloud)
- Next step: Building a virtual heterogeneous computing system

High-end, Low-cost and Mid-range FPGAs



Here, we will omit “XC”.

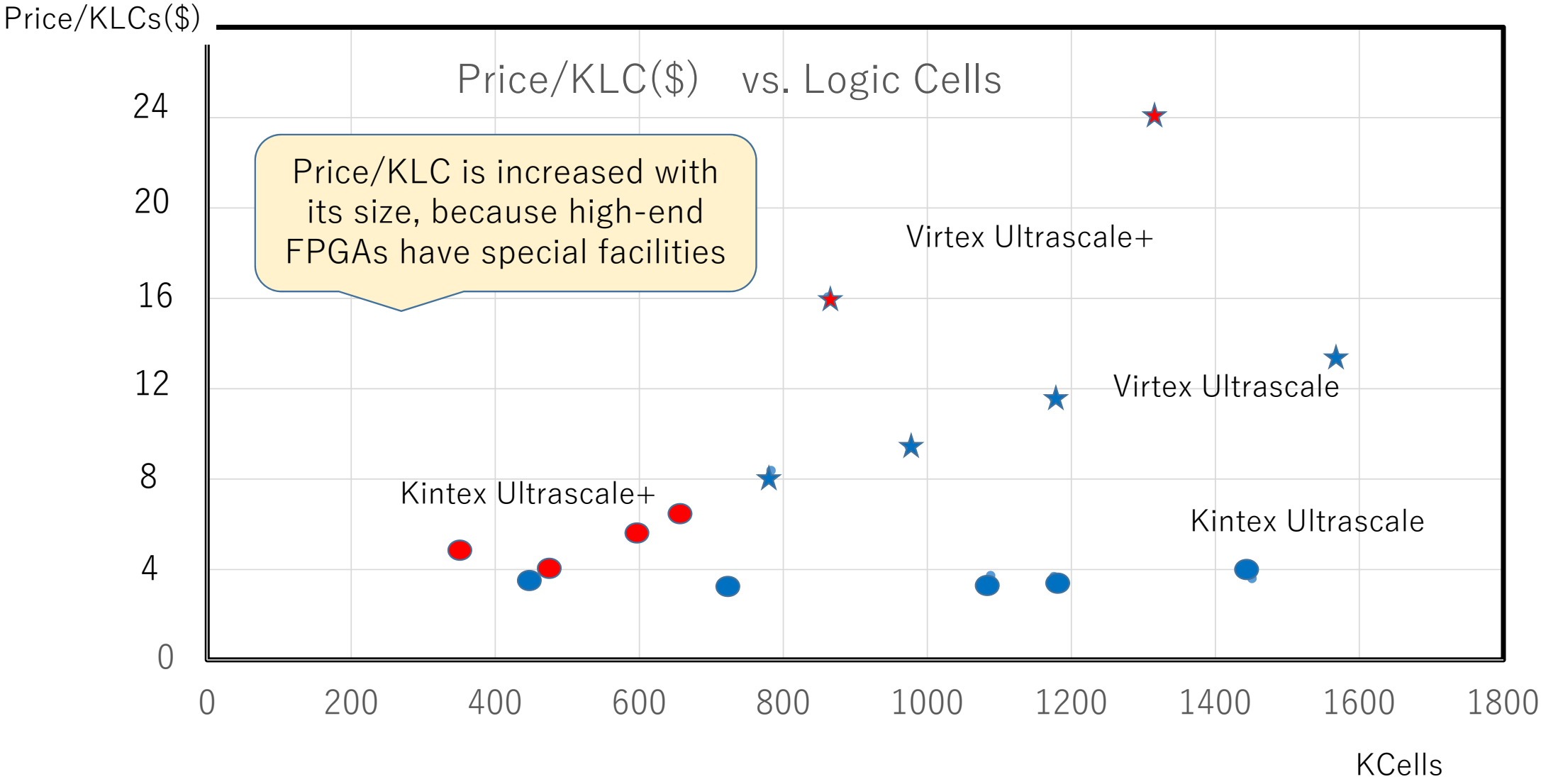
KU085 means XCKU085.

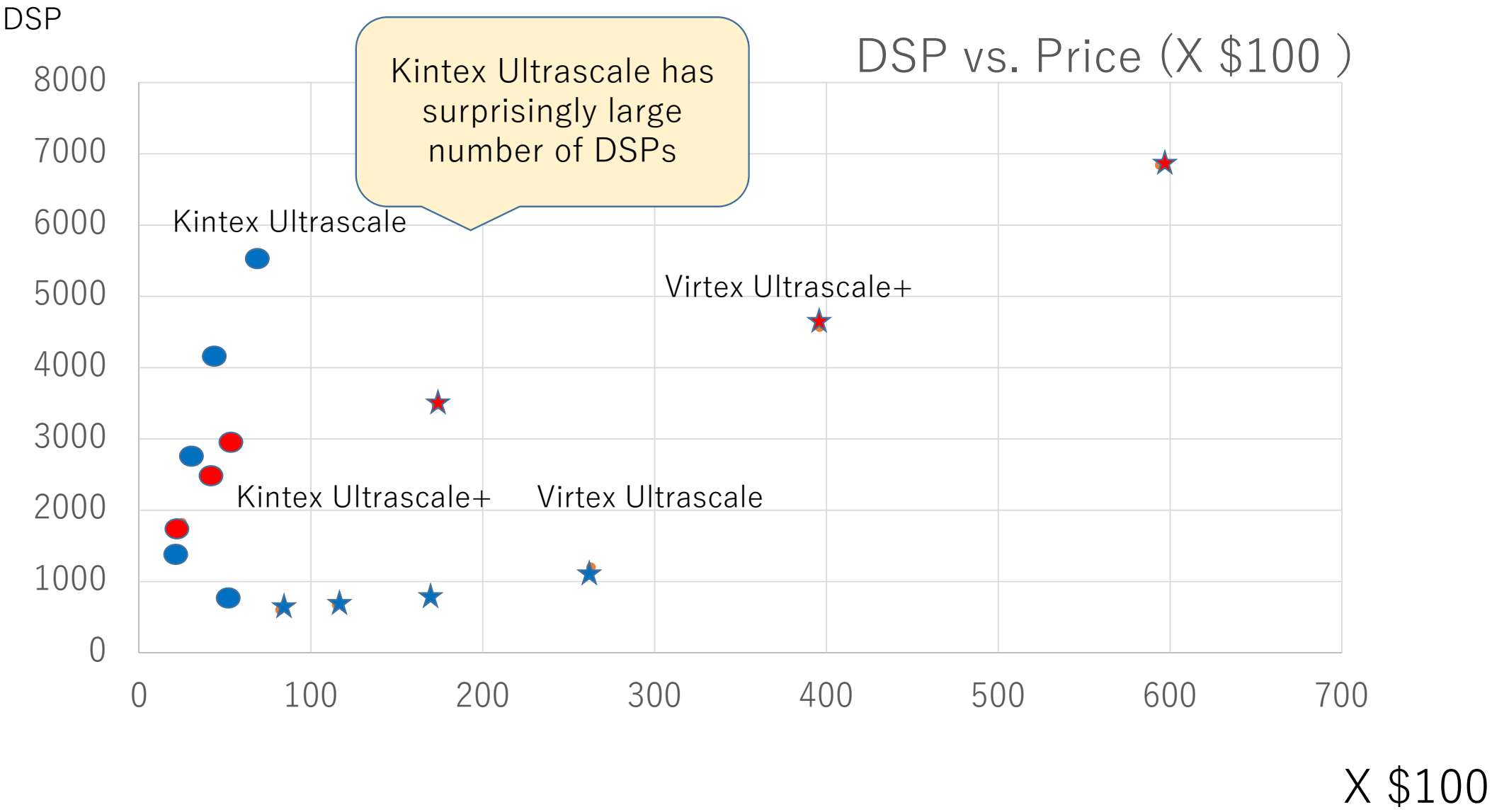
1.Multiple middle scale FPGAs vs. a single powerful FPGA

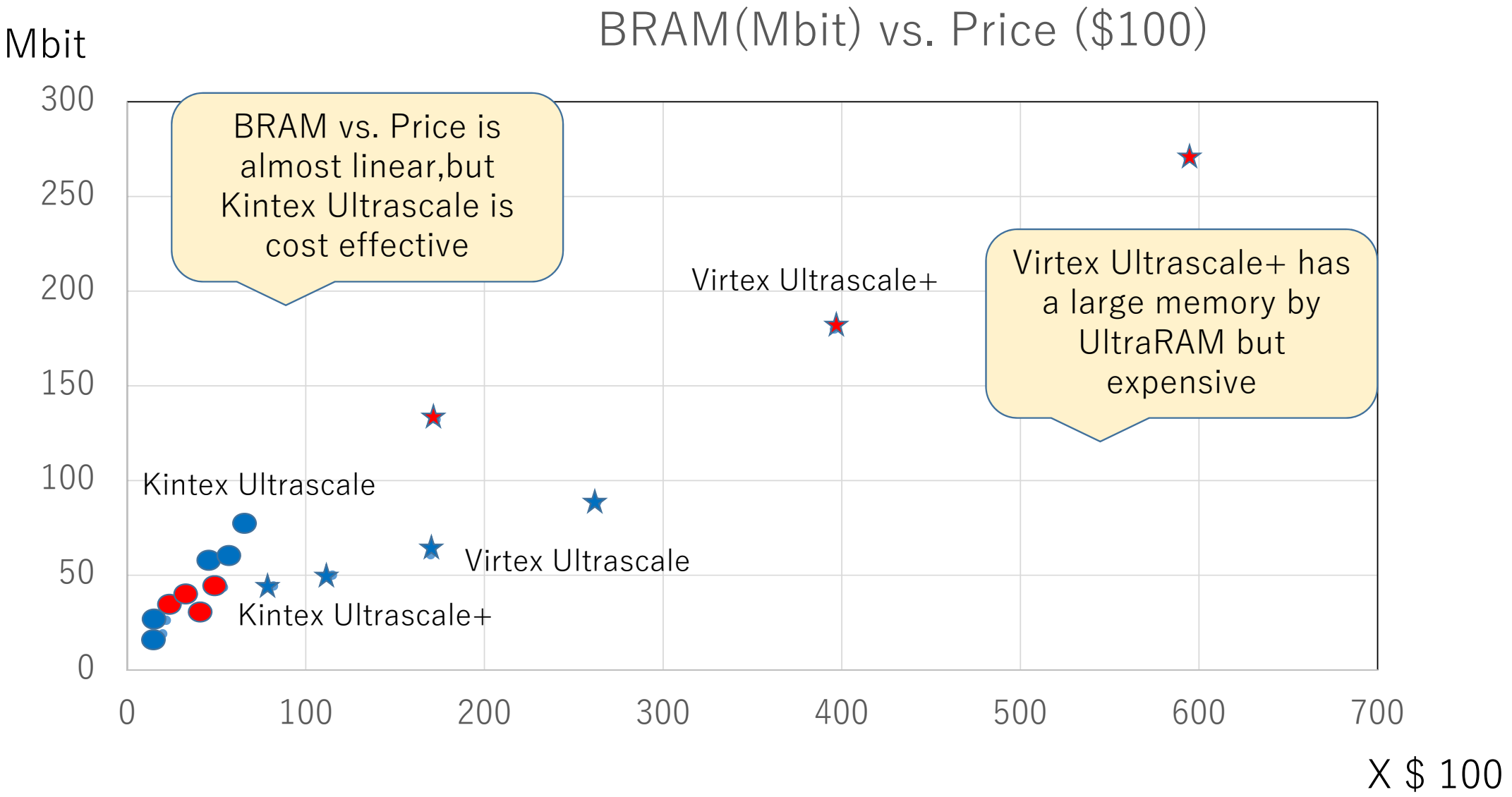
	Kintex Ultrascale KU085	Kintex Ultrascale KU115	Virtex Ultrascale VU440	Virtex Ultrascale+ VU9P
Logic Cell (K)	1088 (3.4)	1451(4.3)	5541(10.6)	2586(6.69)
DSP	4100(0.9)	5520(1.14)	2880(20.4)	6840(2.53)
BRAM(Mb)	56.9(65.4)	75.9(83.0)	88.6(664.6)	345.9(50.05)
Price ($)	3720	6297	58890	17314

Price is from digikey
( ) is price for each unit.

- Most of price/resource of KU085 is the lowest.
- Two KU085s can provide 1.5 LCs of KU115 with almost the same price.
- Five KU085s can provide almost the same LCs of VU440 with about 1/3 price.


