



Reconfigurable Out-of-Order Mechanism Generator for Unstructured Grid Computation in Computational Fluid Dynamics

Dept. of Information and Computer Science, Keio Univ.

Takayuki Akamine, Kenta Inakagata, Hideharu Amano

Dept. of Electrical and Electronics Engineering, Ryukyu Univ.

Yasunori Osana

Aerospace Research and Development Directorate

Japan Aerospace Exploration Agency

Naoyuki Fujita





Agenda

1. Motivation

- CFD and FaSTAR,
- and Unstructured Grid

2. RAW hazards

3. Out-of-Order Mechanism

4. Evaluation

5. Conclusion



Background

- CFD(Computational Fluid Dynamics)
 - Computer simulation for analyzing fluid behavior
 - Design Tool for aircraft components such as engines and bodies
 - Huge amount of computation
- FaSTAR(FAST Aerodynamic Routines)
 - A CFD package program developed by JAXA
 - Adopts an unstructured grid for grid form
 - Reaches a limit of acceleration by software parallel execution
- FPGA acceleration
 - Highly flexible to fit an algorithm
 - A promising approach for accelerating FaSTAR

Related Work

- Kentaro Sano, et al. [FCCM2007]
 - A Systolic Architecture for Computational Fluid Dynamics on FPGAs
 - Implementation of the fractional method
- Hirokazu Morishita, et al. [FPT2008]
 - Exploiting Memory Hierarchy for a Computational Fluid Dynamics Accelerator on FPGAs
 - Implementation of MUSCLE (Monotone Upstream-centered Schemes for Conservation Laws) in UPACS
- Diego Sanchez-Roman, et al. [SPL2011]
 - An Euler solver accelerator in FPGA for computational fluid dynamics applications
 - Implementation of an Euler solver using unstructured mesh with Impulse C

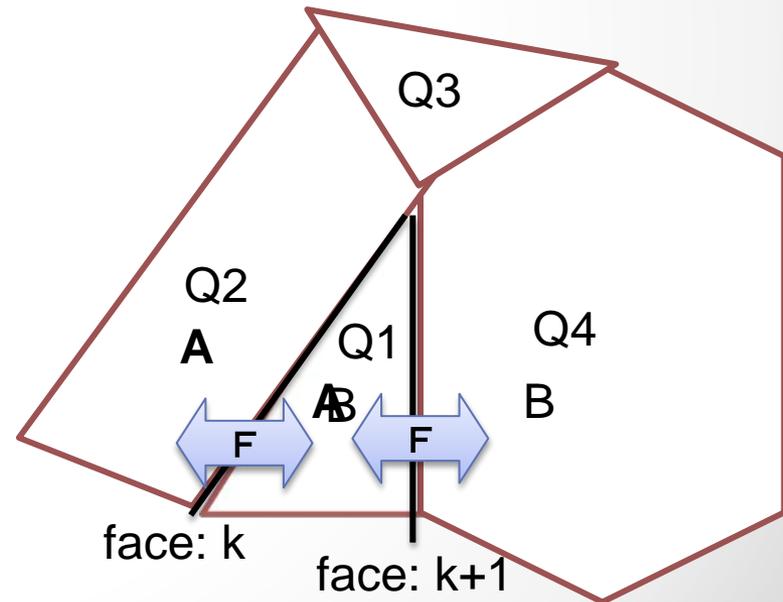
In this work,

1. Our target is a CFD package using unstructured grid
2. In order to avoid the overhead of high-level synthesis, a HDL generator will be proposed

Unstructured Grid

- Unstructured Grid
 - It is good at expressing complicated shapes
 - Grid data are generated automatically
- How to access unstructured grid
 - Accessing a face, two contiguous grids sharing the face (cell-A, cell-B) are accessed
 - Faces are accessed in ascending order
 - Spatial locality of variables
 - Each grid has different number of faces
 - Irregular memory access

