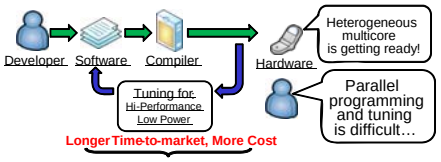
Poster Session by Ambient GCOE RA, July 14, 2008

"Compiler Cooperative Heterogeneous Multicore Processor"

Akihiro Hayashi, Yasutaka Wada, Hiroaki Shikano, Jun Shirako, Keiji Kimura, Hironori Kasahara
Dept. of Computer Science and Engineering

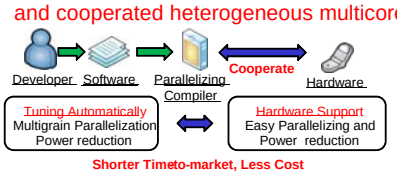
Motivation

- Heterogeneous Multicore
  - Multicore processor + Accelerator
  - Mainstream in Consumer Electoronics



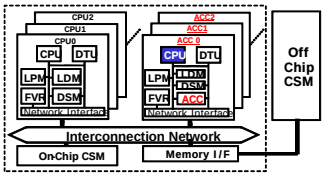
- How can we cope with this problem?

- Our approach : Parallelizing Compiler and cooperated heterogeneous multicore



- Simulation Result
  - 4CPU+4ACC (Hitachi:300MHz, 0.7W)
  - Pentium 4 (Intel:3.2GHz, 72W)
    - ⇒realize equivalent performance!

OSCAR Heterogeneous Multicore Processor

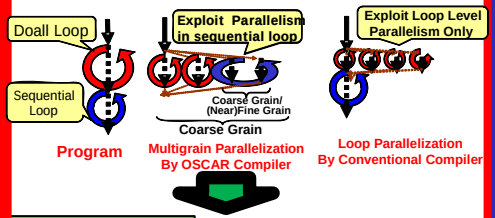


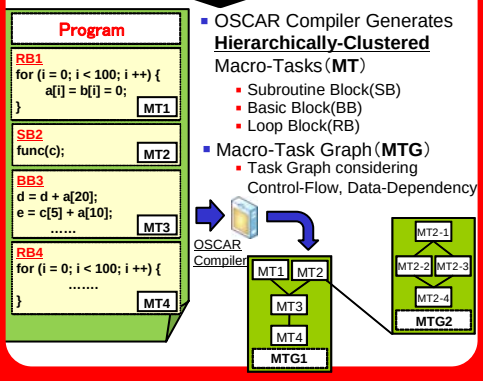
- Accelerator Core
  - (CPU Controller) + (Memory) + (ACC)
- LPM (Local Program Memory)
- LDM (Local Data Memory)
- DSM (Distributed Shared Memory)
- CSM (Centralized Shared Memory)
- DTU (Data Transfer Unit)
- FVR (Frequency/Voltage register)

We are Proposing OSCAR Parallelizing Compiler and OSCAR Heterogeneous multicore

Multigrain Parallelization

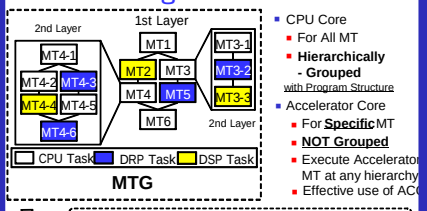
- Exploitation of **Multi-level** Parallelism
  - Coarse-grain Parallelism
    - Subroutine, Loop, Basic Block
  - Fine-grain Parallelism
    - Loop Level Parallelism
  - Near-fine-grain Parallelism
    - Statement Level Parallelism



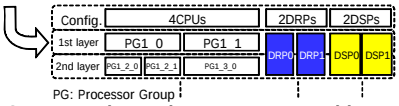


- OSCAR Compiler Generates **Hierarchically-Clustered** Macro-Tasks (**MT**)
  - Subroutine Block(SB)
  - Basic Block(BB)
  - Loop Block(RB)
- Macro-Task Graph (**MTG**)
  - Task Graph considering Control-Flow, Data-Dependency

Multigrain Parallelization For Heterogeneous Multicore

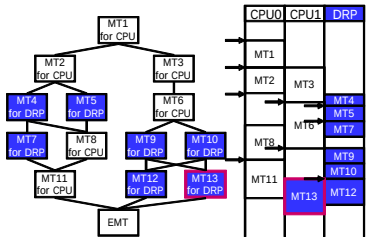


- CPU Core
  - For All MT
  - **Hierarchically - Grouped** with Program Structure
- Accelerator Core
  - For **Specific** MT
  - **NOT Grouped**
  - Execute Accelerator MT at any hierarchy
  - Effective use of ACC



Core-grouping on heterogeneous multicore to realize multigrain parallelization

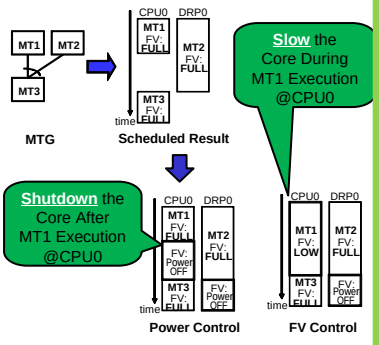
Static Scheduling Scheme for Heterogeneous Multicore



- All Accelerator MT is **NOT** always assigned to DRP
  - Optimization for higher throughput
  - Accelerator MT is determined by Compiler Directive

Power Reduction Scheme For Low Power Consumption

- Fastest Execution Mode



State	FULL	MID	LOW	OFF
Frequency	1	1 / 2	1 / 4	0
Voltage	1	0.87	0.71	0
Energy	1	3 / 4	1 / 2	0
Power Leakage	1	1	1	0

F/V state definition

Compiler achieves low power dissipation by controlling appropriate F/V state for each MT

Details about simulation

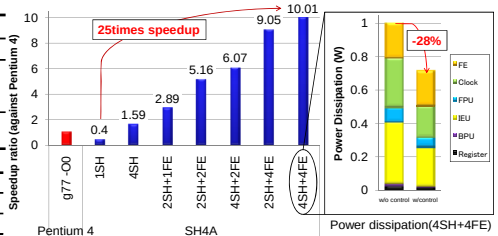
- Evaluated Application : MP3 Encoder
- OSCAR Heterogeneous Multicore

Processor	SH4A, FE-GA(DRP)
Frequency	300 MHz (SH,FE,Bus, Memory)
Memory Access Latency	LDM: 1 cycle DSM : 1 cycle(local), 4 cycle(remote) Off-Chip CSM: 16 cycle
Power Dissipation (per 1 core)	SH4A: 150 mW, FE-GA: 210 mW (@300MHz, 1.0V, 90nm process

- Pentium 4

Processor	Intel Pentium 4
Environment	Linux 2.2.4 + g77 (used performance counter)
Frequency	3.2 GHz
Cache Size	L1: 8KB, L2: 512 KB
Power Dissipation	70 W (@3.2GHz, 1.5V, 0.13 umprocess)

Performance on a OSCAR Heterogeneous Multicore



- 25times speedup @ 4SH+4FE against a single SH4A
- 0.7W:28% Power reduction by Compiler Control

Conclusions

- Parallelizing Compiler and cooperated heterogeneous multicore
  - Effective use of accelerator
  - Controlling approproate F/V state
  - Realize High -performance, Low power dissipation
    - 4SH+4FE @300MHz = Pentium 4 @3.2GHz with 1/100 power dissipation
- Future Work
  - Applying our method to real -chip

Waseda University Global COE Program
"International Research and Education Center for Ambient SoC"