

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

Masaru Nishimura, Yuxi Tan, Yoshiki Yamaguchi  
 Graduate School of Science and Technology, University of Tsukuba

2023 RISC-V Days Tokyo @UTokyo, 6/20



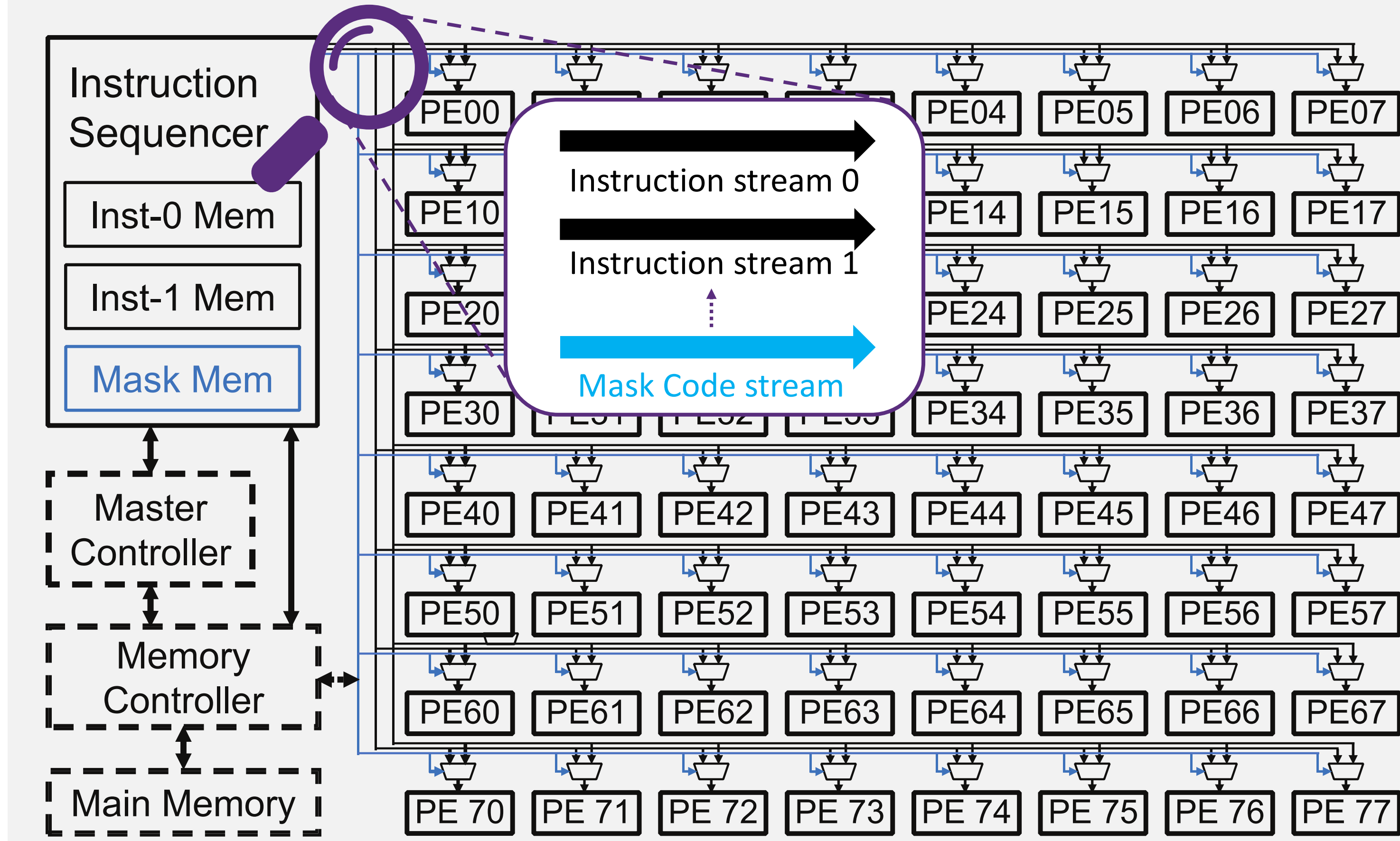
## 1. Motivation & Target

- Introduce **DIMD (Dual Instruction Multiple-Data)**, **Torus/Mesh Interconnect**, and **HBM (High Bandwidth Memory)** and **Memory Hierarchy**.
- SIMD processors often suffer from **Idle time** caused by **data transfer** & limited number of instructions at same time. How can we deal with them?

## 2. Architecture Overview

### DIMD (Dual Instruction Multiple-Data)

Issue 2 instructions with 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.