In FaSTAR, grid data has data locality,
but memory access has no regularity





Agenda

1. Motivation

- CFD and FaSTAR,
- and Unstructured Grid

2. RAW hazards

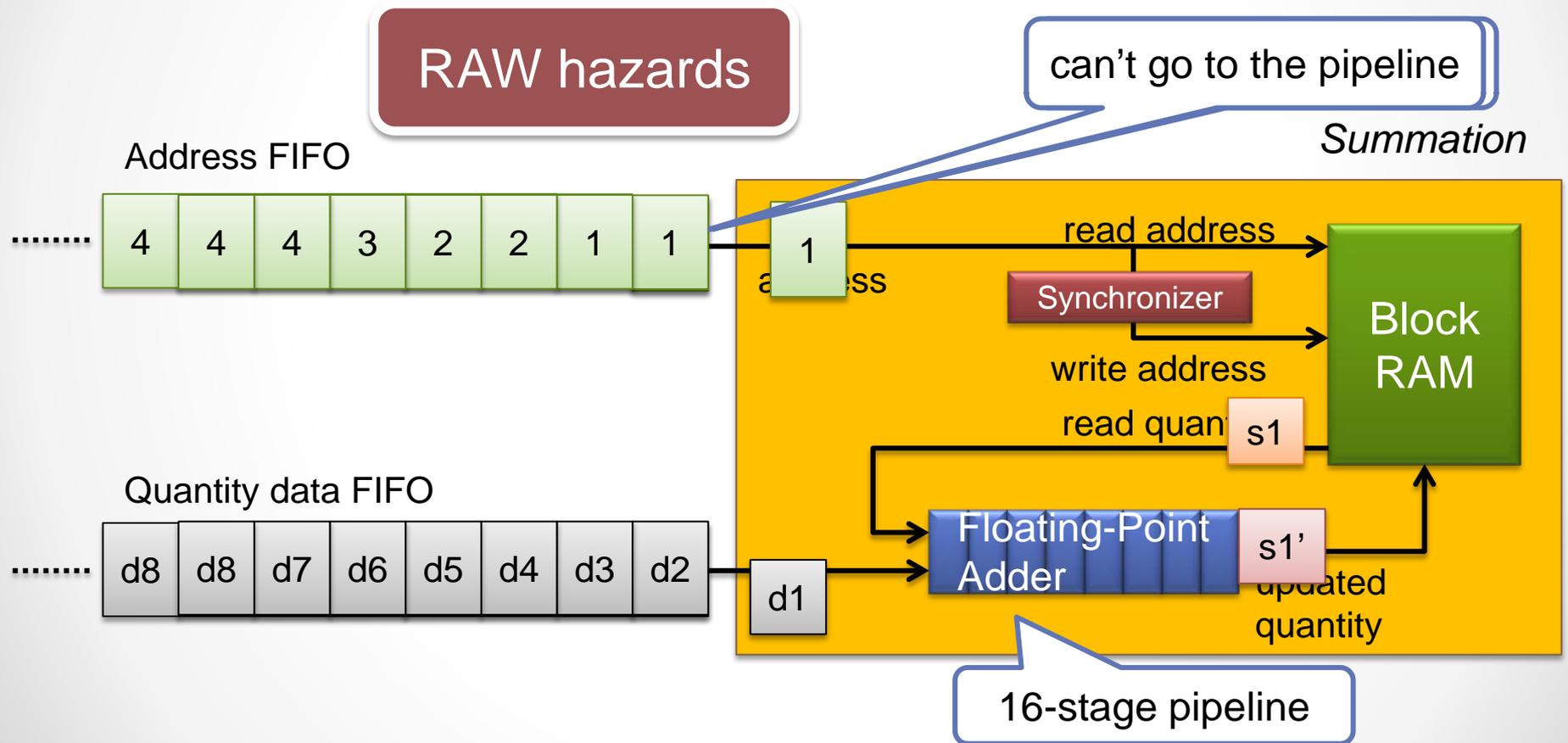
3. Out-of-Order Mechanism

4. Evaluation

5. Conclusion

RAW hazards

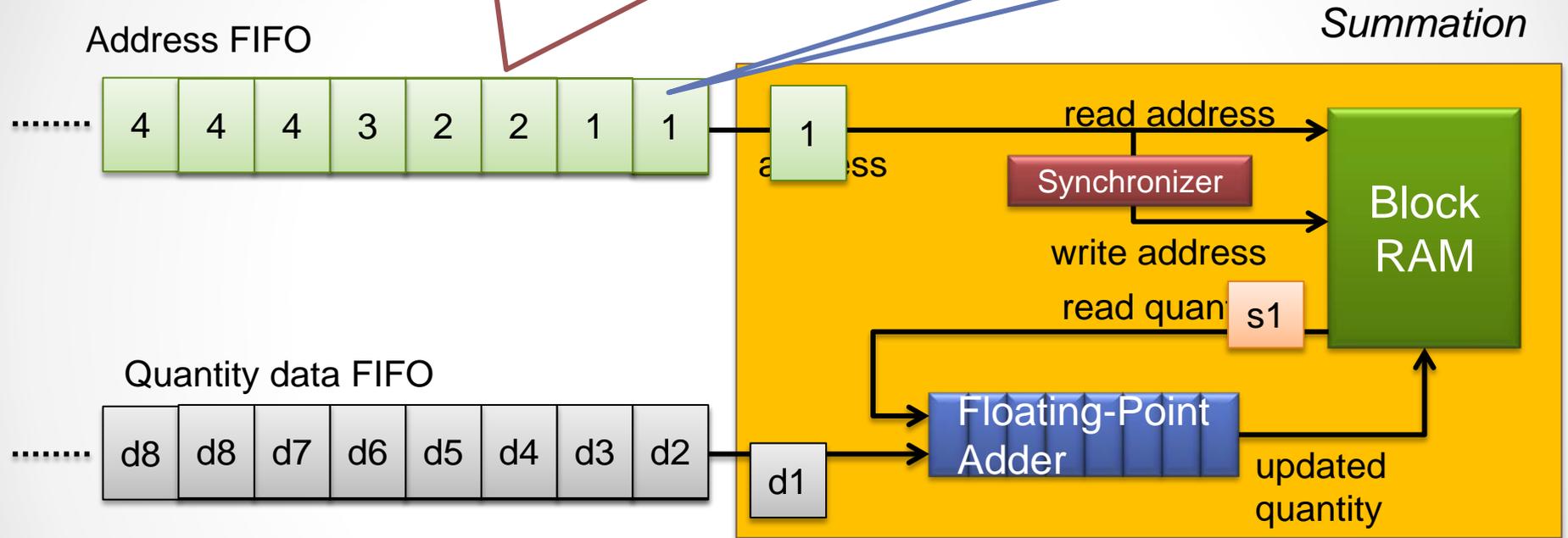
- RAW hazard is critical problem in FaSTAR



Avoiding RAW hazards

Not related to "s1",
can go to the pipeline

can't go to the
pipeline



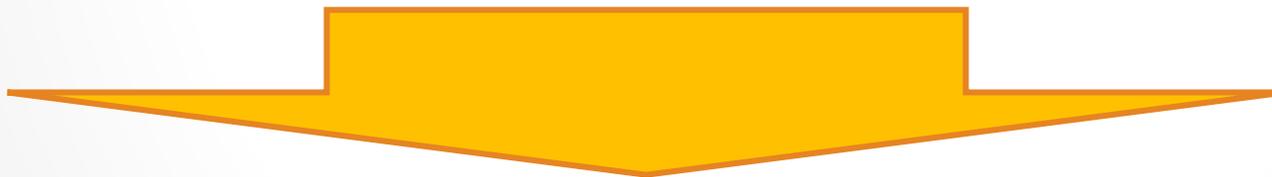
- RAW hazards would be avoided by changing the order of data
- Data should be reordered dynamically

