Implementation and Performance Optimization of Lattice QCD Tool Kit on The Cell/B.E.

Shinji Motoki

Graduate School of Biosphere Science Hiroshima University

In collaboration with

Y. Nakagawa (Niigata University)
K. Nagata (Univ. of Tokyo)
A. Nakamura (RIISE Hiroshima University)

The XXVII International Symposium on Lattice Field Theory - LAT2010

Plan of the Talk

- Introduction and Motivation
- Cell/B.E. Architecture
- Numerical simulation details
- Summary and future plans

Introduction and Motivation

- This work is a part of the Lattice Tool Kit (LTK) project, which is intended to provide sets of free QCD codes, and anyone can use them for writing her/his own program.
- The aim of this study is a development of a SU(3) matrix multiplication code on the Cell/B.E., a new computer architecture with heterogeneous multi-core processor.
- We expect that the Cell/B.E. can be a good cost effective environment for a quenched QCD study such as color confinement, transport coefficients etc..

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

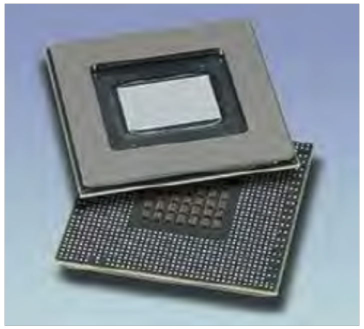
HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Cell/B.E. Architecture

- Developed by IBM SONY and TOSHIBA

  – Consist of 8 Synergistic Processor Element and
    1 PowerPC Processor Element

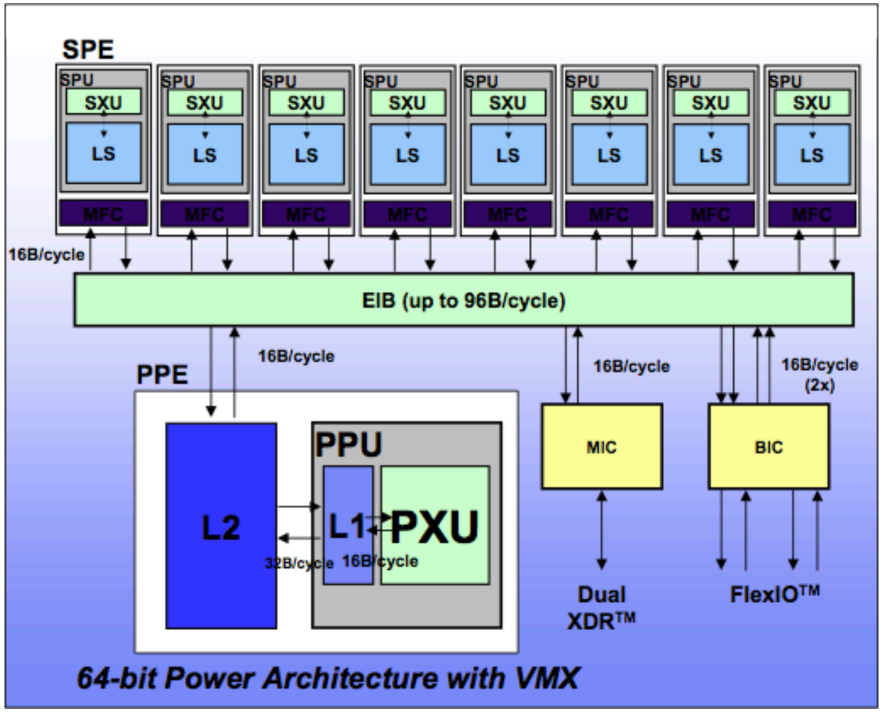
  – High performance Floating point Operations
    - Cell B.E. : 230GFlops (SP) 21GFlops (DP)
    - PowerXCell 8i : **230GFLOPS (SP) 108.5GFLOPS (DP)**


Shinji MOT
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Cell/B.E. Architecture



Multi-core computer:
A Cell/B.E. consists of eight operation system processor cores (SPE) and one system controll processor core (PPE).

SIMD operation:
The SPE is specialized to calculations in the use and has the new architecture with the ability of the SIMD operation.

Small local memory LS:
SPE has a Local Store of 256 KBytes which worked as an inside memory of SPE.

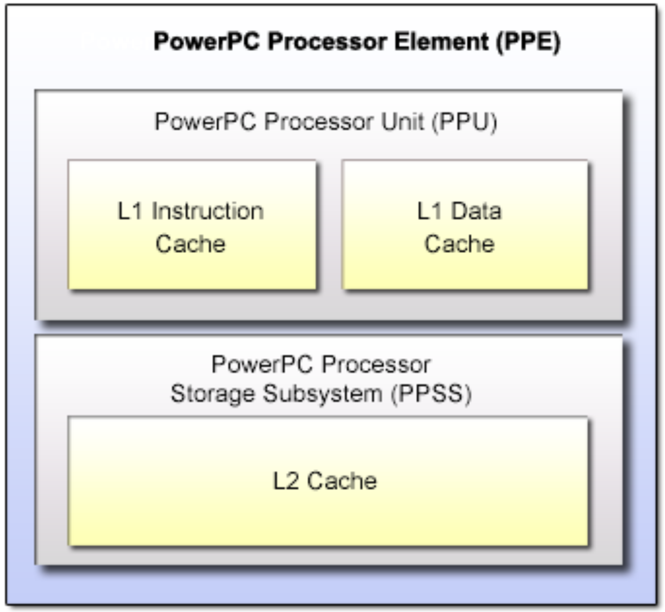
EIB connection and DMA:
PPE and all SPE are connected by a high-speed bus Element Interconnect Bus.
SPE uses Direct Memory Access (DMA) transfer for data transmission. The DMA is used to forward data directly between memory and LS

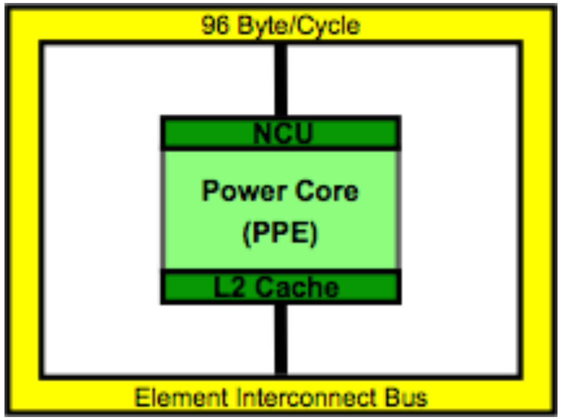
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

PPE Features

HIROSHIMA UNIVERSITY

- General purpose, 3.2GHz 64-bit RISC
- 2-Way hardware multi threaded
- L1 32kB i; 32kB d;
- L2 512kB
- Coherent load / store
- VMX-32
- 32 Unified registers
- In-order-execution



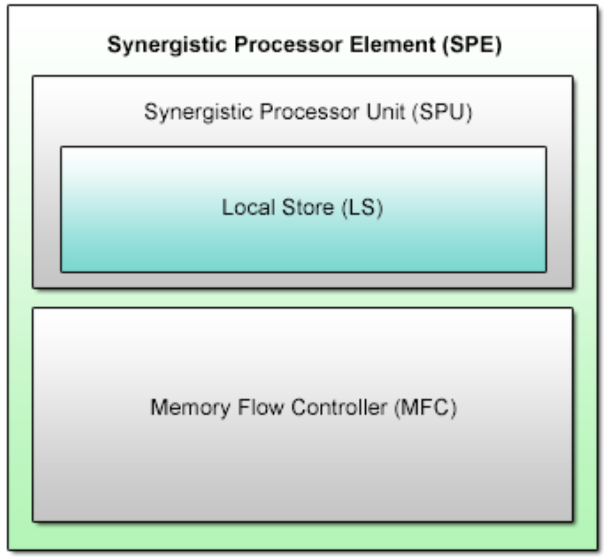


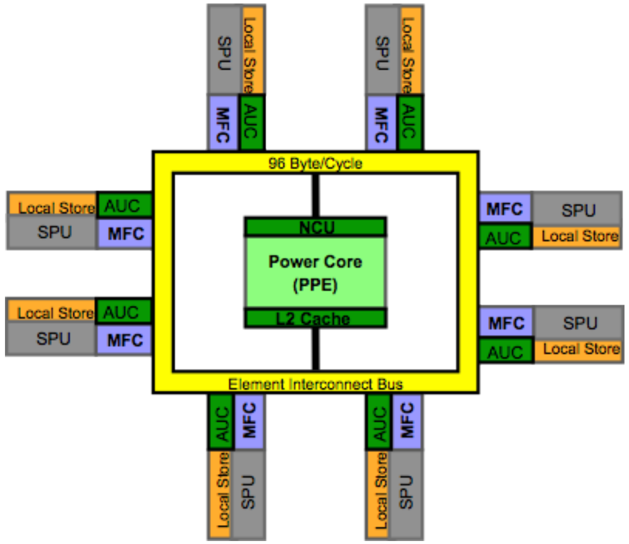
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

SPE Features

HIROSHIMA UNIVERSITY

- 256KB Local Store (LS)
- 2 instructions / cycle
- 128 general registers
-
- 128 bit Data Path (for SIMD)
  - 128bit logic operation/cycle
  - 4 SP(2 DP) floating point operations / cycle
- Latency
  - load 6 cycles
  - SP Float Mul(Add) 7 cycles
  - DP Float Mul(Add) 9(or 13) cycles





Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Numerical simulation details

HIROSHIMA UNIVERSITY

- SU(3) Matrix multiplication

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \times \begin{pmatrix} d_1 & d_2 & d_3 \\ e_1 & e_2 & e_3 \\ f_1 & f_2 & f_3 \end{pmatrix}$$

SU(3) matrix multiplication code on the Cell/B.E.. This calculation is an essential numerical part of any quench QCD calculation

Complex (SP)

573,440 matrix multiplications in single precision.

IBM QS20

Result of only use PPE is 365(msec).

