Homogeneous and Heterogeneous Multicore / Manycore Processors, Parallelizing Compiler and Multiplatform API for Green Computing

Hironori Kasahara
Professor, Dept. of Computer Science & Engineering
Director, Advanced Multicore Processor Research Institute
Waseda University, Tokyo, Japan

IEEE Computer Society Board of Governors
URL: http://www.kasahara.cs.waseda.ac.jp/

MPSoC2011, July7, 2011
Multi/Many-core Everywhere

Multi-core from embedded to supercomputers

- Consumer Electronics (Embedded)
  - Mobile Phone, Game, TV, Car Navigation, Camera,
    - IBM/ Sony/ Toshiba Cell, Fujitsu FR1000,
    - Panasonic Uniphier, NEC/ARM MPCore/MP211/NaviEngine,
    - Renesas 4 core RP1, 8 core RP2, 15core Hetero RP-X,
    - Plurality HAL 64(Marvell), Tilera Tile64/ -Gx100(->1000cores),
    - DARPA UHPC (2017: 80GFLOPS/W)

- PCs, Servers
  - Intel Quad Xeon, Core 2 Quad, Montvale, Nehalem(8cores),
  - Larrabee(32cores), SCC(48cores), Night Corner(50 core+:22nm),
  - AMD Quad Core Opteron (8, 12 cores)

- WSs, Deskside & Highend Servers
  - IBM(Power4,5,6,7), Sun (SparcT1,T2), Fujitsu SPARC64fx8

- Supercomputers
  - BG/Q (A2:16cores) Water Cooled20PFLOPS, 3-4MW (2011-12),
  - BlueWaters(HPCS) Power7, 10 PFLOP+(2011.07),
  - Tianhe-1A (4.7PFLOPS,6coreX5670+ Nvidia Tesla M2050),
  - Godson-3B (1GHz40W 8core128GFLOPS) -T (64 core,192GFLOPS:2011)
  - RIKEN Fujitsu “K” 10PFLOPS(8core SPARC64VIII fx, 128GGFLOPS)

High quality application software, Productivity, Cost performance, Low power consumption are important

Ex, Mobile phones, Games

Compiler cooperated multi-core processors are promising to realize the above futures

The 27thTop 500 (20.6.2011),
No.1, Fujitsu “K” 548,352 cores
(Current Peak 8.774 PFLOPS)
LINPACK 8.162 PFLOPS (93.0%)
**METI/NEDO National Project**

**Multi-core for Real-time Consumer Electronics**

**<Goal>** R&D of compiler cooperative multi-core processor technology for consumer electronics like Mobile phones, Games, DVD, Digital TV, Car navigation systems.

**<Period>** From July 2005 to March 2008

**<Features>**
- Good cost performance
- Short hardware and software development periods
- Low power consumption
- Scalable performance improvement with the advancement of semiconductor
- Use of the same parallelizing compiler for multi-cores from different vendors using newly developed API

API: Application Programming Interface

**<Period>** (2005.7 〜 2008.3)**

**Features**
- Good cost performance
- Short hardware and software development periods
- Low power consumption
- Scalable performance improvement with the advancement of semiconductor
- Use of the same parallelizing compiler for multi-cores from different vendors using newly developed API

**<API>**
- Toshiba
- Panasonic
- NEC

**<Method>**
- Use of the same parallelizing compiler for multi-cores from different vendors using newly developed API

**<Result>**
- Development of high-performance multi-core processors
- Development of low-power multi-core processors
- Development of high-speed multi-core processors
- Development of high-reliability multi-core processors

**<Application>**
- Multi-core processors for consumer electronics
- Multi-core processors for automotive electronics
- Multi-core processors for industrial electronics

**<Impact>**
- Significant improvement in performance and efficiency
- Increased market competitiveness of consumer electronics
- Advanced technology for automotive and industrial applications

**<Acknowledgments>**
- Hitachi, Renesas, Fujitsu, Toshiba, Panasonic, NEC
**Renesas-Hitachi-Waseda Low Power 8 core RP2**