Goal of this work

- Reducing hazards in 5 solvers
 1. Surface integral of flux(*Surface*)
 2. Green Gauss
 3. Summation of fluxes' eigenvalue(*Sum Eigen*)
 4. Maximum of fluxes' eigenvalue(*Max Eigen*)
 5. Coefficient Initialization(*Coefficient*)

All suffer from RAW hazards

- An out-of-order(OoO) mechanism



- An OoO mechanism generator



Agenda

1. Motivation

- CFD and FaSTAR,
- and Unstructured Grid

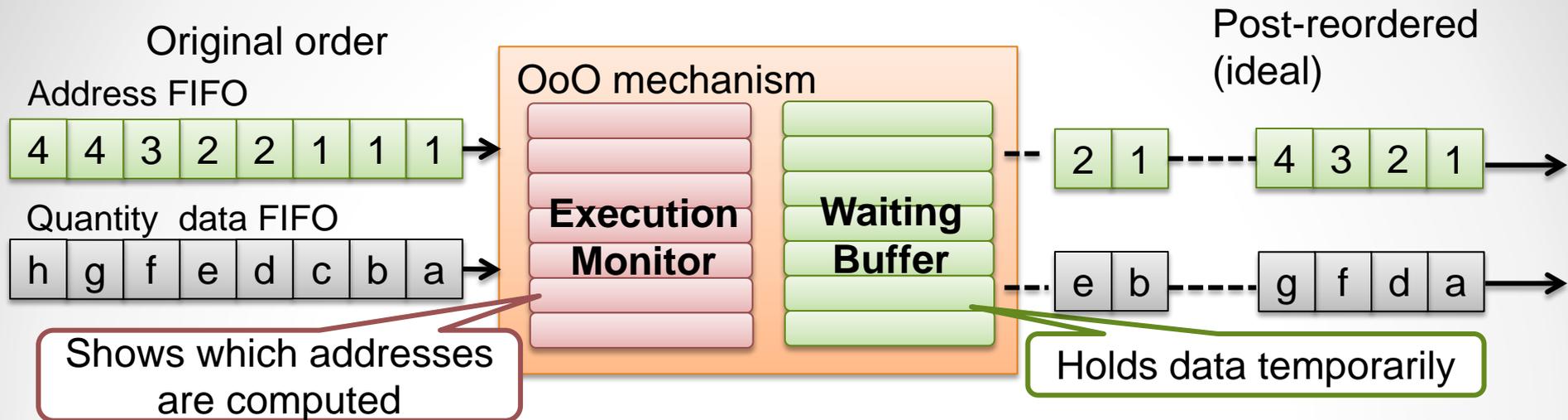
2. RAW hazard

3. Out-of-Order Mechanism

4. Evaluation

5. Conclusion

Overview



- Changing the order of data dynamically so as to resolve data dependencies
- Unlike common OoO mechanisms in general purpose CPUs†, the mechanism is devoted to single arithmetic unit

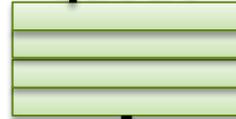
†“ *An Efficient Algorithm for Exploiting Multiple Arithmetic Units*”,
Robert Tomasulo, IBM Journal of Research and Development(1967)