### Format of mask code

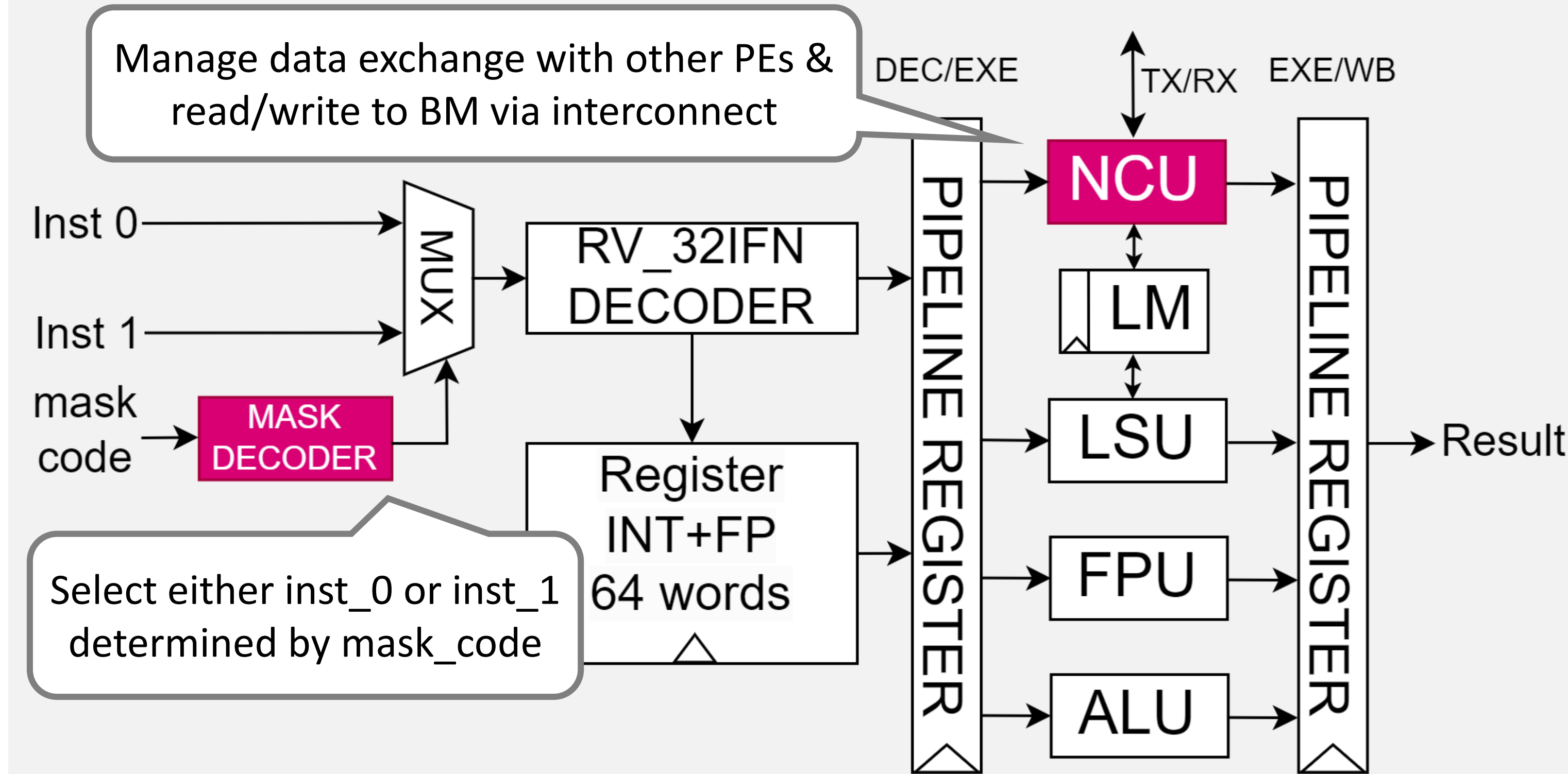
Mask code consists of **serial numbers** and **numbering scheme**.

