Custom Computing for Efficient Acceleration of HPC Kernels

Kentaro Sano

Graduate School of Information Sciences
Tohoku University, Japan

Outline

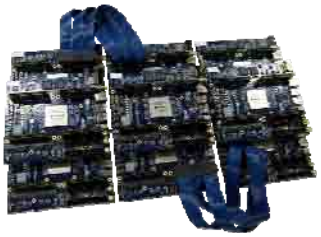
- **FPGAs are capable for HPC?**

  Yes, then how to scale them?

- **FPGA-based Stream Processor**

- **FPGA-based Real-time Data Compressor**

- **Summary**



FPGA-based stream processor

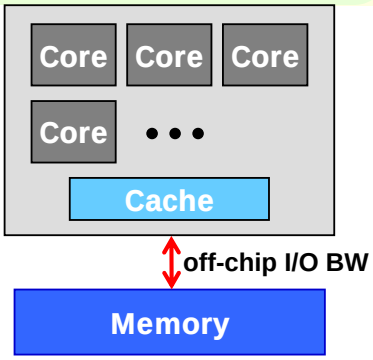
FPGA-based data-compressor

Introduction

General-purpose μ-processors

Increased cores, but
low growing rate of off-chip I/O bandwidth

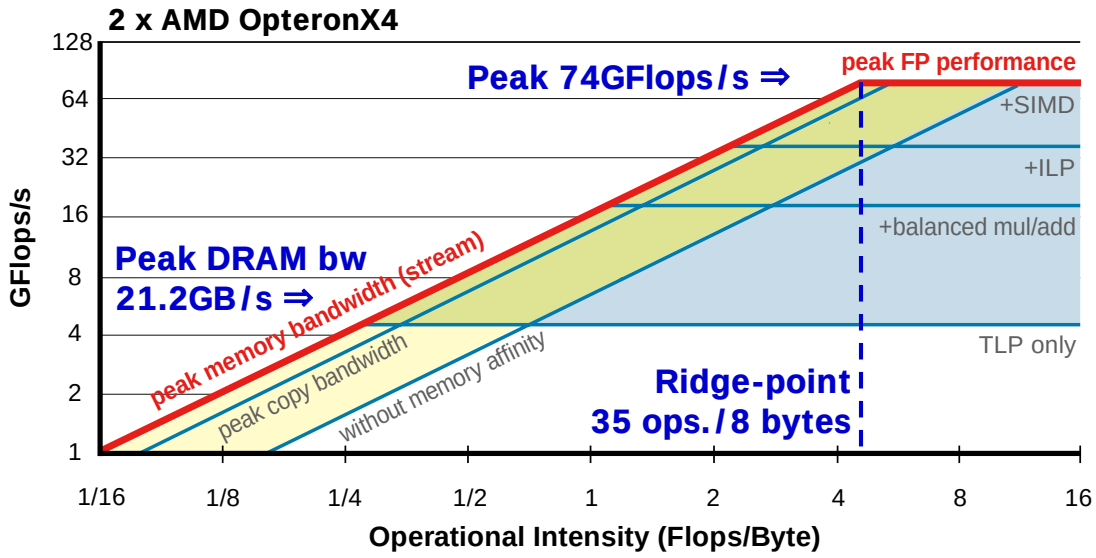
- ✓ Gap between the memory bandwidth and the peak arithmetic performance
- ✓ Only a fraction of the peak performance



Samuel Williams, Andrew Waterman and David Patterson, "Roofline: An Insightful Visual Performance Model for Multicore Architectures," Comm. of ACM, Vol.52, No.64, 65-76, 2009.

GFlops vs. Operational Intensity

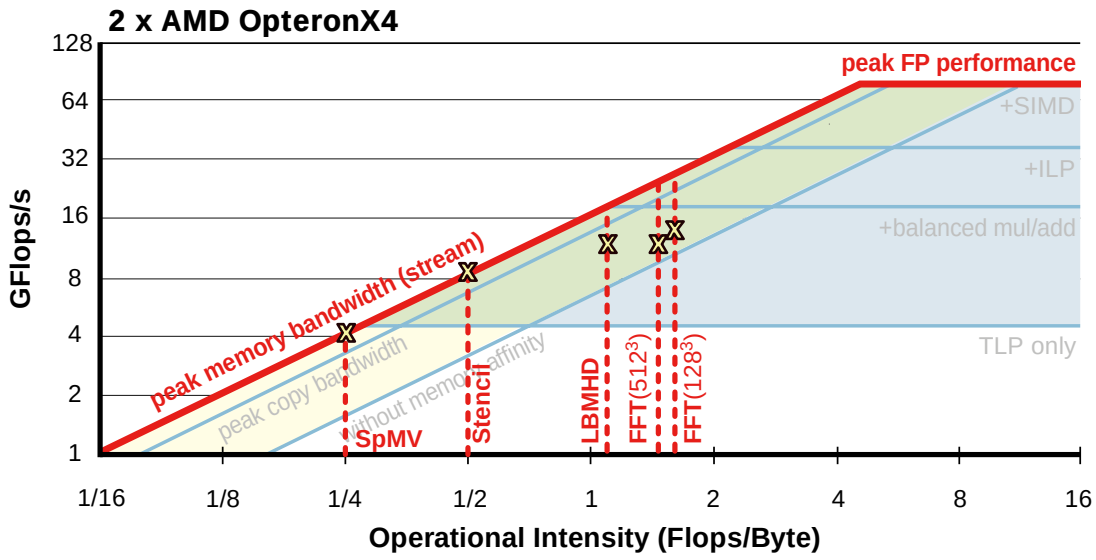
Roofline Model : Samuel Williams, Andrew Waterman and David Patterson, "Roofline: An Insightful Visual Performance Model for Multicore Architectures," Comm. of ACM, Vol.52, No.64, 65-76, 2009.



= # of floating-point ops. per DRAM access (in byte)

Performance of Applications

Roofline Model : Samuel Williams, Andrew Waterman and David Patterson, "Roofline: An Insightful Visual Performance Model for Multicore Architectures," Comm. of ACM, Vol.52, No.64, 65-76, 2009.



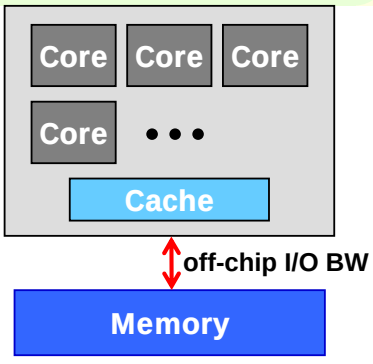
insufficient intensity : Actual performance << Peak performance

Introduction (contd.)

General-purpose μ-processors

Increased cores, but
low growing rate of off-chip I/O bandwidth

- ✓ Gap between the memory bandwidth and the peak arithmetic performance
- ✓ Only a fraction of the peak performance



Their massively-parallel systems

- ✓ Communication overhead
- ✓ Limited scalability
- ✓ Very inefficient computation