Developed in 2007 in METI/NEDO project

**Process Technology**
- 90nm, 8-layer, triple-Vth, CMOS

**Chip Size**
- 104.8mm²
- (10.61mm x 9.88mm)

**CPU Core Size**
- 6.6mm²
- (3.36mm x 1.96mm)

**Supply Voltage**
- 1.0V–1.4V (internal)
- 1.8/3.3V (I/O)

**Power Domains**
- 17 (8 CPUs, 8 URAMs, common)

---

IEEE ISSCC08: Paper No. 4.5, M.ITO, ... and H. Kasahara, “An 8640 MIPS SoC with Independent Power-off Control of 8 CPUs and 8 RAMs by an Automatic Parallelizing Compiler”
OSCAR Multi-Core Architecture

CMP₀ (chip multiprocessor 0)

CPU

PE₀

PE₁

PEₙ

LDM/ D-cache

DTC

DSM

Network Interface

Intra-chip connection network
(Multiple Buses, Crossbar, etc)

CSM / L2 Cache

Inter-chip connection network
(Crossbar, Buses, Multistage network, etc)

I/O Devices

CSM:
central shared mem.

LDM:
local data mem.

DSM:
distributed shared mem.

LPM:
local program mem.

DTC:
Data Transfer Controller

FVR:
frequency / voltage control register
8 Core RP2 Chip Block Diagram

Cluster #0

Core #0
- CPU
- FPU

Local memory
I:8K, D:32K

URAM 64K

Core #1
- CPU
- FPU

Local memory
I:8K, D:32K

URAM 64K

Core #2

Core #3

Barrier
Sync. Lines

Core #4

Core #5

Core #6

Core #7

Cluster #1

URAM 64K

Local memory
I:8K, D:32K

Local memory
I:8K, D:32K

Local memory
I:8K, D:32K

On-chip system bus (SuperHyway)

LCPG: Local clock pulse generator
PCR: Power Control Register
CCN/BAR: Cache controller/Barrier Register
URAM: User RAM (Distributed Shared Memory)
Demo of NEDO Multicore for Real Time Consumer Electronics at the Council of Science and Engineering Policy on April 10, 2008

CSTP Members
Prime Minister: Mr. Y. FUKUDA
Minister of State for Science, Technology and Innovation Policy: Mr. F. KISHIDA
Chief Cabinet Secretary: Mr. N. MACHIMURA
Minister of Internal Affairs and Communications: Mr. H. MASUDA
Minister of Finance: Mr. F. NUKAGA
Minister of Education, Culture, Sports, Science and Technology: Mr. K. TOKAI
Minister of Economy, Trade and Industry: Mr. A. AMARI
1987 OSCAR (Optimally Scheduled Advanced Multiprocessor)
OSCAR PE (Processor Element)

SYSTEM BUS

BUS INTERFACE

LOCAL BUS 1

LOCAL BUS 2

INSTRUCTION BUS

DMA

LPM

DSM

LSM

LDM

IPU

FPU

REG

DP

DMA : DMA CONTROLLER
LPM : LOCAL PROGRAM MEMORY (128KW * 2BANK)
INSC : INSTRUCTION CONTROL UNIT
DSM : DISTRIBUTED SHARED MEMORY (2KW)
LSM : LOCAL STACK MEMORY (4KW)
LDM : LOCAL DATA MEMORY (256KW)
DP : DATA PATH
IPU : INTEGER PROCESSING UNIT
FPU : FLOATING PROCESSING UNIT
REG : REGISTER FILE (64 REGISTERS)
1987 OSCAR PE Board
OSCAR Memory Space (Global and Local Address Space)
OSCAR Parallelizing Compiler

To improve effective performance, cost-performance and software productivity and reduce power

Multigrain Parallelization

coarse-grain parallelism among loops and subroutines, near fine grain parallelism among statements in addition to loop parallelism