0.3 Gflops...

Theoretical peak speed is 250 Gflops (in this case)

Numerical simulation details

HIROSHIMA UNIVERSITY

- SU(3) Matrix multiplication

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \times \begin{pmatrix} d_1 & d_2 & d_3 \\ e_1 & e_2 & e_3 \\ f_1 & f_2 & f_3 \end{pmatrix}$$

SU(3) matrix multiplication code on the Cell/B.E.. This calculation is an essential numerical part of any quench QCD calculation

Complex (SP)

573,440 matrix multiplications in single precision.

IBM QS20

Result of only use PPE is 365(msec).

0.3 Gflops...

Theoretical speed is 250 Gflops (in this case)

We Need improve calculation

Optimization

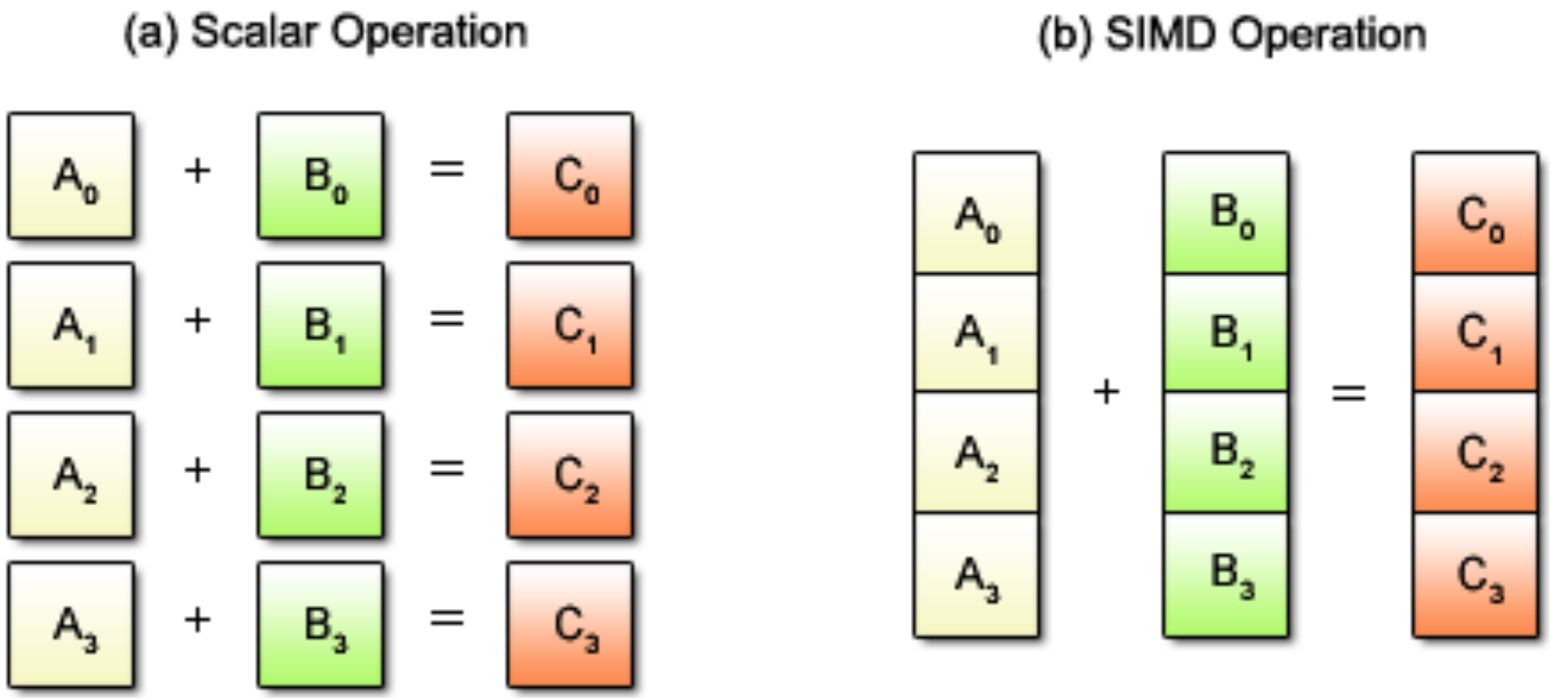
Our Optimization is two steps

- Calculation part Optimize
  - Use SIMD Operation
  - Loop Unrolling and Pipe Line Optimize
- Data Transfer Part Optimize
  - Use Double Buffering
  - Use Unitarity

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Optimization(1) SIMDization



The SIMD operation is the operation technique that can process plural data by one instruction.

Optimization(1) SIMDization

HIROSHIMA UNIVERSITY

for(i = 0; i < 3; i++){
for(j = 0; j < 3; j++){
for(n = 0; n < NV; n++){

$$\vec{D}_{\Re} = \vec{A}^{(n)}_{\Re}(i,0)\cdot\vec{B}^{(n)}_{\Re}(0,j)$$
$$\vec{D}_{\Im} = \vec{A}^{(n)}_{\Re}(i,0)\cdot\vec{B}^{(n)}_{\Im}(0,j)$$

$$\vec{D}_{\Re} = -\vec{A}^{(n)}_{\Im}(i,0)\cdot\vec{B}^{(n)}_{\Im}(0,j)+\vec{D}_{\Re}$$
$$\vec{D}_{\Im} = \vec{A}^{(n)}_{\Im}(i,0)\cdot\vec{B}^{(n)}_{\Re}(0,j)+\vec{D}_{\Im}$$

$$\vec{D}_{\Re} = \vec{A}^{(n)}_{\Re}(i,1)\cdot\vec{B}^{(n)}_{\Re}(1,j)+\vec{D}_{\Re}$$
$$\vec{D}_{\Im} = \vec{A}^{(n)}_{\Re}(i,1)\cdot\vec{B}^{(n)}_{\Im}(1,j)+\vec{D}_{\Im}$$

$$\vec{D}_{\Re} = -\vec{A}^{(n)}_{\Im}(i,1)\cdot\vec{B}^{(n)}_{\Im}(1,j)+\vec{D}_{\Re}$$
$$\vec{D}_{\Im} = \vec{A}^{(n)}_{\Im}(i,1)\cdot\vec{B}^{(n)}_{\Re}(1,j)+\vec{D}_{\Im}$$

$$\vec{D}_{\Re} = \vec{A}^{(n)}_{\Re}(i,2)\cdot\vec{B}^{(n)}_{\Re}(2,j)+\vec{D}_{\Re}$$
$$\vec{D}_{\Im} = \vec{A}^{(n)}_{\Re}(i,2)\cdot\vec{B}^{(n)}_{\Im}(2,j)+\vec{D}_{\Im}$$

$$\vec{C}^{(n)}_{\Re}(i,j) = -\vec{A}^{(n)}_{\Im}(i,2)\cdot\vec{B}^{(n)}_{\Im}(2,j)+\vec{D}_{\Re}$$
$$\vec{C}^{(n)}_{\Im}(i,j) = \vec{A}^{(n)}_{\Im}(i,2)\cdot\vec{B}_{\Re}(2,j)+\vec{D}_{\Im}$$

}}}

In order to extract SPE's calculational power, the full use of its SIMD function is essential.
We must provide our input matrix data in a form which fits the SIMD operation.

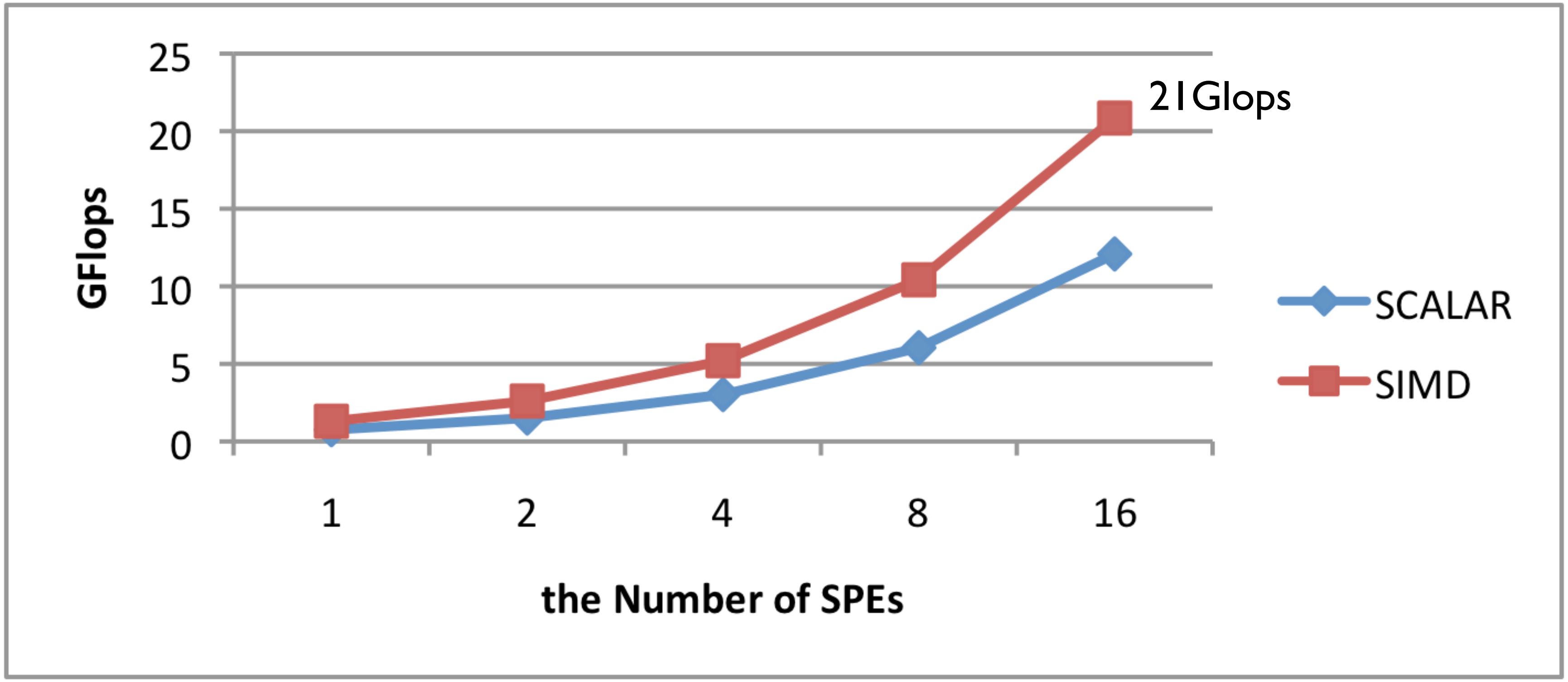
The SPE can handle several data at once by one instruction with vector data type.
The data treated in the Cell/B.E. are 16 bytes fixation, and it can handle four data at once by a single instruction in the single precision floating point arithmetic.