Price/KLCs($)
Price/KLC($) vs. Logic Cells
Price/KLC is increased with its size, because high-end FPGAs have special facilities
Virtex Ultrascale+
Virtex Ultrascale
Kintex Ultrascale+
Kintex Ultrascale
24
20
16
12
8
4
0
0
200
400
600
800
1000
1200
1400
1600
1800
KCells


DSP
DSP vs. Price (X $100)
Kintex Ultrascale has surprisingly large number of DSPs
Kintex Ultrascale
Virtex Ultrascale+
Kintex Ultrascale+
Virtex Ultrascale
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
700
X $100

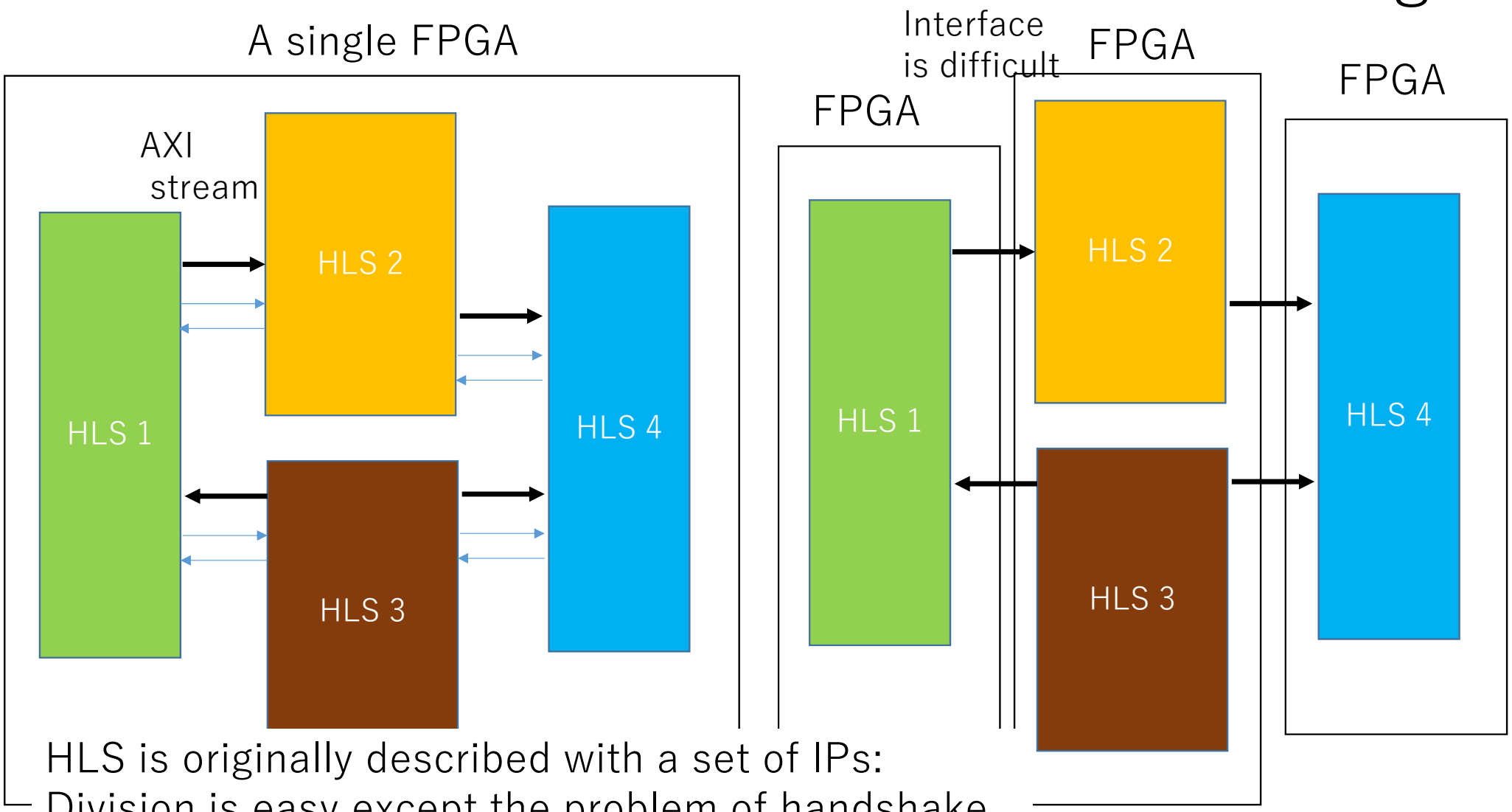

BRAM(Mbit) vs. Price ($100)
Mbit
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
X $ 100
BRAM vs. Price is almost linear,but Kintex Ultrascale is cost effective
Virtex Ultrascale+
Virtex Ultrascale+ has a large memory by UltraRAM but expensive
Kintex Ultrascale
Virtex Ultrascale
Kintex Ultrascale+

Cost including serial links

- KU085 has the best price per resource!
  - Logic cells: $3.4 for 1K LC.
  - DSP: $0.9 for a DSP.
  - BRAM: $65.4 for 1Mb.
- Let's use a number of common serial links GTH (12.5Gbps).
  - Of course, faster serial links (32Gbps GTY, 58GbpsGTM) are available, but cost becomes high.
- Firefly cable ($59 for 4 links) is available.
  - No drivers/receivers are needed.
  - Aurora IPs from Xilinx can be used.

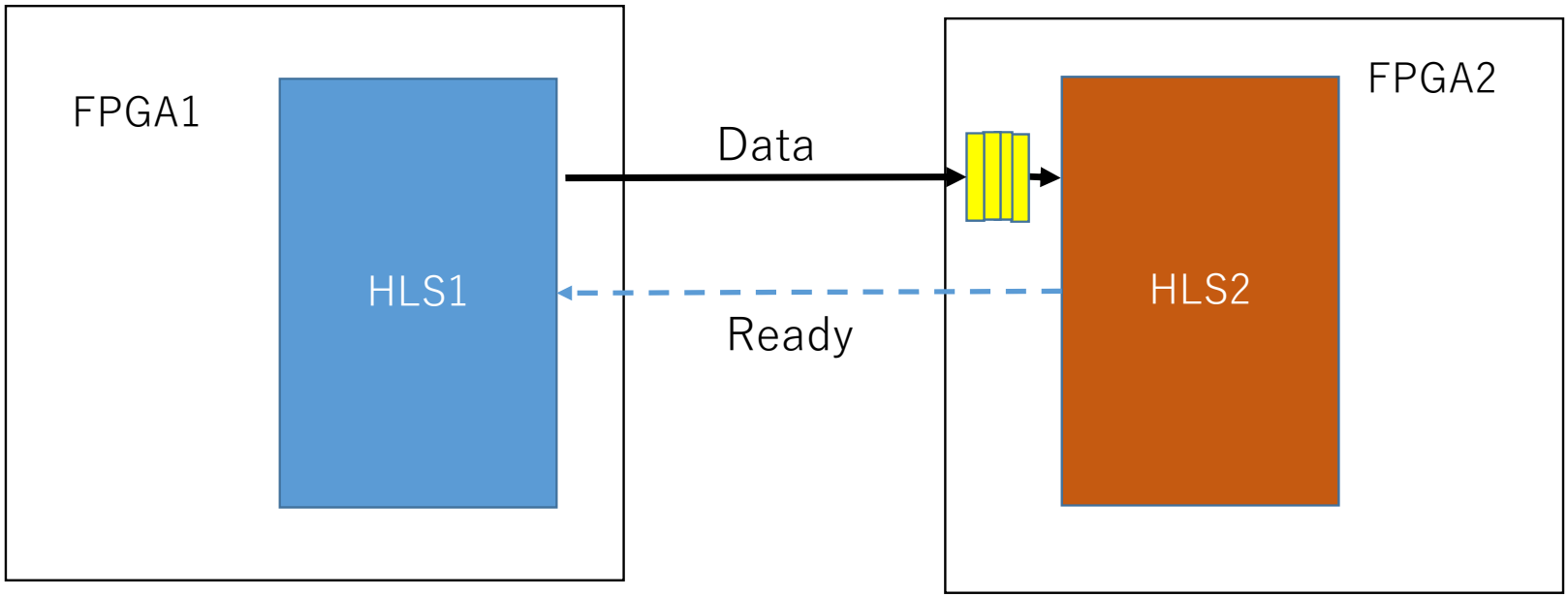
- Conclusion: "Using multiple middle scale FPGAs" is a cost efficient solution.
  - Open issue
    - Cost of switches
    - Operational Speed

2. Virtualization at the IP based HLS design



HLS is originally described with a set of IPs:
Division is easy except the problem of handshake.

The handshake problem



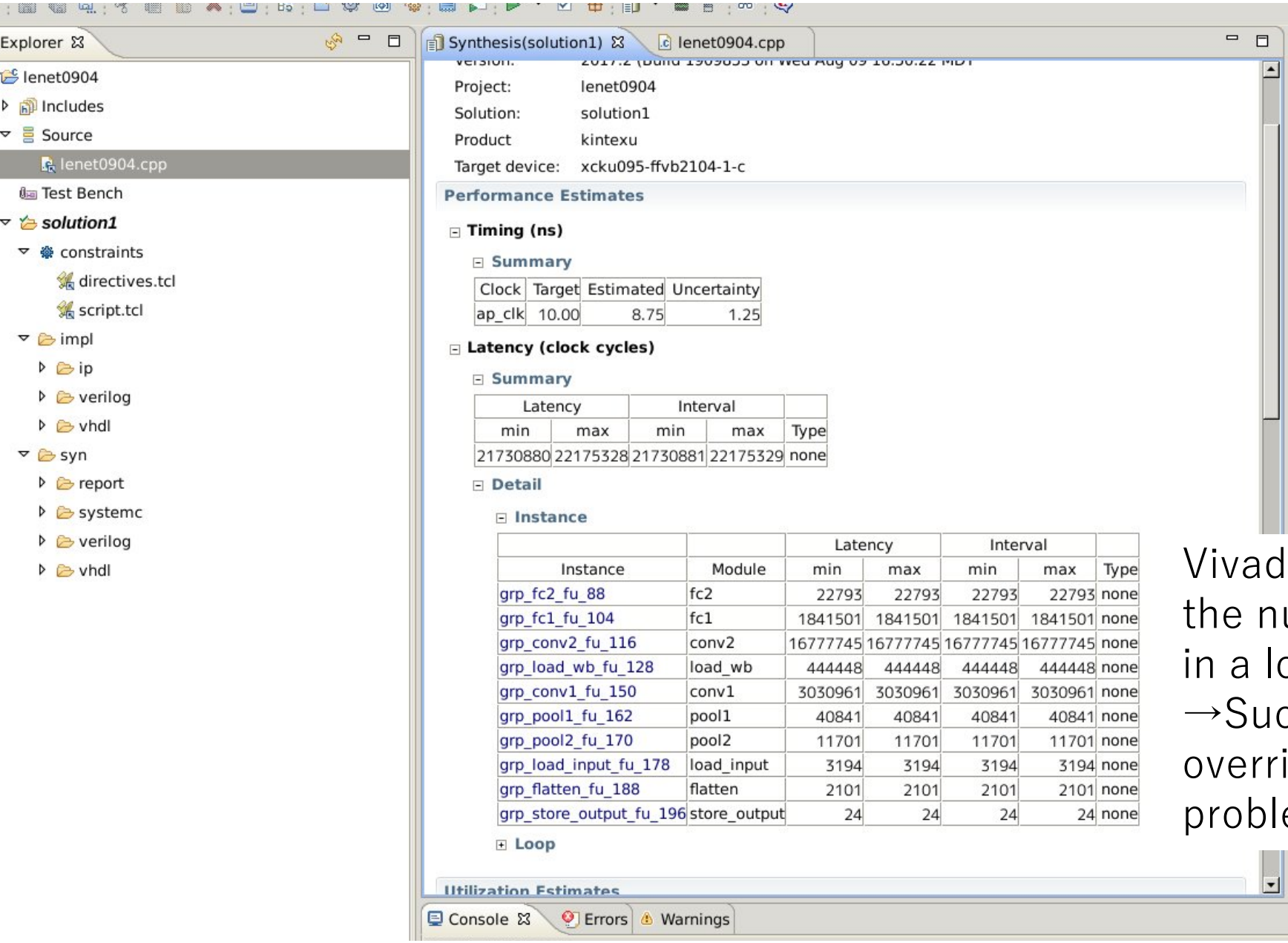
- Valid signal can be omitted by checking the data arrival.
- Without a ready signal, the possibility of input FIFO overflow remains.

Overriding the handshake problem

- Virtual ready wire: providing a virtual wire between the receiver and the sender.
  - Direct approach but the overhead of synchronization may increase.
- Providing a required memory inside HLS module.
  - Convenient for streaming processing but the HLS programmer must take care of it.
- A pre-processor can insert delay or synchronization code according to the evaluation results from Vivado HLS.

