# 拡張型 SIMD RISC-V アーキテクチャによるアイドル状態の 演算コア削減とその高効率化

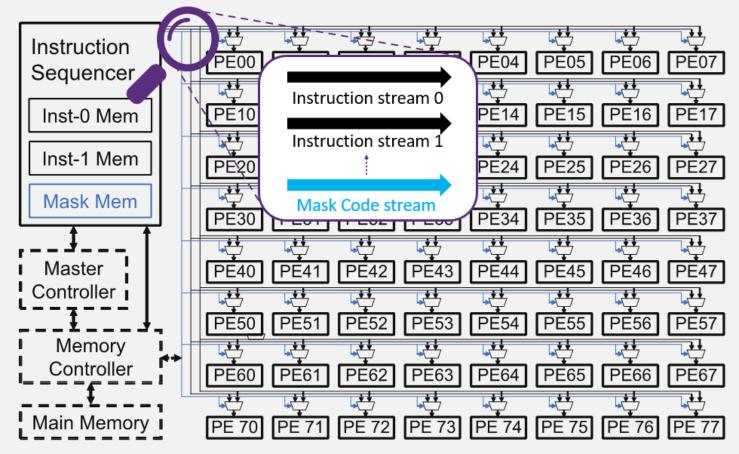
Optimizing Efficiency in Extended SIMD RISC-V Based Architectures through Minimization of Idle Computational Cores

> Masaru NISHIMURA, Yuxi TAN, Yoshiki Yamaguchi University of Tsukuba

# Motivation & Target

- SIMD processors often suffer from Idle time caused by data transfer & limited number of instructions at same time. How can we deal with them ?
- Introduce DIMD (Dual Instruction Multiple-Data), Torus/Mesh Interconnect, and HBM (High Bandwidth Memory) and Memory Hierarchy.
- Adopt RISC-V as PE(Processing Element)'s ISA that can be utilize matured ecosystems for software programming.
- Application range: HPC, Al Computing, and Embedded systems, etc.

# Architecture Overview - DIMD



### **DIMD (Dual Instruction Multiple-Data)**

Issue 2 instructions and mask code to all PE at same cycle.

Then each PE determine which instruction should be executed.

#### Benefits

Reduce idle time caused by limited number of instructions at SIMD. Effective for stencil computation, FFT, etc.

### Architecture Overview / DIMD

#### Format of mask code

Mask code consists of serial numbers and numbering scheme.

