Software-Cooperative Power-Efficient Heterogeneous Multi-core for Media Processing


*Hitachi, Ltd.  **Waseda University

ASP-DAC 2008
Contents

1 Introduction
2 Heterogeneous multi-core architecture
3 Parallelizing compiler
4 Performance evaluation
5 Summary
Digital semiconductor trends

- Integration of more and more functions into a chip
- Gate density continues to increase (~32nm)
- Frequency saturation and power become issues
Our approach: to achieve high-performance by parallel architecture utilizing high-gate density with support of parallelizing compiler for high software productivity.
Power-aware HMCP architecture

- Multiple types of processors
  - CPUs + accelerators (ACC)

- Unified hierarchical memory architecture
  - LDM: local data memory
  - DSM: distributed shared memory
  - LPM: local program memory
  - CSM: centralized shared memory

- Programmable data transfer unit (DTU)
- Power control register (FVR)
Accelerator core; Flexible Engine (FE)

A dynamically reconfigurable processor as an accelerator
Data transfer unit (DTU)

- Concurrent data transfer with CPU computation
- Programmability by transfer commands on local memories
  - Put/get commands
  - Flag check/set commands

Interconnection network

Local Memory

Command #1
- FLAG CHECK
- Flag Adr.
- Check Value
- Check Interval
- Next pointer

Command #2
- TRANSFER
- Source Adr.
- Destination Adr.
- Transfer Size
- Next pointer

Command #3
- TRANSFER
- Source Adr.
- Destination Adr.
- Transfer Size
- Next pointer

Command #4
- FLAG SET
- Flag Adr.
- Flag Value
- Next pointer

CPU #m
Local Mem.
Local Mem.

CPU #n
Local Mem.
Local Mem.

Put/Get

Source Adr.
Transfer
Destination Adr.
Next pointer
OSCAR parallelizing compiler

Improve effective performance, cost-performance and productivity and reduce consumed power

- **Multigrain parallelization**
  - Exploitation of parallelism from the whole program by use of coarse-grain parallelism among loops and subroutines, near fine grain parallelism among statements in addition to loop parallelism

- **Data localization**
  - Automatic data distribution for distributed shared memory, cache and local memory on multiprocessor systems

- **Data transfer overlapping**
  - Data transfer overhead hiding by overlapping task execution and data transfer using DMA or data pre-fetching

- **Power reduction**
  - Reduction of consumed power by compiler control of frequency, voltage and power shut down with hardware supports

*Optimally SCheduled Advanced multiprocessor*
Compiling steps

Processor Info. → OSCAR Compiler

Program

Macro task (MT) partitioning
Parallelism analysis
MT scheduling

CPU-MT → DRP-MT → DSP-MT

Local compiler

CPU compiler

DRP compiler / library

DRP compiler / library

Chip

CPU #0  CPU #1  DRP #0  DRP #1  DSP #0  DSP #1

Code generation
10. Generation of coarse grain tasks

- **Macro-tasks (MTs)**
  - Block of pseudo assignments (BPA): Basic block (BB)
  - Repetition block (RB): loop
  - Subroutine block (SB): subroutine

Diagram:
- Near fine grain parallelization
- Loop level parallelization
- Near fine grain of loop body
- Coarse grain parallelization

Program

- 1st. Layer
- 2nd. Layer
- 3rd. Layer

Total System
Sample macro-task graph (MTG)
# Scheduled-tasks execution

<table>
<thead>
<tr>
<th>CPU0 CORE</th>
<th>DTU</th>
<th>CPU1 CORE</th>
<th>DTU</th>
<th>CPU2 CORE</th>
<th>DTU</th>
<th>CPU3 CORE</th>
<th>DTU</th>
<th>DRP0 CORE</th>
<th>DTU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MTG1:
- **MT1-1**: Load
- **MT1-2**: Load
- **MT1-3**: Send
- **MT1-4**: Send

MTG2:
- **MT2-1**: Load
- **MT2-2**: Send
- **MT2-5**: Send
- **MT2-7**: Send

MTG3:
- **MT3-1**: Load
- **MT3-2**: Send
- **MT3-3**: Load
- **MT3-4**: Send
- **MT3-7**: Store
- **MT3-8**: Store

MTG3:
- **MT2-3**: Load
- **MT2-4**: Send
- **MT2-6**: Load
- **MT3-5**: Send
- **MT3-7**: Store
- **MT3-8**: Store

**MTG1** and **MTG2** are scheduled for tasks, while **MTG3** is reserved for scheduled tasks.
Compiler power saving scheme

- Power controlling in parallelized tasks
  - Compiler control of frequency / voltage (F/V) and power shut down by utilizing a parallelized task scheduling result
  - Reduces power while maintaining parallelized performance

Macro task graph

- CPU0
  - MT1: FV: FULL
  - MT2: FV: FULL
  - MT3: FV: FULL

- CPU1
  - MT1: FV: FULL
  - MT2: FV: FULL
  - MT3: FV: FULL

Scheduled tasks

- MT2: FV: LOW

F/V control

- CPU0: FV: FULL
  - MT1: FV: LOW
  - MT2: FV: FULL
  - MT3: FV: FULL

- CPU1: FV: FULL
  - MT2: FV: FULL
  - MT3: FV: FULL

Power off

- CPU0: Power OFF
  - MT1: Power OFF
  - MT2: Power OFF
  - MT3: Power OFF

- CPU1: Power OFF
  - MT2: Power OFF
  - MT3: Power OFF
Evaluated HMCP architecture