18	11	10	3	2	0
inst-1 serial number		inst-0 serial number		numbering scheme	
<b>000: row-based</b>		<b>001: column-based</b>		<b>010: center</b>	
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	0 0 1 1 1 1 1 1	0 1 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 1	0 1 2 3 4 5 6 7	0 1 2 2 2 2 1 0	2 1 0 1 2 3 4 5
2 2 2 2 2 2 2 2	0 1 2 3 4 5 6 7	0 1 2 3 3 2 1 0	0 1 2 3 4 5 6 7	0 1 2 3 3 2 1 0	3 2 1 0 1 2 3 4
3 3 3 3 3 3 3 3	0 1 2 3 4 5 6 7	0 1 2 3 2 2 1 0	0 1 2 3 4 5 6 7	0 1 2 2 2 1 0	4 3 2 1 0 1 2 3
4 4 4 4 4 4 4 4	0 1 2 3 4 5 6 7	0 1 2 3 1 1 0	0 1 2 3 4 5 6 7	0 1 2 2 1 0	5 4 3 2 1 0 1 2
5 5 5 5 5 5 5 5	0 1 2 3 4 5 6 7	0 1 2 3 0 0 0	0 1 2 3 4 5 6 7	0 1 1 1 1 1 1 0	6 5 4 3 2 1 0 1
6 6 6 6 6 6 6 6	0 1 2 3 4 5 6 7	0 1 2 3 0 0 0	0 1 2 3 4 5 6 7	0 0 0 0 0 0 0 0	7 6 5 4 3 2 1 0
7 7 7 7 7 7 7 7	0 1 2 3 4 5 6 7	0 1 2 3 0 0 0	0 1 2 3 4 5 6 7		
<b>100: 45 degree</b>		<b>101: 135 degree</b>		<b>110: 225 degree</b>	