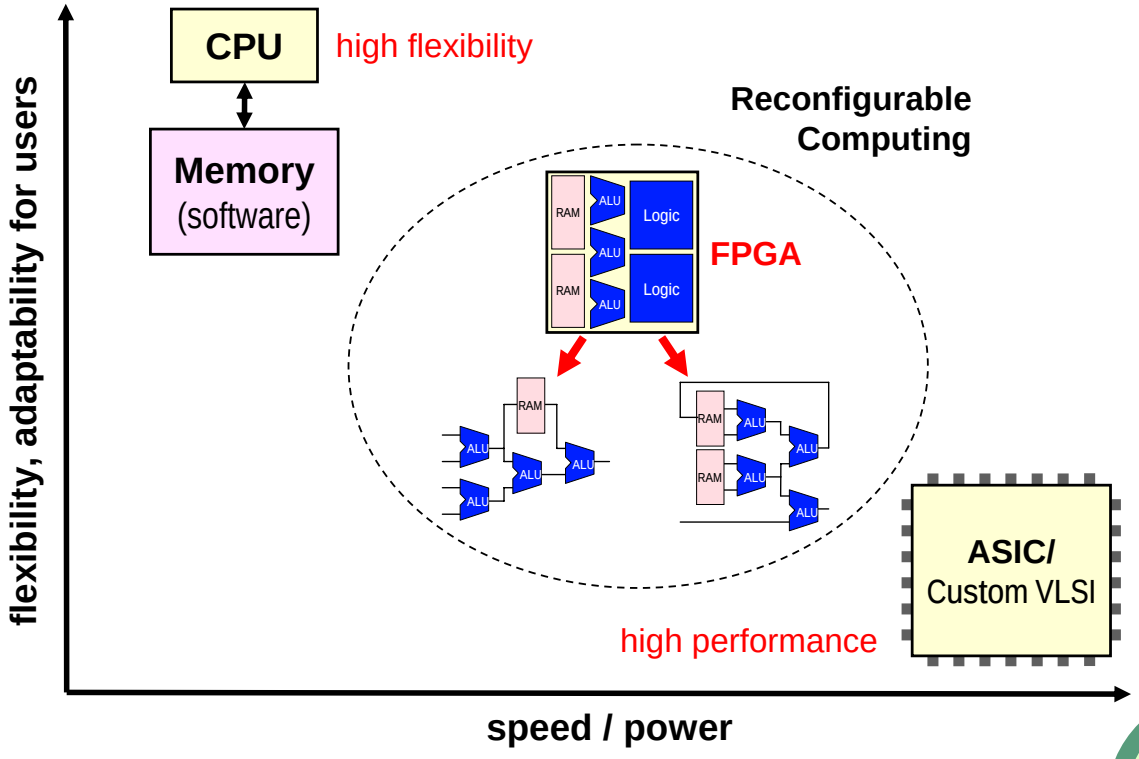
... a few % of the peak GFlops
of an entire large-scale system

If we give resource-balanced HW adaptively to each application, happy? YES!

But how?

Custom Computing Machine!

Platform for Custom Computing Machines

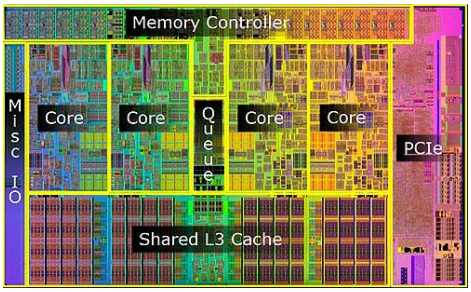


FPGA (Field-Programmable Gate Array)

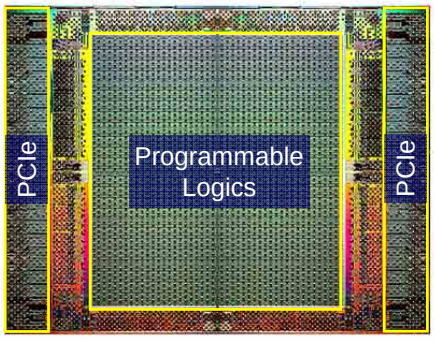
Core i7 processor

Stratix IV FPGA

FPGA (Field-Programmable Gate Array)



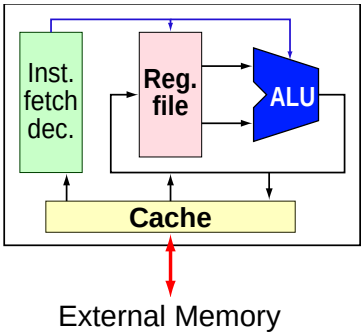
Die photo of Core i7 processor



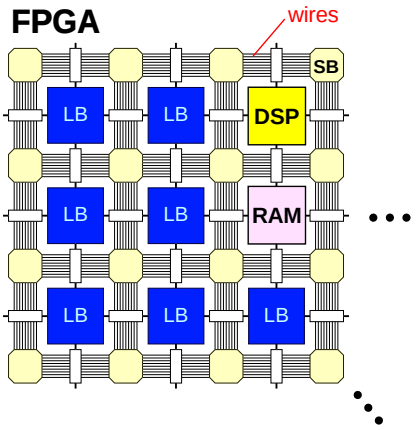
Die photo of Stratix IV FPGA

FPGA (Field-Programmable Gate Array)

Microprocessor (each core)



FPGA



Components for program execution:

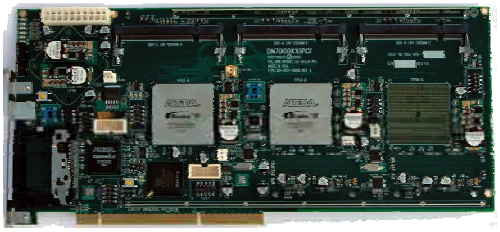
- ✓ register files
- ✓ ALUs (data-paths)
- ✓ control logic
- ✓ cache memory

Components to make logics:

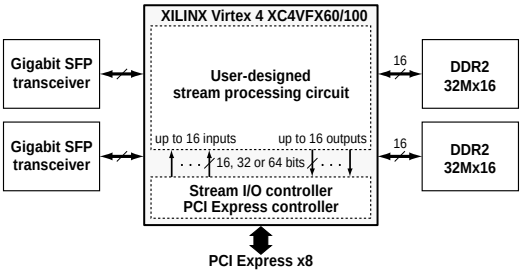
- ✓ logic blocks (LBs)
- ✓ DSP blocks (integer comput.)
- ✓ block RAMs
- ✓ wires & switch-blocks (SBs)

FPGA Boards

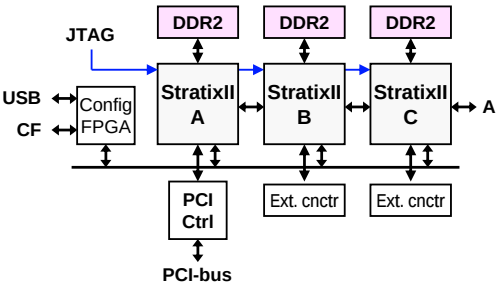
MAX-1
http://www.maxeler.com/content/



DN7000K10PCI
http://www.dinigroup.com/DN7000k10pci.php



PCI-Express I/F on FPGA



Separate PCI I/F chip

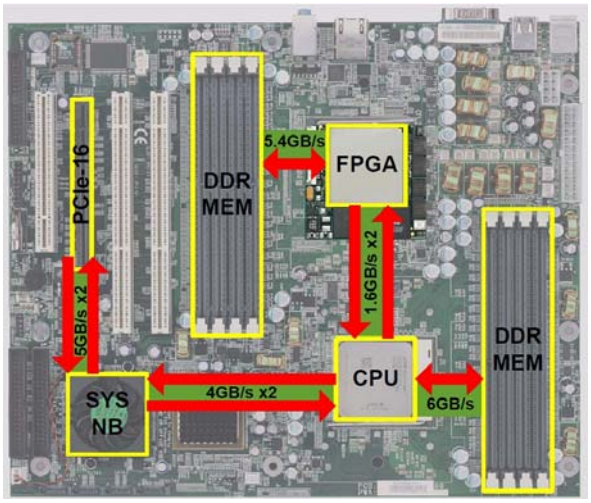
FPGA Co-Processors (XtremeData, Inc.)

FPGA module for CPU sockets

Wide bandwidth for main memory
(via FSB / Hyper Transport Link)

XD2000F
Altera StratixII
FPGAs for AMD
Opteron Socket

XD2000i
Altera StratixIII
FPGAs for
Intel Socket



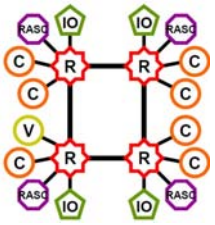
http://www.xtremedatainc.com/

FPGA Clusters

SGI Altix RASC RC100

2 Xilinx Virtex-4 LX200

Dual NUMAlink ports



FPGA
NUMA link
Processors
I/O
Visualization

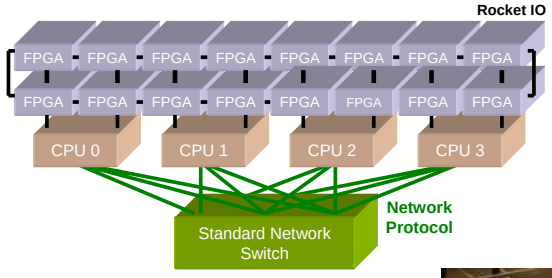
Altix System Architecture

Maxwell (The University of Edinburgh)

64 Xilinx Virtex-4 LX100

with 32 Intel Xeon processors

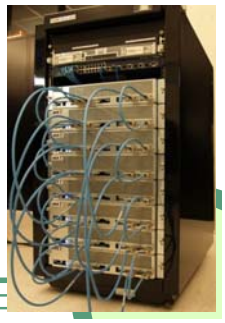
- PCI-X for Host to FPGAs
- 2D torus network of Rocket I/O