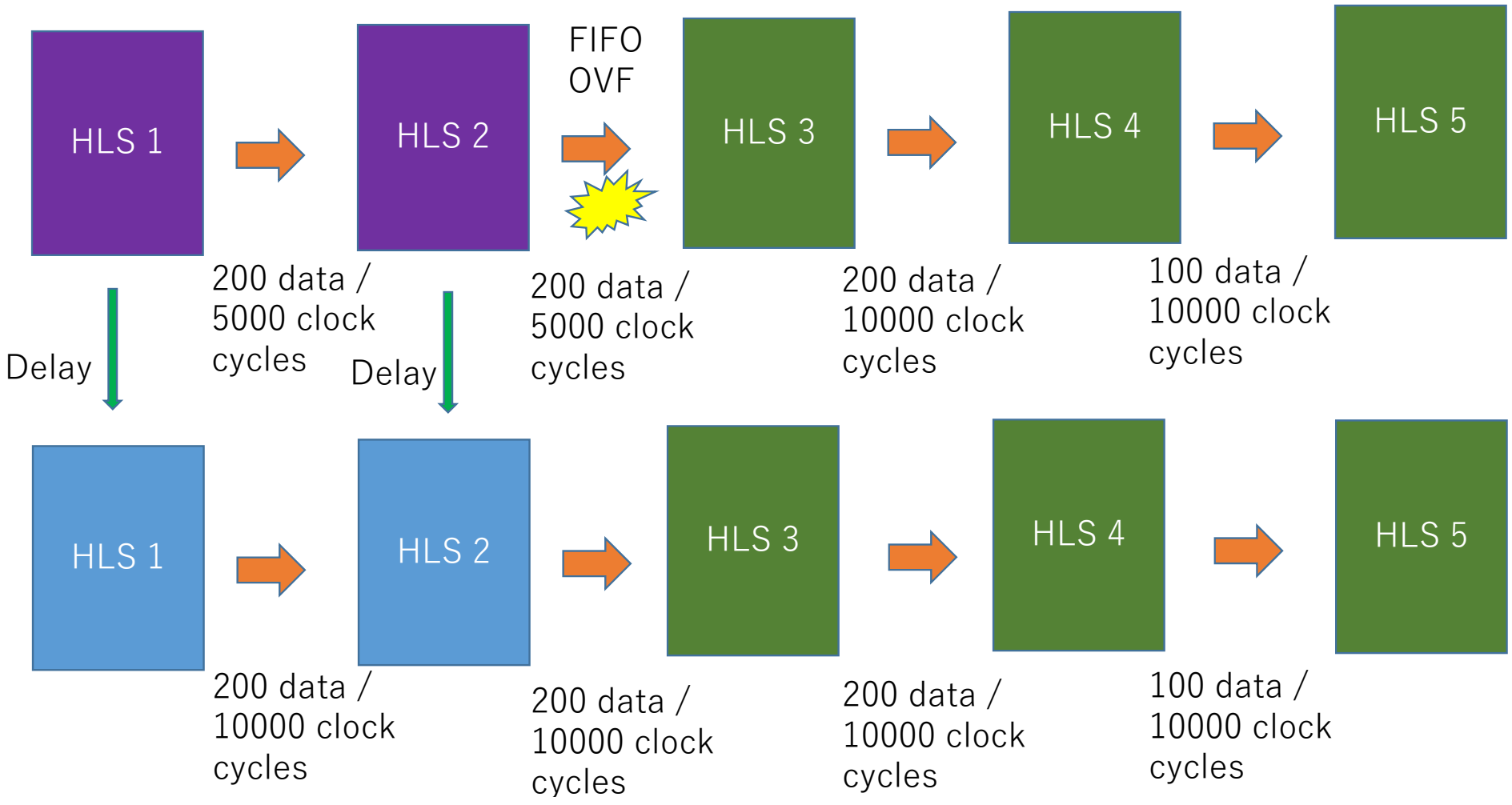
→ All methods require fixed latency/throughput communication.

Our approach: circuit switching with Static Time Division Multiplexing.



Vivado HLS evaluates the number of clock cycle in a loop.
→Such information helps override the handshake problem.

An example: Streaming processing

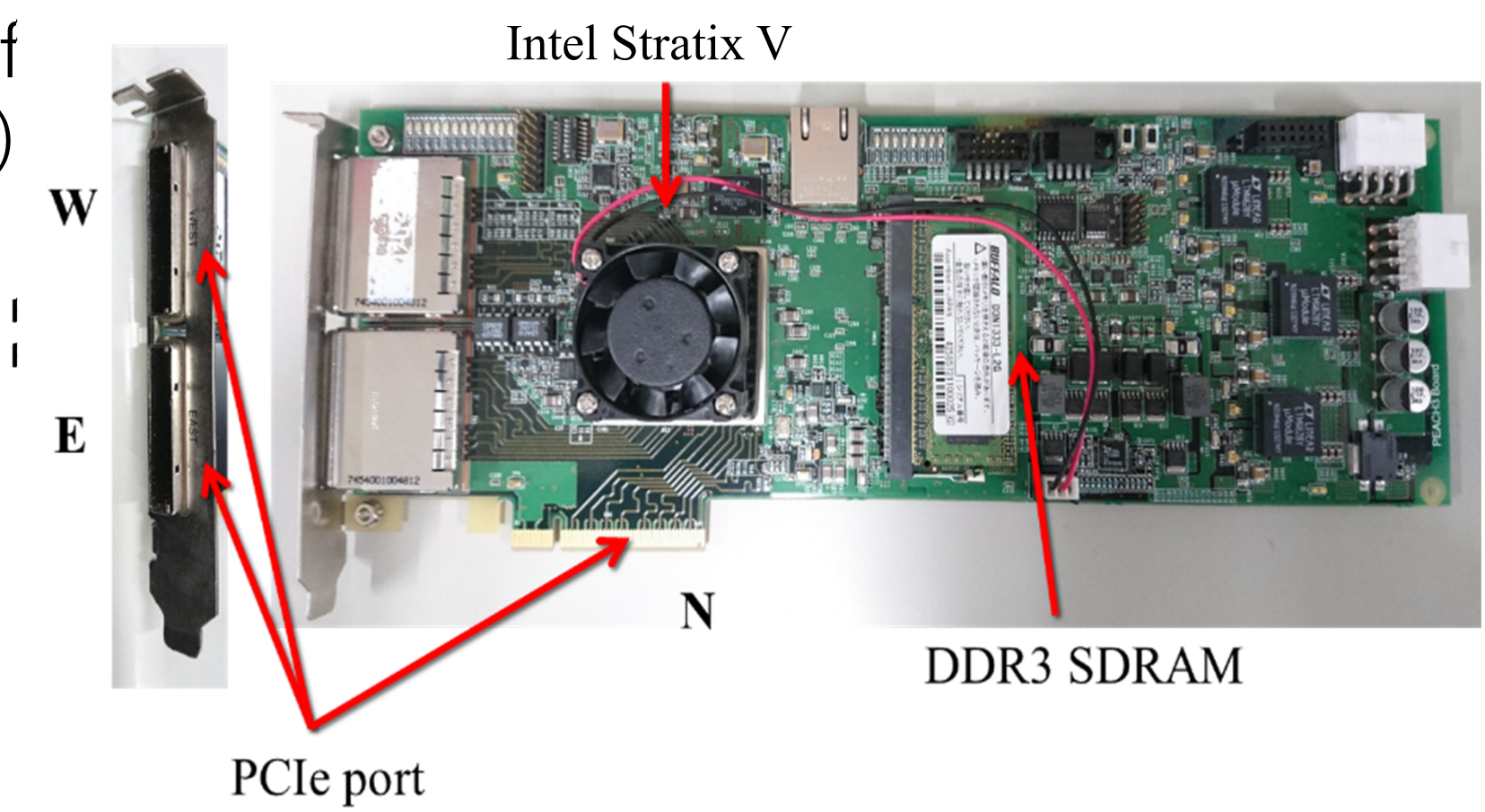


3. Integrating switch and accelerator tightly in an FPGA: Accelerators in Switch

- What is the benefit of FPGAs compared to GPUs ?
  - Switching capability is much superior to that of GPUs.
  - Of course, recent GPUs provide NVLinks or other powerful interface.
    - However, they are only for expensive GPUs, and the function is limited.
  - Various type of switches can be implemented on FPGAs.
  - FPGAs are widely used for high speed switches and network interface.
- Tightly coupled switch and accelerator in an FPGA.
- Separation with Partial Reconfiguration

→ Accelerator in Switch [FPL2017]

An example of AiS (PEACH3)



- Implemented as a module on the Avalon MM bus.
- Shared memory is used for exchange of data

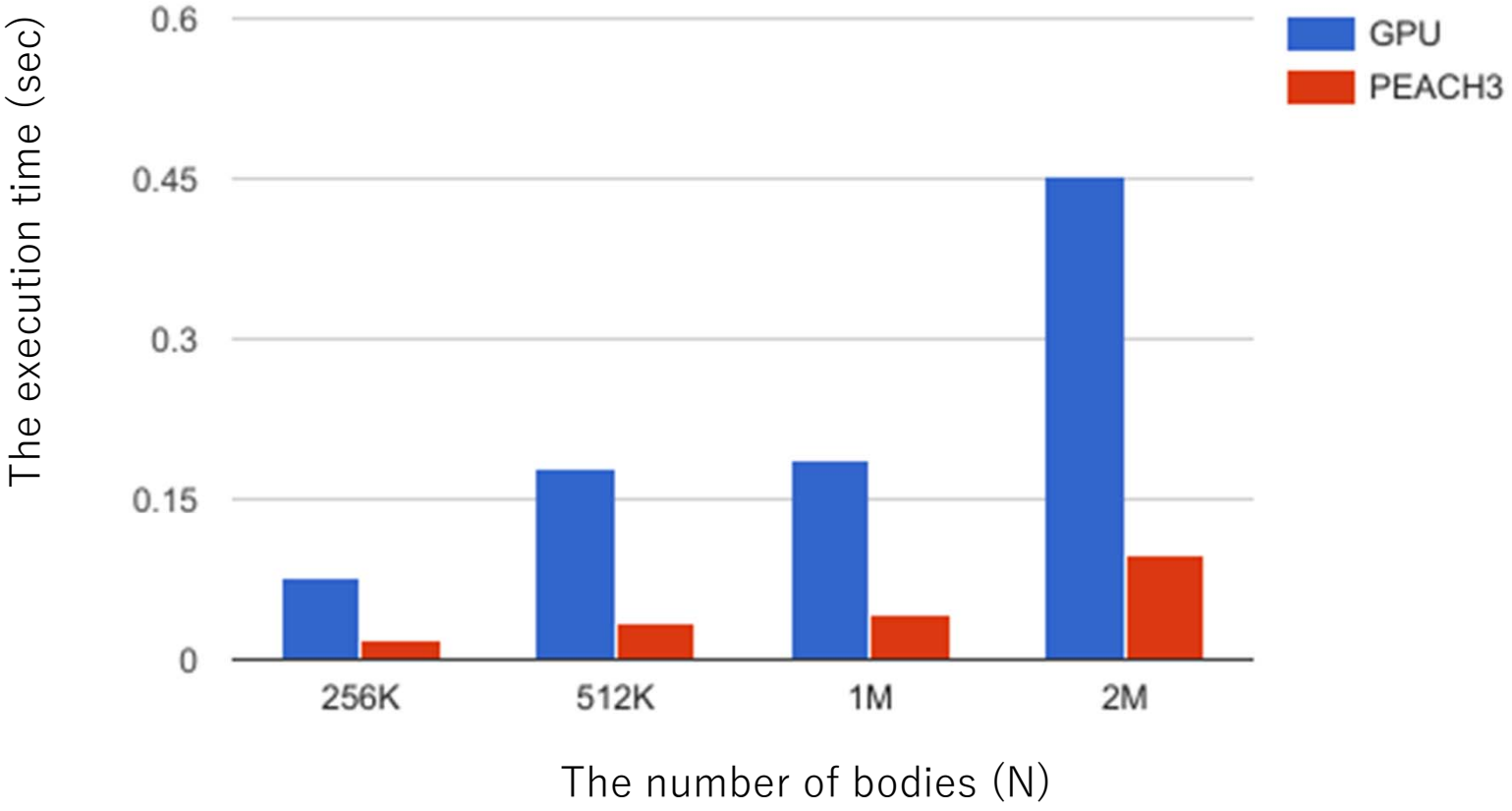
ST

PCIe gen3 x8

FIFO FIFO

- Reduction / Locally Essential Tree generation were implemented in the AiS part.