Architecture

Assumption

- 4 stage-pipeline arithmetic
- Two-dimensional vector quantity

Input queue



4-entry execution monitor

Pointer decides which entry would be updated

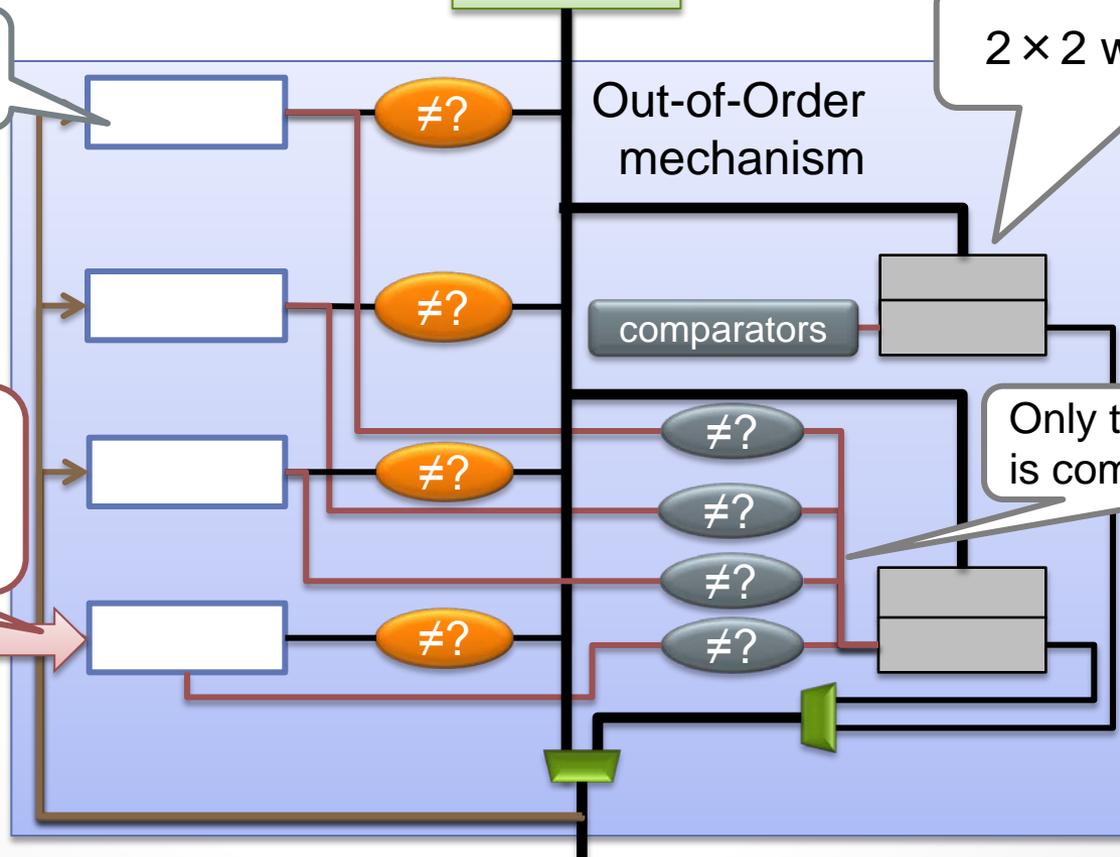
Out-of-Order mechanism

2 × 2 waiting buffer

comparators

Only the first element is compared

Arithmetic Unit



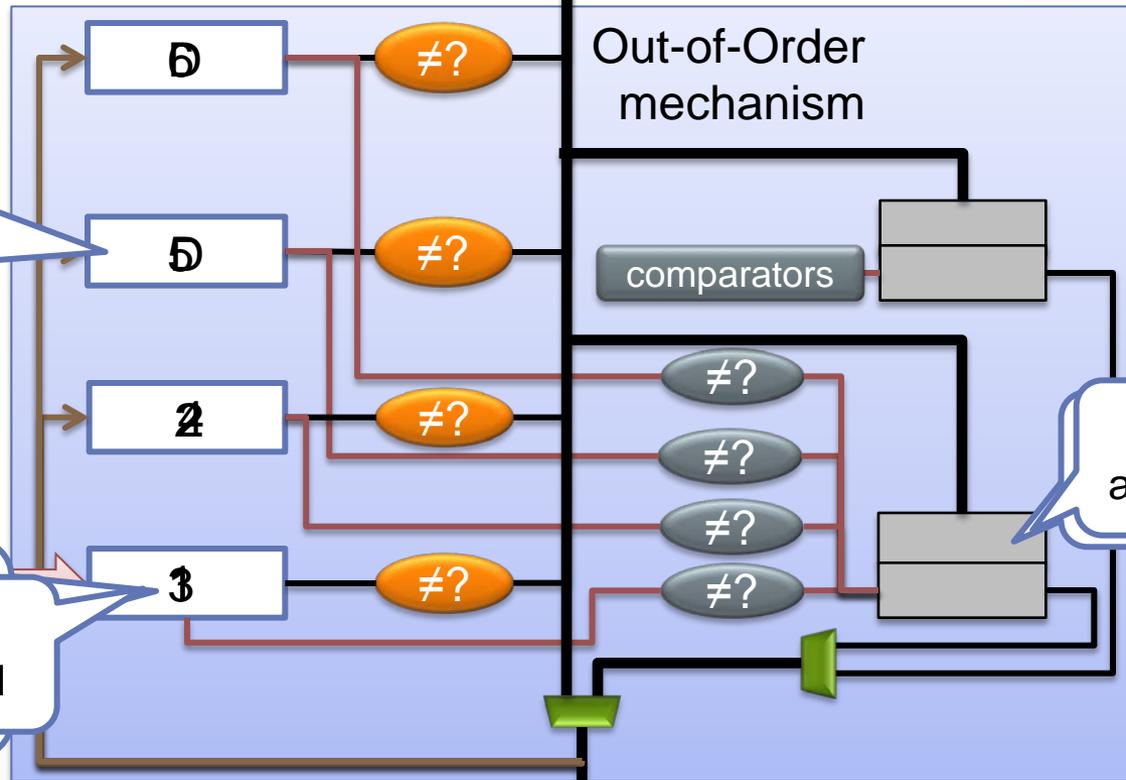
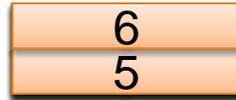
Behavior

Assumption

- 4 stage-pipeline arithmetic
- Two-dimensional vector quantity

input
queue

10 clock



same address
→ incoming data
have to wait

Updated
→ Processing finished

Go to the
arithmetic unit

Arithmetic
Unit



Parameters

- Parameterization supports us to generate various types of OoO mechanism
- *latency* is set as same number of pipeline depth in an arithmetic unit
 - *latency* decides the number of entries in an execution monitor
- *dimension* is quantity vector dimension
- *set* is the parameter decided by a designer
 - “*set* × *dimension*” equals to the number of waiting buffer