BEE2 (Berkley Emulation Engine)

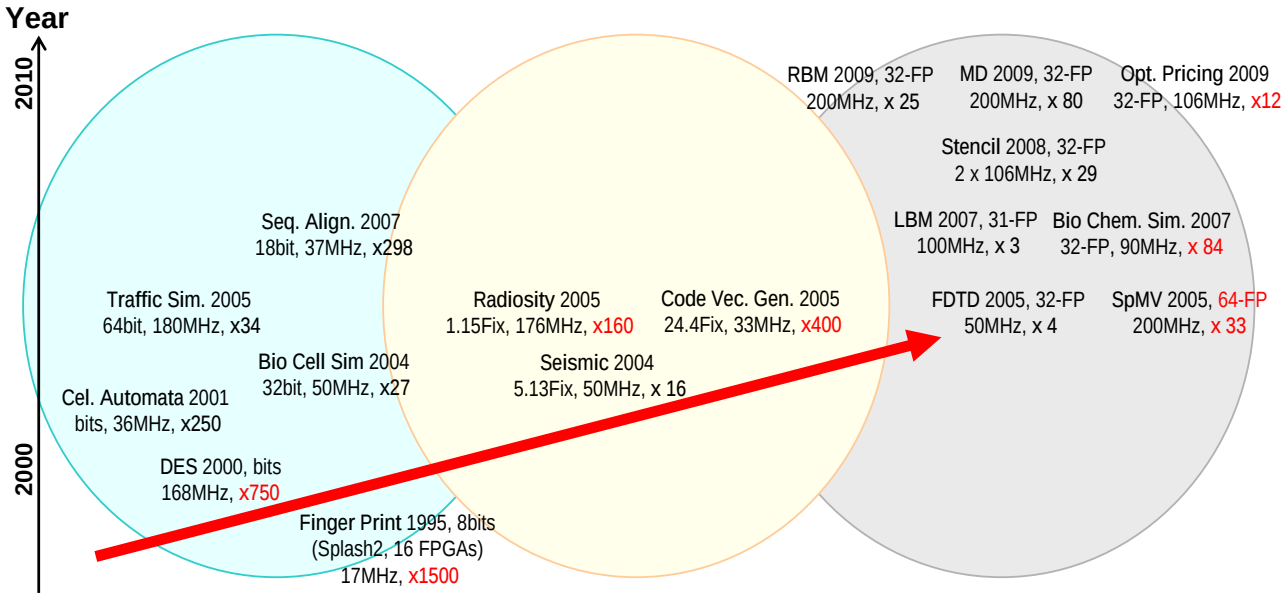
5 Xilinx Virtex-II pro 70 per module

- CAD tool acceleration
- image/signal processing
- scientific computing



Application Domains Suitable for FPGAs

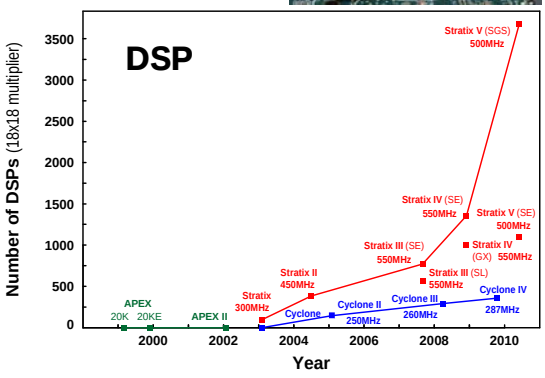
Capable for massively parallel floating-point computation, while suffering from overhead of frequency and area...

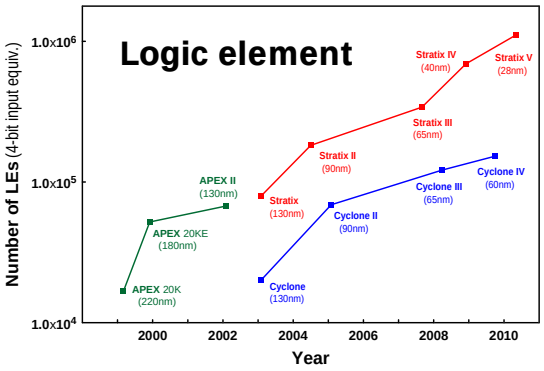


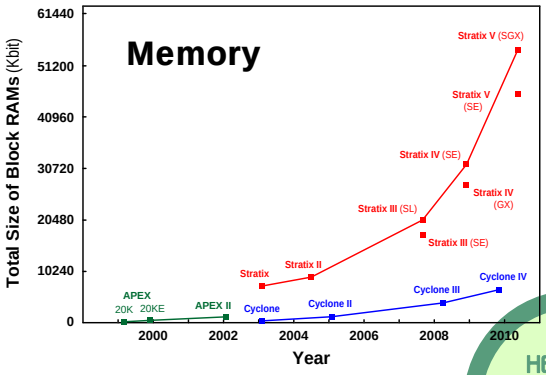
FPGA's Potential for HPC

State-of-the-art FPGAs

- ✓ Reconfigurable commodity device
- ✓ Larger & faster
- ✓ Lower initial-cost than VLSIs
- ✓ High potential for custom computing with floating-point operations





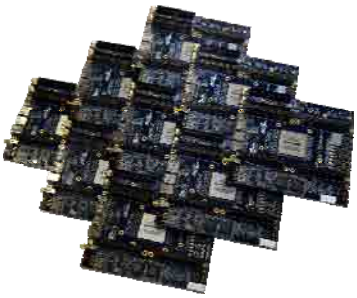


How to Build Scalable Machines?

- **Remove a bottleneck in scaling a system**
- **What causes the bottleneck? -- Bandwidth!**
  - ✓ Not only arithmetic performance, but also bandwidth must be considered for scalable custom machines.
- **We have to explore custom structures/architectures for individual applications to build scalable and extensible machines with multiple devices.**

Our Approaches toward Custom HPC

FPGA-based machines with tailored structures for scalable high-performance custom computation

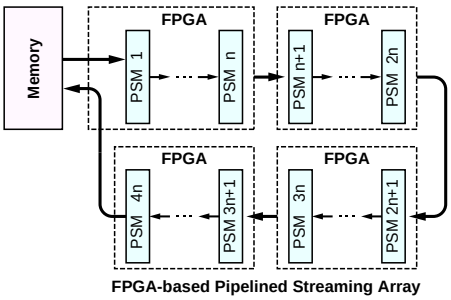


Systolic-array processor
for finite difference method

Stream processor with extensible custom pipelines

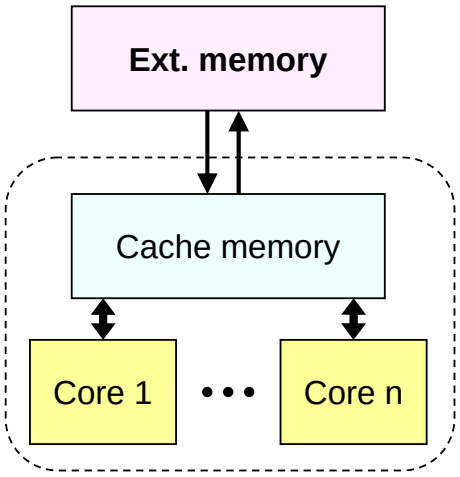
Real-time numerical data compressor for improving memory bandwidth

Stream Processor with Extensible Custom Pipelines for HPC Kernels



FPGA-based Pipelined Streaming Array

Problem of Many-Core Processors



Many cores require many data.

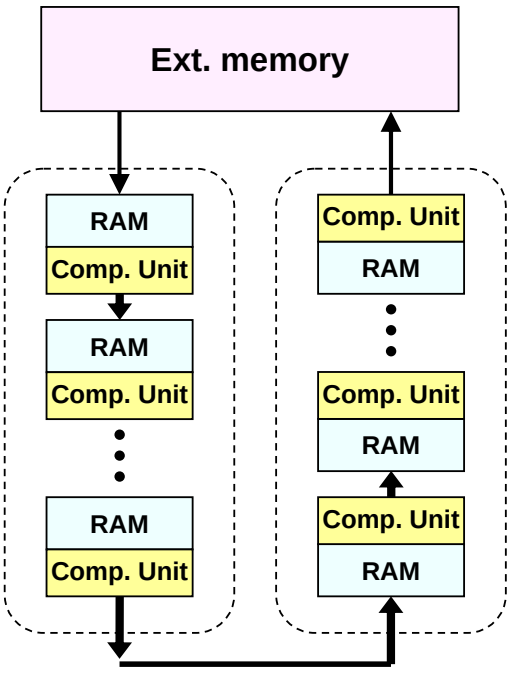
but

Insufficient memory bandwidth

↓

Cores are not fully utilized,
resulting in low scalability.