18	11	10	3	2	0
inst-1	serial number	inst-	0 serial number	numberir	ng scheme

	000: row-based 001:column-based							ed	010: center							011: diagonal															
0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1	0	1	2	3	4	5	6	7	0	1	1	1	1	1	1	0	1	0	1	2	3	4	5	6
2	2	2	2	2	2	2	2	0	1	2	3	4	5	6	7	0	1	2	2	2	2	1	0	2	1	0	1	2	3	4	5
3	3	3	3	3	3	3	3	0	1	2	3	4	5	6	7	0	1	2	3	3	2	1	0	3	2	1	0	1	2	3	4
4	4	4	4	4	4	4	4	0	1	2	3	4	5	6	7	0	1	2	3	3	2	1	0	4	3	2	1	0	1	2	3
5	5	5	5	5	5	5	5	0	1	2	3	4	5	6	7	0	1	2	2	2	2	1	0	5	4	3	2	1	0	1	2
6	6	6	6	6	6	6	6	0	1	2	3	4	5	6	7	0	1	1	1	1	1	1	0	6	5	4	3	2	1	0	1
7	7	7	7	7	7	7	7	0	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1	0
	100: 45 degree 101: 135 degree									110: 225 degree							111: 315 degree														
	10	0:	45	de	gr	ee		·	101	1: 1	35	i d	eg	ree	•	1	110	): 2	22	5 d	eg	ree	Э	1	111	::	315	5 d	eg	ree	е
0	10 1	0: 2	45 3	de 4	egr 5	ee 6	7	7	101 6	1: 1 5	35 4	5 d 3	eg 2	ree 1	) 0	7	110 7	): 2 7	22: 7	5 d 7	eg 7	ree 7	e 7	1 7	111 7	: : 7	315 7	5 d 7	eg 7	ree 7	e 7
-	10 1 1	0: 2 2	45 3 3	de 4 4	_		_	7 7 7		1:1 5 5	35 4 4	-	eg 2 2	ree 1 1		7 7 7	110 7 6	): 2 7 6	225 7 6	5 d 7 6	_	ree 7 6	e 7 6	7 6	11 7 6	: 7 6	315 7 6	5 d 7 6	eg 7 6		e 7 7
-	10 1 1 2	0: 2 2 2	45 3 3	4	5	6	7	7	6	5	4	3	eg 2 2 2	ree 1 1 2		7	7	7	7	7	7	7	7	7 6 5	11 7 6 5	7 6 5	7	5 d 7 6 5	7	7	7
0 1	10 1 1 2 3	0: 2 2 2 3	45 3 3 3 3	4	5 5	6 6	7 7	7 7	6 6	5 5	4 4	3 3	eg 2 2 2 3	ree 1 1 2 3	0 1	7 7	7 6	7 6	7	7	7	7	7 6	7 6 5 4	11 7 6 5 4	7 6 5 4	7	5 d 7 6 5 4	7 6	7 6	7
0 1 2	10 1 2 3 4	0: 2 2 2 3 4	45 3 3 3 4	4 4 4	5 5 5	6 6 6	7 7 7	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	2 2 2	1 1 2	0 1 2	7 7 7	7 6 6	7 6 5	7 6 5	7	7	7	7 6 5	7 6 5 4 3	11 7 6 5 4 3	7 6 5 4 3	7	5 d 7 6 5 4 4	7 6 5	7 6 6	7
0 1 2	10 1 2 3 4 5	0: 2 2 3 4 5	45 3 3 3 4 5	4 4 4 4	5 5 5 5	6 6 6 6	7 7 7 7	7 7 7 7	6 6 6	5 5 5 5	4 4 4 4	3 3 3	2 2 2	1 1 2	0 1 2 3	7 7 7	7 6 6	7 6 5 5	7 6 5 4	7	7	7 6 5 4	7 6 5 4	7 6 5 4	11 7 6 5 4 3 2	: 3 6 5 4 3 2	7 6 5 4	7 6 5 4	7 6 5 5	7 6 6	7
0 1 2 3 4	1 1 2 3 4	0: 2 2 3 4 5 6	45 3 3 3 4 5 6	4 4 4 4 4	5 5 5 5 5 5	6 6 6 6	7 7 7 7 7	7 7 7 7 7	6 6 6 6	5 5 5 5 5 5	4 4 4 4	3 3 3	2 2 2	1 1 2 3 4	0 1 2 3 4	7 7 7 7 7	7 6 6 6 6	7 6 5 5 5	7 6 5 4 4	7	7 6 5 4 3	7 6 5 4 3	7 6 5 4	7 6 5 4	11 7 6 5 4 3 2 1	7 6 5 4 3 2 2	7 6 5 4 3	7 6 5 4 4	7 6 5 5 5	7 6 6 6	7
0 1 2 3 4 5	1 1 2 3 4	0: 2 2 3 4 5 6 7	45 3 3 3 4 5 6 7	4 4 4 4 5	5 5 5 5 5 5 5	6 6 6 6 6	7 7 7 7 7	7 7 7 7 7 7	6 6 6 6 6	5 5 5 5 5 5 5	4 4 4 4 5	3 3 3 4 5	2 2 3 4 5	1 1 2 3 4 5	0 1 2 3 4 5	7 7 7 7 7 7	7 6 6 6 6	7 6 5 5 5 5 5	7 6 5 4 4 4	7 6 5 4 3 3	7 6 5 4 3 2	7 6 5 4 3	7 6 5 4	7 6 5 4	11 7 6 5 4 3 2 1 1	7 6 5 4 3 2 2 2	7 6 5 4 3 3	7 6 5 4 4 4 4	7 6 5 5 5 5 5	7 6 6 6 6	7