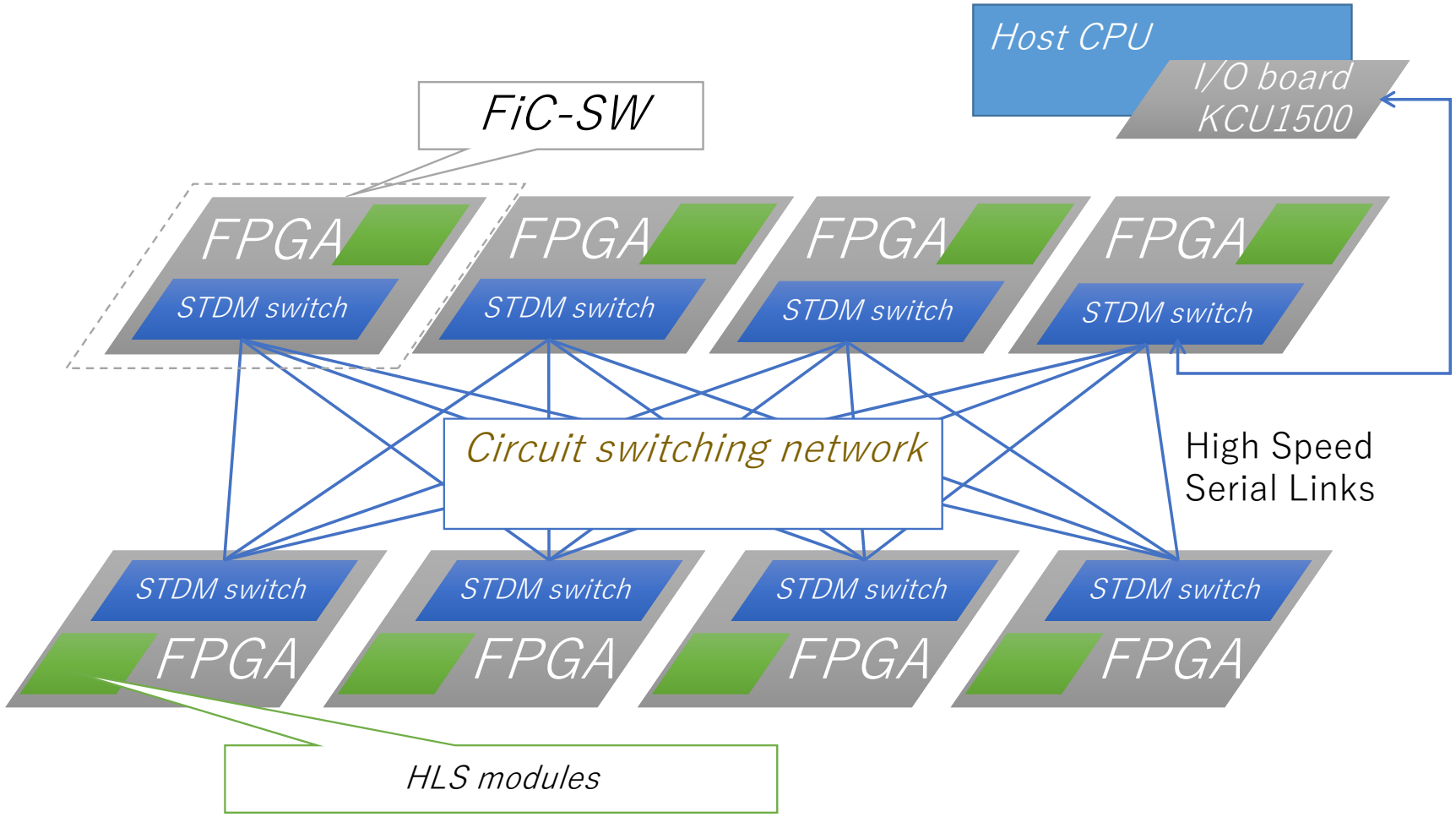
The execution time for LET generation



Today's talk

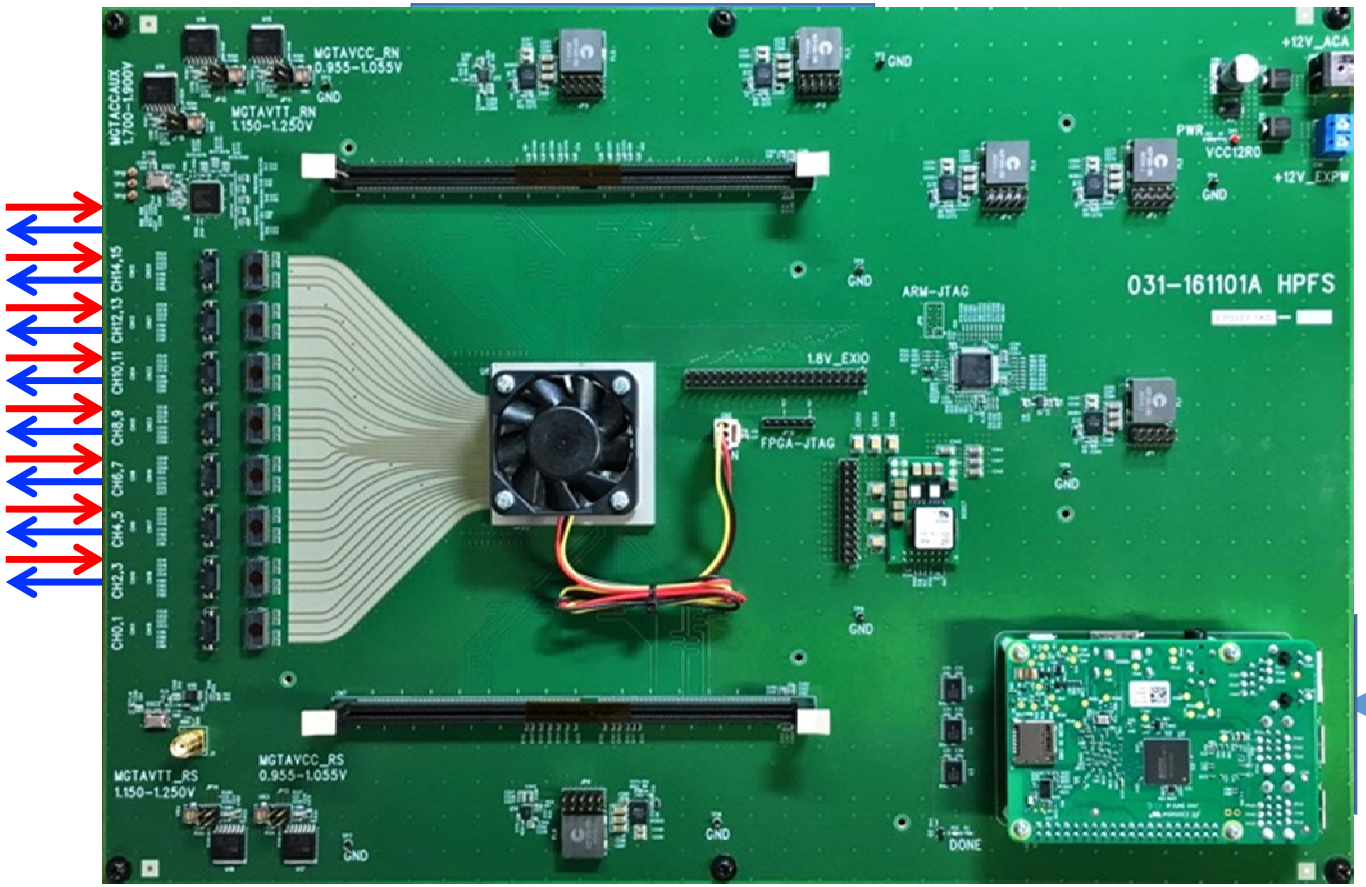
- Building a virtual large FPGA
  - Concept 1: Use middle-range FPGAs and common serial links
  - Concept 2: Virtualize at the level of HLS description
  - Concept 3: Couple accelerators and a switch tightly in an FPGA
    → Accelerator-in-Switch
- Our prototype: FiC (Flow-in-Cloud)
- Next step: Building a virtual heterogeneous computing system

Flow-in-Cloud (FiC) overview



Flow-in-Cloud (FiC) SW Board

FiC Network
8x4 9.9Gbps

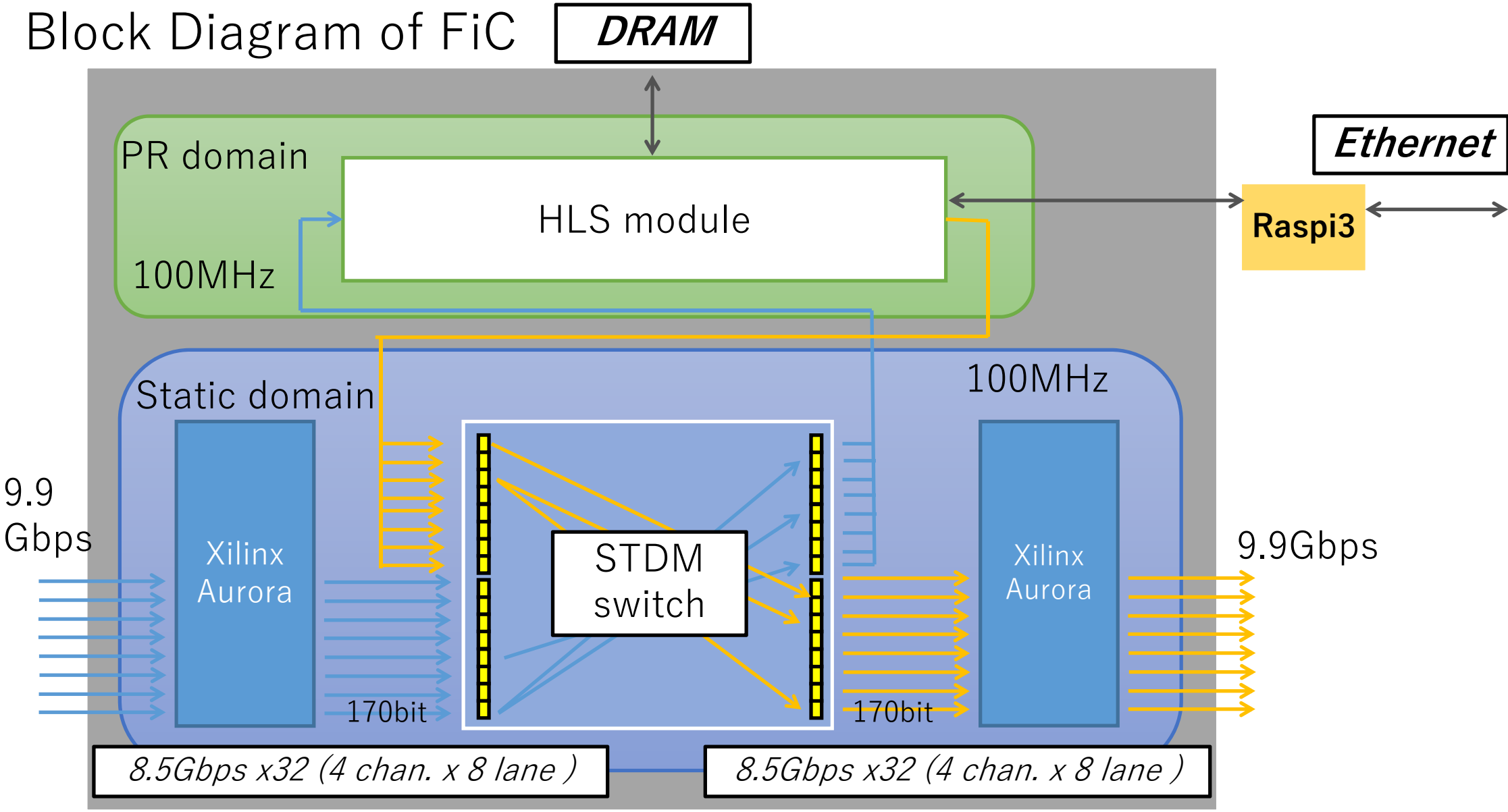


Here, we call each link “channel”, and a bundle of 4 channels “lane”.

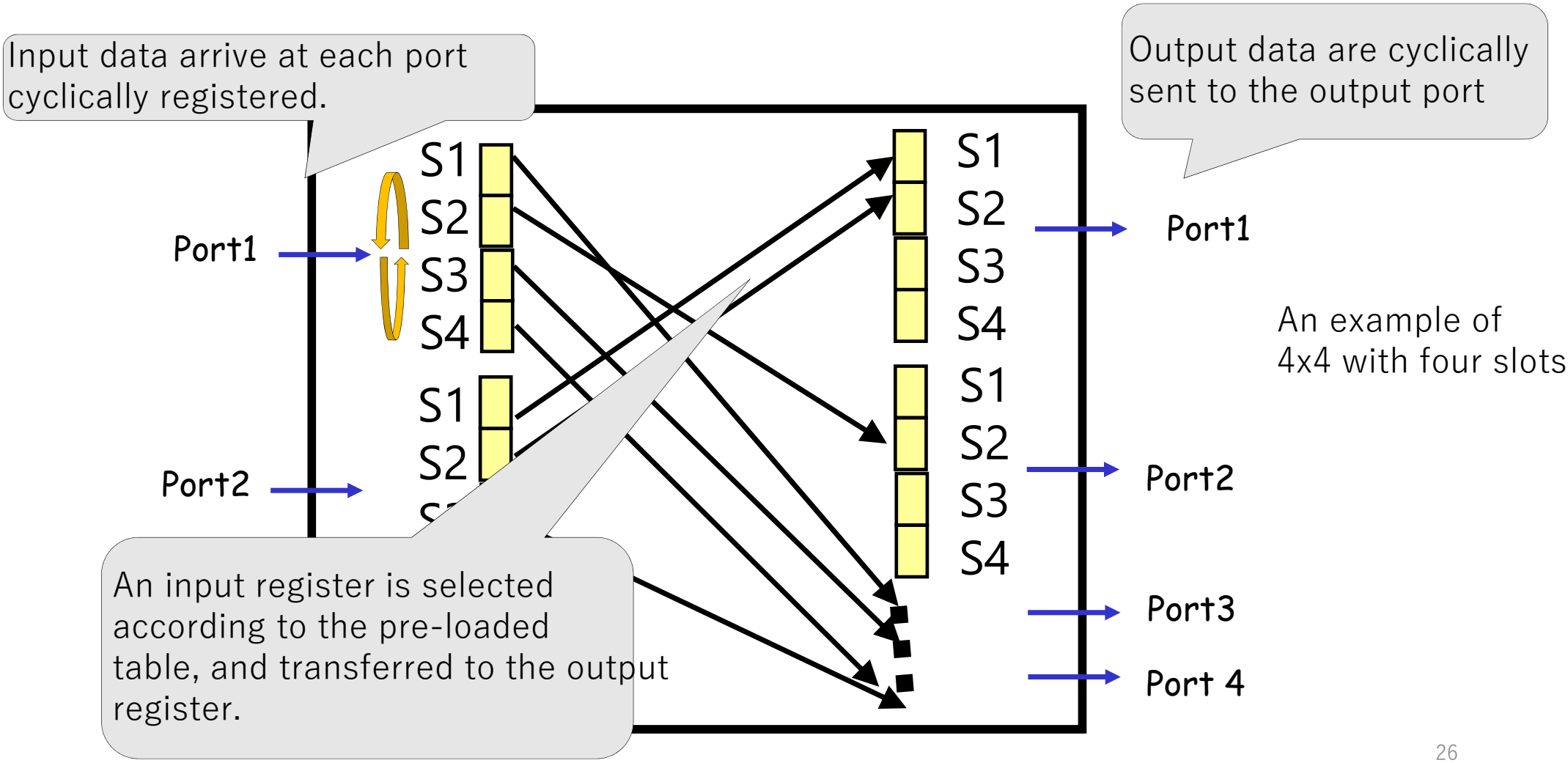
A board has 8 lanes each of which has 4 channels