Custom Machine with Stream Computing



Computing unit requires only
the outputs of the previous unit

↓

All units can operate with
the constant bandwidth.

More units, higher performance.

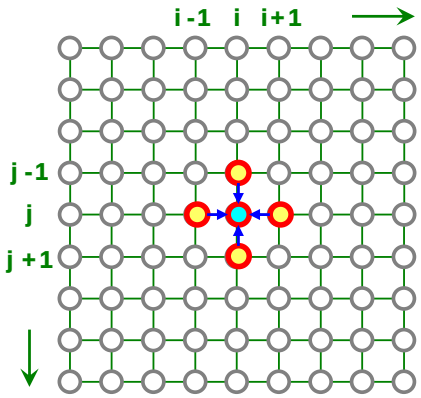
What kind of kernels can be streamed?

Time-Marching Stencil Computation

```
for (n=0; n< N_max ; n++)   <-- Time marching
  for (j=0; j< J_max ; j++)
    for (i=0; i< I_max ; i++)
      v'_{i,j} = f (v_{i,j}, v_{i+1,j} , v_{i-1,j} , v_{i,j+1} , v_{i,j-1})
                                   Stencil computation
```

Grid traverse

2D Time-marching Stencil Computation



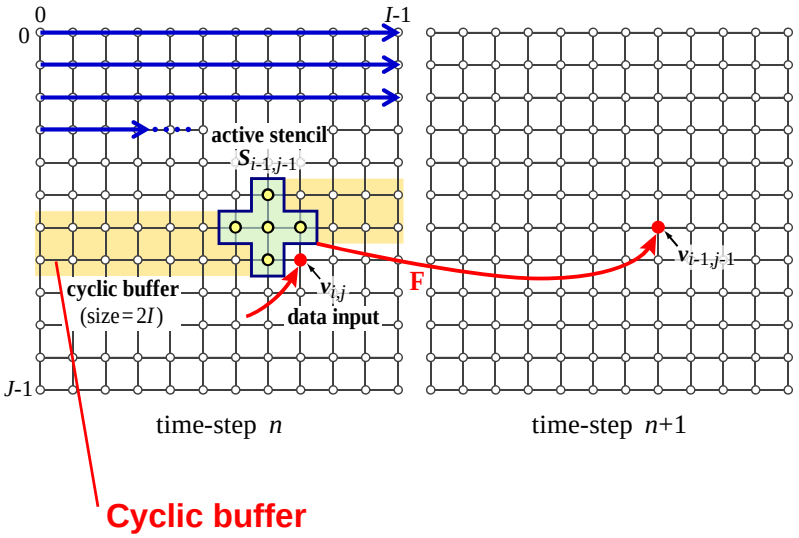
Computational grid

Stencil Computation

- ✓ Update all the grid-points with local stencil computation
- ✓ Repeat them for time-marching

(Examples: Cellular Automata, Jacobi computation, LGM, LBM)

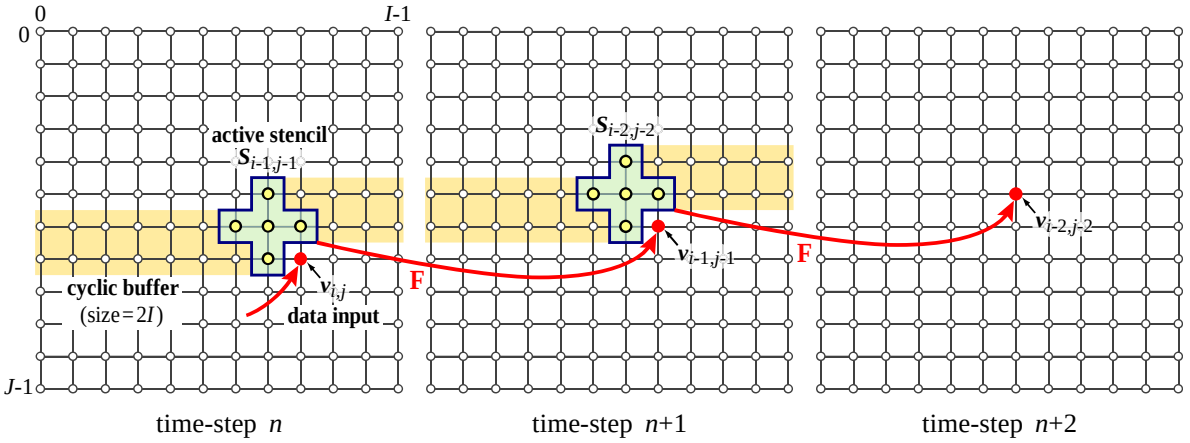
Make It Stream!



Cyclic buffer

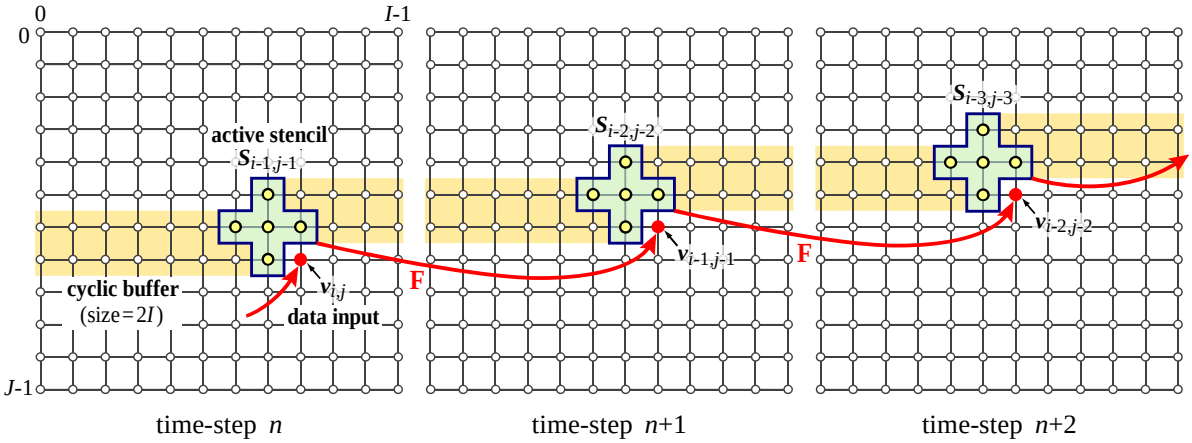
Tomoyoshi Kobori and Tsutomu Maruyama, "A High Speed Computation System for 3D FCHC Lattice Gas Model with FPGA," Proceedings of FPL2003, 755-765, 2003.

Make It Stream!



Tomoyoshi Kobori and Tsutomu Maruyama, "A High Speed Computation System for 3D FCHC Lattice Gas Model with FPGA," Proceedings of FPL2003, 755-765, 2003.

Make It Stream!

