SMYLERef: A Reference Architecture for Manycore-Processor SoCs

Masaaki Kondo
(The University of Electro-Communications)

Acknowledgement

• SMYLE Project
  – Scalable ManY-core for Low-Energy computing
  – This research was supported in part by New Energy and Industrial Technology Development Organization

• Special thanks to SMYLE project members
Scalability of Parallel Programs

- In-Order Core@1GHz (up to 128 cores) w/ private 32KB L1 & 512KB L2
- 2D Mesh NoC, 100 ns DRAM latency

Benchmark: SPLASH-2
Simulator: Graphite (MIT)

Poor scalability in most of parallel applications

Issues in Manycore Processors

- Poor scalability in most of single applications
  - Lack of inherent parallelism
  - Memory access bottleneck
  - Barrier synchronization overhead

Our Goal: Manycore Processor SoC
- Efficient parallel processing for high-performance and low-power
- Exploiting Data / Thread / Task / Application level parallelism
- Effective memory hierarchy management
- High-speed barrier synchronization
Design Concept of SMYLEref

• VAM: Virtual Accelerator on Many-core
  – Flexible and effective mapping of multiple tasks
  – Uses many simple and low-power cores

Flexible VAM Allocation
• Increasing core usage
• Making good use of Locality

Using various parallelism
• Intra-VAM: data/thread parallelism
• Inter-VAM: task/app. parallelism

SMYLEref Manycore-Processor

• SMYLEref: a reference architecture for VAM
  – Bus-based multicore processor forms a cluster
  – Clusters are connected by a two-dimensional on-chip network (NoC)
  → Reduce hardware overhead of routers and NoC
Structure of Clusters and NoC

- Processor core: *Geyser core*
  - Developed in a national research project “Innovative Power Control for Ultra Low-Power High-Performance System LSIs” (PI: Prof. Nakamura at U.Tokyo)
  - Based on MIPS R3000, evaluated with real LSI implementation

- Cluster
  - Processor Cluster
    • 8 processor cores,
    • distributed shared L2 cache
    • a router for 2D-mesh NoC
  - Peripheral Cluster
    • DRAM controller
    • I/O controller
    • dedicated router for 2D-mesh NoC

Hardware Extensions for VAM

- Reconfigurable L1 data cache
  - Each L1 cache is used either as cache or Scratch-Pad-Memory
  - Determined by SMYLE compiler depending on applications

- Indexing management for distributed shared L2 cache
  - Base: each L2$ slice is shared by all clusters
  - Option: allocate set of L2$ slices to a particular VAM
    • Introduce dynamic address translation for L2 cache indexing
    • Avoid L2 cache contention between VAMs

- Group hardware barrier
  - Supports hardware barrier synchronization in arbitrary group of cores within VAM
Flexible HW Barrier Support

- Tree style dedicated barrier network
  - Realizes high-speed barrier sync. within each VAM
  - Parallel barrier operation for VAMs

Programming Environment for SMYLEref

- A number of parallel programming models, languages, and frameworks
  - OpenMP, MPI, OpenCL, Intel Threading Building Blocks, Nvidia CUDA, etc

- **OpenCL** is a natural choice
  - Open, royalty-free standard by Khronos Group
  - Based on C Language
  - Support of heterogeneous architectural platforms
  - Platform independent
    - Intel’s multi-core CPUs
    - Nvidia’s GPUs
    - AMD’s GPUs
    - SONY/IBM/Toshiba’s Cell B.E.
  - Supports both data and task-level parallelisms
Limitations of Existing OpenCL for GPUs

- Parallel execution of multiple applications is impossible
  - A single application occupies the entire device (all cores) at a time
- Hard to guarantee real-time constraints
  - Large performance overhead for context creation and dispatch
  - Such overhead is hardly predictable in multi-tasking systems
Key Features of SMYLE OpenCL