Agenda

1. Motivation

- CFD and FaSTAR,
- and Unstructured Grid

2. RAW hazards

3. Out-of-Order Mechanism

4. Evaluation

5. Conclusion

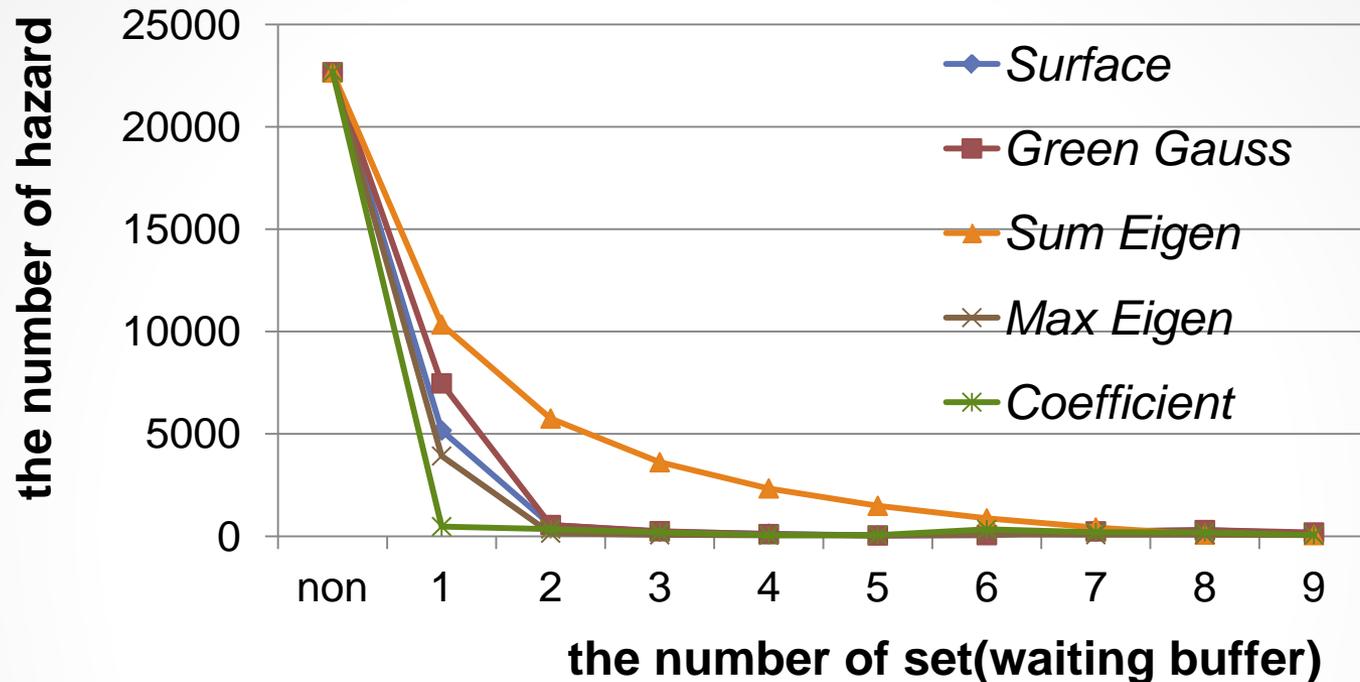
Evaluation Environment

- ▶ FPGA : Xilinx Virtex-4 XC4VLX100
- ▶ Double-precision floating-point units based on IEEE754
- ▶ Floating point units, BlockRAMs and FIFOs are generated by Xilinx CORE Generator
- ▶ Synthesis : Xilinx ISE 13.2
- ▶ Test dataset : grid data around 22,883 faces

Table1. Parameters on each solver

computation	operation type	latency	dimension	Iteration
Surface	Adder	16	5	1000
Green Gauss	Adder	16	4	1000
Sum Eigen	Adder	16	1	1000
Max Eigen	Comparator	3	1	1000
Coefficient	Adder	16	6	1

