Custom Computing with Reconfigurable Technologies for Efficient Acceleration of CFD Kernels

Kentaro Sano
Graduate School of Information Sciences
Tohoku University, Japan

Outline

- Why Custom Computing?
- FPGA-based Stream Processor
- FPGA-based Real-time Data Compressor
- Summary

FPGA-based data-compressor

FPGA-based stream processor
**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

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

But how?

**Custom Computing Machine!**

Their massively-parallel systems

- Communication overhead
- Limited scalability
- Very inefficient computation

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

---

Roofline model of two AMD OpteronX4

Platform for Custom Computing Machines

- CPU: high flexibility
- Memory: (software)

Reconfigurable Computing

- FPGA
- ASIC/Custom VLSI

Speed / power

flexibility, adaptability for users

What’s FPGA (Field-Programmable Gate Array)?

- Core i7 processor
- Stratix IV FPGA
What’s FPGA (Field-Programmable Gate Array)?

**Components to make logics:**
- logic blocks (LBs)
- DSP blocks (integer comput.)
- block RAMs
- wires & switch-blocks (SBs)

**Components for program execution:**
- register files
- ALUs (data-paths)
- control logic
- cache memory

---

Die photo of Core i7 processor
Die photo of Stratix IV FPGA
**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

**Our Approaches toward Custom HPC**

FPGA-based custom machines with tailored structures for app. specific computing and memory

- **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 CFD Kernels

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

2D Time-marching Stencil Computation

\[
\begin{align*}
\text{for} \ (n=0; \ n<N_{\text{max}}; \ n++) & \quad \text{Time marching} \\
\text{for} \ (j=0; \ j<J_{\text{max}}; \ j++) \\
\text{for} \ (i=0; \ i<I_{\text{max}}; \ i++) \\
& \quad \text{Grid traverse} \\
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
\text{Stencil computation} \\
\text{Update all the grid-points with local stencil computation} \\
\text{Repeat them for time-marching} \\
(\text{Examples: Jacobi computation, LGM, LBM})
\end{cases}
\end{align*}
\]
**Make It Stream!**

Pipelined execution of streamed multiple time-steps

Required bandwidth is constant!


**Stream Processor w/ Extensible Stages**

Multiple-stage execution with one memory-read.
Single-Stage Stream Processor for Lattice Boltzmann Method (LBM)

Algorithm for LBM

- Macroscopic calc.
- Equilibrium calc.
- Collision
- Translation
- Boundary

End?

Relaxation to locally equilibrium state

Diffuse particles

Boundary condition

FPGA board
Design of Pipeline Stage Unit

\[ t = t + \Delta t \]

- **Init.**
- **Macroscopic calc.**
- **Equilibrium calc.**
- **Collision**
- **Translation**
- **Boundary**

**Parallel & regular streaming by HW (FPGA)**

**Software (CPU)**

**Host Computer**

**Streaming Engine**
- Macro calc
- Equilibrium calc
- Collision for all the points

**f_i^\text{new}**


---

Implementation of Single Stage

- Xilinx Virtex4 FPGA (2005)
- **Single stage of streaming LBM** (operating at 100MHz)

Real-time 2D LBM computation by FPGA
Computed Results (Circle, RE=300)

FPGA

CPU

T=10000  T=50000

Computed Results (Wing, RE=300)

FPGA

CPU

T=4000  T=50000
**Speedup by Single Stage**

![Graph showing speedup by single stage with three stages: FPGA@100MHz, Opteron@2.2GHz, Pentium4@3.4GHz.](image)

- FPGA@100MHz
- Opteron@2.2GHz
- Pentium4@3.4GHz

Bandwidth of PCI-Express limits the performance.

---

**Real-time Data Compressor**

**to Improve Memory Bandwidth for Streamed CFD Kernels**

FPGA-based data compressor

```
0100111011001
0011110100110...
```

0100111011001
0011110100110...
**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

**Compression with Polynomial Prediction**

**Overview of compression algorithm**

More accurate prediction gives integer-difference closer to zero, which can be recorded with fewer bits.
**Compression Ratios for 2D LBM Case**

We estimate x4 wider bandwidth on average.

**Summary**

- **FPGA-based custom computing**
  - Why we focus on?
  - How good expected to be?

- **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!
Future Directions of Custom Computing

- **Performance issues**
  - Further exploration required on
    - Algorithm level
    - Architecture level
    - Circuit level

- **Productivity issues**
  - Programming/Design tools
  - Standards for programming, OS, HW
  - Costs

- **Future work**
  - Prototype implementation & evaluation
  - Developing tools for HW and SW design
    (HW compiler for SW codes, etc.)

**Design of 3-stage compressor**

**Design of cubic predictor**

- **Systolic-array processor**
- **Stream processor**

**Thank you!**

**Real-time data compressor**