- Cycle-accurate simulator is utilized
- CPU: SuperH (SH)
- DRP: FE w. sub CPU
  - 300 MHz @ 90-nm tech.
- Memory latency
  - Local memory: 1-cycle latency (local), 4-cycle latency (remote)
  - On-chip SM: 4-cycle latency
  - Off-chip SM: 16-cycle latency
- Wattch-based power model
  - Parameters were introduced from RTL-level power simulation on SH processors
**Evaluated application; MP3 encoder**

- **MP3 Audio Encoding** as firstly evaluated application targeting car navigation systems as an example
  - Encoding processed in each individual audio frame and inter-frame parallelism exists
  - 16 frames of 16-bit 44.1-KHz input and 128-kbps stream output

**Process flow of MP3 encoding**

- PCM Data
  - **Subband Analysis**
    - MDCT
    - Psycho-Acoustic Analysis
  - **Quantization**
  - **Huffman Coding**
  - **Bit Encoding**
    - **MP3 Bit Stream**

**Profiling Result**

- **Bit Encoding** 1.7%
- **Huffman Coding** 0.4%
- **Subband Analysis** 25.4%
- **Psycho-Acoustic Analysis** 1.2%
- **MDCT** 4.0%

**Quantization** 67.4%

MDCT: Modified Discrete Cosine Transform
Effect of FE

- 24 macro tasks assigned for FEs in MP3 encoder

**CPU/ FE execution cycles per frame**

- **Subband analysis**
  - (subband 1 task) x 71.9

- **Psycho-acoustic analysis**
  - (psycho Total) x 108.2

- **MDCT**
  - (mdct Total) x 59.7

- **Quantization**
  - (alloc_bits 5 tasks) x 4.4
  - (outer_loop 14 tasks)
  - (inner_loop 2 tasks)

FE effect
Average x 6.9
Performance evaluation

![Graph showing speed-up rate against one CPU for different configurations.]

- **Homogeneous**
  - 1CPU: 1.0
  - 2CPU: 2.0
  - 4CPU: 4.0
  - 8CPU: 7.6
  - 2CPU+1FE: 7.1
  - 2CPU+2FE: 12.6
  - 4CPU+2FE: 14.6
  - 2CPU+4FE: 22.3
  - 4CPU+4FE: 24.5

- **Heterogeneous**
  - 2CPU: 5
  - 4CPU: 10
  - 8CPU: 15
  - 16CPU: 20
  - 32CPU: 25
  - 64CPU: 30

The graph compares the speed-up rates for different CPU configurations, showing a significant increase in performance with the addition of multiple CPUs and I/O accelerators.
Macro task trace (SHx2+DRPx2)

CPU#0

CPU#1

FE#0

FE#1

Filter, MDCT, psycho-acoustic analysis

Quantization for frame #0-15

Bit stream generation

Time
Power control mode

Status of the FV power mode

- **CPU**: Avg. power 150 mW @ 300 MHz, 1.0 V (FULL)
  - FULL, MID, LOW, OFF supported
- **FE**: Avg. power 210 mW @ 300 MHz, 1.0 V (FULL)
  - FULL, OFF supported

<table>
<thead>
<tr>
<th></th>
<th>FULL</th>
<th>MIDDLE</th>
<th>LOW</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clock frequency</strong></td>
<td>1 (300 MHz)</td>
<td>1/2 (150 MHz)</td>
<td>1/4 (75 MHz)</td>
<td>0 (Clock off)</td>
</tr>
<tr>
<td><strong>Supply voltage</strong></td>
<td>1 (1.0 V)</td>
<td>0.87 (0.87 V)</td>
<td>0.71 (0.71 V)</td>
<td>0 (Power off)</td>
</tr>
<tr>
<td><strong>Leakage power</strong></td>
<td>1 (1.0 %)</td>
<td>1 (1.0 %)</td>
<td>1 (1.0 %)</td>
<td>0 (No power)</td>
</tr>
</tbody>
</table>
Power control effects by compiler

Energy consumption [J]

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Energy Consumption [J]</th>
<th>Percentage Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o FV 2CPU+1FE</td>
<td>0.27</td>
<td>-37.1%</td>
</tr>
<tr>
<td>w/ FV 2CPU+1FE</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>w/o FV 2CPU+2FE</td>
<td>0.22</td>
<td>-26.4%</td>
</tr>
<tr>
<td>w/ FV 2CPU+2FE</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>w/o FV 2CPU+4FE</td>
<td>0.20</td>
<td>-19.2%</td>
</tr>
<tr>
<td>w/ FV 2CPU+4FE</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>w/o FV 4CPU+4FE</td>
<td>0.24</td>
<td>-28.4%</td>
</tr>
<tr>
<td>w/ FV 4CPU+4FE</td>
<td>0.17</td>
<td></td>
</tr>
</tbody>
</table>
Macro task trace with power control

- Clock frequency of CPU#0 and #1 is lowered for bit-stream generation tasks
Summary

- **Power-efficient heterogeneous multi-core architecture supported by OSCAR parallelizing compiler was studied**
  - Various types of processor cores on a chip such as CPUs and Flexible Engines as accelerators
  - Unified hierarchical memory architecture throughout the PEs controlled by software to improve performance, power efficiency and programming/compiling efficiency
  - Power control registers reducing power consumption

- **Performance evaluation was performed using MP3 audio encoder**
  - 24.5-folded speed-up in performance on 4 CPUs and 4 FEs against sequential execution on one CPU
  - As much as 37.1% reduction in energy consumption was achieved when the compiler power saving scheme was applied

*Thank you for your attention!!*