Data Localization

Automatic data management for distributed shared memory, cache and local memory

Data Transfer Overlapping

Data transfer overlapping using Data Transfer Controllers (DMAs)

Power Reduction

Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.
Generation of coarse grain tasks

- **Macro-tasks (MTs)**
  - Block of Pseudo Assignments (BPA): Basic Block (BB)
  - Repetition Block (RB): natural loop
  - Subroutine Block (SB): subroutine
Earliest Executable Condition Analysis for coarse grain tasks (Macro-tasks)

A Macro Flow Graph

A Macro Task Graph
Automatic processor assignment in su2cor

- Using 14 processors
  - Coarse grain parallelization within DO400 of subroutine LOOPS

\[ N_{PG}, N_{PE} = [PG, PE] \]
MTG of Su2cor-LOOPS-DO400

- Coarse grain parallelism PARA_ALD = 4.3
Data-Localization
Loop Aligned Decomposition

- Decompose multiple loop (Doall and Seq) into **CARs** and **LRs** considering inter-loop data dependence.
  - Most data in **LR** can be passed through LM.
  - **LR**: Localizable Region, **CAR**: Commonly Accessed Region
Data Localization

MTG

MTG after Division

A schedule for two processors
An Example of Data Localization for Spec95 Swim

### (a) An example of target loop group for data localization

```fortran
DO 200 J=1,N
  DO 200 I=1,M
    UNEW(I+1,J) = UOLD(I+1,J) +
    TTDTS8*(Z(I+1,J)+Z(I,J))*(CV(I+1,J)+CV(I,J)+CV(I,J+1)) -
    TDTSDX*(H(I+1,J)+H(I,J))
    VNEW(I,J+1) = VOLD(I,J+1) -
    TTDTS8*(Z(I,J+1)+Z(I,J))*(CU(I+1,J)+CU(I,J)+CU(I,J+1)) -
    TDTSDY*(H(I,J)+H(I,J+1))
    PNEW(I,J) = POLD(I,J) -
    TDTSDX*(CU(I,J+1)-CU(I,J)) -
    TDTSDY*(CV(I,J)+CV(I,J+1))
  200 CONTINUE
```

```fortran
DO 210 J=1,N
  UNEW(1,J) = UNEW(M+1,J)
  VNEW(M+1,J+1) = VNEW(1,J+1)
  PNEW(M+1,J) = PNEW(1,J)
  210 CONTINUE
```

```fortran
DO 300 J=1,N
  DO 300 I=1,M
    UOLD(I,J) = U(I,J) +
    ALPHA*(UNEW(I,J)-2.*U(I,J)+UOLD(I,J))
    VOLD(I,J) = V(I,J) +
    ALPHA*(VNEW(I,J)-2.*V(I,J)+VOLD(I,J))
    POLD(I,J) = P(I,J) +
    ALPHA*(PNEW(I,J)-2.*P(I,J)+POLD(I,J))
  300 CONTINUE
```

### (b) Image of alignment of arrays on cache accessed by target loops

- **Cache line conflicts occur among arrays which share the same location on cache.**
- **Image of alignment of arrays on cache accessed by target loops.**
Data Layout for Removing Line Conflict Misses by Array Dimension Padding

Declaration part of arrays in spec95 swim

**before padding**

PARAMETER (N1=513, N2=513)

COMMON U(N1,N2), V(N1,N2), P(N1,N2),
  * UNEW(N1,N2), VNEW(N1,N2),
  1 PNEW(N1,N2), UOLD(N1,N2),
  * VOLD(N1,N2), POLD(N1,N2),
  2 CU(N1,N2), CV(N1,N2),
  * Z(N1,N2), H(N1,N2)

**after padding**

PARAMETER (N1=513, N2=544)