- Low runtime overhead
  - Very short start time
    - Static creation of contexts and objects
    - Static mapping of applications
- Multi-level parallel execution
  - Multiple applications
  - In each application
    - Task-parallel execution
    - Data-parallel execution
SMYLE OpenCL Toolkit

• Cross-Compiler
  – GCC as is

• Runtime Library
  – A limited set of OpenCL APIs have been implemented for the SMYLEref architecture

• Task Mappers
  – Single-context (exclusive) static mapping
  – Multi-context static mapping

• SMYLEref Native Simulator
  – Can execute OpenCL programs w/ runtime library on Linux-based host PC

Evaluation Platform for Manycores

• Requirement
  – Evaluate/verify many number of cores with high scalability
  – Evaluate programs with realistic working set including OS
  – Flexibility, Cost, etc.

• Candidate of the Platform
  – Software Simulator / LSI implementation / FPGA Prototyping

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Accuracy</th>
<th>Flexibility</th>
<th>Development Cost</th>
<th>Evaluation Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>Low</td>
<td>Medium</td>
<td>Very High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Real LSI</td>
<td>Medium</td>
<td>Very High</td>
<td>Low</td>
<td>Very High</td>
<td>Very High</td>
</tr>
<tr>
<td>FPGA</td>
<td>Very High</td>
<td>Very High</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
</tr>
</tbody>
</table>

FPGA prototyping is fairly advantageous
Development Environment

- FPGA board: Xilinx ML605 Evaluation board with Virtex-6
- HDL: Verilog HDL
- Logic Synthesis, Mapping, P&R: Xilinx ISE 14.2

<table>
<thead>
<tr>
<th>ML605 Evaluation Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA device</td>
</tr>
<tr>
<td>SDRAM</td>
</tr>
<tr>
<td>I/O port</td>
</tr>
<tr>
<td>Clock-input</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Virtex-6 (XCVLX240T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
</tr>
<tr>
<td>Logic Cells</td>
</tr>
<tr>
<td>CLB Slices</td>
</tr>
<tr>
<td>Block RAM</td>
</tr>
<tr>
<td>Num of user I/O</td>
</tr>
</tbody>
</table>

Design of Evaluation Platform

SMYLEref Architecture

Evaluation Platform with FPGA
Photographic View of Evaluation Platform

Preliminary Evaluation

- Hardware configuration
  - 8 core x 16 clusters (128 cores in total) + 1 peripheral cluster
  - Core clock: 10MHz, Bus and router clock: 5MHz, DDR3 SDRAM: 100MHz
  - without hardware coherence

- Software Environment
  - Benchmark: FFT and LU from SPLASH2
  - Compiler: gcc 4.4.6 targeted for MIPS-1
  - Floating point operation: software emulation (Soft Float)

- Parallel processing API
  - In house simple pthread library for SMYLEref Evaluation Platform
Evaluation Result: Parallel Speedup

- Correctly working up to 128 cores
- Poor scalability due to unsupported cache coherence
  - Heap data is always uncacheable
  - Needs cache flush in barrier synchronization and atomic operation

Scalability of Evaluation Environment

- Evaluation time comparison SMYLEref v.s. Software simulator
  - Software simulator: MARSS-x86 simulator
- SMYLEref on FPGA has very good scalability as an evaluation environment
Scalability of OpenCL Benchmark

- 1 host-core + 127 device core

Execution Time Breakdown

- Start time overhead (red bar) is almost constant to the number of cores
Summary

- **SMYLEref for Manycore-Processor SoCs**
  - Key Concept: Virtual Accelerator on Many-core (VAM)
  - Flexible and effective mapping of multiple tasks

- **OpenCL Programming Environment for SMYLEref**
  - Exploit data, thread, and application-level parallelisms
  - Low runtime overhead & multi-task

- **Evaluation platform on FPGA**
  - Can evaluate parallel programs up to 128 cores
  - OpenRISC version of SMYLEref evaluation environment will be available under BSD license