Ethernet

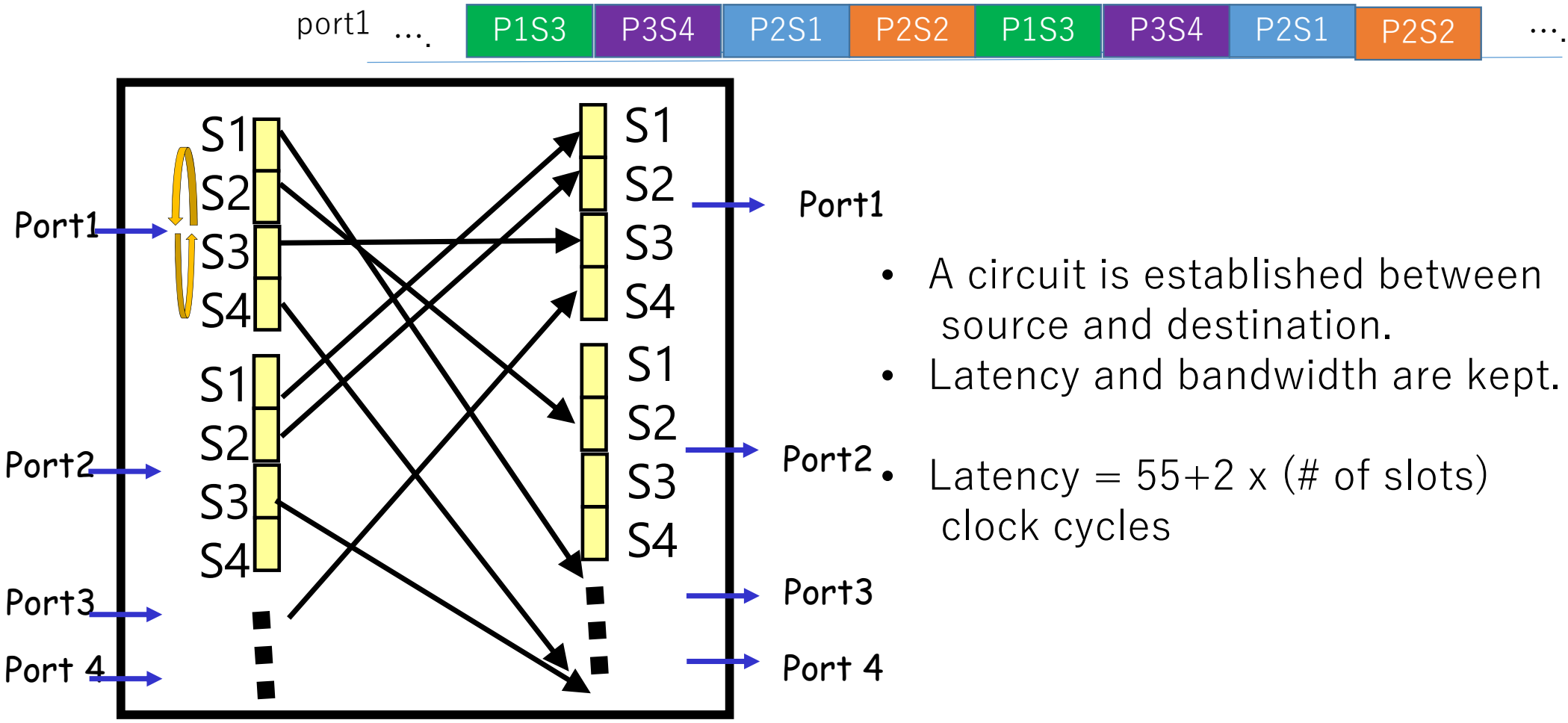
Control Network


Block Diagram of FiC
DRAM
PR domain
HLS module
100MHz
Ethernet
Raspi3
Static domain
100MHz
9.9
Gbps
Xilinx
Aurora
STDM
switch
Xilinx
Aurora
9.9Gbps
170bit
170bit
8.5Gbps x32 (4 chan. x 8 lane)
8.5Gbps x32 (4 chan. x 8 lane)

STDM (Static Time Division Multiplexing)

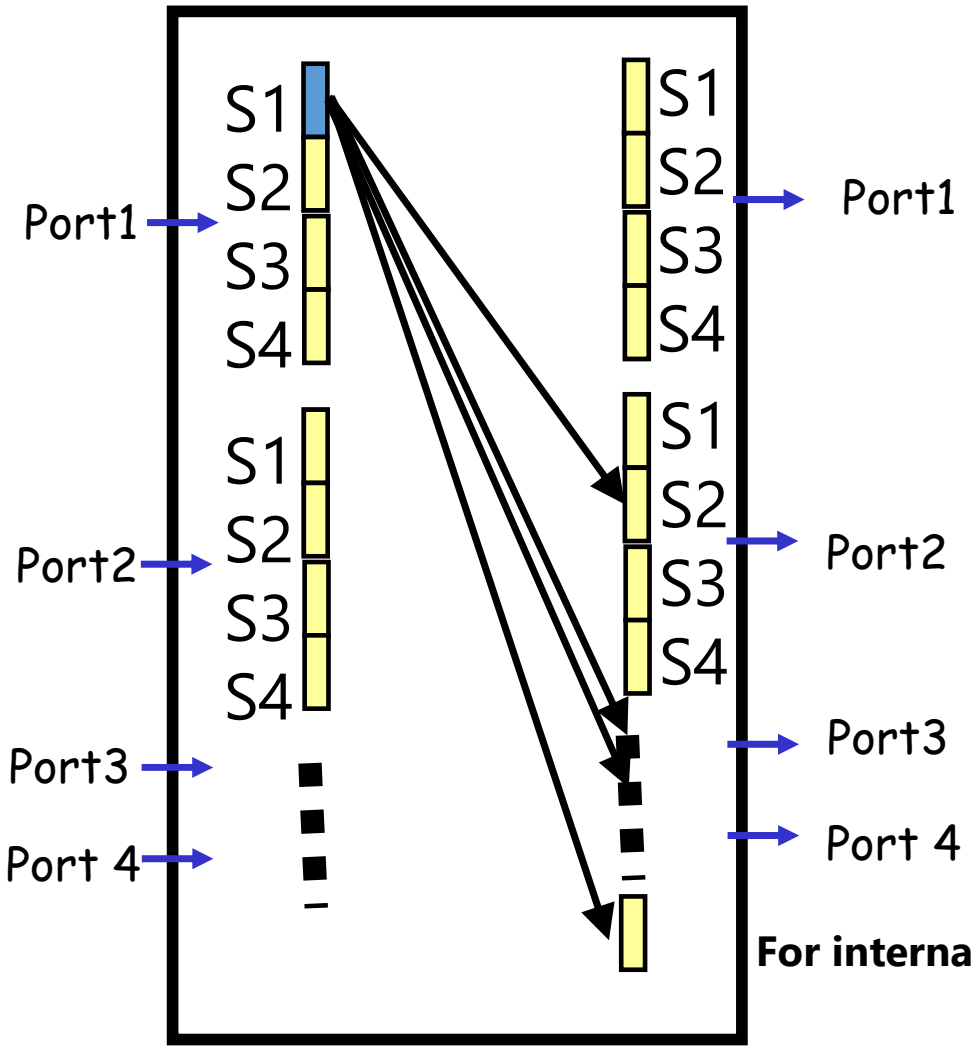


STDM (Static Time Division Multiplexing)



- A circuit is established between source and destination.
- Latency and bandwidth are kept.
- Latency = 55+2 x (# of slots) clock cycles

Multicast using the STDM

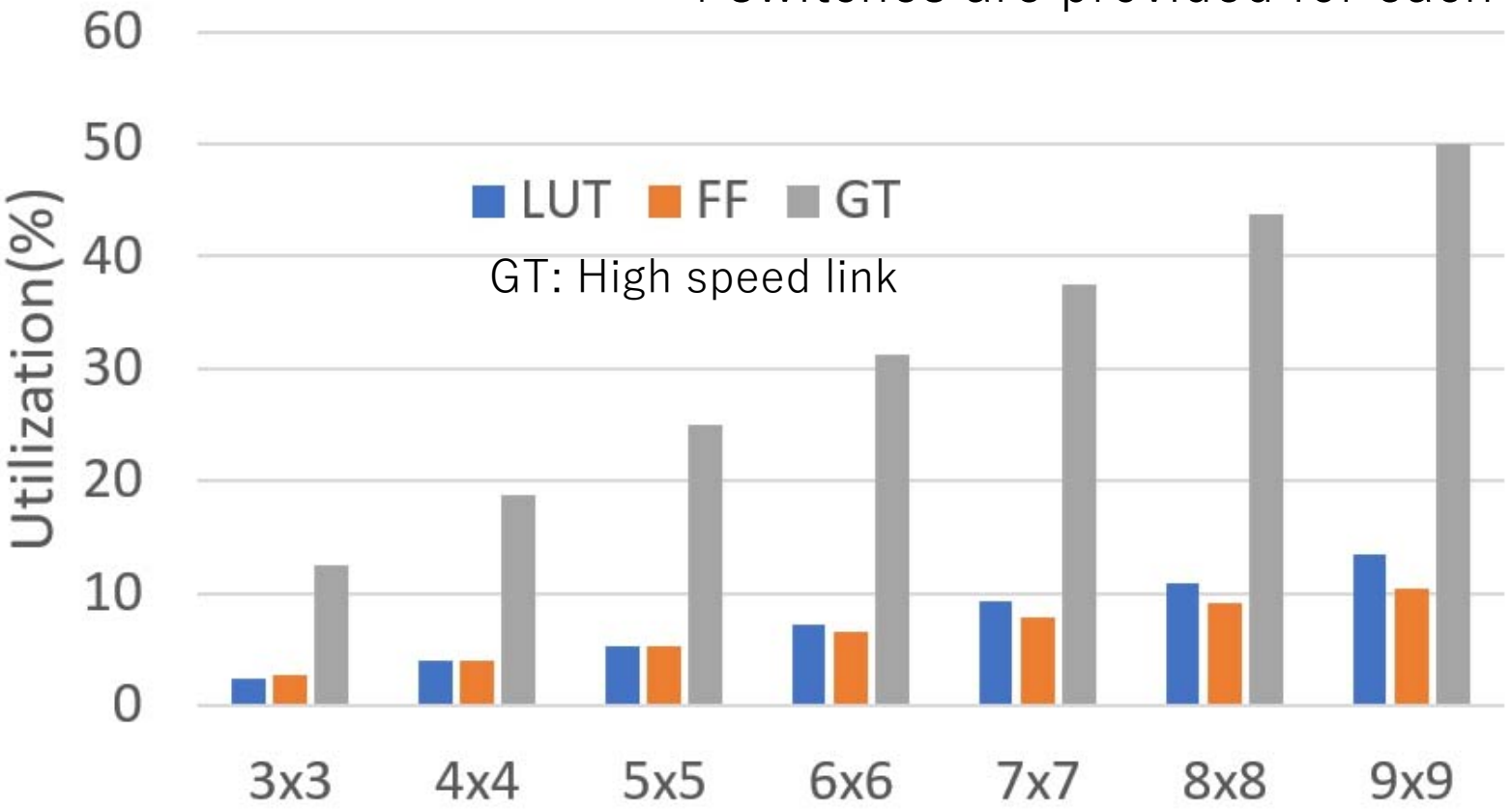


Multicast is done efficiently.

Multiple outputs can receive the same data in a specific slot.

The resource usage

4 switches are provided for each channel.



Enough design is remained for HLS design.

How boards should be connected?

- Any type of interconnection is OK.
- However, there are two limitations:
  - 4 channels are bundled into a lane.
  - For HLS modules, the size should be less than four 9x9 switches.
- 4 channels in a lane are used independently.
  - An HLS module has four independent ports, or four HLS modules with a port are implemented at maximum.

Network with 8-degree

→　Natural solution: 4 dimensional torus

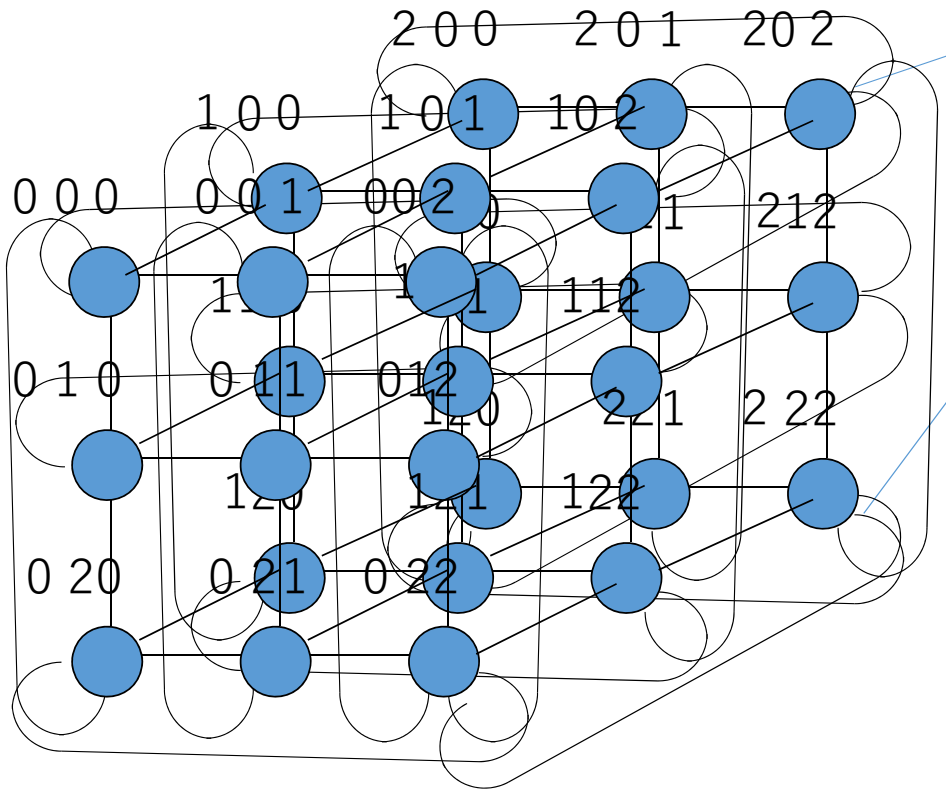
The diameter is large.