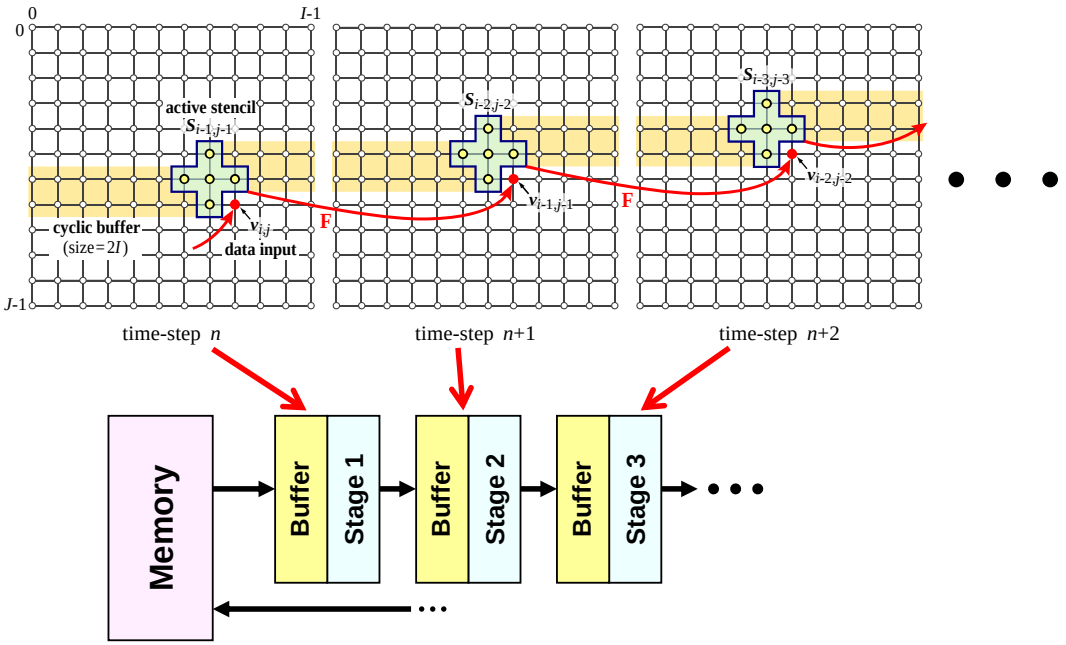


Pipelined execution of streamed multiple time-steps

Required bandwidth is constant!

Tomoyoshi Kobori and Tsutomu Maruyama, "A High Speed Computation System for 3D FCHC Lattice Gas Model with FPGA," Proceedings of FPL2003, 755-765, 2003.

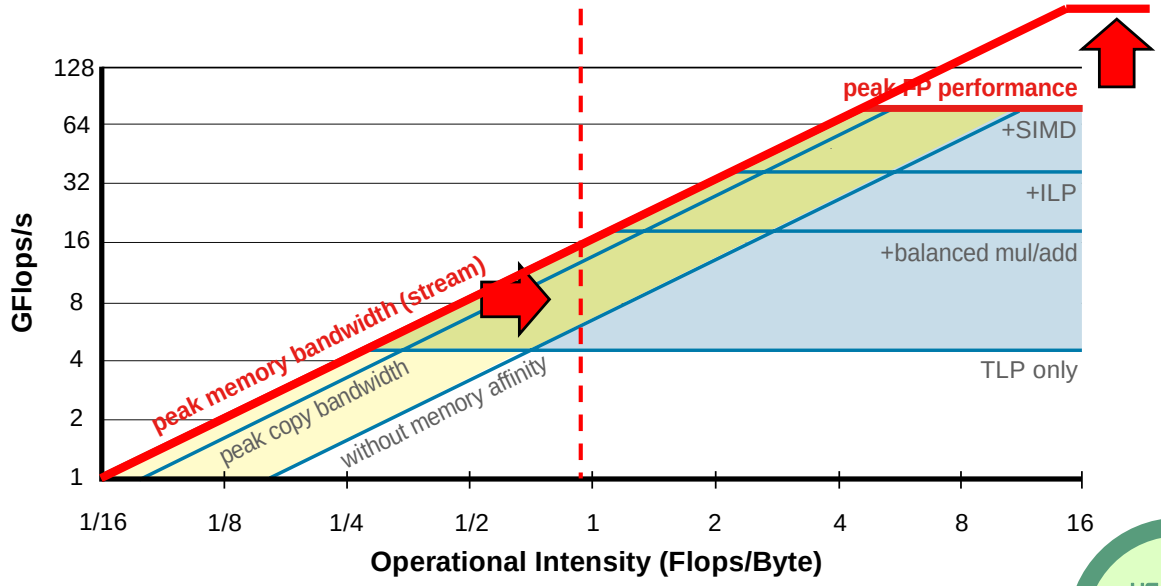
Stream Processor w/ Extensible Stages



Multiple-stage execution per memory-read.

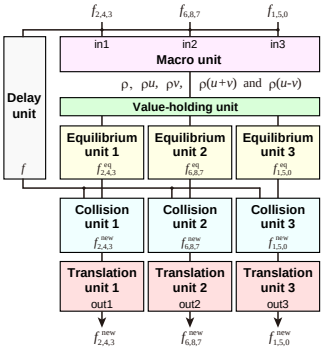
Roofline Model of Extensible Stream Processor

1. Increase the peak performance with a constant memory BW
2. Increase the operational intensity of problems

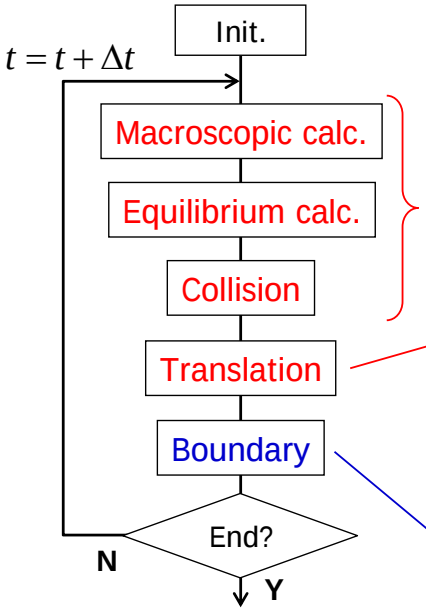


Single-Stage Stream Processor for Lattice Boltzmann Method (LBM)