We pack the matrices, a and b, of 16KByte in a structure. In order to fit the algorithm done on SPE, we separate the real and imaginary parts of a complex matrix, and pack 112 matrices.

```
// DMA Send Data Struct(16kbyte Packed)
typedef struct _s_gprod0_send_t
{
        float ar[3][3][112];
        float ai[3][3][112];
        float br[3][3][112];
        float bi[3][3][112];
} s_gprod0_send_t;
```

Result of SIMDization

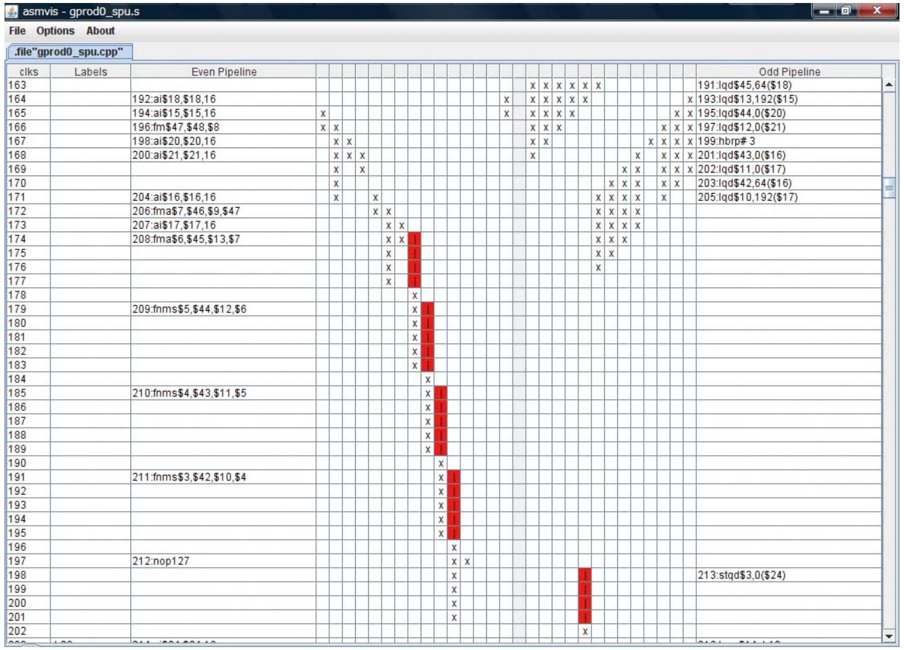
HIROSHIMA UNIVERSITY

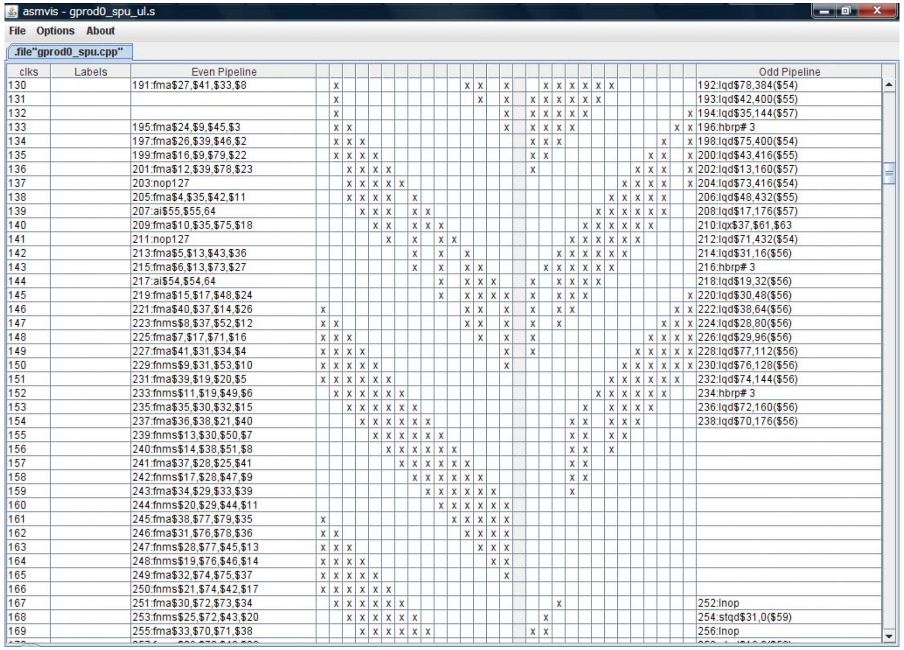


When we used only the PPE, it took 0.3GFlops
but, it was 12GFlops, we used scalar calculation with 16SPE.
Furthermore, we used SIMD 16SPEs it took about 21GFlops.

Optimization(2) **Loop Unrolling**

HIROSHIMA UNIVERSITY

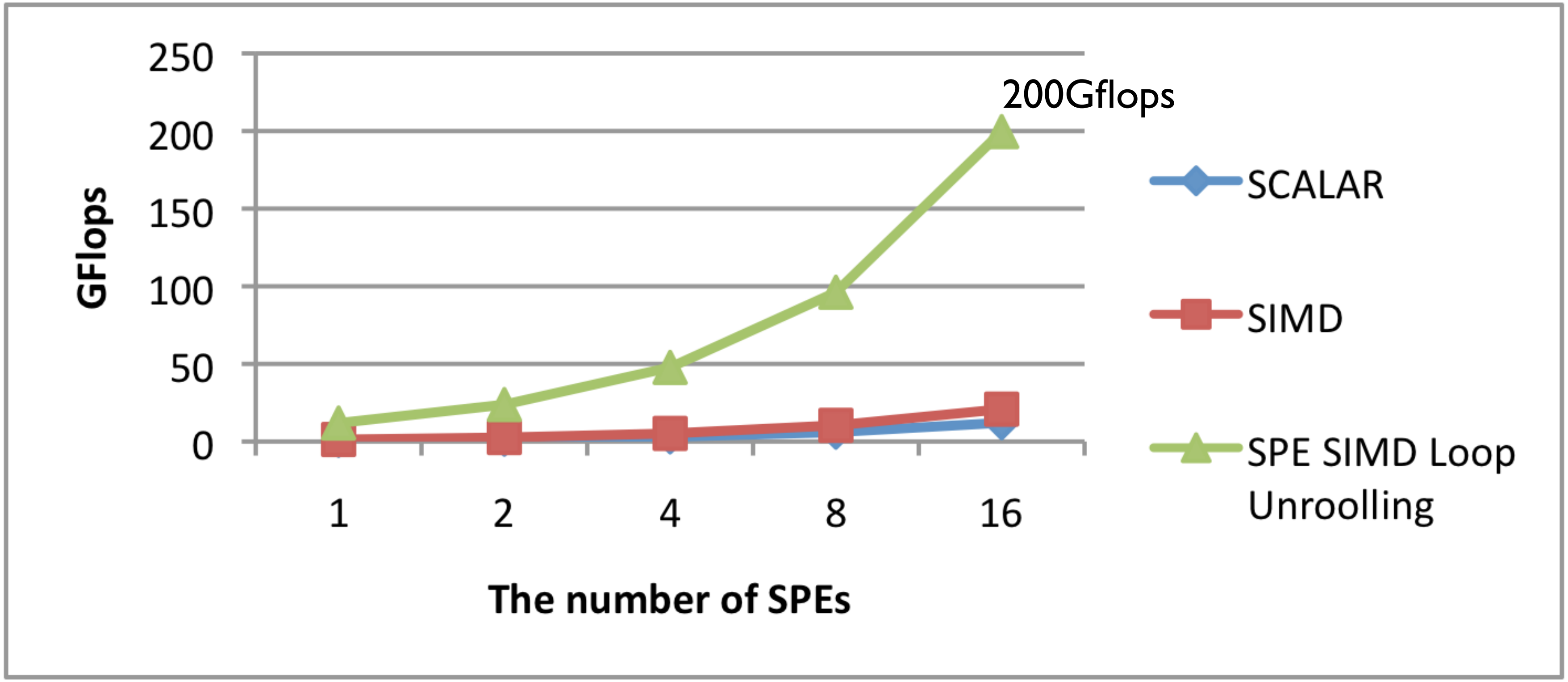




SPE has 128 general registers of 128 bit.
Thanks to the many registers, loop unrolling
and the SPE has the two pipeline and two instructions can be executed at once.
Left is before optimization, red dots is stand for resistor conflict point.
Right is after optimization, We develop a loop manual operation.
After use this Optimization, we achieve 80% theoretical peak speed