COMMON U(N1,N2), V(N1,N2), P(N1,N2),
  * UNEW(N1,N2), VNEW(N1,N2),
  1 PNEW(N1,N2), UOLD(N1,N2),
  * VOLD(N1,N2), POLD(N1,N2),
  2 CU(N1,N2), CV(N1,N2),
  * Z(N1,N2), H(N1,N2)
Power Reduction by Power Supply, Clock Frequency and Voltage Control by OSCAR Compiler

- Shortest execution time mode

- Realtime processing mode with dead line constraints
An Example of Machine Parameters for the Power Saving Scheme

• Functions of the multiprocessor
  – Frequency of each proc. is changed to several levels
  – Voltage is changed together with frequency
  – Each proc. can be powered on/off

<table>
<thead>
<tr>
<th>state</th>
<th>FULL</th>
<th>MID</th>
<th>LOW</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency</td>
<td>1/2</td>
<td>1/4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>voltage</td>
<td>0.87</td>
<td>0.71</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>dynamic energy</td>
<td>3/4</td>
<td>1/2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>static power</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

• State transition overhead (Example: not for RP2)

<table>
<thead>
<tr>
<th>state</th>
<th>FULL</th>
<th>MID</th>
<th>LOW</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>0</td>
<td>40k</td>
<td>40k</td>
<td>80k</td>
</tr>
<tr>
<td>MID</td>
<td>40k</td>
<td>0</td>
<td>40k</td>
<td>80k</td>
</tr>
<tr>
<td>LOW</td>
<td>40k</td>
<td>40k</td>
<td>0</td>
<td>80k</td>
</tr>
<tr>
<td>OFF</td>
<td>80k</td>
<td>80k</td>
<td>80k</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>FULL</th>
<th>MID</th>
<th>LOW</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>0</td>
<td>20</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>MID</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>LOW</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>OFF</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>0</td>
</tr>
</tbody>
</table>

delay time [u.t.]  energy overhead [μJ]
Power Reduction Scheduling

Fig. 6. V/F control of applu(4proc.)
Generated Multigrain Parallelized Code
(The nested coarse grain task parallelization is realized by only OpenMP “section”, “Flush” and “Critical” directives.)

Centralized scheduling code

1st layer

Distributed scheduling code

2nd layer

Thread group0

Thread group1

END SECTIONS
Compilation Flow Using OSCAR API

- **Application Program**
  - Fortran or Parallelizable C (Sequential program)

- **Waseda Univ. OSCAR Parallelizing Compiler**
  - Coarse grain task parallelization
  - Global data Localization
  - Data transfer overlapping using DMA
  - Power reduction control using DVFS, Clock and Power gating

- **Compilation Flow**
  - **OSCAR API for Real-time Low Power High Performance Multicores**
    - Directives for thread generation, memory, data transfer using DMA, power managements
  - Generation of parallel machine codes using sequential compilers
  - Multicore from Vendor A
  - Multicore from Vendor B
  - Shred memory servers

- **Backend Compiler**
  - API Analyzer
  - Existing sequential compiler

- **OpenMP Compiler**

- **Hitachi, Renesas, Fujitsu, Toshiba, Panasonic, NEC**

**OSCAR: Optimally Scheduled Advanced Multiprocessor API:** Application Program Interface
OSCAR API

Now Open! (http://www.kasahara.cs.waseda.ac.jp/)

- Targeting mainly realtime consumer electronics devices
  - embedded computing
  - various kinds of memory architecture
    - SMP, local memory, distributed shared memory, ...

- Developed with Japanese 6 companies
  - Fujitsu, Hitachi, NEC, Toshiba, Panasonic, Renesas
  - Supported by METI/NEDO

- Based on the subset of OpenMP
  - very popular parallel processing API
  - shared memory programming model

- Six Categories
  - Parallel Execution (4 directives from OpenMP)
  - Memory Mapping (Distributed Shared Memory, Local Memory)
  - Data Transfer Overlapping Using DMA Controller
  - Power Control (DVFS, Clock Gating, Power Gating)
  - Timer for Real Time Control
  - Synchronization (Hierarchical Barrier Synchronization)
Performance of OSCAR Compiler on IBM p6 595 Power6 (4.2GHz) based 32-core SMP Server

