

OSCAR Parallelizing and Power Reducing Compiler for Multicores

Hironori Kasahara

Professor, Dept. of Computer Science & Engineering

Director, Advanced Multicore Processor Research Institute

Waseda University (早稲田大学), Tokyo, Japan

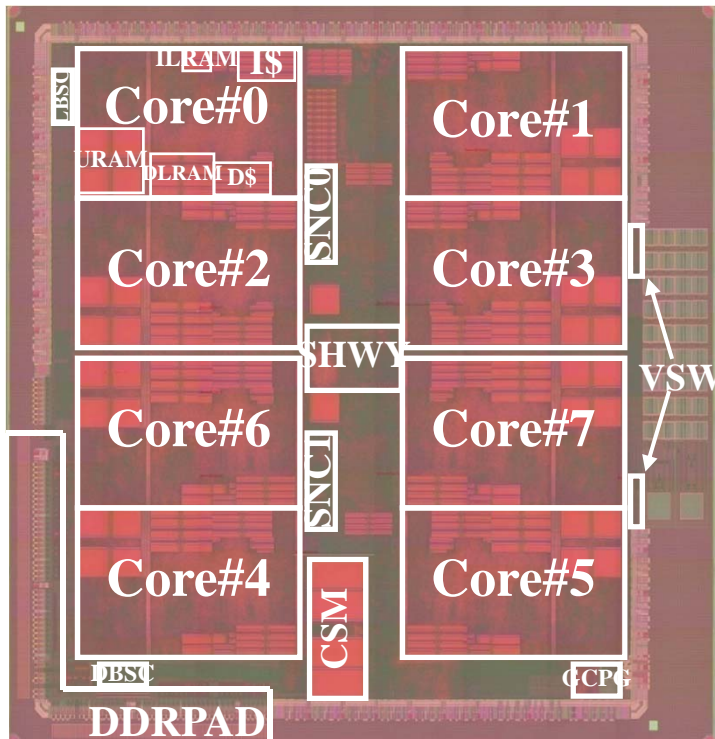
IEEE Computer Society

President Elect 2017, President 2018

URL: <http://www.kasahara.cs.waseda.ac.jp/>

Multicores for Performance and Low Power

Power consumption is one of the biggest problems for performance scaling from smartphones to cloud servers and supercomputers (“K” more than 10MW) .



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”

$$\text{Power} \propto \text{Frequency} * \text{Voltage}^2$$

(Voltage \propto Frequency)

➔ Power \propto Frequency³

If Frequency is reduced to 1/4
(Ex. 4GHz \rightarrow 1GHz),
Power is reduced to 1/64 and
Performance falls down to 1/4 .

<Multicores>

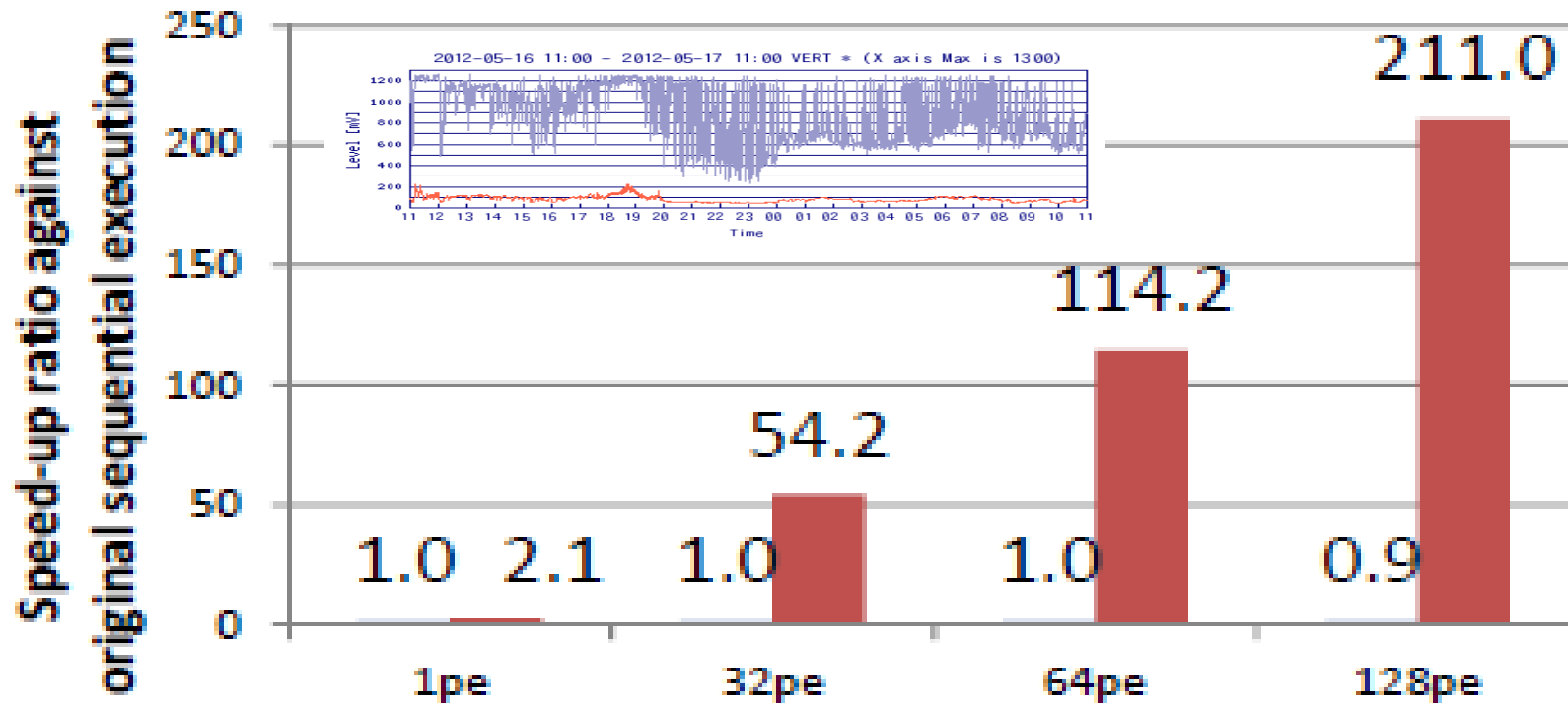
If 8cores are integrated on a chip,
Power is still 1/8 and
Performance becomes 2 times .