Result of Loop Unrolling

HIROSHIMA UNIVERSITY



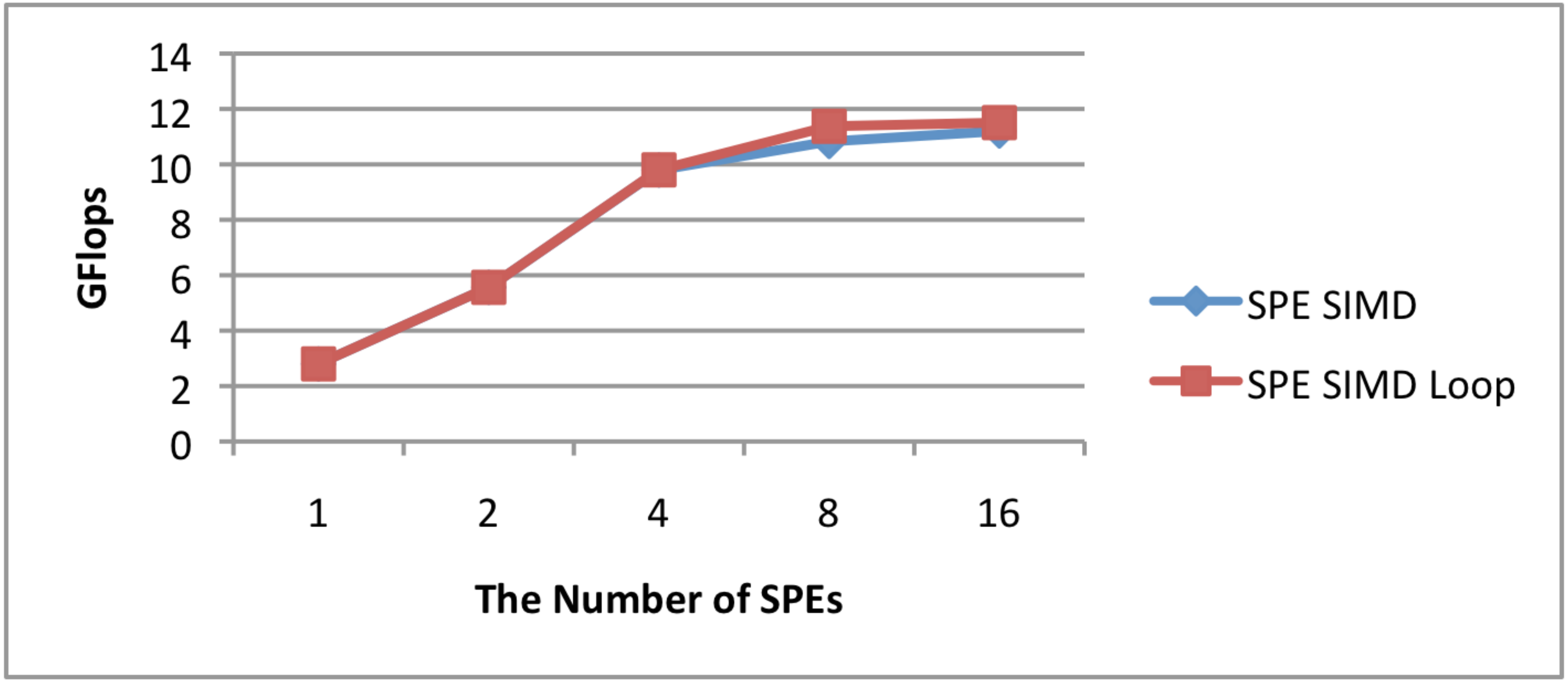
When we used 16 SPEs with the loop unrolling technique, it took 200GFlops.
After use this Optimization,　we achieve 80% theoretical peak speed.

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

What is Bottleneck

HIROSHIMA UNIVERSITY



We finally achieve 200GFlops in calculation part,
But it took becomes12GFlops together with data transfer

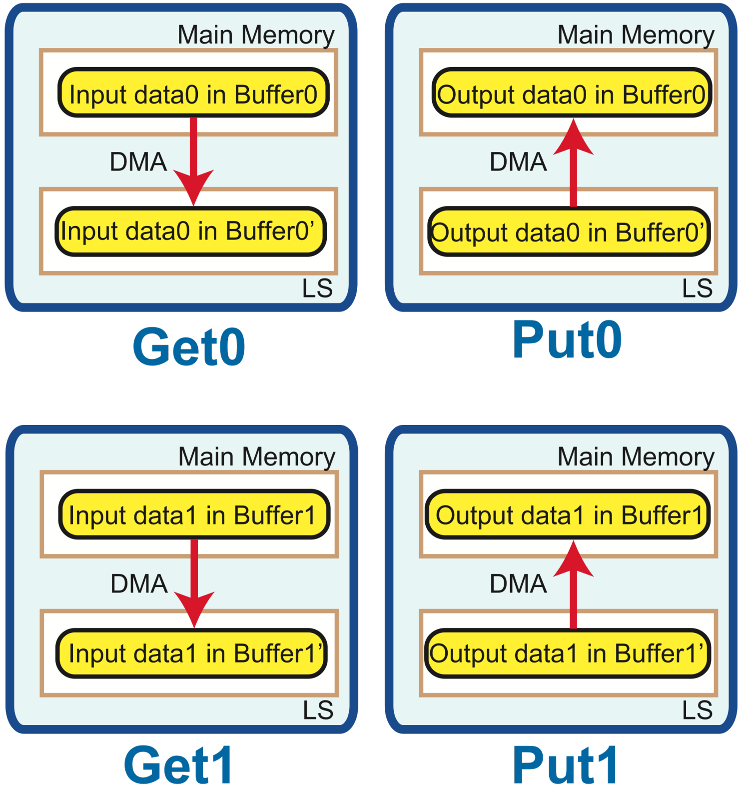
What is Bottleneck

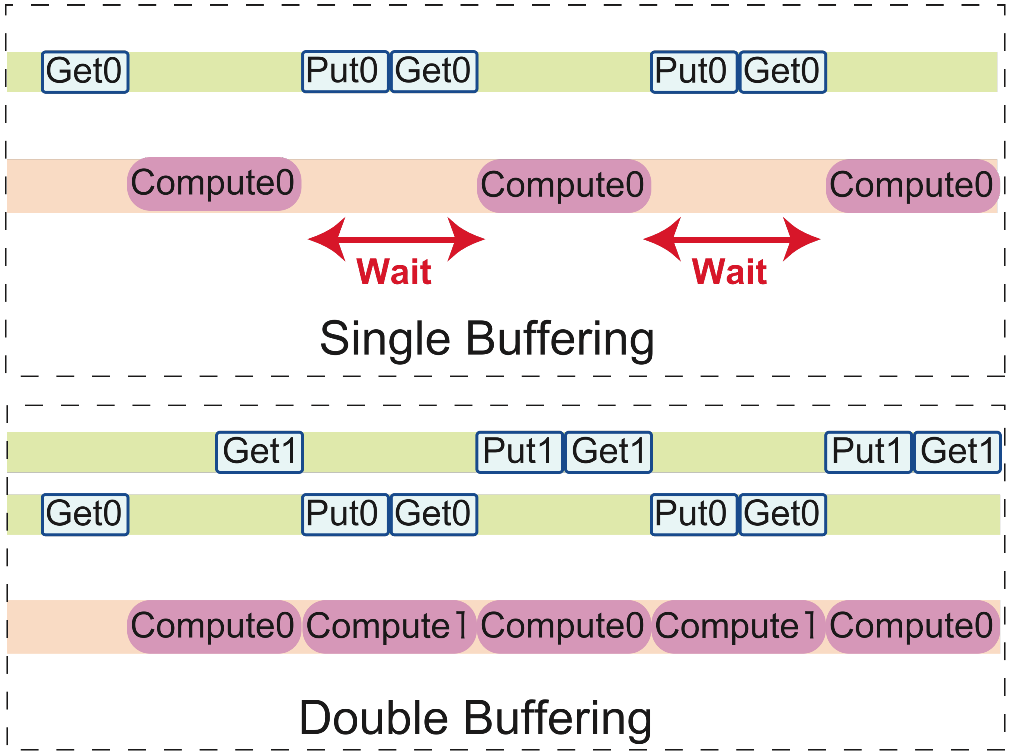
Time base frequency = 26.665 MHz
All SPEs completed successfully!
Results for: /opt/cell/sdk/usr/bin/benchmarks/dmabench --numspes 16 seqdmar

dma_size	ticks	pclocks	microsecs	aggr GB/s
8	22.2	2662	0.83	0.1539
16	22.3	2678	0.84	0.3058
32	22.4	2688	0.84	0.6093
64	22.1	2654	0.83	1.2345
128	22.3	2673	0.84	2.4509
256	22.2	2660	0.83	4.9274
512	22.4	2690	0.84	9.7446
1024	21.9	2625	0.82	19.9673
2048	34.7	4158	1.30	25.2159
4096	74.1	8898	2.78	23.5686
8192	155.5	18657	5.83	22.4805
16384	332.1	39853	12.45	21.0484