Inst-0 Inst-1

Inst-idle

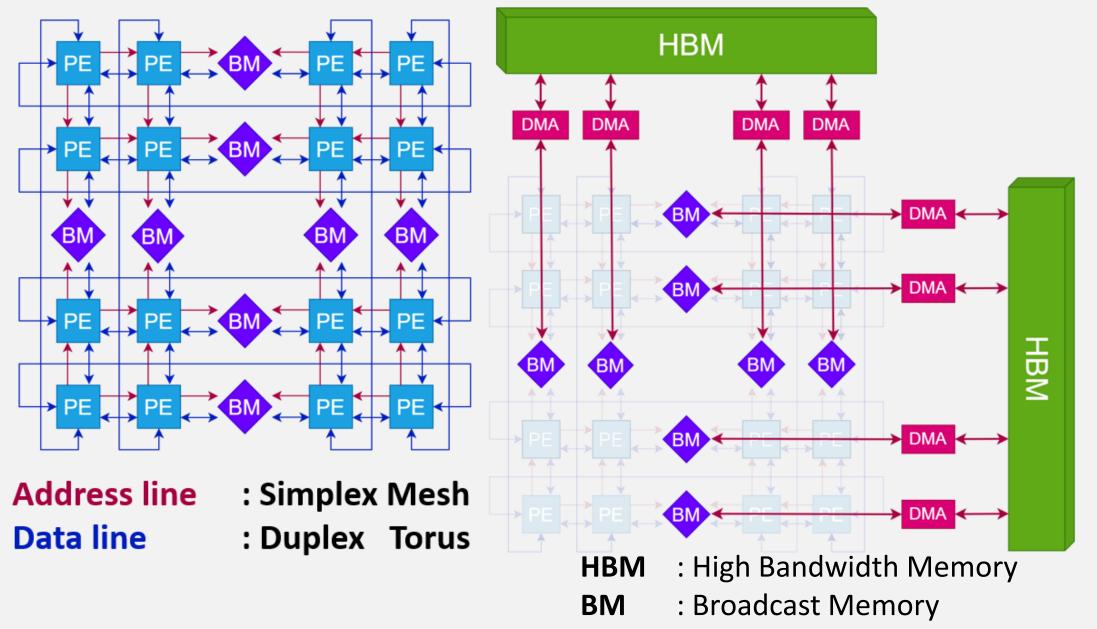
Example of Operation

01100000 00011111 100

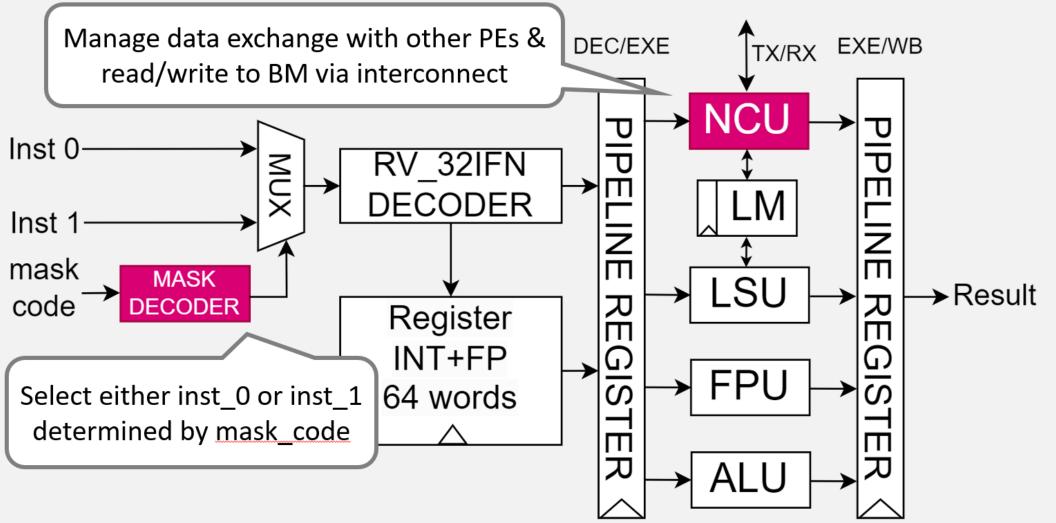
#### Architecture Overview

## **Torus/Mesh Interconnect**

## **HBM and Memory