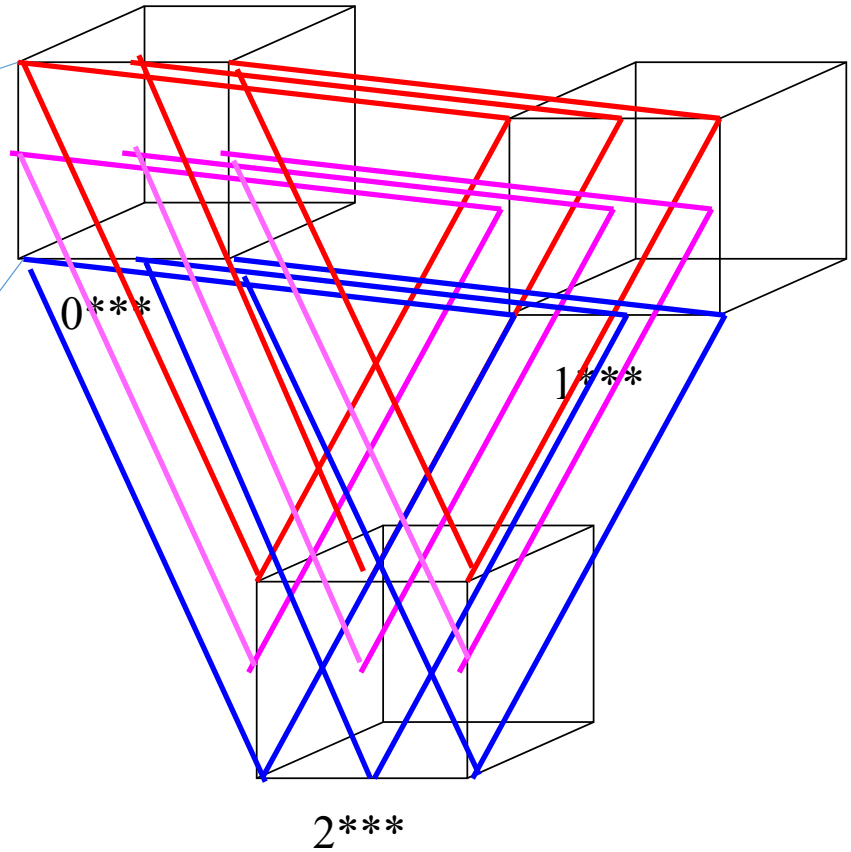
→　Alternative: Full mesh Connected Cycles（FCC）

Dragonfly-like network but more economical.

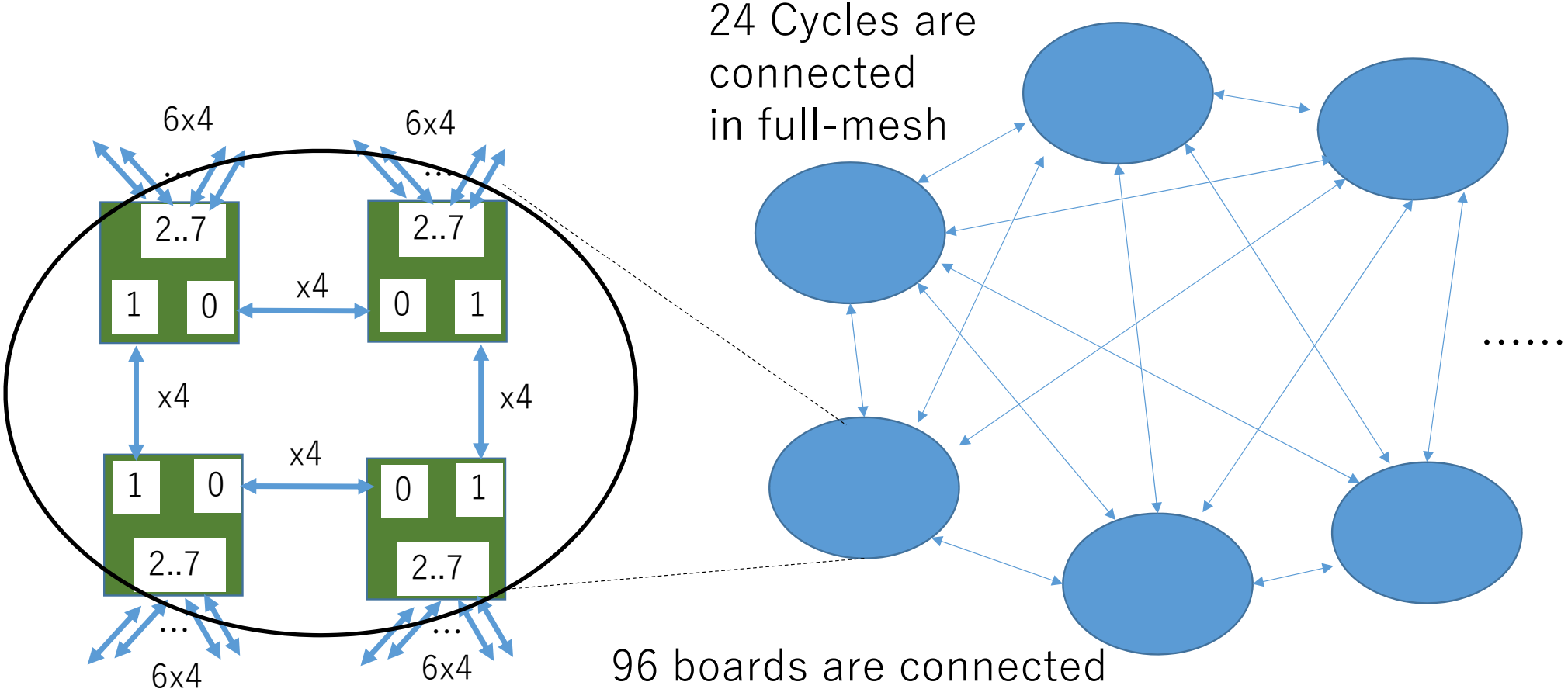
4-DTorus: the case of 3x3x3x3=81boards

Suitable if local traffic is dominant.
Diameter is relatively large: 8





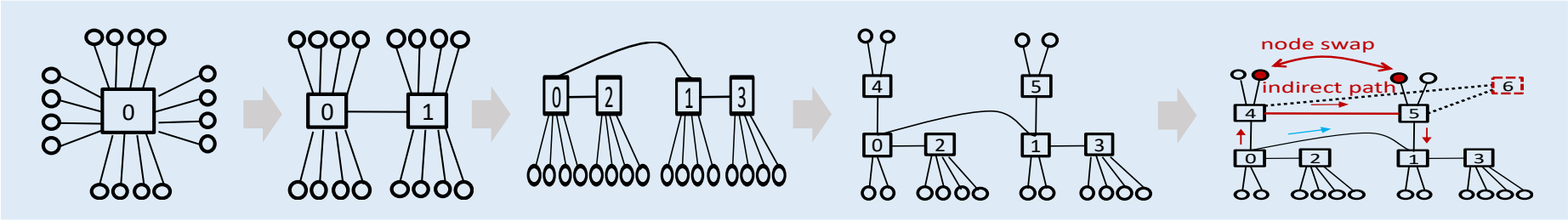
4x24 Full mesh Connected Cycles (FCC)

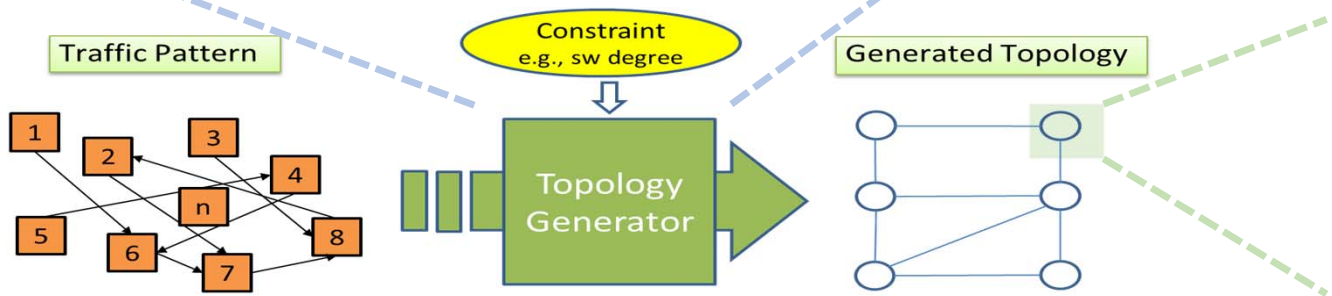


Estimation of the network performance

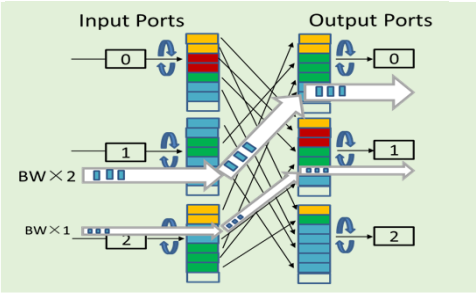
	3x3x3x3 Torus	4x24 FCC
Number of boards	81	94
Slots (Bit complement/tornado/reversal)	1	1
Slots (All to all)	12	8
Diameter	8	5
Max Latency (All to all nsec)	5360	3550

Topology Optimization for Traffic Pattern*





Circuit Switching



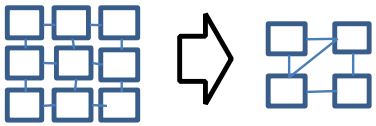
Generated topo. with # of slots

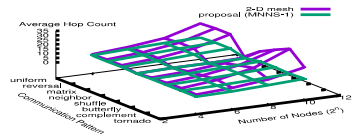
# of slots = 2	# of slots = 4	# of slots = 9	# of slots = 12
# of switches = 12	# of switches = 44	# of switches = 118	# of switches = 606
# of links = 18	# of links = 78	# of links = 184	# of links = 1063
Avg. SW hops = 1.313	Avg. SW hops = 2.563	Avg. SW hops = 4.543	Avg. SW hops = 5.329