OpenMP codes generated by OSCAR compiler accelerate IBM XL Fortran for AIX Ver.12.1 about **3.3 times** on the average

Compile Option:


(*2) Sequential: -O5 -q64 –qarch=pwr6, XLF: -O5 –q64 –qarch=pwr6 –qsmp=auto, OSCAR: -O5 –q64 –qarch=pwr6 –qsmp=noauto

(Others) Sequential: -O5 –qarch=pwr6, XLF: -O5 –qarch=pwr6 –qsmp=auto, OSCAR: -O5 –qarch=pwr6 –qsmp=noauto
Performance of OSCAR Compiler Using the Multicore API on Intel Quad-core Xeon

- OSCAR Compiler gives us 2.1 times speedup on the average against Intel Compiler ver.10.1
Performance of OSCAR compiler on 16 cores SGI Altix 450 Montvale server

Compiler options for the Intel Compiler:
- for Automation parallelization: -fast -parallel.
- for OpenMP codes generated by OSCAR: -fast -openmp

- OSCAR compiler gave us 2.3 times speedup against Intel Fortran Itanium Compiler revision 10.1
Performance of OSCAR compiler on NEC NaviEngine(ARM-NEC MPcore)

- OSCAR compiler gave us 3.43 times speedup against 1 core on ARM/NEC MPCore with 4 ARM 400MHz cores
Performance of OSCAR Compiler Using the multicore API on Fujitsu FR1000 Multicore

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG2enc</td>
<td>1.00</td>
<td>1.94</td>
<td>3.76</td>
<td>3.75</td>
<td>2.55</td>
<td>2.55</td>
<td>3.47</td>
<td>3.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPEG2dec</td>
<td>1.00</td>
<td>1.80</td>
<td>2.12</td>
<td>2.54</td>
<td>1.94</td>
<td>1.94</td>
<td>2.81</td>
<td>3.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP3enc</td>
<td>1.00</td>
<td>1.98</td>
<td>2.90</td>
<td>3.75</td>
<td>2.54</td>
<td>2.54</td>
<td>3.76</td>
<td>3.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPEG 2000enc</td>
<td>1.00</td>
<td>1.98</td>
<td>2.90</td>
<td>3.75</td>
<td>2.55</td>
<td>2.55</td>
<td>3.47</td>
<td>3.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.38 times speedup on the average for 4 cores against a single core execution
Performance of OSCAR Compiler Using the Developed API on 4 core (SH4A) OSCAR Type Multicore

3.31 times speedup on the average for 4 cores against 1 core
Processing Performance on the Developed Multicore Using Automatic Parallelizing Compiler

Speedup against single core execution for audio AAC encoding

*) Advanced Audio Coding

<table>
<thead>
<tr>
<th>Numbers of processor cores</th>
<th>Speedups</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>1.9</td>
</tr>
<tr>
<td>4</td>
<td>3.6</td>
</tr>
<tr>
<td>8</td>
<td>5.8</td>
</tr>
</tbody>
</table>
Power Reduction by OSCAR Parallelizing Compiler for Secure Audio Encoding

AAC Encoding + AES Encryption with 8 CPU cores

Without Power Control (Voltage : 1.4V)

With Power Control (Frequency, Resume Standby: Power shutdown & Voltage lowering 1.4V-1.0V)

Avg. Power 5.68 [W] 88.3% Power Reduction Avg. Power 0.67 [W]
Power Reduction by OSCAR Parallelizing Compiler for MPEG2 Decoding

MPEG2 Decoding with 8 CPU cores

Without Power Control
(Voltage : 1.4V)

Avg. Power
5.73 [W]

73.5% Power Reduction

With Power Control
(Frequency, Resume Standby: Power shutdown & Voltage lowering 1.4V-1.0V)

Avg. Power
1.52 [W]
Low Power High Performance Multicore Computer with Solar Panel