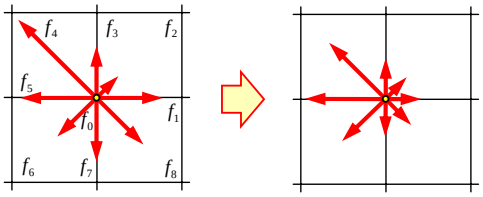
FPGA board



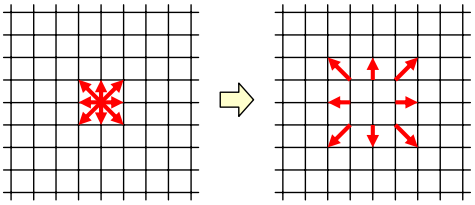
Algorithm for LBM



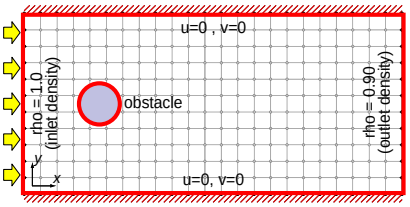
Relaxation to locally equilibrium state



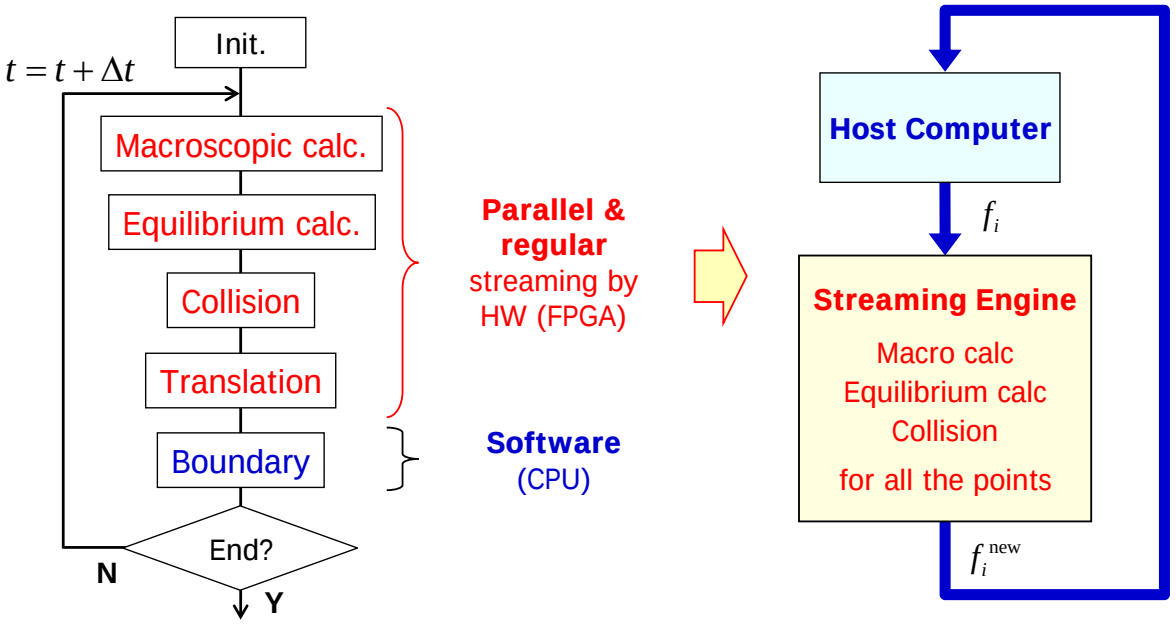
Diffuse particles



Boundary condition



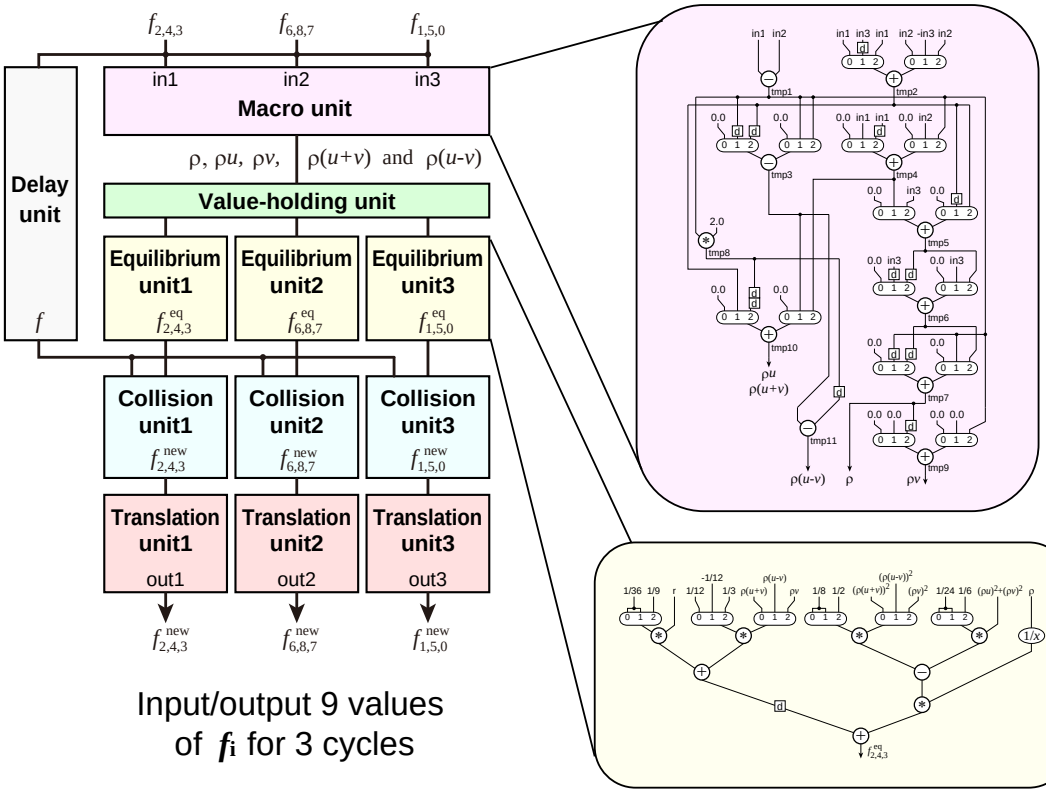
Mapping to Stream Processor



Streaming by FPGA

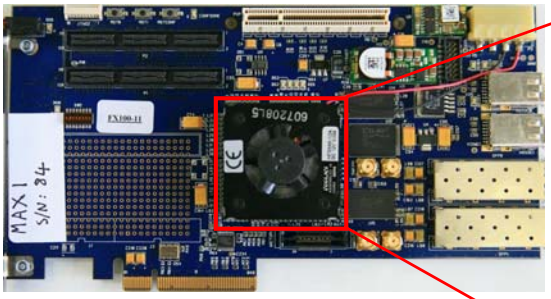
Kentaro Sano, et al., "FPGA-based Streaming Computation for Lattice Boltzmann Method,"
Procs of the International Conference on Field-Programmable Technology, pp.233-236, 2007.

Design of Pipeline Stage Unit

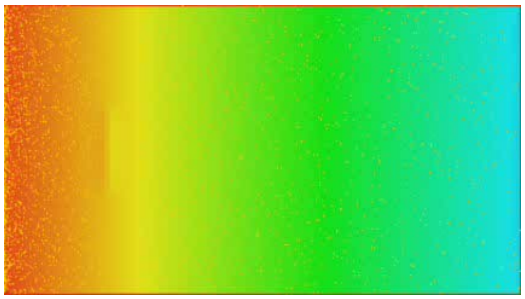


Input/output 9 values
of f_i for 3 cycles

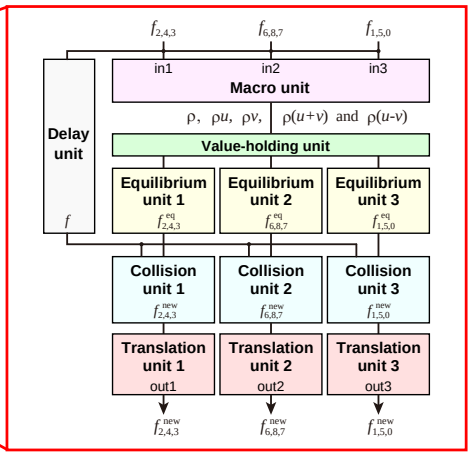
Implementation of Single Stage



Xilinx Virtex4 FPGA (2005)



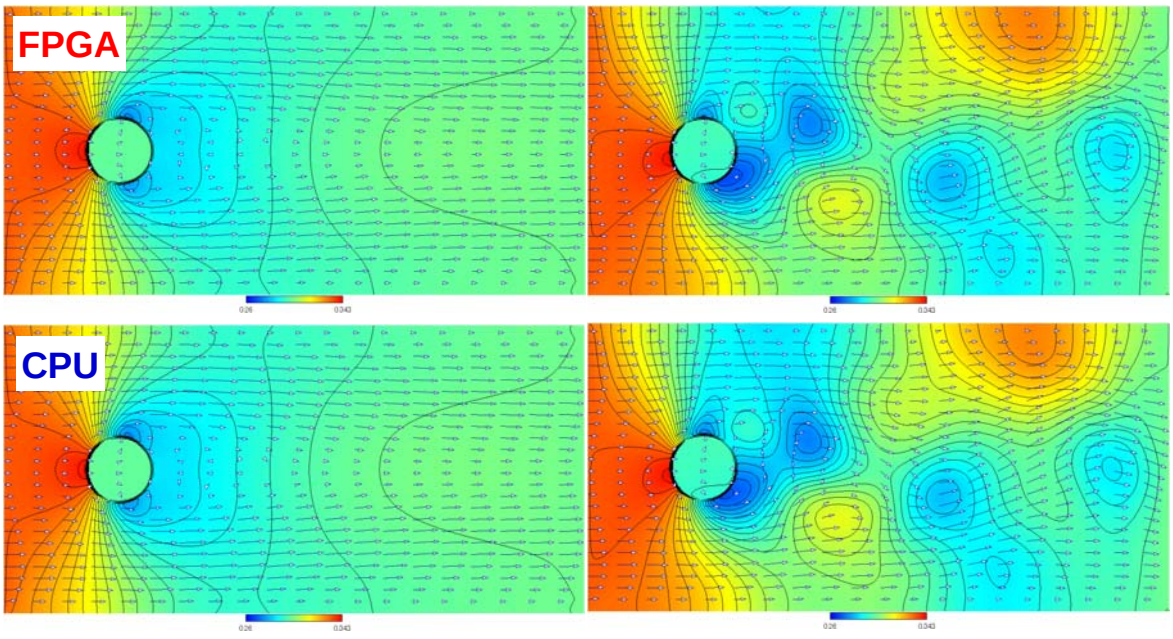
Real-time 2D LBM computation by FPGA



Single stage of streaming LBM
(pipeline operating at **100MHz**)