0 1 2 3 4 5 6 7	7 6 5 4 3 2 1 0	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7
1 1 2 3 4 5 6 7	7 6 5 4 3 2 1 1	7 6 6 6 6 6 6 6	7 6 6 6 6 6 6 6	7 6 6 6 6 6 6 6	7 6 6 6 6 6 6 6
2 2 2 3 4 5 6 7	7 6 5 4 3 2 2 2	7 6 5 5 5 5 5 5	7 6 5 5 5 5 5 5	7 6 5 5 5 5 5 5	7 6 5 5 5 5 5 5
3 3 3 3 4 5 6 7	7 6 5 4 3 3 3 3	7 6 5 4 4 4 4 4	7 6 5 4 4 4 4 4	7 6 5 4 4 4 4 4	7 6 5 4 4 4 4 4
4 4 4 4 4 5 6 7	7 6 5 4 4 4 4 4	7 6 5 4 3 3 3 3	7 6 5 4 3 3 3 3	7 6 5 4 3 3 3 3	7 6 5 4 3 3 3 3
5 5 5 5 5 5 6 7	7 6 5 5 5 5 5 5	7 6 5 4 3 2 2 2	7 6 5 4 3 2 2 2	7 6 5 4 3 2 2 2	7 6 5 4 3 2 2 2
6 6 6 6 6 6 6 7	7 6 6 6 6 6 6 6	7 6 5 4 3 2 1 1	7 6 5 4 3 2 1 1	7 6 5 4 3 2 1 1	7 6 5 4 3 2 1 1
7 7 7 7 7 7 7 7	7 7 7 7 7 7 7 7	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0

### Example of Operation

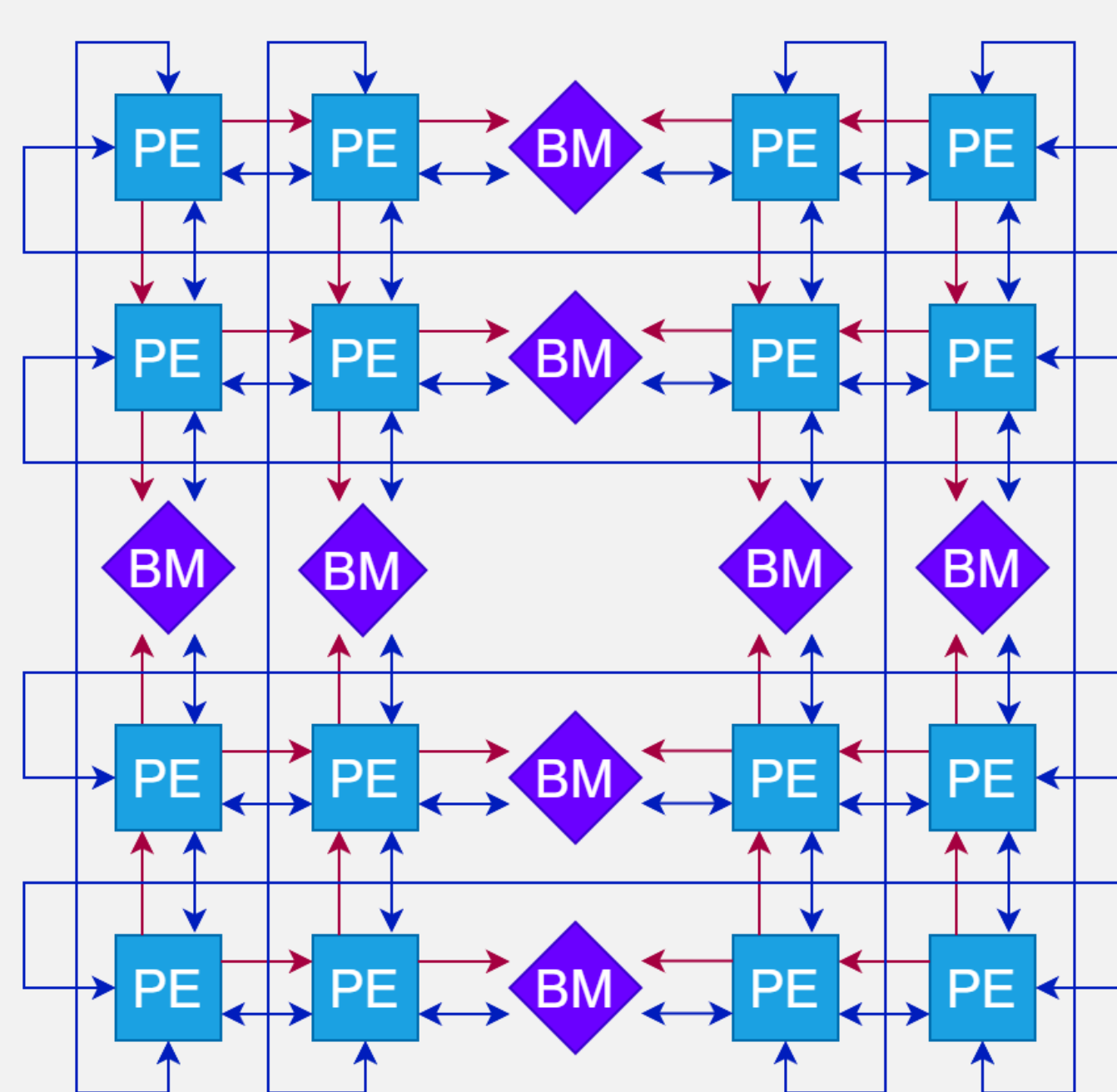
01100000 00011111 100 ■ Inst-0 ■ Inst-1 ■ Inst-idle