Optimization(3) Double Buffering

HIROSHIMA UNIVERSITY

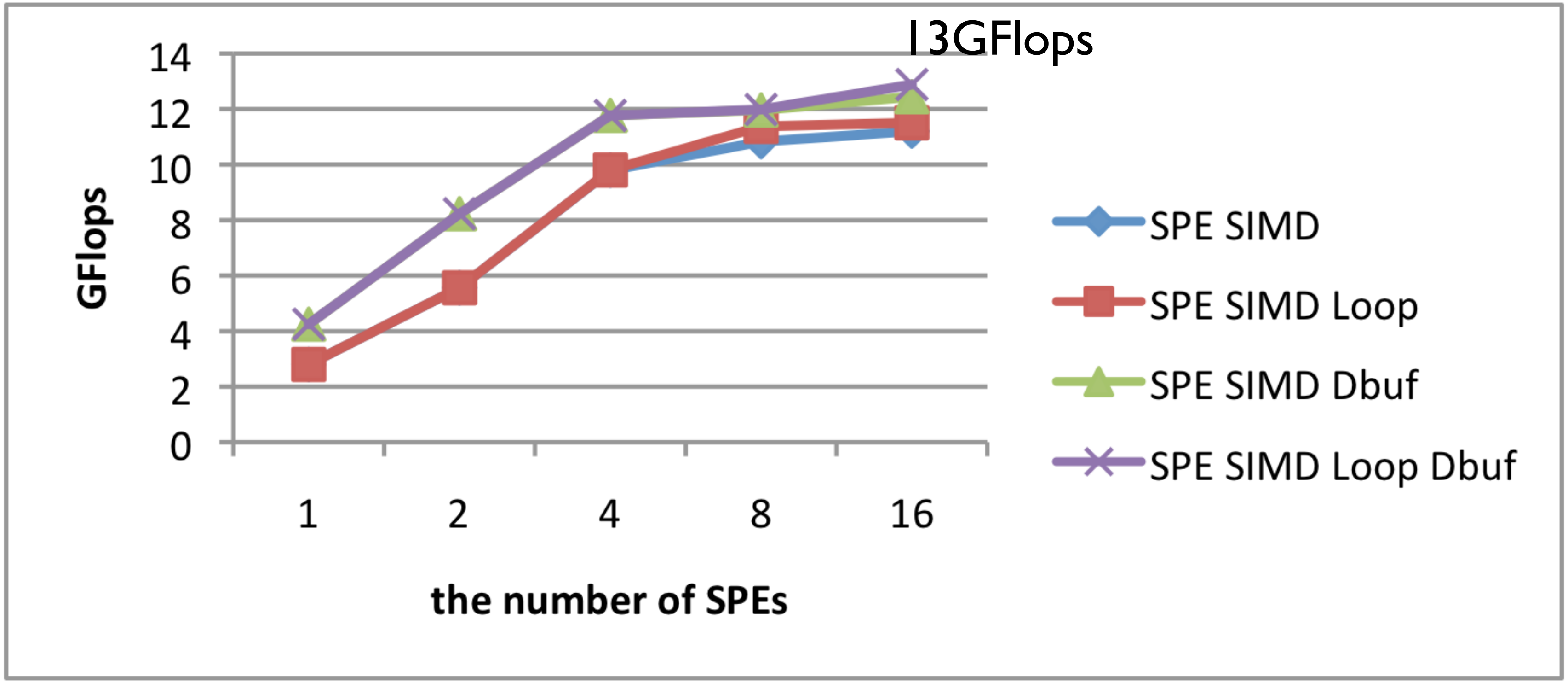




When we used without buffering, data transfer and calculation is sequential control

Double buffering is use two buffers alternately because calculation wait time reduce As possible

Result of Double Buffering



Optimization(4) Reconstruction

HIROSHIMA UNIVERSITY

Using Unitarity can reduce transfer data

$$\begin{pmatrix} v_1 & w_1 & z_1 \\ v_2 & w_2 & z_2 \\ v_3 & w_3 & z_3 \end{pmatrix} \Rightarrow \begin{pmatrix} v_1 & w_1 \\ v_2 & w_2 \\ v_3 & w_3 \end{pmatrix}, \quad z = (v \times w)^*$$

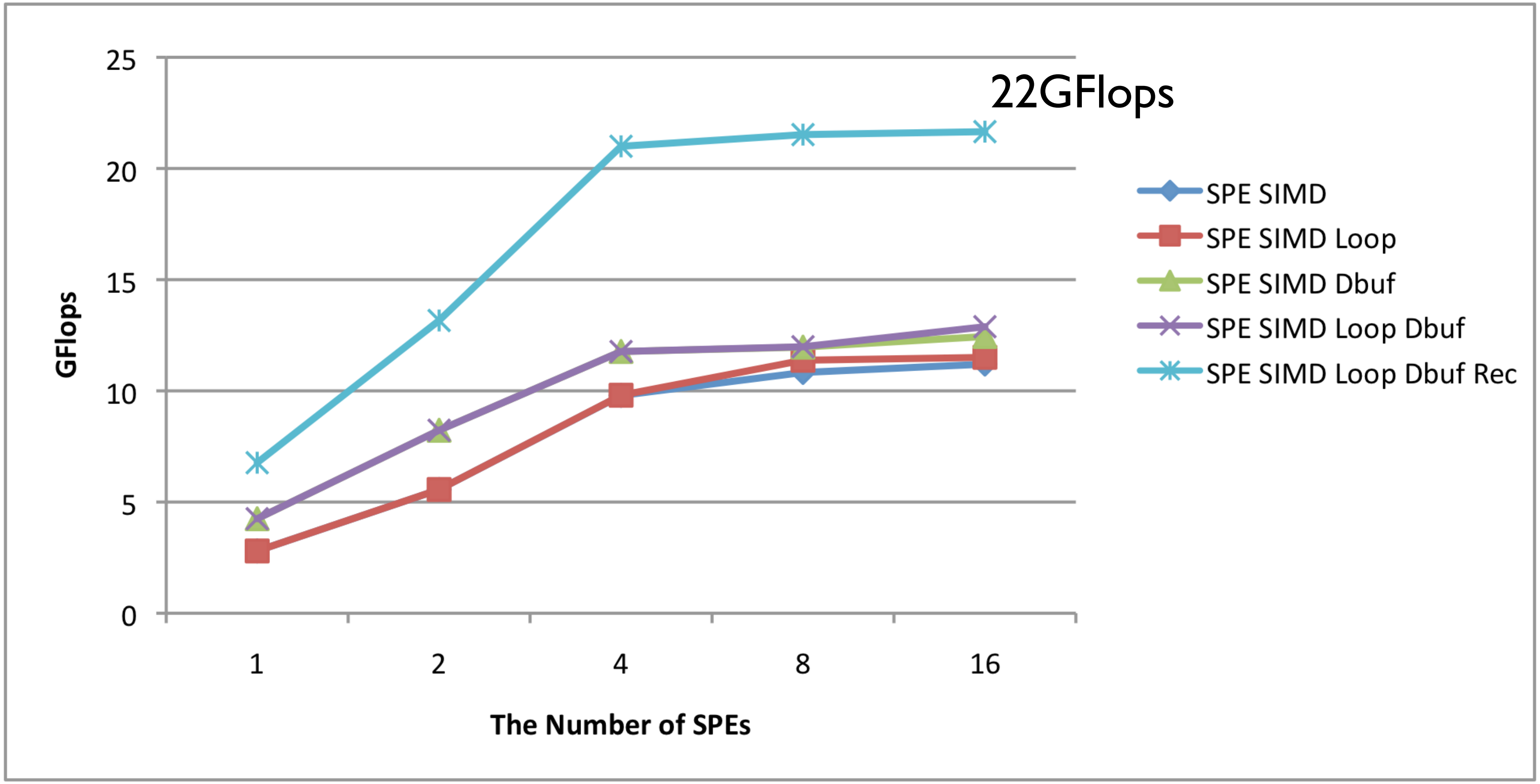
This techniques can reduce 30% of transfer data
and reconstruction calculation is on the fly

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Optimization(4) Reconstruction

HIROSHIMA UNIVERSITY



Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

Summary and future plans

HIROSHIMA UNIVERSITY

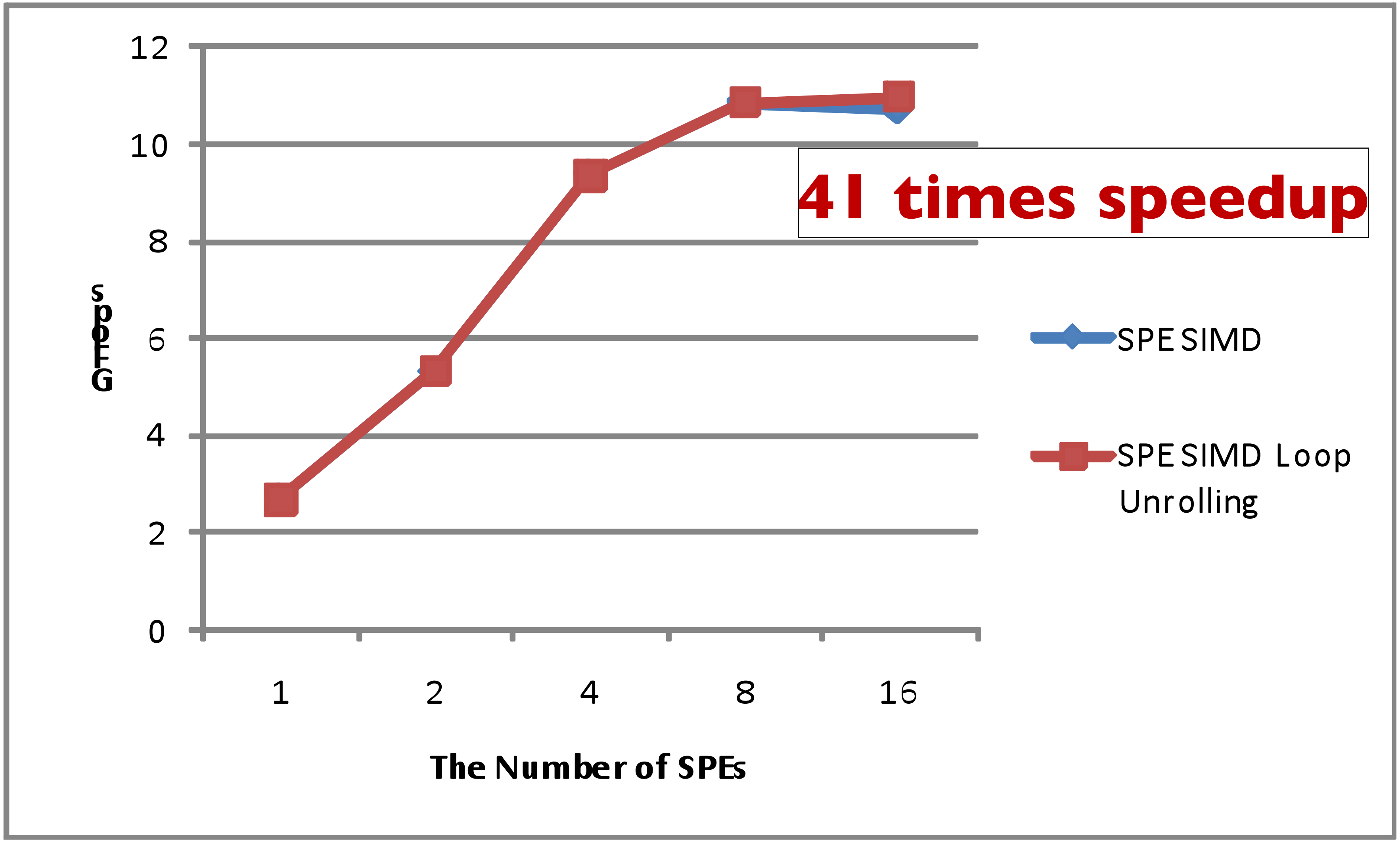
- We show large number of SU(3)matrix multiplications on Cell/ B.E.(200GFlops in calculation part)
- Calculation optimize is achieve limit
- In Result 70 times speed up (with transfer data)
- But 22GFLOPS yet.(only 9% of theoretical peak speed in this case)
- DMA data transfer is bottleneck
- We need reduce data transportation and we think how keep data long time at LS as possible