Result: comparison with mesh

# of switches *reduced by up to* ***56.3%***

Avg. hop count *Reduced by up to* ***83.7%***

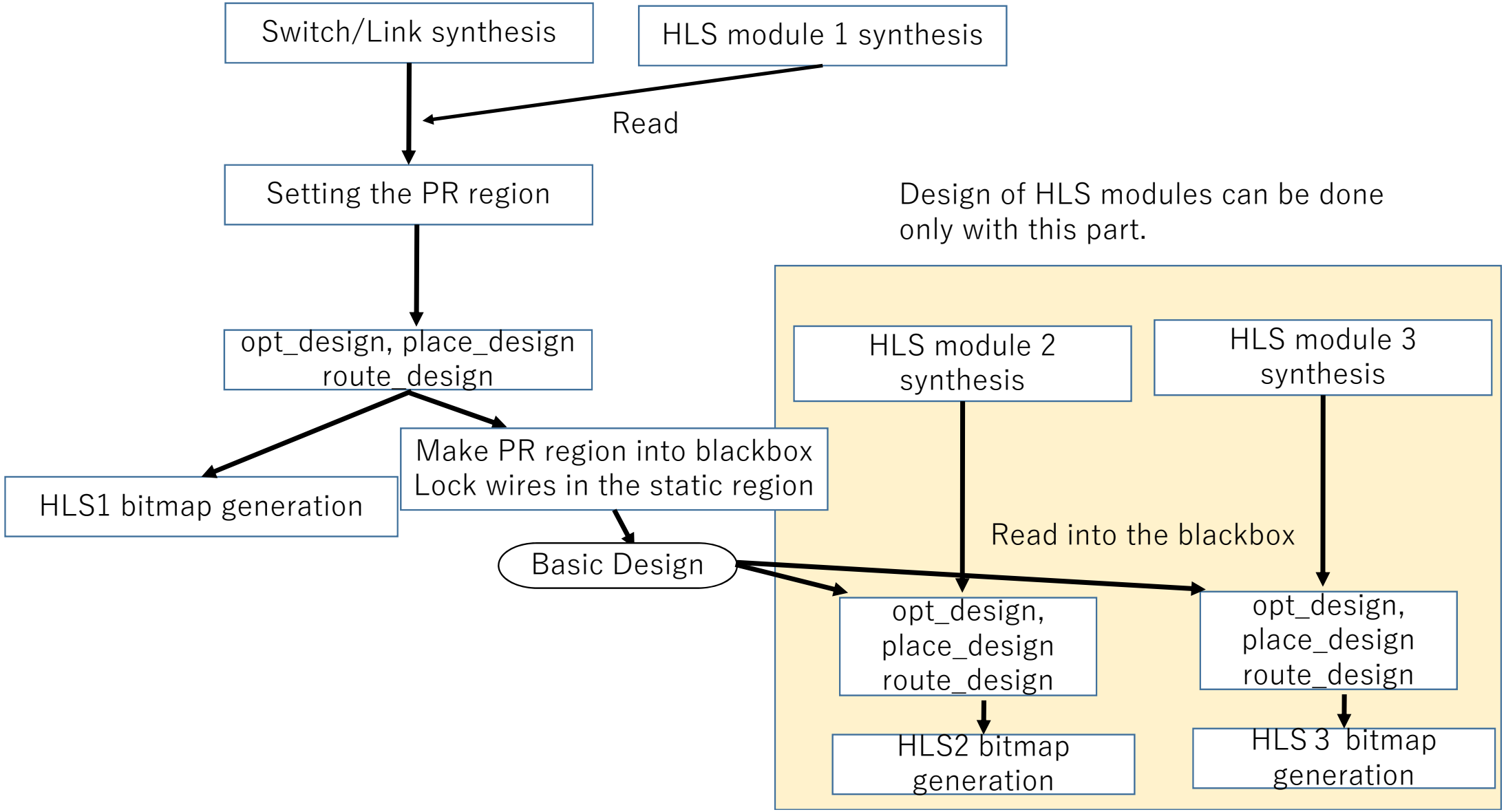


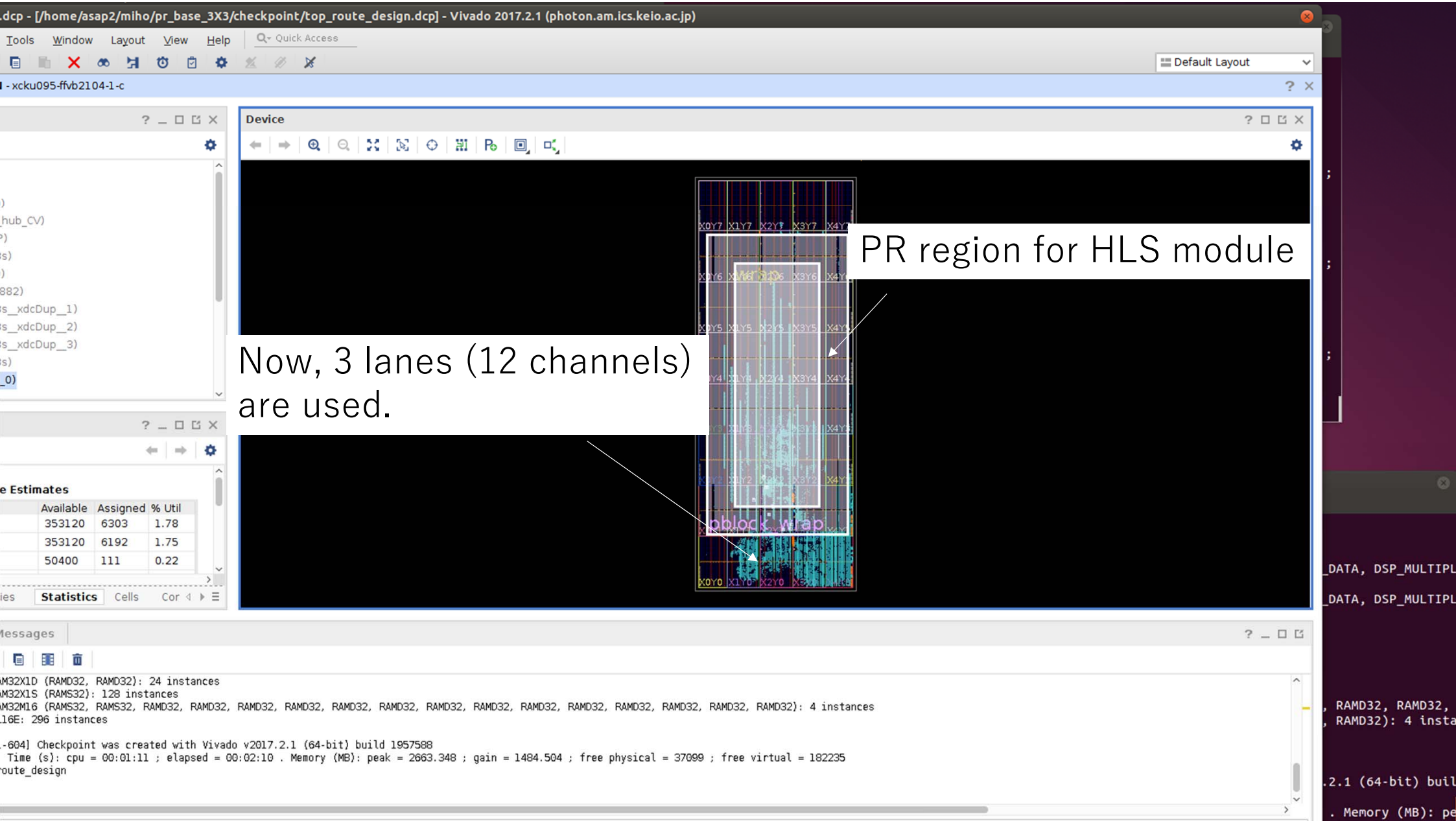


This slide is supported from Dr.Hu and Prof.Koibuchi

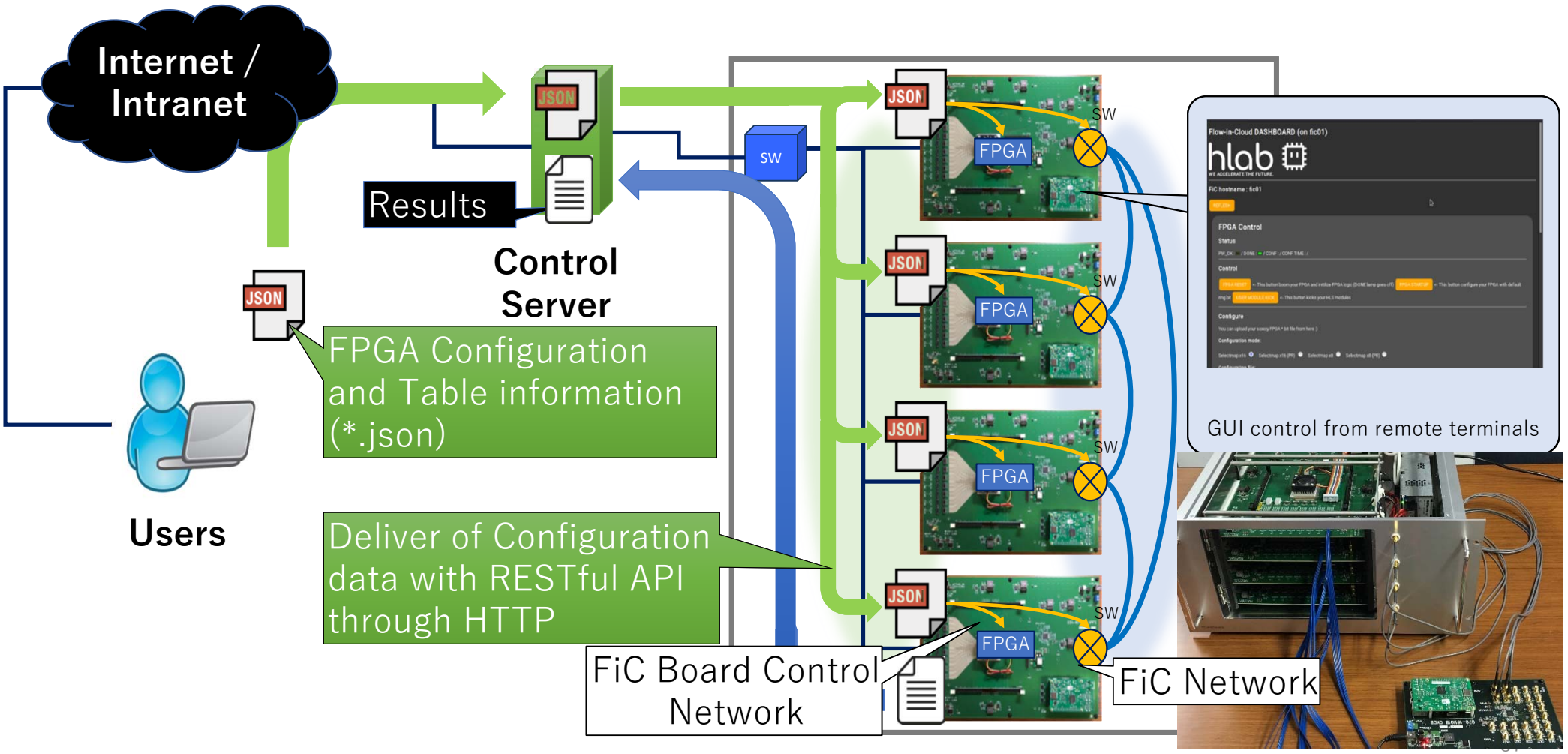
[*] Yao Hu, Tomohiro Kudoh, Michihiro Koibuchi, "A Case of Electrical Circuit Switched Interconnection Network for Parallel Computers", The 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT' 17), pp. 276-283, Taipei Taiwan, December 18-20, 2017.

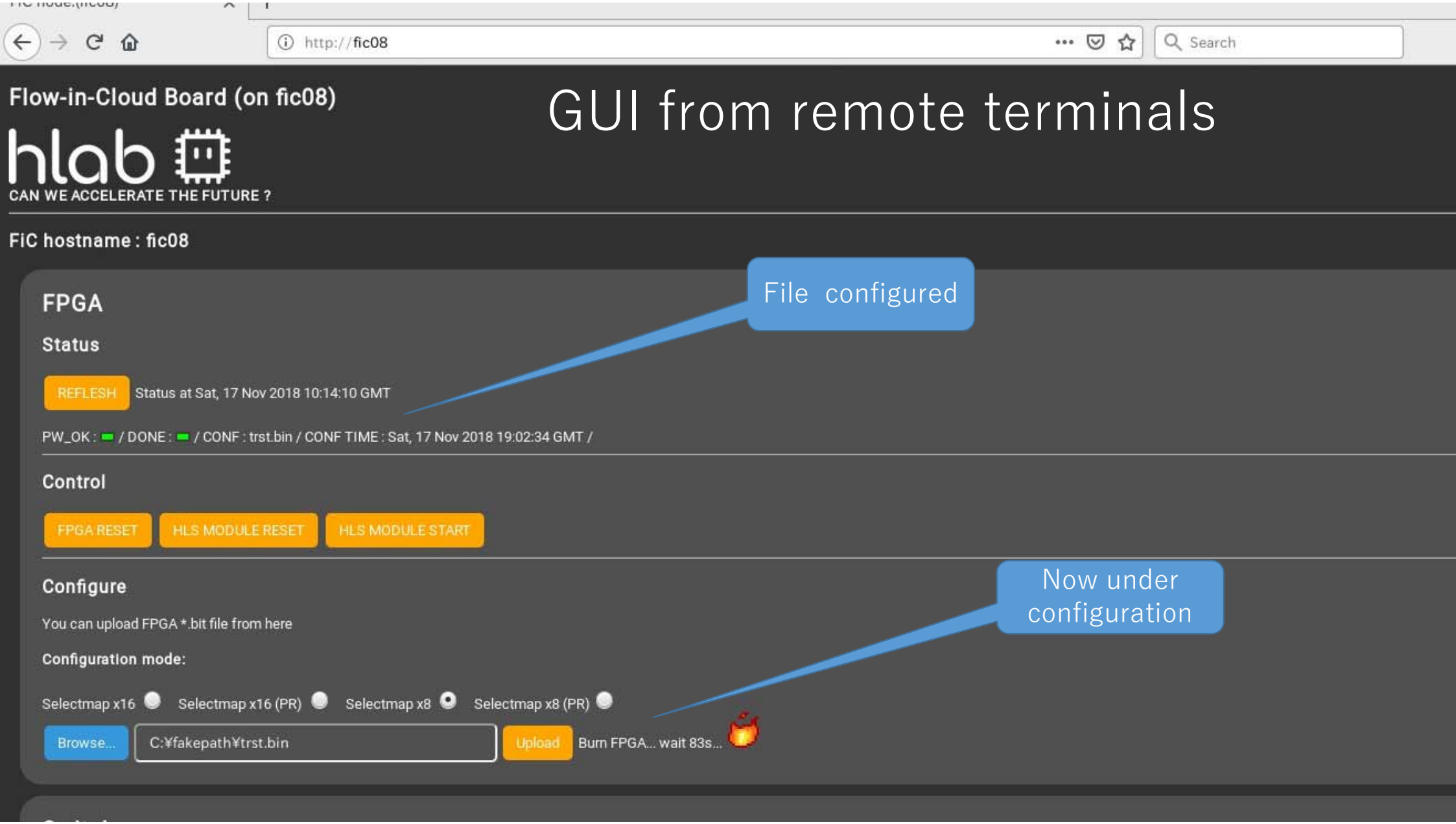
Partial Reconfiguration for separating HLS from switches.