Hierarchy**



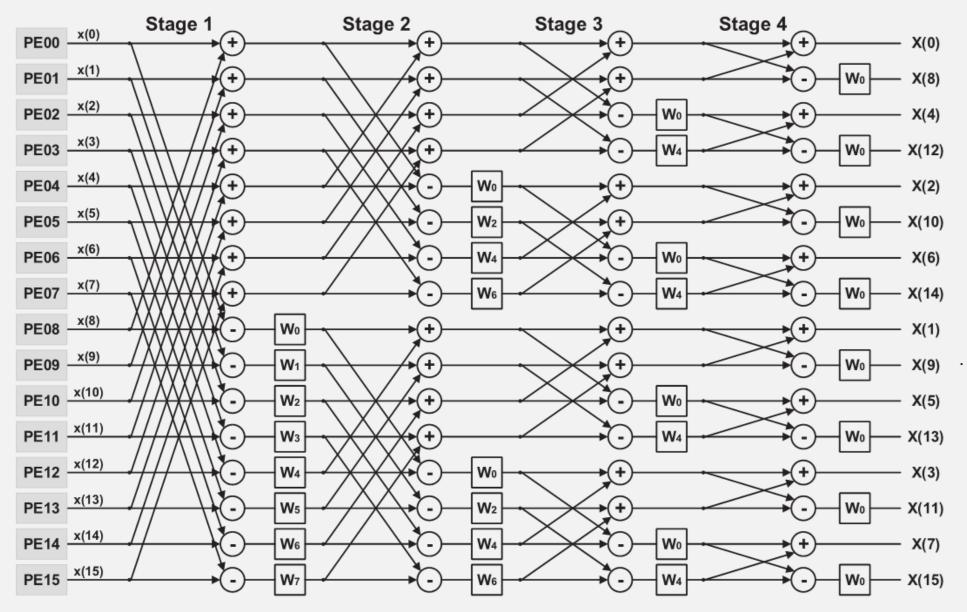
# 3. Architecture of the Processing Element

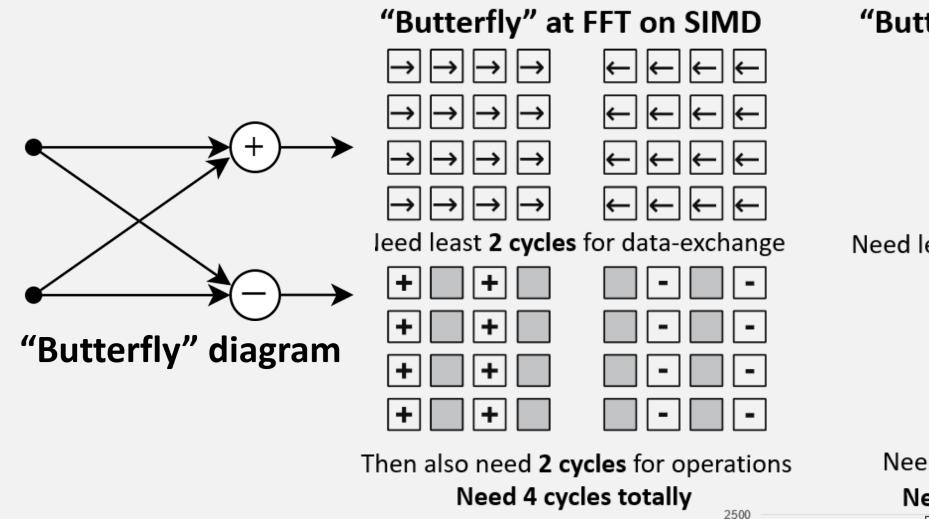


Adopt RISC-V as PE's ISA. We've fully designed processor dedicated to PE operations.