Earthquake Simulation "GMS" on Fujitsu M9000 Sparc CC-NUMA Server



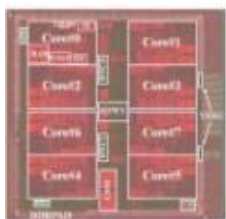
■ original (sun studio) ■ proposed method



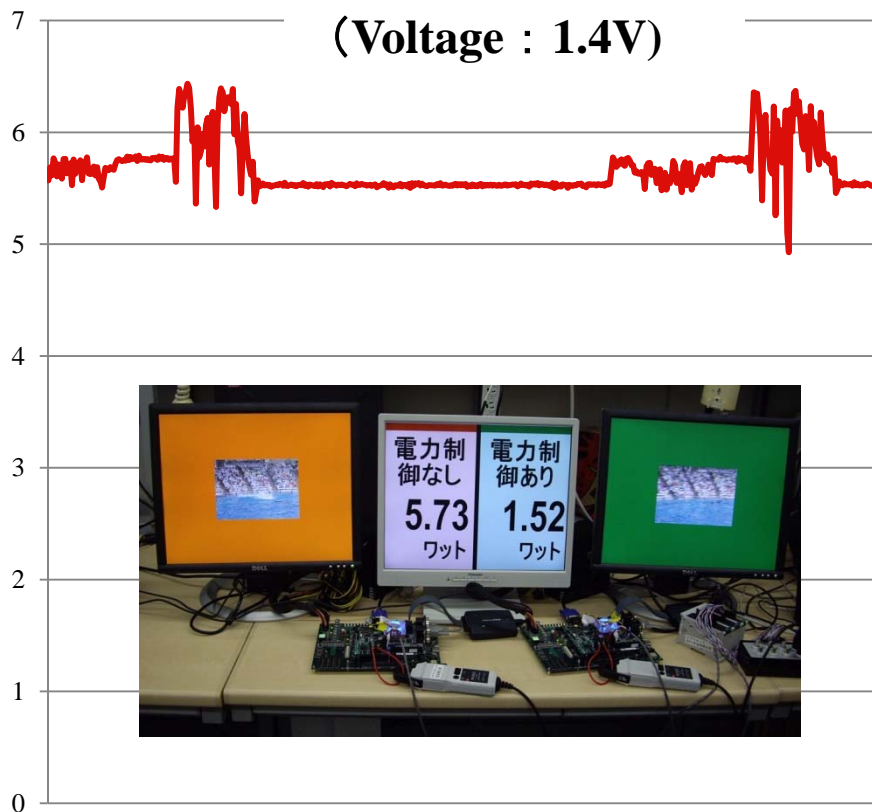
With 128 cores, OSCAR compiler gave us 100 times speedup against 1 core execution and 211 times speedup against 1 core using Sun (Oracle) Studio compiler.