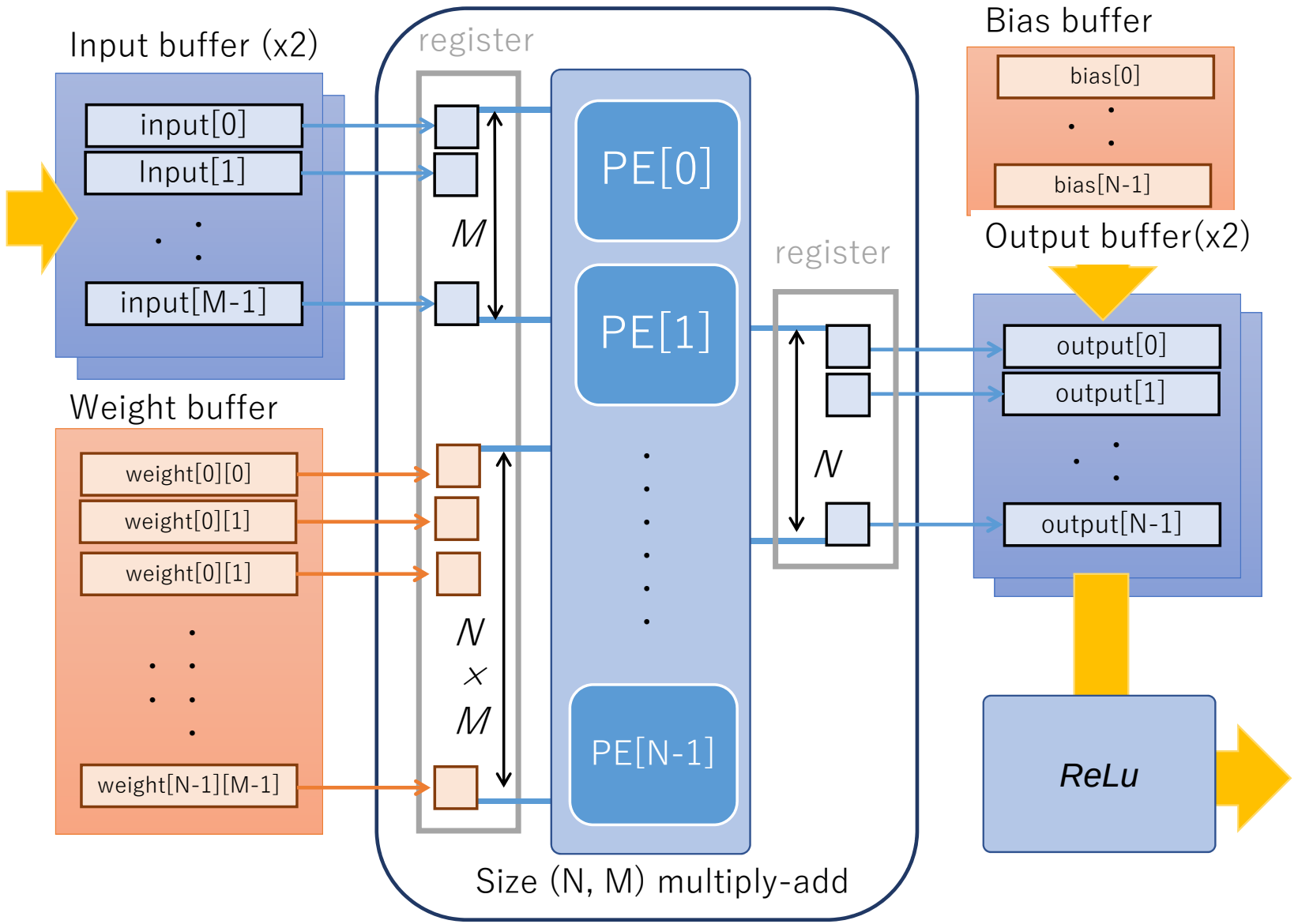

.dcp - [/home/asap2/miho/pr_base_3X3/checkpoint/top_route_design.dcp] - Vivado 2017.2.1 (photon.am.ics.keio.ac.jp)
Tools
Window
Layout
View
Help
Quick Access
Default Layout
- xcku095-ffvb2104-1-c
Device
PR region for HLS module
Now, 3 lanes (12 channels) are used.
Available
Assigned
% Util
353120
6303
1.78
353120
6192
1.75
50400
111
0.22
Statistics
Cells
Messages
AM32X1D (RAMD32, RAMD32): 24 instances
AM32X1S (RAMS32): 128 instances
AM32M16 (RAMS32, RAMS32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32, RAMD32): 4 instances
16E: 296 instances
-604] Checkpoint was created with Vivado v2017.2.1 (64-bit) build 1957588
Time (s): cpu = 00:01:11 ; elapsed = 00:02:10 . Memory (MB): peak = 2663.348 ; gain = 1484.504 ; free physical = 37099 ; free virtual = 182235
route_design

The current FiC system




http://fic08
Search
Flow-in-Cloud Board (on fic08)
hlab
CAN WE ACCELERATE THE FUTURE ?
GUI from remote terminals
FiC hostname : fic08
FPGA
Status
REFLESH
Status at Sat, 17 Nov 2018 10:14:10 GMT
PW_OK : / DONE : / CONF : trst.bin / CONF TIME : Sat, 17 Nov 2018 19:02:34 GMT /
File configured
Control
FPGA RESET
HLS MODULE RESET
HLS MODULE START
Configure
You can upload FPGA *.bit file from here
Configuration mode:
Selectmap x16
Selectmap x16 (PR)
Selectmap x8
Selectmap x8 (PR)
Browse...
C:¥fakepath¥trst.bin
Upload
Burn FPGA... wait 83s...
Now under configuration

Implementation Example (LeNet Fully Connected layer)



Fully connection layer of the Lenet

	Frequency（MH z）	Power（W）	image/sec	GOPS	GOPS/W
1 board	100	17.89	23551	120.58	6.74
4 boards	100	71.56	94226	482.34	6.74

	Frequency（MHz）	Power（W）	GOPS	GOPS/W
FiC 4boards	100	71.56	482.34	6.74
Stratex-V [1]	120	25.8	136.5	5.29
KCU060　[2]	200	25.0	172.0	6.88

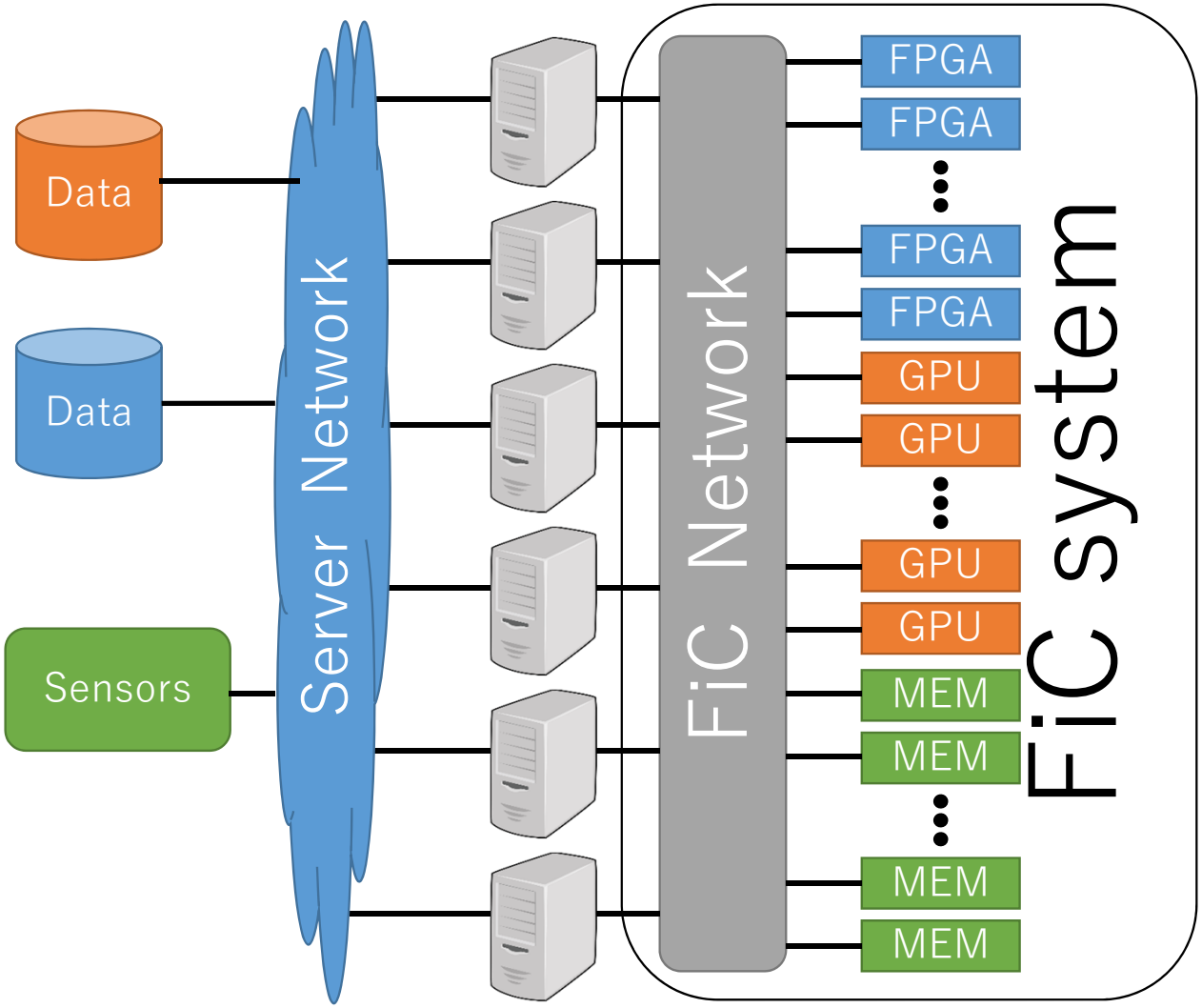
Higher performance is achieved with the similar power efficiency compared with a single FPGA system.

[1] N.Suda, V.Chandra, G.Dasika, et.al “Throughput-optimized open-based FPGA Accelerator for large-scale comvolutional neural networks,” FPGA2016.
[2] C.Zhang, Z.Fang, P.Zhao, P.Pan, J.Cong, “Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks,” ICCCAD2017.

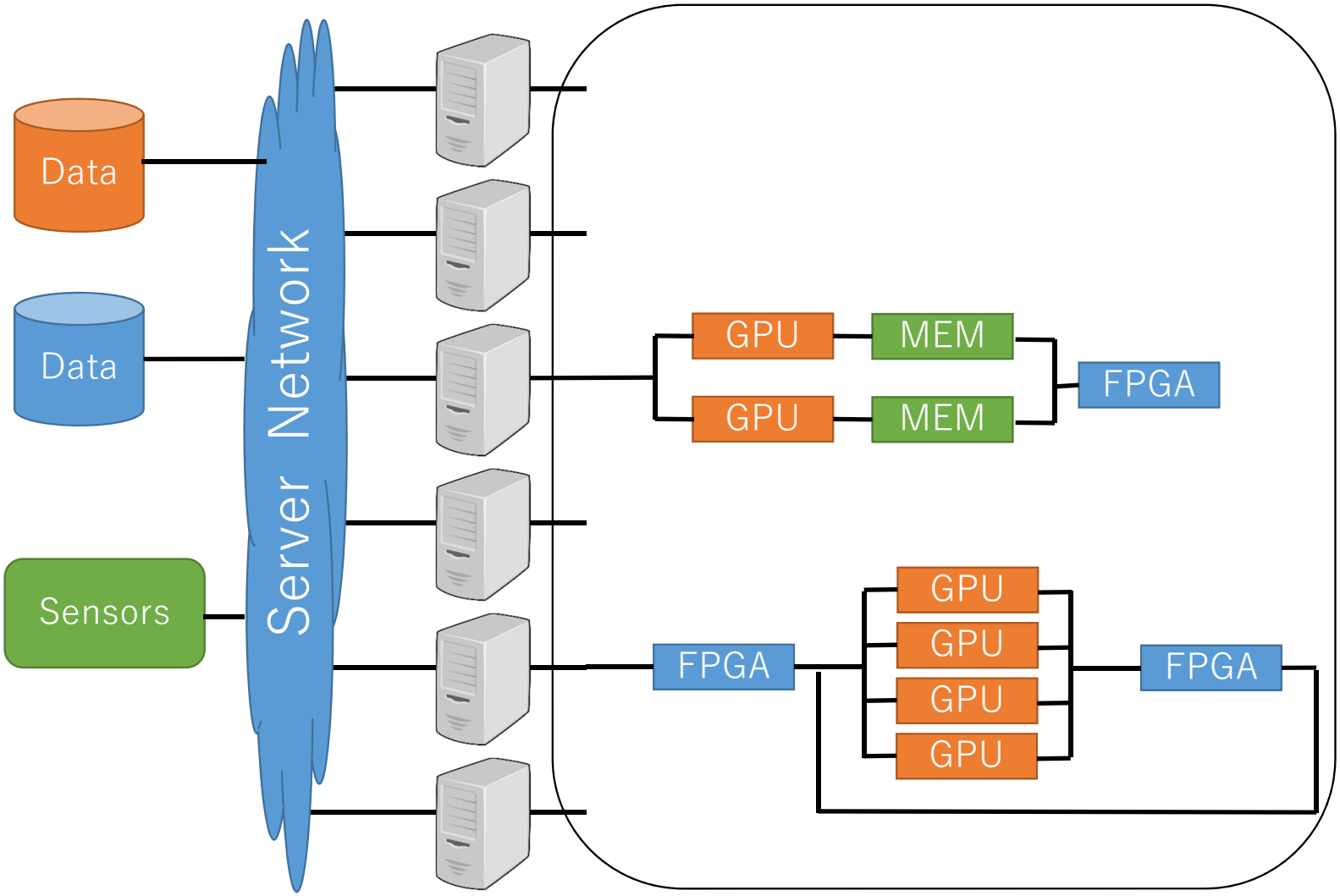
Today's talk

- Building a virtual large FPGA
  - Concept 1: Use middle-range FPGAs and common serial links
  - Concept 2: Virtualize at the level of HLS description
  - Concept 3: Couple accelerators and a switch tightly in an FPGA
    → Accelerator-in-Switch
- Our prototype: FiC (Flow-in-Cloud)
- Next step: Building a virtual heterogeneous computing system

Next Step:
Accelerator bare-metal cloud



FLOW-OS is now under development by the national institute of industrial science and technology (AIST)



Conclusion

- A virtual large FPGA:
  - Scalable performance with
  - similar power/performance and
  - smaller cost/performance compared to conventional FPGA-in-clouds.
- Future direction

  → Accelerator bare metal cloud

  Integration of FPGAs and GPUs.