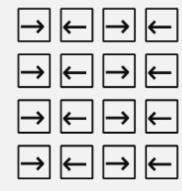
NCU	: Neighbor Communication Unit
LM	: Local Memory
N extension	: support inst. corresponding to NCU.

## 4. Comparison of FFT performance with SIMD

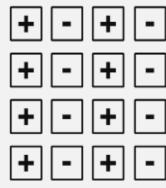




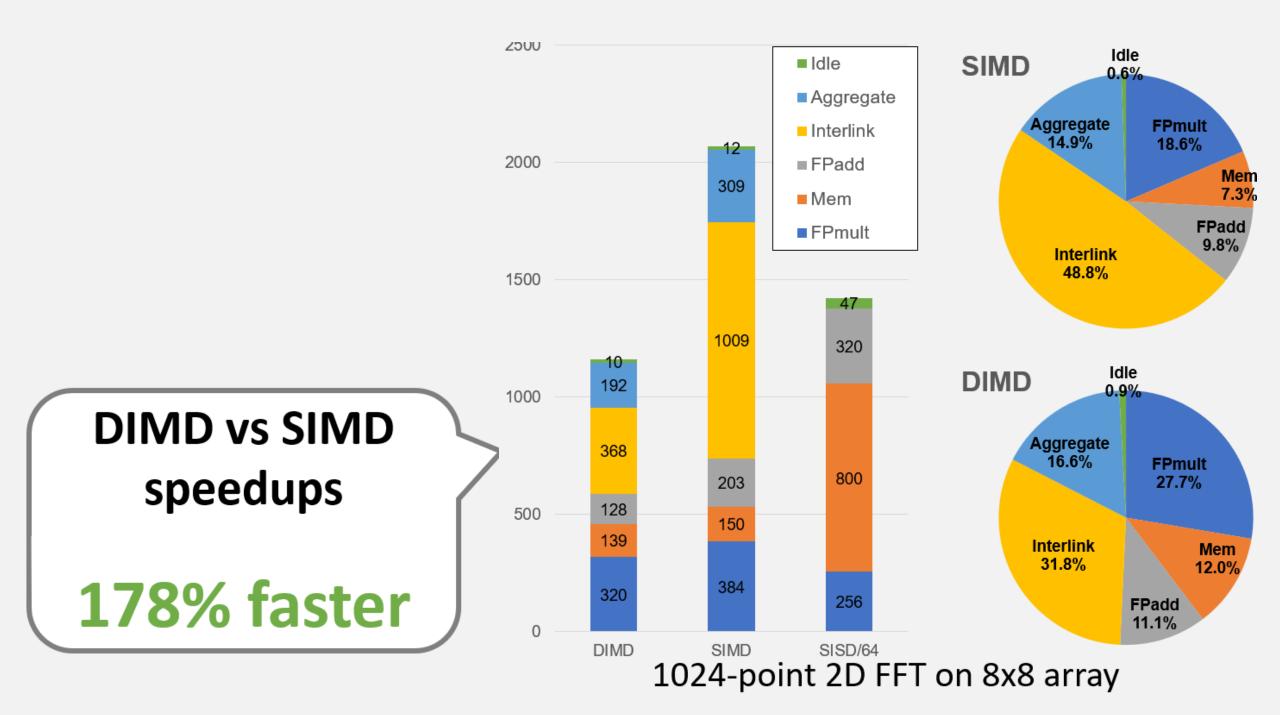
### "Butterfly" at FFT on DIMD



Need least 1 cycle for data-exchange



Need 1 cycles for operations Need only 2 cycles totally





We are pursuing to implement this architecture on high-end FPGA, Xilinx ALVEO U280.

Estimating we can achieve more than 400 PEs scale many core processors.