Power Reduction of MPEG2 Decoding to 1/4 on 8 Core Homogeneous Multicore RP-2 by OSCAR Parallelizing Compiler

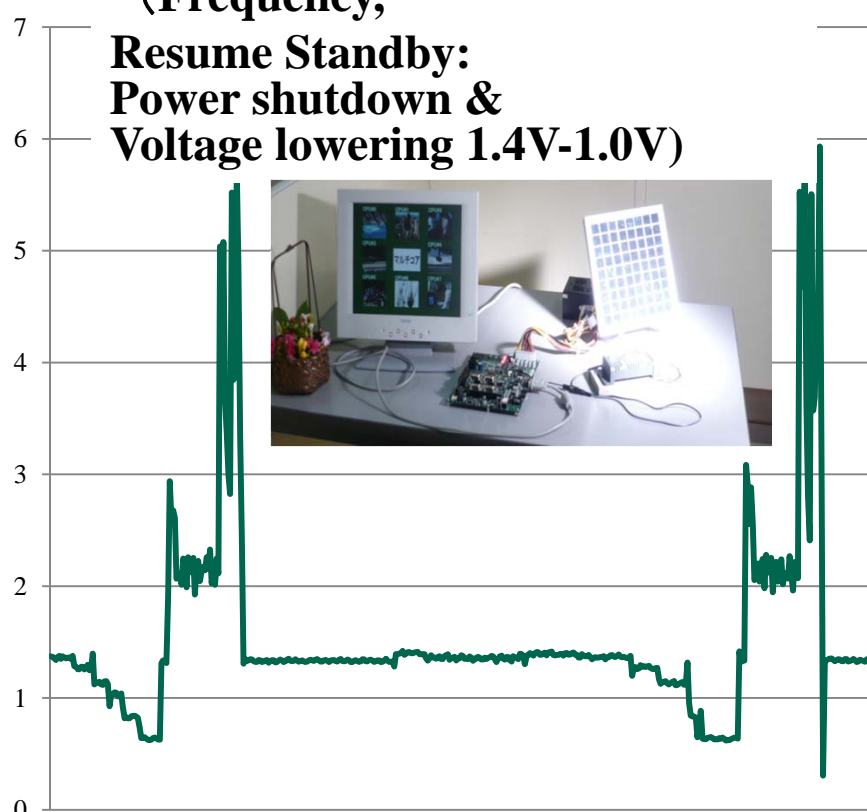
MPEG2 Decoding with 8 CPU cores



Without Power Control
(Voltage : 1.4V)



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



Avg. Power
5.73 [W]

73.5% Power Reduction



Avg. Power
1.52 [W]

Demo of NEDO Multicore for Real Time Consumer Electronics at the Council of Science and Engineering Policy on April 10, 2008

第74回総合科学技術会議【平成20年4月10日】



第74回総合科学技術会議の様子(1)



第74回総合科学技術会議の様子(2)



第74回総合科学技術会議の様子(3)



第74回総合科学技術会議の様子(4)

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

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

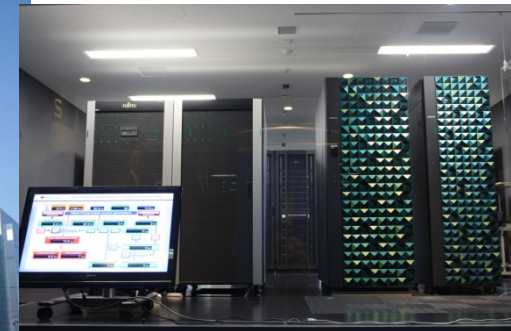
- More than 64 cores
- Natural air cooling (No fan)
Cool, Compact, Clear, Quiet
- Operational by Solar Panel

<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



Hitachi SR16000:

Power7 128coreSMP

Fujitsu M9000

SPARC VII 256 core SMP



Beside Subway Waseda Station,
Near Waseda Univ. Main Campus

Industry-government-academia collaboration in R&D and target practical applications

Protect Lives

For smart life

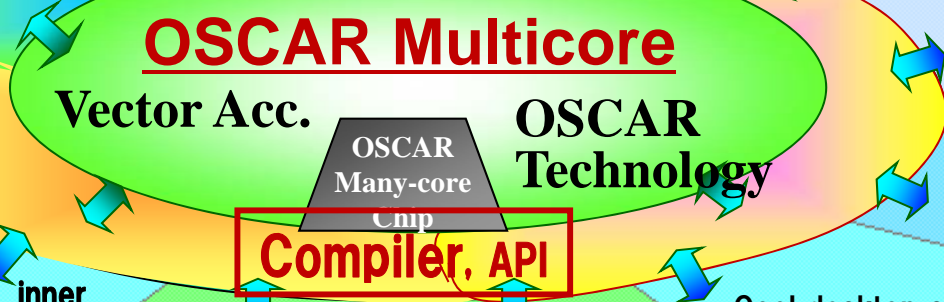


Protect Environment

Waseda University :R&D
Many-core system technologies with ultra-low power consumption