- Clean Energy Autonomous
- Servers operational in deserts
OSCAR API-Applicable Heterogeneous Multicore Architecture

- DTU
  - Data Transfer Unit
- LPM
  - Local Program Memory
- LDM
  - Local Data Memory
- DSM
  - Distributed Shared Memory
- CSM
  - Centralized Shared Memory
- FVR
  - Frequency/Voltage Control Register
Static Scheduling of Coarse Grain Tasks for a Heterogeneous Multi-core
### An Image of Static Schedule for Heterogeneous Multi-core with Data Transfer Overlapping and Power Control

<table>
<thead>
<tr>
<th>CPU0</th>
<th>CPU1</th>
<th>CPU2</th>
<th>CPU3</th>
<th>DRP0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE</td>
<td>DTU</td>
<td>CORE</td>
<td>DTU</td>
<td>CORE</td>
</tr>
<tr>
<td>LOAD</td>
<td>MT1-2</td>
<td>LOAD</td>
<td>SEND</td>
<td>MT3-1</td>
</tr>
<tr>
<td>SEND</td>
<td>MT1-3</td>
<td>SEND</td>
<td>SEND</td>
<td>MT3-2</td>
</tr>
<tr>
<td>SEND</td>
<td>MT1-4</td>
<td>SEND</td>
<td>SEND</td>
<td>MT3-3</td>
</tr>
<tr>
<td>OFF</td>
<td>MT2-1</td>
<td>OFF</td>
<td>MT3-4</td>
<td>MT3-6</td>
</tr>
<tr>
<td>OFF</td>
<td>MT2-2</td>
<td>OFF</td>
<td>MT3-7</td>
<td>MT3-8</td>
</tr>
<tr>
<td>OFF</td>
<td>MT2-3</td>
<td>OFF</td>
<td>MT3-8</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**MTG1**

- MT1-1: LOAD
- MT1-2: LOAD
- MT1-3: SEND
- MT1-4: SEND

**MTG2**

- MT2-1: LOAD
- MT2-2: SEND
- MT2-3: SEND
- MT2-4: SEND
- MT2-5: SEND
- MT2-6: SEND
- MT2-7: Send
- MT2-8: Store

**MTG3**

- MT3-1: LOAD
- MT3-2: LOAD
- MT3-3: LOAD
- MT3-4: SEND
- MT3-5: SEND
- MT3-6: SEND

**DRP0**

- OFF

**Legend**

- LOAD
- STORE
- SEND
Heterogeneous Multicore RP-X
presented in SSCC2010 Processors Session on Feb. 8, 2010

Renesas, Hitachi, Tokyo Inst. Of Tech. & Waseda Univ.
Parallel Processing Performance Using OSCAR Compiler and OSCAR API on RP-X (Optical Flow with a hand-tuned library)

CPU performs data transfers between SH and FE

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Speedups against a single SH processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1SH</td>
<td>1</td>
</tr>
<tr>
<td>2SH</td>
<td>2.29</td>
</tr>
<tr>
<td>4SH</td>
<td>3.09</td>
</tr>
<tr>
<td>8SH</td>
<td>5.4</td>
</tr>
<tr>
<td>2SH+1FE</td>
<td>18.85</td>
</tr>
<tr>
<td>4SH+2FE</td>
<td>26.71</td>
</tr>
<tr>
<td>8SH+4FE</td>
<td>32.65</td>
</tr>
</tbody>
</table>

3.5 [fps]
Power Reduction in a real-time execution controlled by OSCAR Compiler and OSCAR API on RP-X (Optical Flow with a hand-tuned library)

Without Power Reduction

70% of power reduction

With Power Reduction by OSCAR Compiler

Average: 1.76[W]

Average: 0.54[W]

1 cycle: 33[ms] → 30[fps]
8 Core RP2 Chip Block Diagram

On-chip system bus (SuperHyway)