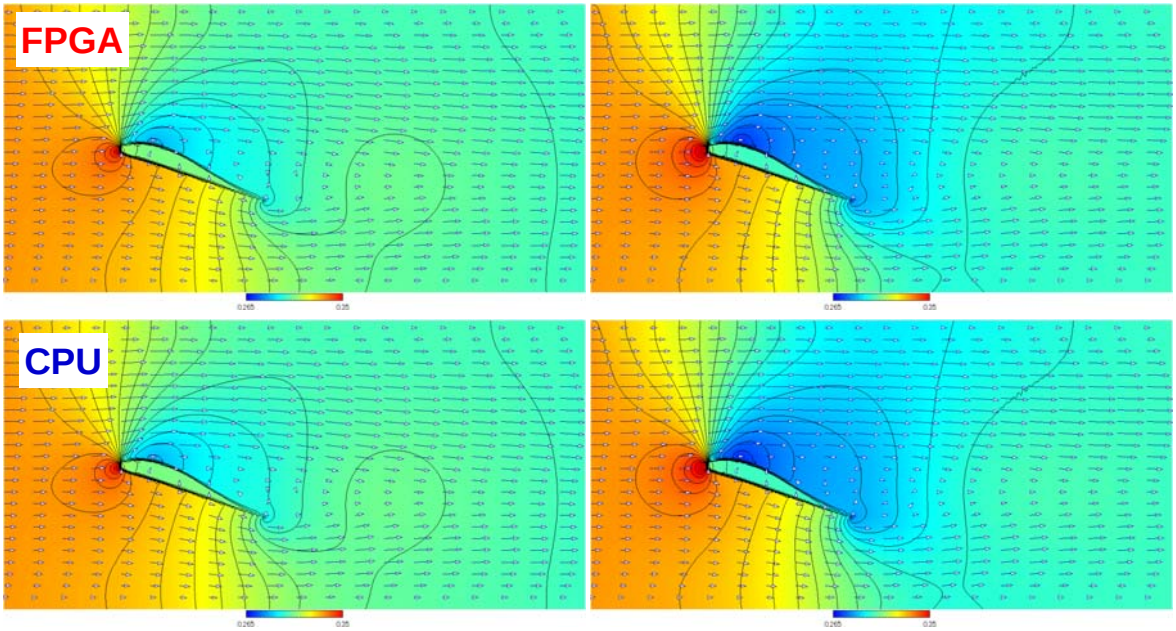
Floating-point operations
(single precision custom 31bit FP)
adder 27 multiplier 17
divider 1

Computed Results (Circle, RE=300)



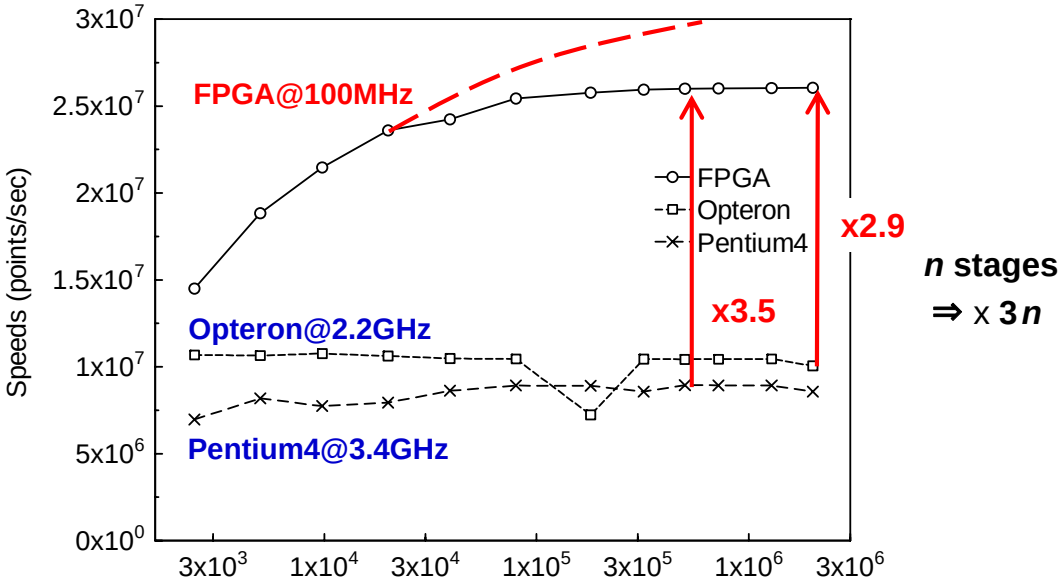
T=10000 T=50000

Computed Results (Wing, RE=300)



T=4000 T=50000

Speedup by Single Stage

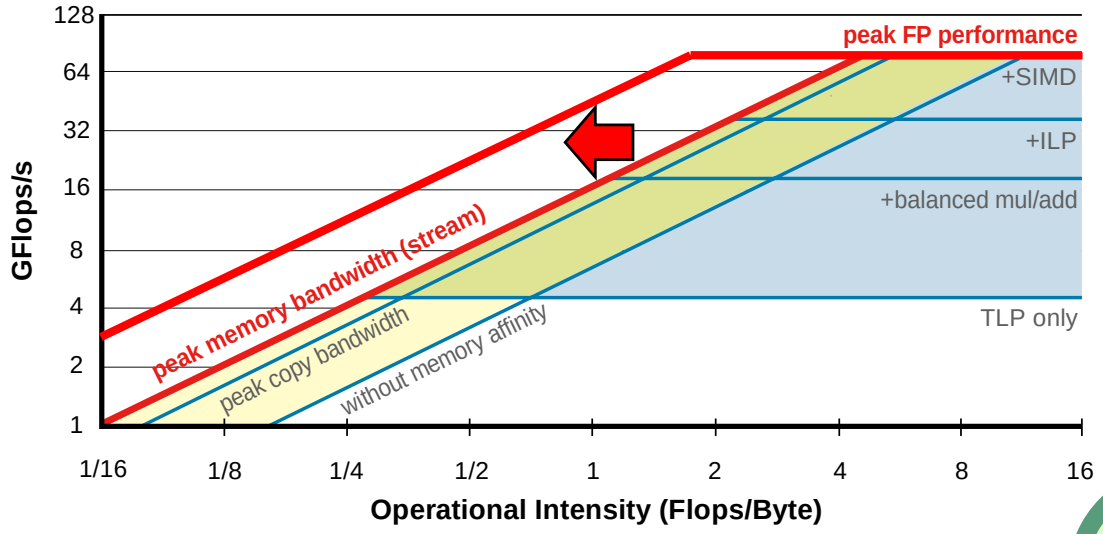


Bandwidth of PCI-Express limits the performance.

1.8GB/s : 40% stall of 100MHz pipeline,
and higher frequency > 100MHz available.

Can We Improve Memory Bandwidth?

Real-time data compressor



Real-time Data Compressor to Improve Memory Bandwidth for Streamed CFD Kernels

FPGA-based data compressor

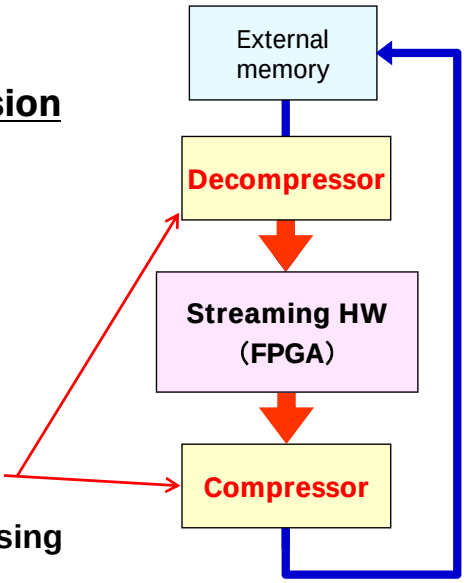
Strategy

Bandwidth enhancement for stream computing by data compression

- ✓ **Lossless compression** of floating-point (FP) data streams
- ✓ **Only high-throughput** required (Stream comp. is tolerant to latency.)