## 3. Architecture of the Processing Element



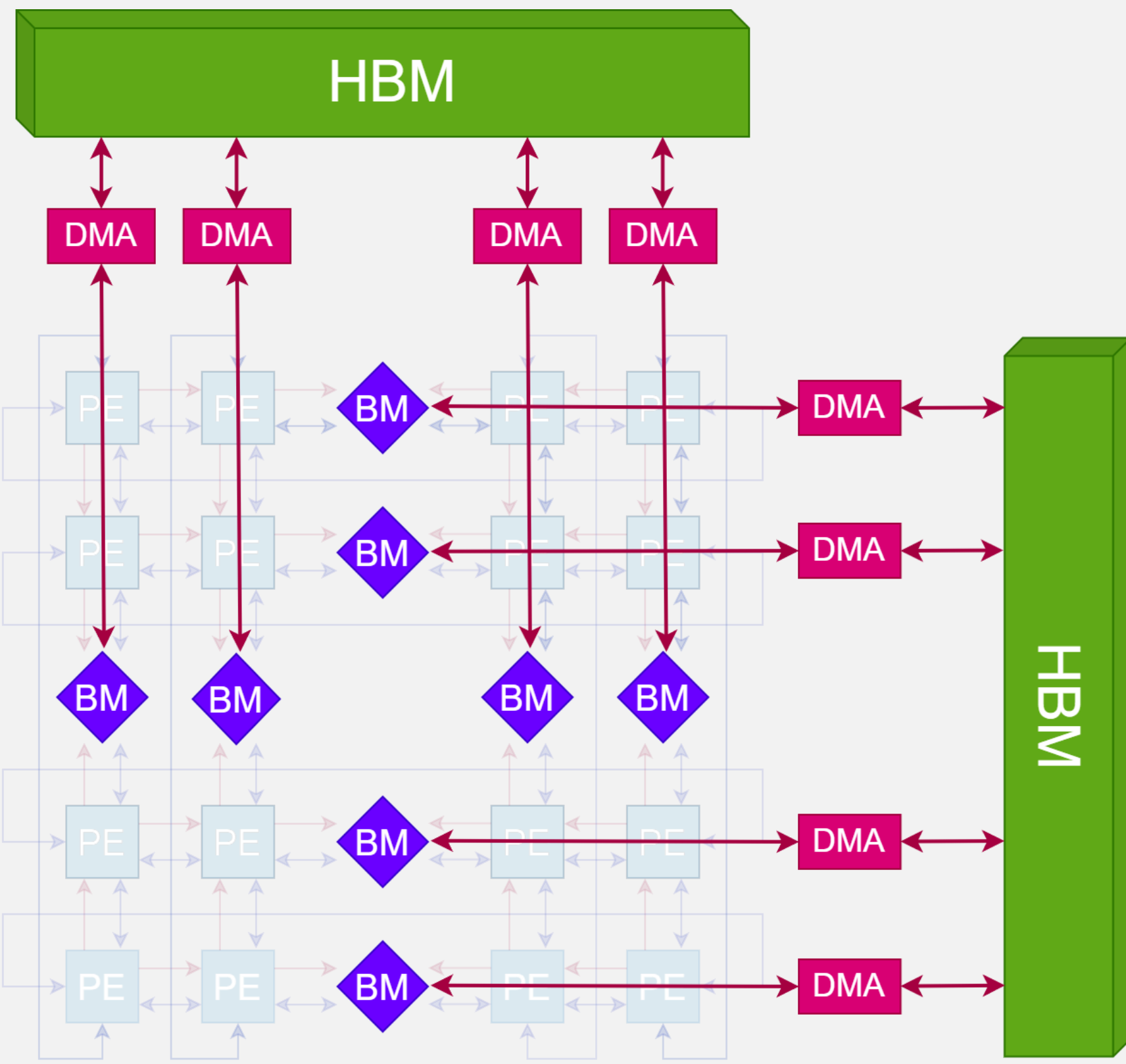
Adopt **RISC-V** as PE's ISA. We've fully designed processor dedicated to PE operations.