LCPG: Local clock pulse generator
PCR: Power Control Register
CCN/BAR: Cache controller/Barrier Register
URAM: User RAM (Distributed Shared Memory)
Software Cache Coherent Control by OSCAR Compiler and API on RP2

![Graph showing speedups against 1 processor for AAC Encoder, MPEG2 Decoder, and MPEG2 Encoder with varying number of processors.](image-url)
Performance of OSCAR Compiler on IBM p6 595
Power6 (4.2GHz) based 64-core SMP Server

Compile Option:
- Sequential: -O5 -bmaxdata:64000000000 -q64 -qarch=pwr6
- XLF: -O5 -qsmp=auto -bmaxdata:64000000000 -q64 -qarch=pwr6
- OSCAR: -O5 -qsmp=noauto -bmaxdata:64000000000 -q64 -qarch=pwr6

<table>
<thead>
<tr>
<th>SPEC 2000 swim</th>
<th>IBM XL Fortran for AIX Ver.12.1</th>
<th>OSCAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>1.7</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>3.3</td>
<td>2.8</td>
</tr>
<tr>
<td>8</td>
<td>6.9</td>
<td>8.6</td>
</tr>
<tr>
<td>16</td>
<td>12.0</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>14.5</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>23.5</td>
<td>31.0</td>
</tr>
</tbody>
</table>
Fujitsu’s Vector Manycore Design Evaluated in NEDO Manycore Leading Research, Feb. 2010

各アプリ領域の処理を効率良く実行するための方式

Copyright 2010 FUJITSU LIMITED
4.3 低電力マルチコア実装技術（1）

◆ 提案した64コア、Fat-treeアーキテクチャにて、以下のような処理負荷に連動した階層的な低電力制御を行うことで、メニーコアチップ低消費電力実現

Fat-Tree構成メニーコア低電力制御例

(1) CPUコア単位で周波数制御停止

(2) クラスタ単位で停止

(3) 4クラスタ、ルータ、メモリコントローラをすべて停止

(4) 高速メモリI/0も停止
Green Computing Systems R&D Center
Waseda University
Supported by METI (Mar. 2011 Completion)

<R & D Target>
Hardware, Software, Application for Super Low-Power Manycore Processors
- More than 64 cores
- Natural air cooling (No fan)
  - Cool, Compact, Clear, Quiet
- Operational by Solar Panel

<Industry, Government, Academia>
Hitachi, Fujitsu, NEC, Renesas, Olympus, Toyota, Denso, Mitsubishi, Toshiba, etc

<Ripple Effect>
- Low CO₂ (Carbon Dioxide) Emissions
- Creation Value Added Products
  - Consumer Electronics, Automobiles, Servers

Beside Subway Waseda Station, Near Waseda Univ. Main Campus
Research, development and practical utilization through industry-government-academia partnerships (spillover effect)

Environment

Industry

Lives

Cancer Treatment
Carbon Ion Radiotherapy
Cancer Treatment
Carbon Ion Radiotherapy

National Institute of Radiological Sciences (NIRS)

5.58 times speedup by 8 processors
Intel Quadcore Xeon 8 core SMP

5.78 times speedup by 8 processors
IBM Power 7 8 core SMP (Hitachi SR16000)
Conclusions

- OSCAR compiler cooperative real-time low power multicore with high effective performance, short software development period will be important in wide range of IT systems from consumer electronics to automobiles, medical systems, disaster super-realtime simulator (Tsunami), and EX-FLOPS machines.

- For industry
  - A few minutes of compilation of C program using OSCAR Compiler and API without months of programming allows us
    - Several times speedups on market available SMP servers.
    - Scalable speedup on various multicores like 8 core homogeneous RP2 (8SH4A), 15 core heterogeneous RPX (8SH4A, 4FEGA, MX2 & 1VPE), MPCore, FR1000 and so on.
    - 70% power reduction on RP2 and RPX for realtime media processing.
  - OSCAR green compiler, API, multicores and manycores will be continuously developed for saving lives from natural disasters and sickness like cancer.