Green supercomputers



Disaster Survival Supercomputer (Earthquakes, tsunami, Fire-spreading)

Multicore Engine ECU, ADAS (Driver Assistance), Self Driving, HV, EV, FCV

Consumer electronic Internet TV/DVD



Camcorders
Capsule inner cameras
Cameras



Solar Powered Smart phones



Operation/recharging by solar cells



Medical servers

Heavy particle radiation planning, cerebral infarction)

National Institute of Radiological Sciences

Cool desktop servers



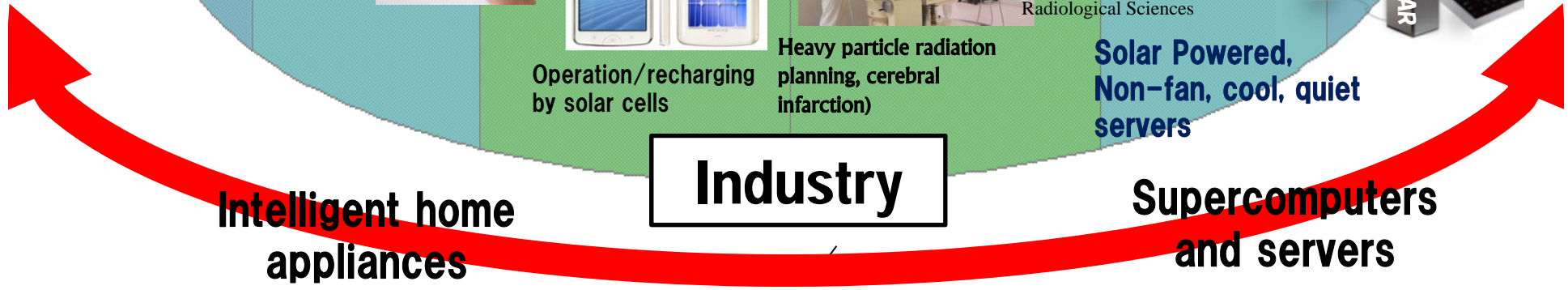
Solar Powered, Non-fan, cool, quiet servers



Industry

Intelligent home appliances

Supercomputers and servers



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

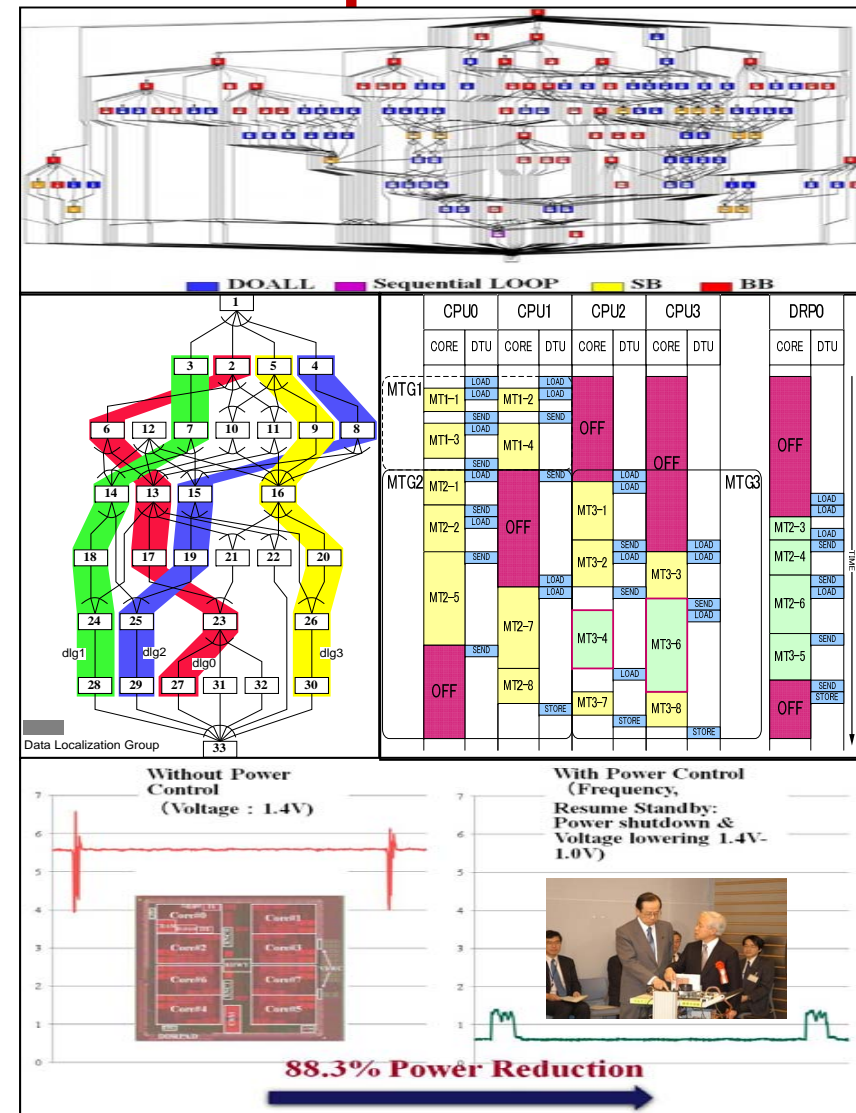
Automatic data management for distributed shared memory, cache and local memory