### Torus/Mesh Interconnect

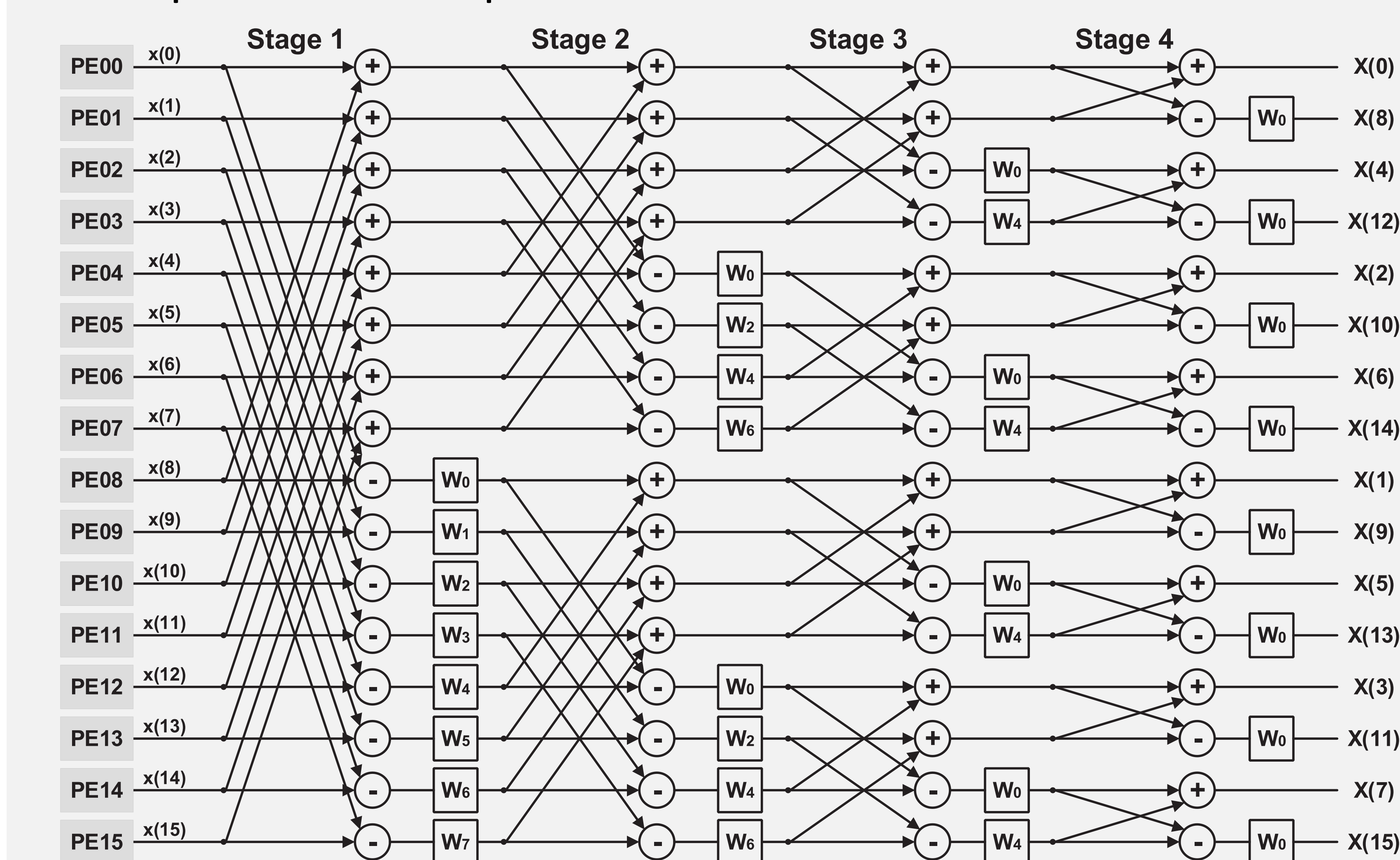


**Address line** : Simplex Mesh  
**Data line** : Duplex Torus