That's all
Thank you

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Result of Loop Unrolling



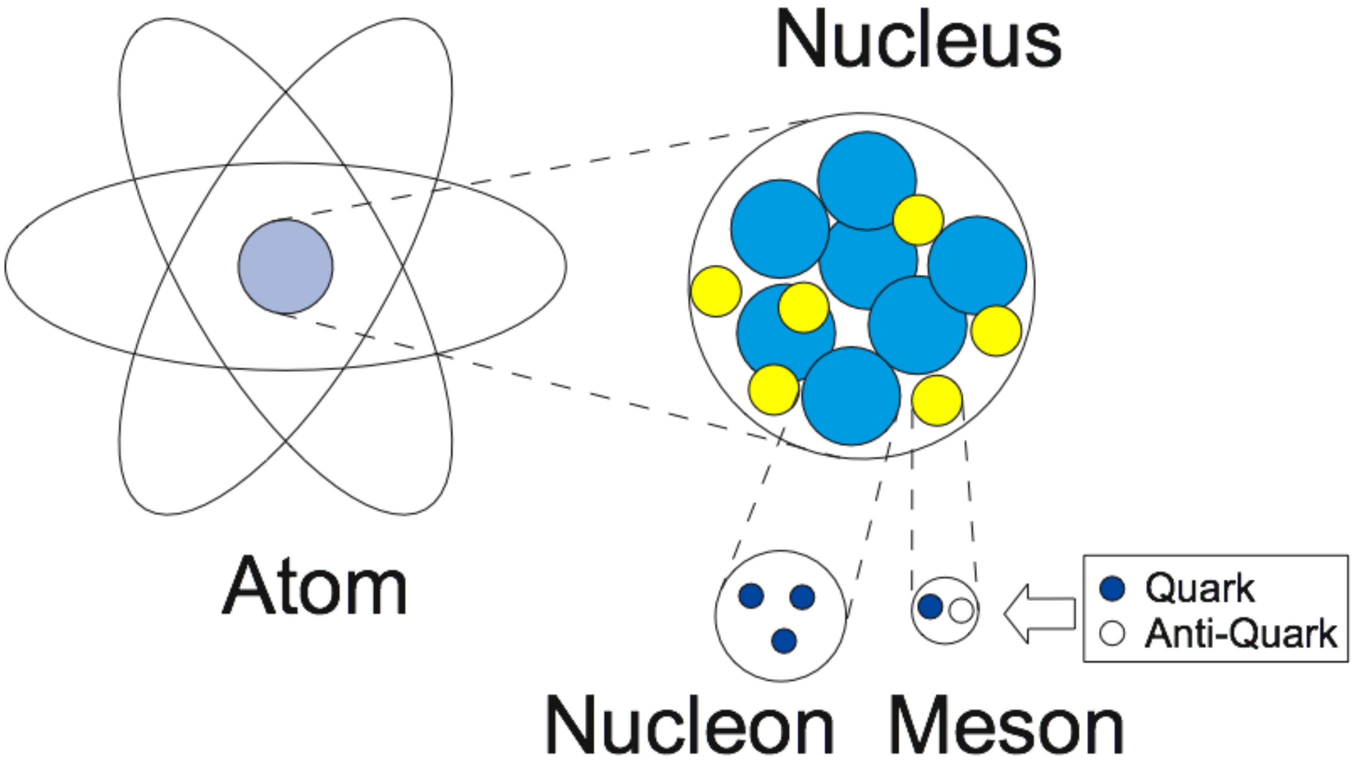
<table>
<tr><td>Byte 0</td><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td><td>Byte 7</td><td>Byte 8</td><td>Byte 9</td><td>Byte 10</td><td>Byte 11</td><td>Byte 12</td><td>Byte 13</td><td>Byte 14</td><td>Byte 15</td></tr>
<tr><td>char [0]</td><td>char [1]</td><td>char [2]</td><td>char [3]</td><td>char [4]</td><td>char [5]</td><td>char [6]</td><td>char [7]</td><td>char [8]</td><td>char [9]</td><td>char [10]</td><td>char [11]</td><td>char [12]</td><td>char [13]</td><td>char [14]</td><td>char [15]</td></tr>
<tr><td colspan="2">halfword [0]</td><td colspan="2">halfword [1]</td><td colspan="2">halfword [2]</td><td colspan="2">halfword [3]</td><td colspan="2">halfword [4]</td><td colspan="2">halfword [5]</td><td colspan="2">halfword [6]</td><td colspan="2">halfword [7]</td></tr>
<tr><td colspan="4">word [0]</td><td colspan="4">word [1]</td><td colspan="4">word [2]</td><td colspan="4">word [3]</td></tr>
<tr><td colspan="8">doubleword [0]</td><td colspan="8">doubleword [1]</td></tr>
</table>

(MSB) (LSB)

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

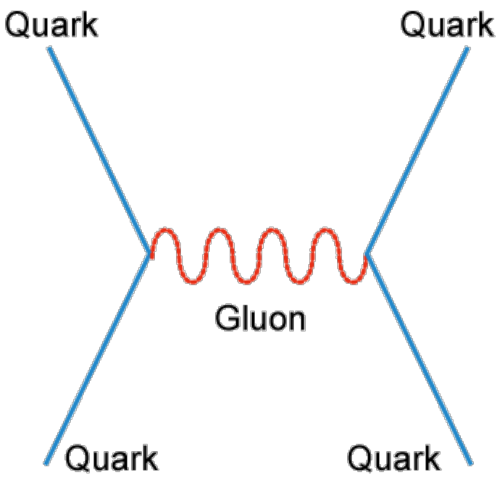
HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

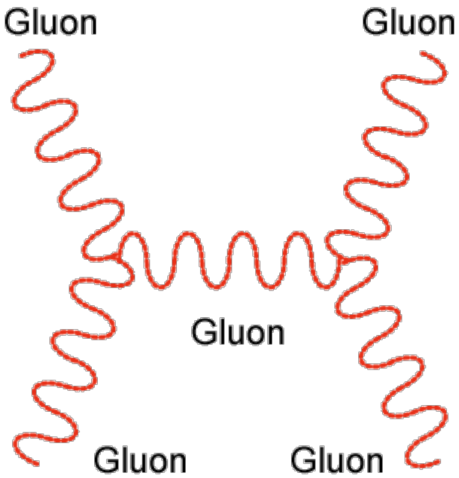
QCD
=Quantum ChromoDynamics
=Dynamics of Quarks and Gluons



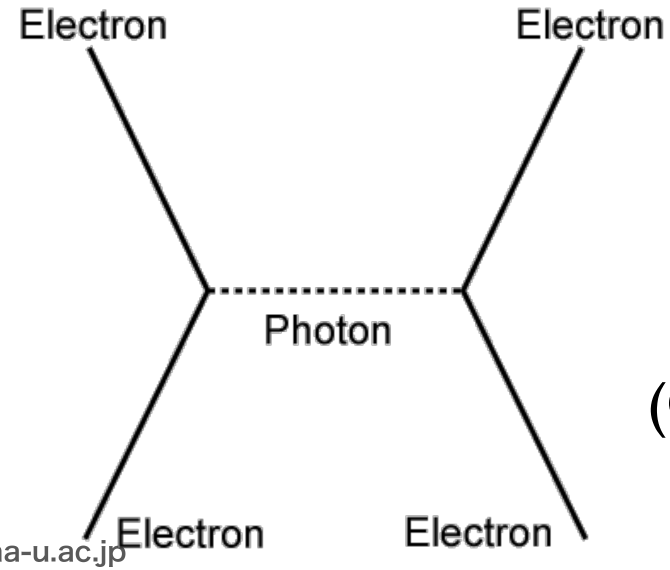
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

QCD Interactins





QCD



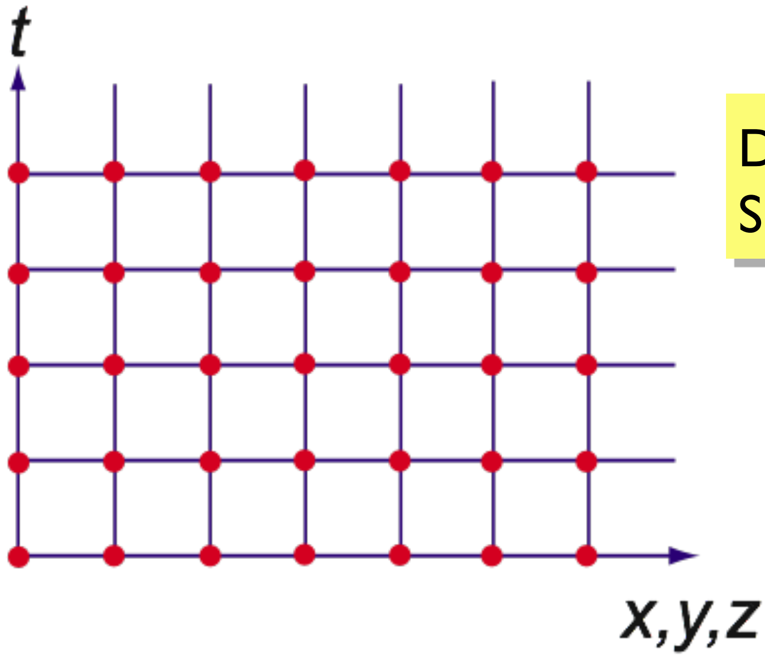
QED

(Quantum Electro-Dynamics)