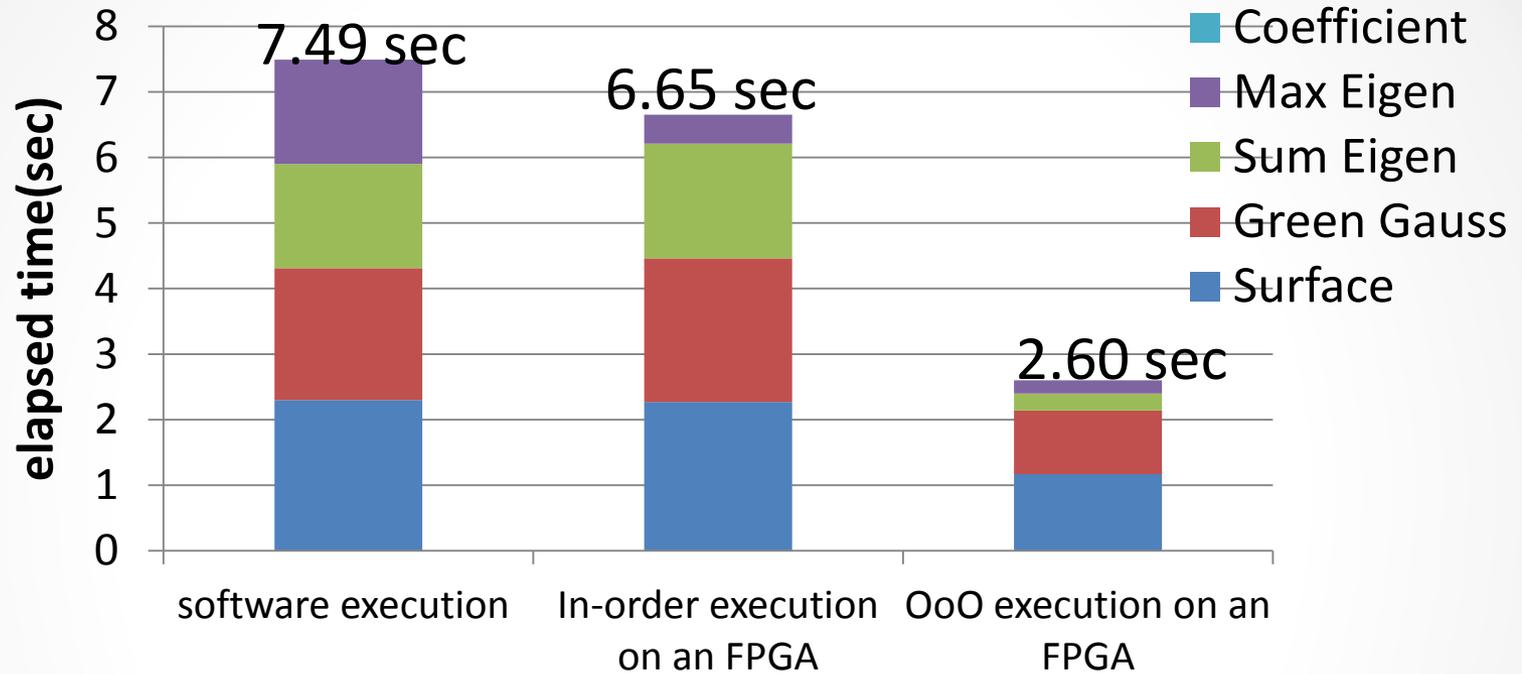
Hazards reduction



graph1 : the number of hazards with increasing set

- Increasing *set* reduced the number of RAW hazards
- In most cases, hazards were eliminated when *set* is 2
- Sum Eigen needed 7 *sets* for removal of hazards

Acceleration evaluation



- **Software execution:**

- Intel Core2Duo CPU(2.66GHz)
- Linux kernel ver2.6
- Intel Fortran Compiler 10.4

- **2.88-fold** speed-up with Intel Core2Duo CPU
- **2.55-fold** speed-up with in-order execution on an FPGA
 - 6.7-fold speed-up in Sum Eigen

Resource Overhead

Computation	Slice usage (in-order)	Slice usage (on-peak)	Increase ratio	Occupation Increase
Surface	10473	12889	23.1%	4.91%
Green Gauss	10528	12113	15.1%	3.22%
Sum Eigen	6215	9106	45.6%	5.88%
Max Eigen	3294	4149	26.0%	1.73%
Coefficient	10530	11617	24.6%	2.21%

- About 27% of resource overhead is required for peak performance on average
 - The total overhead in an FPGA is 3.5% on average
- Increase ratio of Sum Eigen is the biggest, but the impact of total resource is not so large



Agenda

1. Motivation

- CFD and FaSTAR,
- and Unstructured Grid

2. RAW hazard

3. Out-of-Order Mechanism

4. Evaluation

5. Conclusion



Conclusion

- The out-of-order mechanism is applied to five solvers in FaSTAR
 - In most cases, the hazards are eliminated when *set* is 2
 - The mechanism achieved 2.88-fold speed-up to Intel Core2Duo, and 2.55-fold speed-up to in-order execution on an FPGA
- Future work
 - Evaluation should be done with various grid data sample
 - Critical path delay should be reduced for further acceleration by using pipeline processing



Thank you.

Q&A