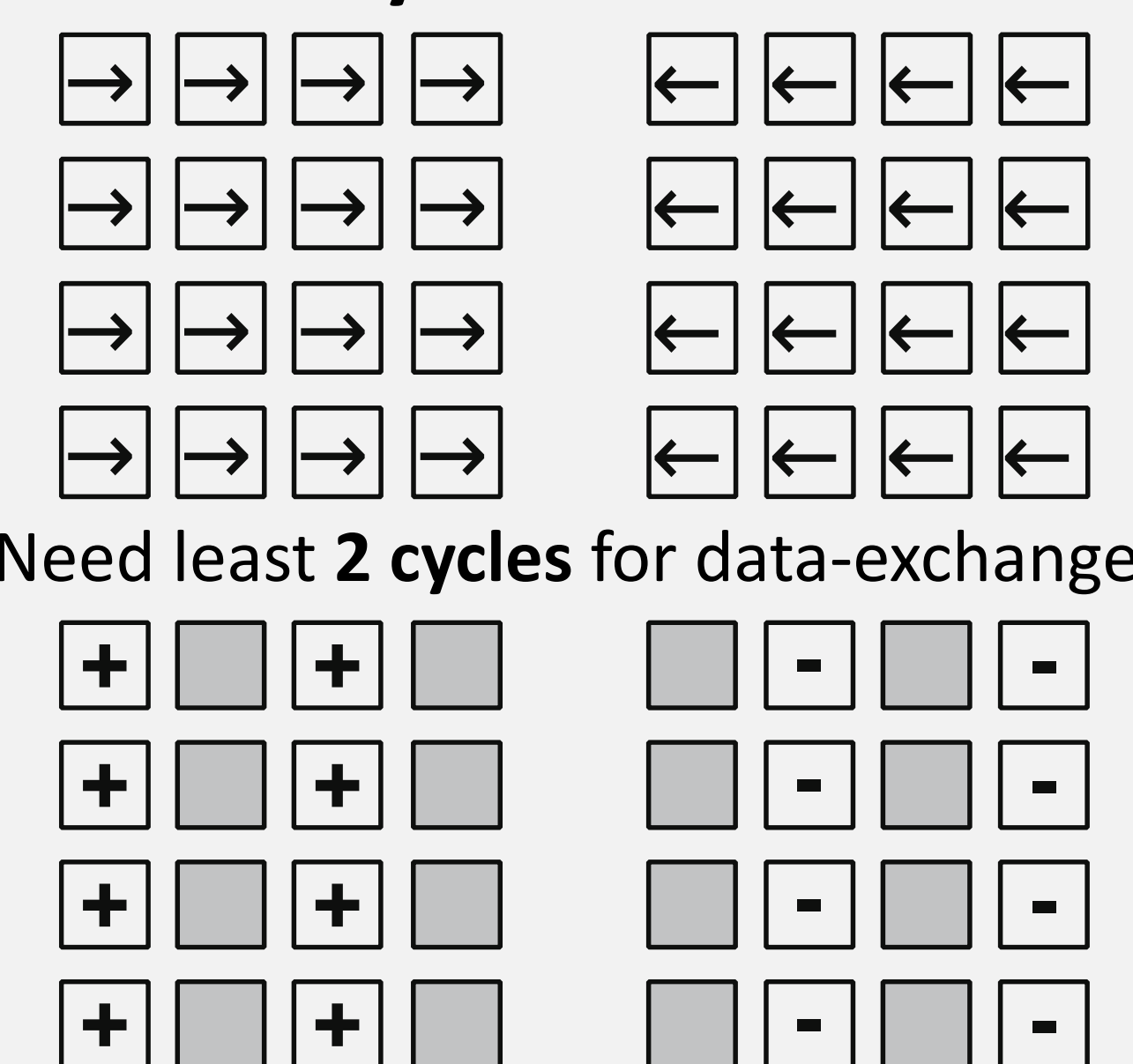
### HBM and Memory Hierarchy



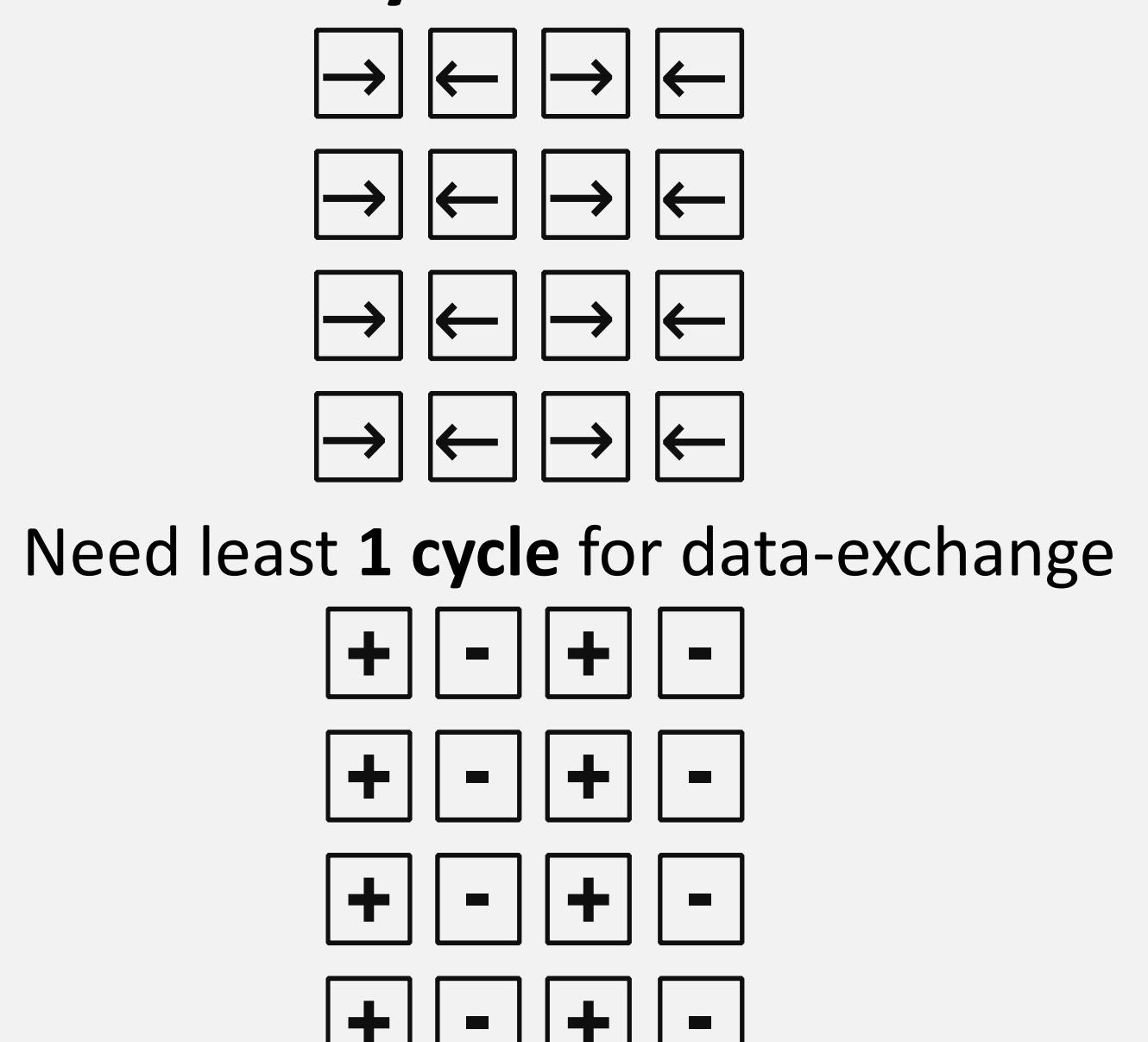
## 4. Comparison of FFT performance with SIMD