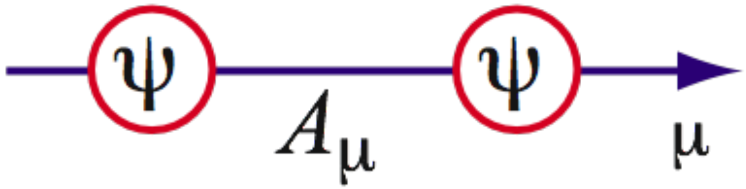
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Lattice QCD is a Field Theory of Space-Time



Discretize the Space-Time

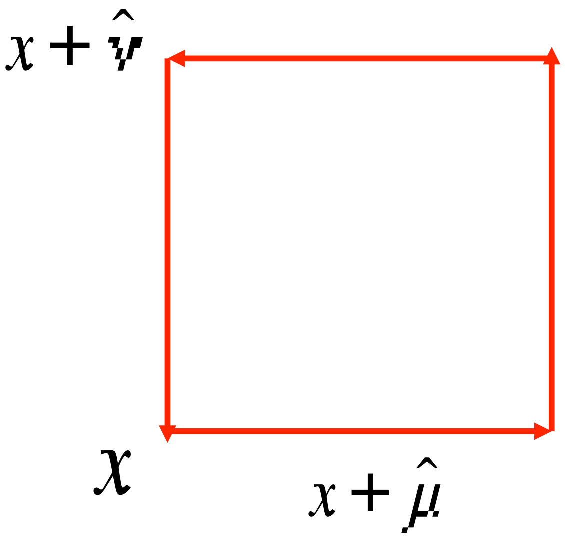


Quark on a Site
Gluon on a Link

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

Lattice QCD (1) plaquettes

$$\sum_{x,\mu} U_\mu(x) U_\nu(x+\hat{\mu}) U_\mu^{+}(x+\hat{\nu}) U_\nu^{+}(x)$$



U_μ :SU(3) Matrix

$$\begin{pmatrix} v_1 & w_1 & z_1 \\ v_2 & w_2 & z_2 \\ v_3 & w_3 & z_3 \end{pmatrix}$$

Lattice QCD(2) quark propagator

HIROSHIMA UNIVERSITY

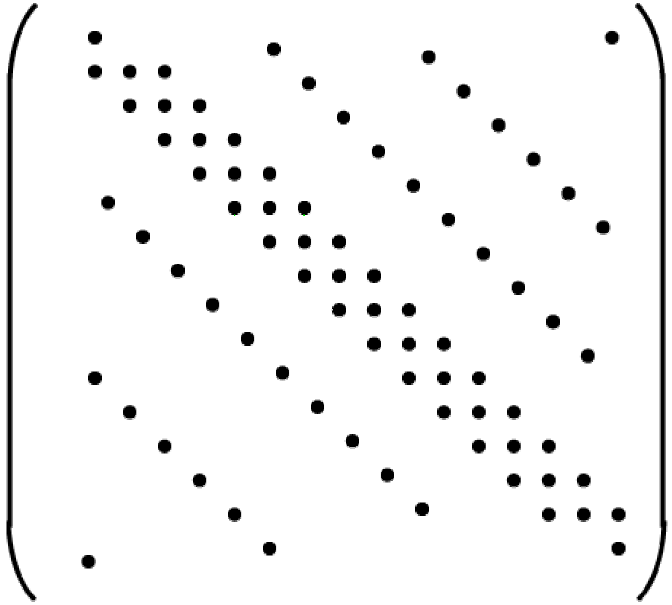
$$S_{quark} = \overline{\psi} W \psi$$

W^{-1} : quark propagator

$W\vec{X} = \vec{B}$ Solved by CG method

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

Quark Matrix *W*: Sparse, but Large



A typical Example

- 12x12 Block Matrix

9-Blocks in one Column

$$n = 3 \times 4 \times N_x \times N_y \times N_z \times N_t$$

No. of Non-Zero Elements:

$108 \times n$ (Standard case)

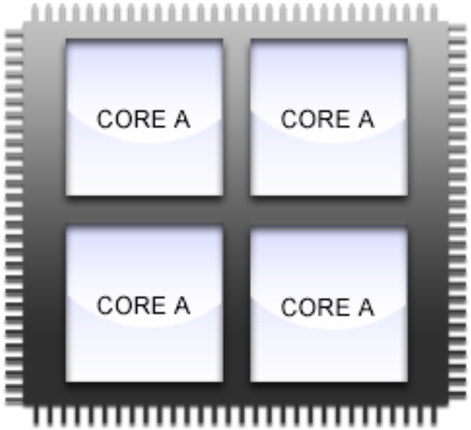
$$n = 3 \times 4 \times 32 \times 32 \times 32 \times 32 \approx 1.26 \times 10^7$$

Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

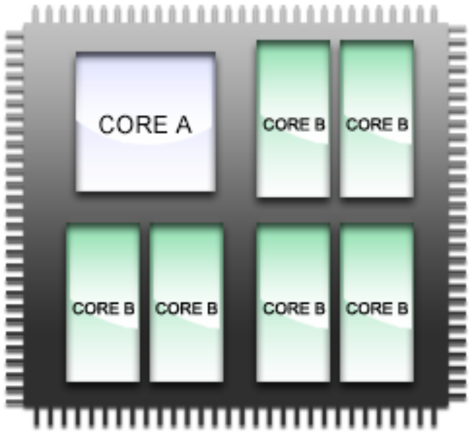
HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Multi-Core