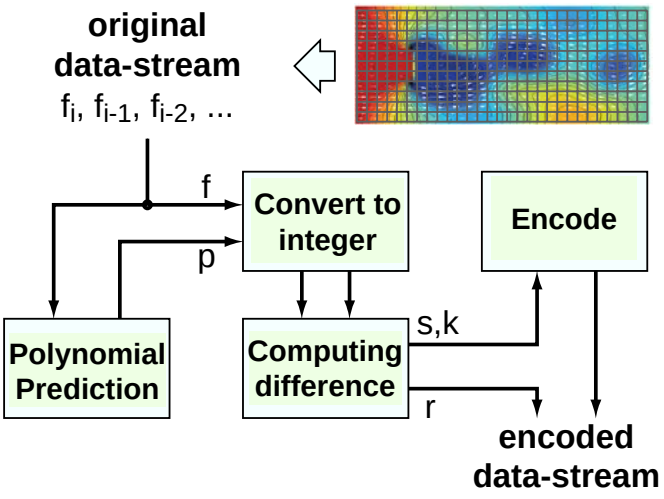
FPGA-based compression hardware

- ✓ High-throughput by **hardware processing**
- ✓ **Compact circuit** that can be attached to memory-I/F unit



Stream Computing

Compression with Polynomial Prediction

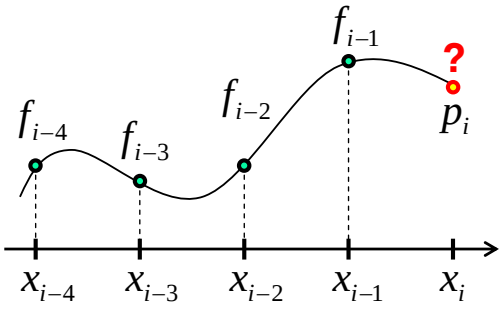


Overview of compression algorithm

More accurate prediction gives integer-difference closer to zero, which can be recorded with fewer bits.

1D Polynomial Predictors

[Ibarria2003]



Sequence of FP values

Underlying func. can be locally modeled with polynomial ones.

$$f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$$

1D polynomial predictors

$$p_i = c_{-1} f_{i-1} + c_{-2} f_{i-2} + \cdots + c_{i-n} f_{i-n}$$

Coefficients by Lagrange polynomial

✓ 0th order (n=1)	$p_i = f_{i-1}$	**Constant**
✓ 1st order (n=2)	$p_i = 2f_{i-1} - f_{i-2}$	**Linear**
✓ 2nd order (n=3)	$p_i = 3f_{i-1} - 3f_{i-2} + f_{i-3}$	**Quadratic**
✓ 3rd order (n=4)	$p_i = 4f_{i-1} - 6f_{i-2} + 4f_{i-3} - f_{i-4}$	**Cubic**

Difference Computation

32 bits or 64 bits

	sign	exponent	significand
Original **F**	1	0 0 1 0 1 1 0 0	0 0 1 0 1 1 0 1
Prediction **P**	1	0 0 1 0 1 1 0 0	0 0 1 0 1 0 1 1

|F – P| = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 **1 0** , **0**

leading-zero count **LZC = 13** (5bits)

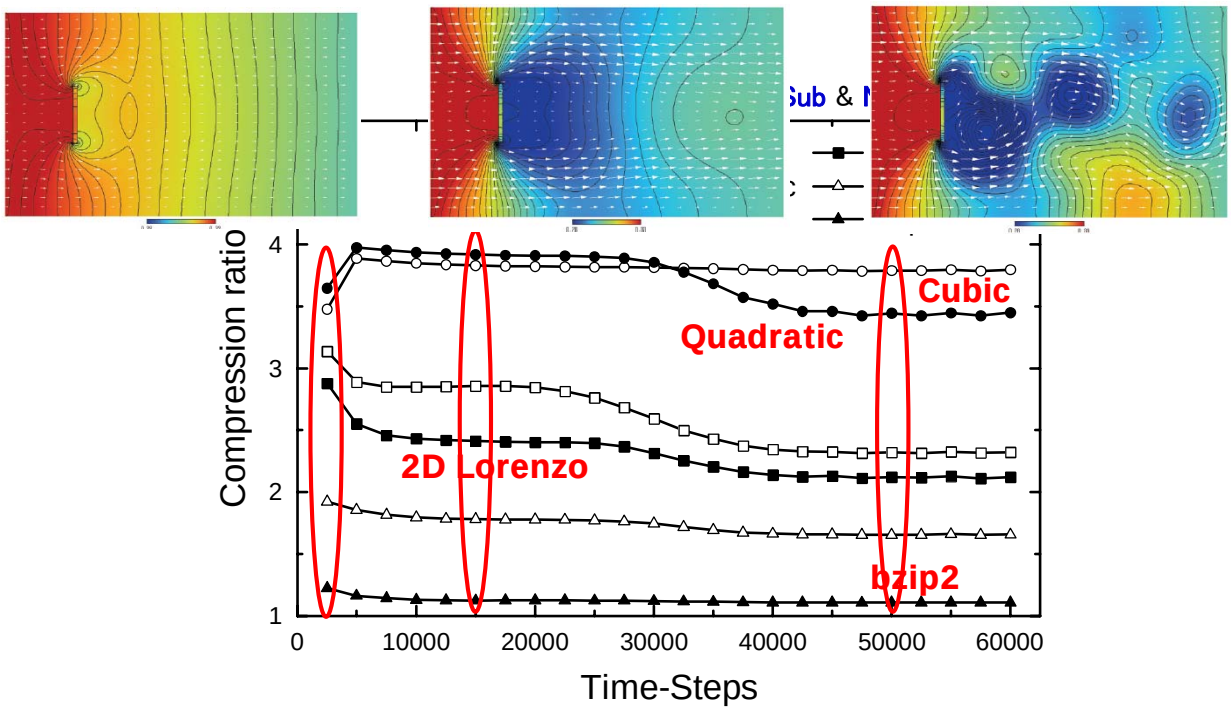
residual bits **r = '10'**

sign (F-P>0) **s = 0**

32 bits → 8bits (LZC:5bits, residual:2bits, sign:1bit)

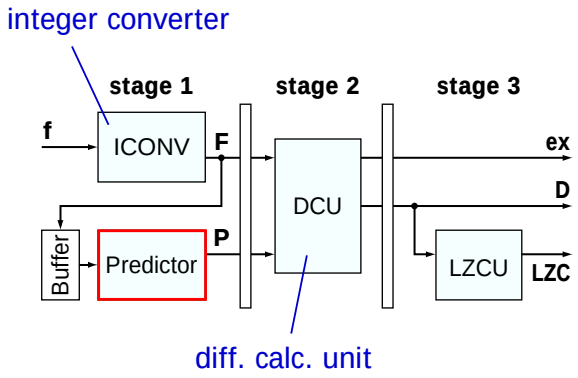
Compression ratio = 4

Compression Ratios for 2D LBM Case



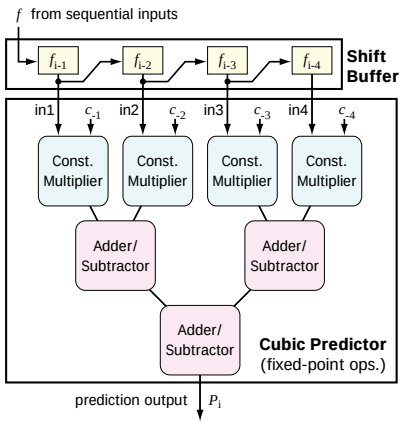
We estimate x4 wider bandwidth on average.

Design of High-Throughput Compressor



3-stage compressor

- ✓ No feedback loop
- ✓ High-throughput by pipelining



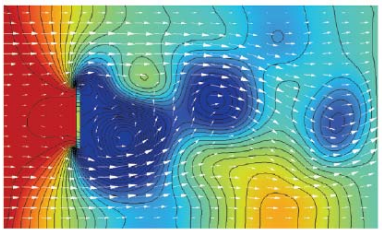
Buffer & Cubic predictor

- ✓ Fixed-point calculation

Developing a parallel compressor/decompressor

Summary

- **Efficient acceleration of HPC kernels**
  - ✓ requires balanced comp. & bandwidth.
  - ✓ FPGA-based custom computing provides.
  - ✓ How to design such custom machines?
    → architecture & algorithm mapping
- **Stream processor with extensible stages**
  - ✓ Higher scalability with slow memory
  - ✓ 2D LBM example
- **Data compression for wider BW**
  - ✓ Prediction-based compression
  - ✓ 4 x Compression = 4 times faster memory!
- **Custom computing can change the roofline to balance comp. and BW!**



Future Directions of HPCC

- **Performance issues**
  - ✓ Further exploration required on Algorithm, Architecture and Circuit levels.
  - ✓ New possibilities to implement custom machines
    - ➢ Coarser-grained reconfigurable devices?
    - ➢ Structured ASICs? -- ALTERA Hardcopy
    - ➢ Custom VLSIs by EPL (electron projection lithography)?
- **Productivity issues**
  - ✓ Design -- Programming/Design tools
    - ➢ Programming language is important for HPC. (OpenMP? CUDA? OpenCL? Chapel?)
  - ✓ Utilization -- Standards for programming, OS, HW
- **A lot of things to do!**