### "Butterfly" at FFT on SIMD

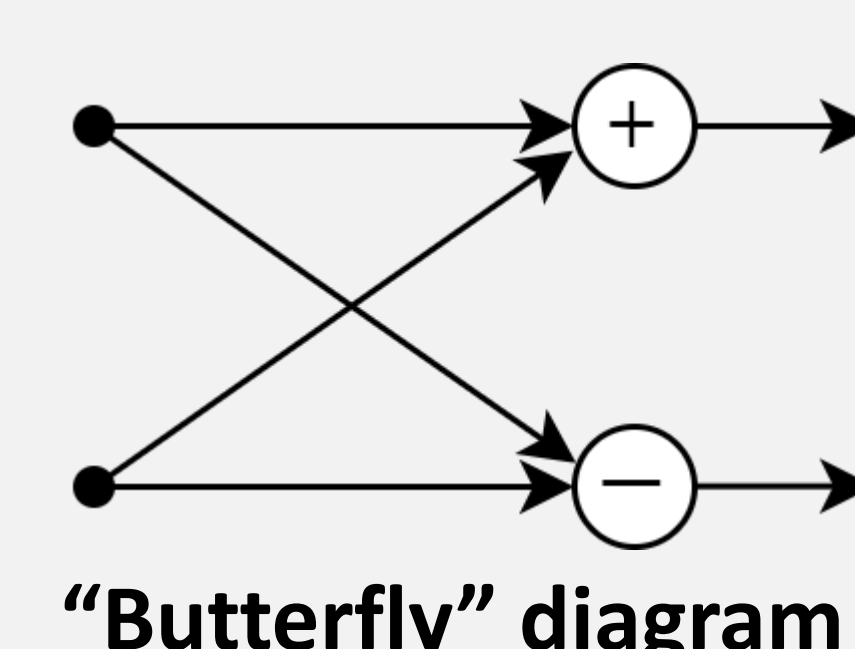


### "Butterfly" at FFT on DIMD



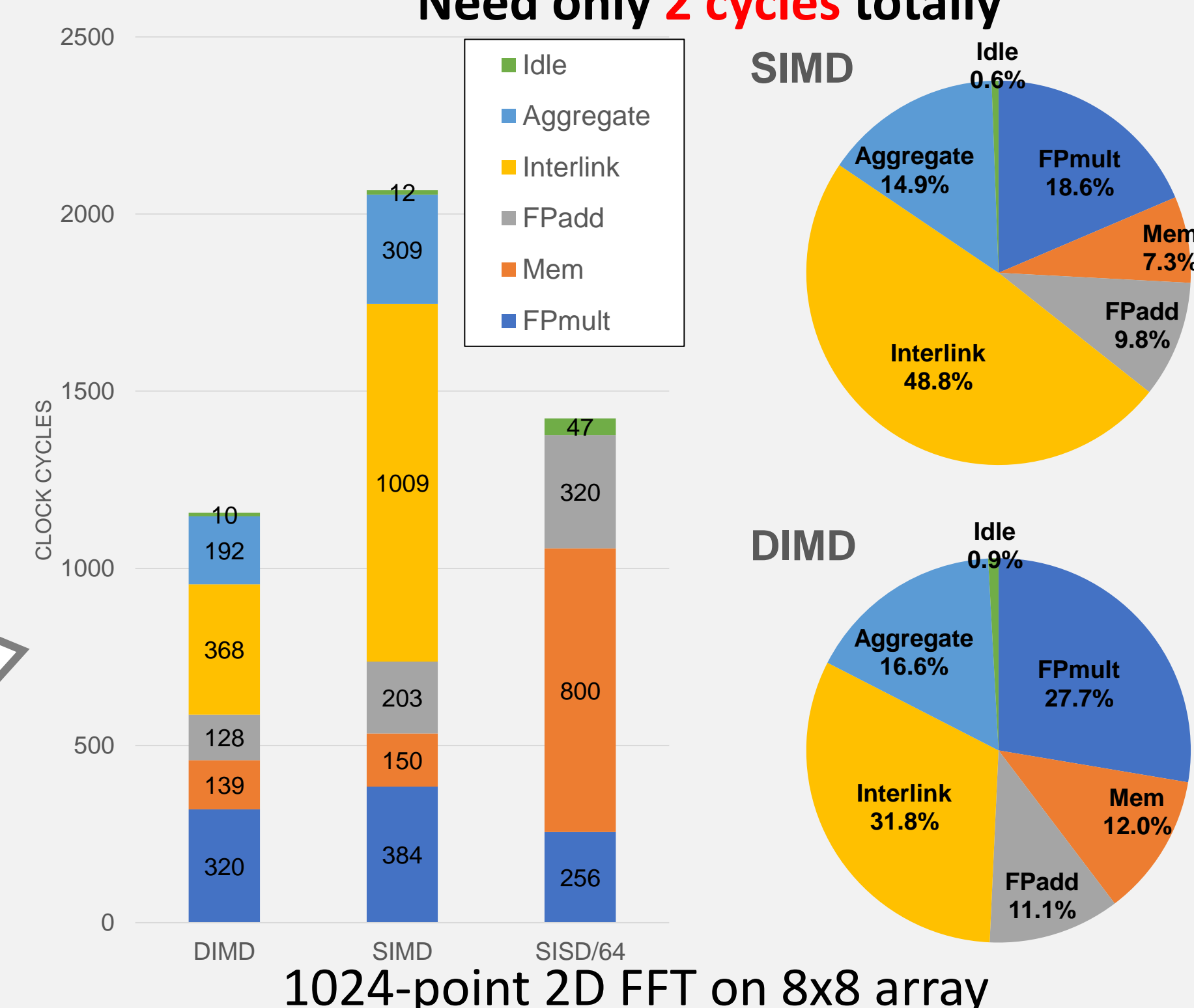
Then also need 2 cycles for operations  
**Need 4 cycles totally**

Need 1 cycles for operations  
**Need only 2 cycles totally**



**DIMD vs SIMD speedups**

**178% faster**



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.