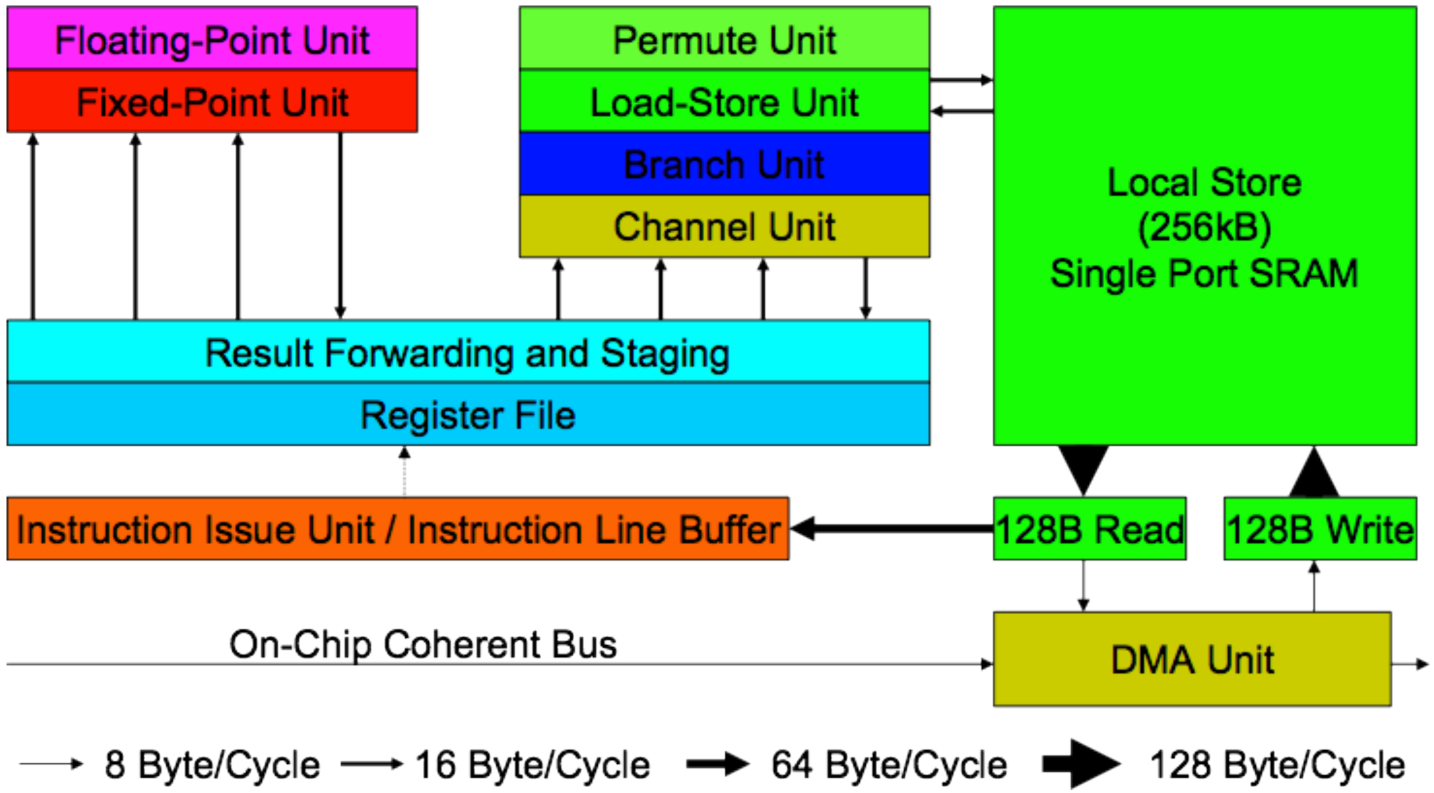
Homogeneous multi core processor



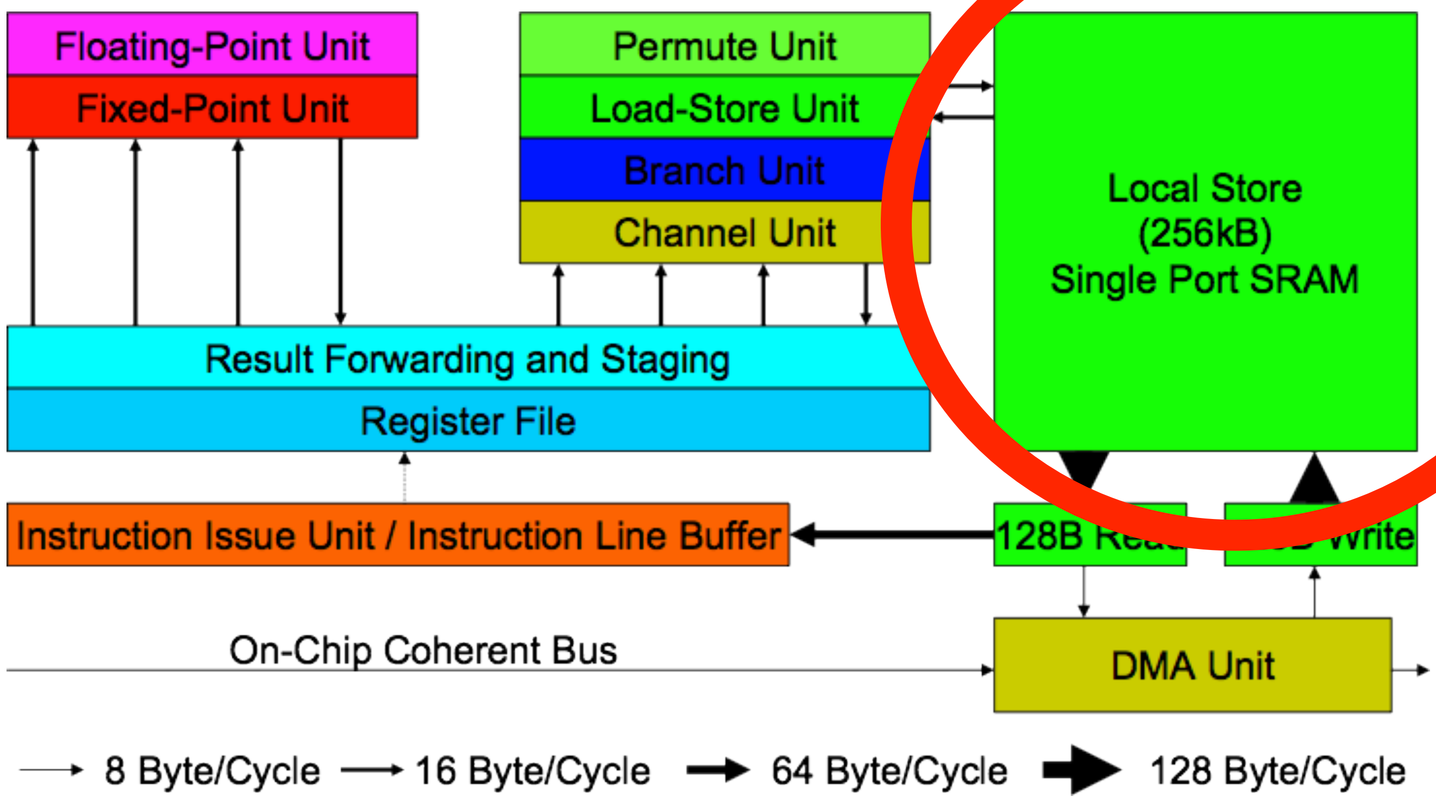
Heterogeneous multi core processor



SPE Block Diagram



SPE Block Diagram



Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

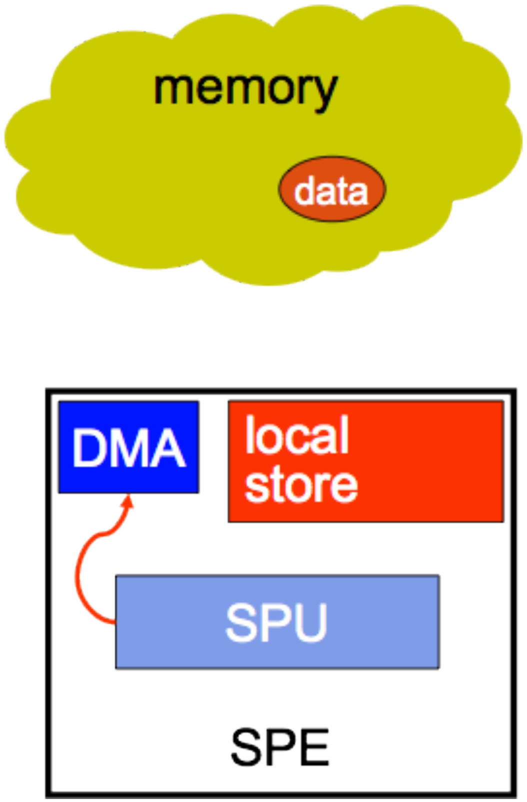
HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Data in and Out of the SPE LS

HIROSHIMA UNIVERSITY

- SPU needs data

1. SPU initiates DMA request for data



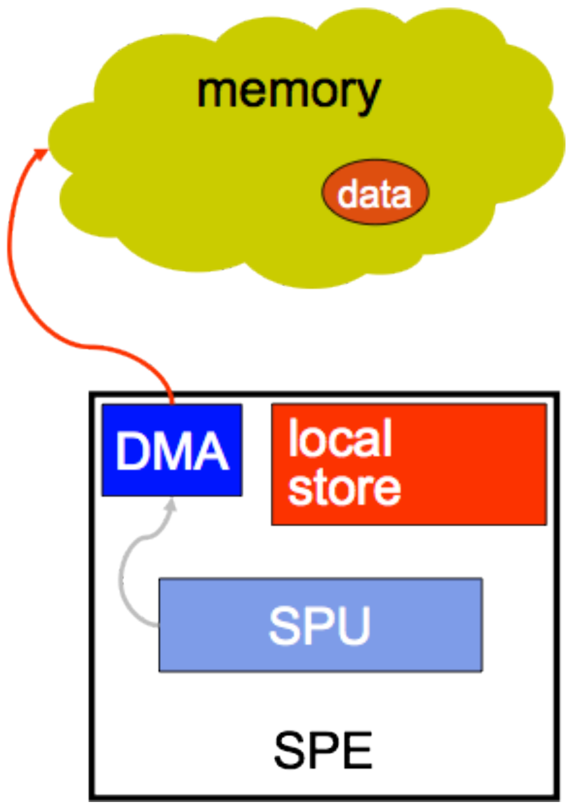
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Data in and Out of the SPE LS

- SPU needs data

1. SPU initiates DMA request for data
2. DMA requests data from memory



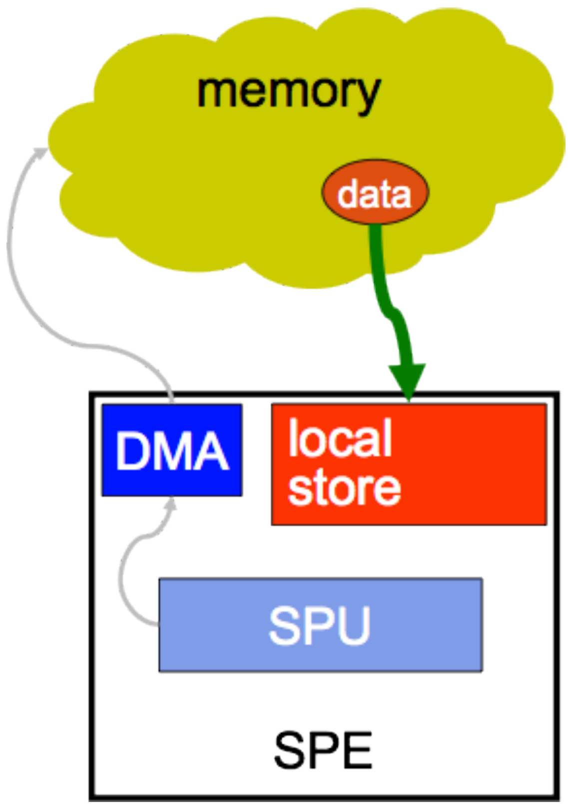
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Data in and Out of the SPE LS

- SPU needs data

1. SPU initiates DMA request for data
2. DMA requests data from memory
3. Data is **copied** to local store



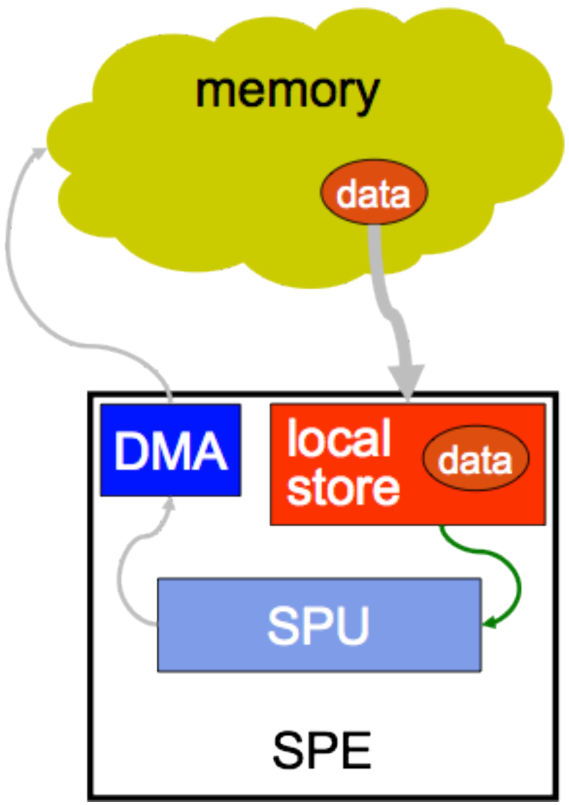
Shinji MOTOKI
motoki-shinji@hiroshima-u.ac.jp

HIROSHIMA UNIVERSITY
Graduate School of Biosphere Science

Data in and Out of the SPE LS

- SPU needs data

1. SPU initiates DMA request for data
2. DMA requests data from memory
3. Data is copied to local store
4. SPU can access data from local store



Data in and Out of the SPE LS

- SPU needs data

1. SPU initiates DMA request for data
2. DMA requests data from memory
3. Data is copied to local store
4. SPU can access data from local store

- SPU operates on data then **copies** data from local store back to memory in a similar process