Data Transfer Overlapping

Data transfer overlapping using Data Transfer Controllers (DMAs)

Power Reduction

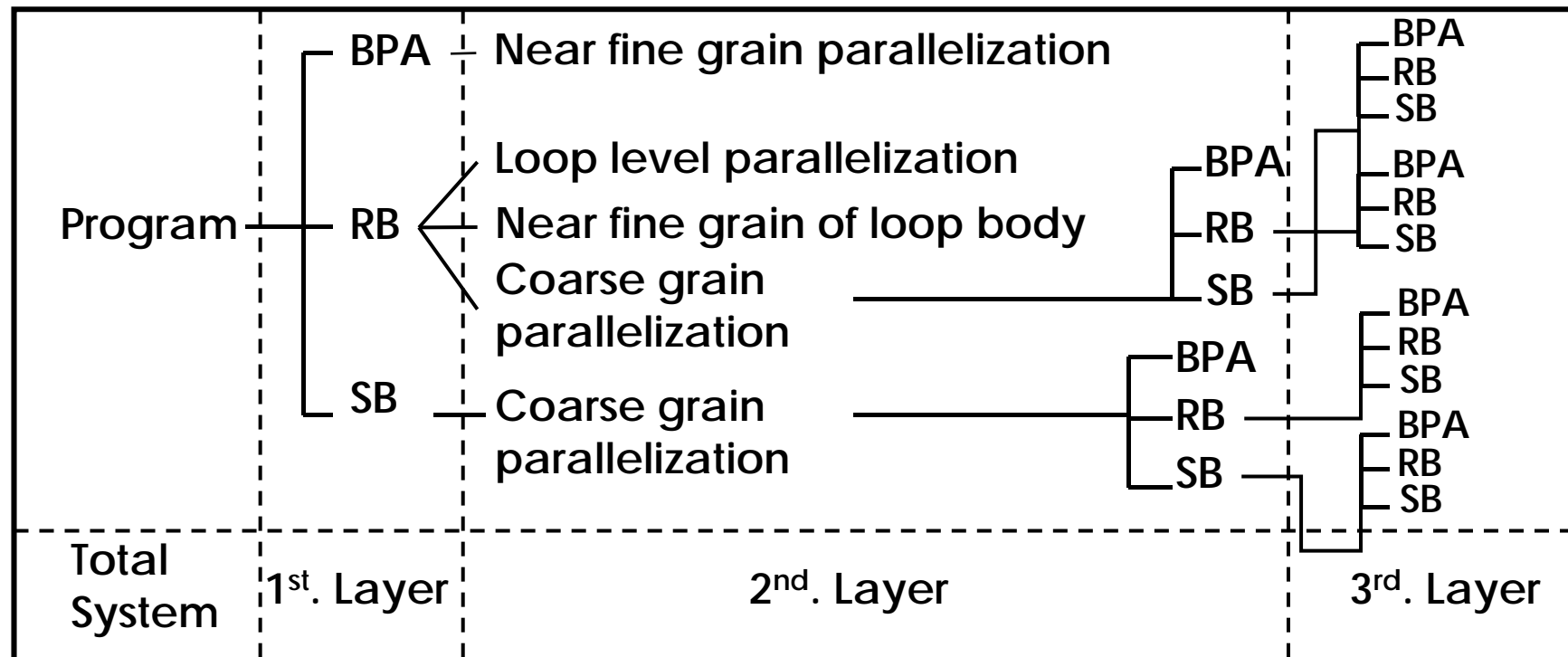
Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.



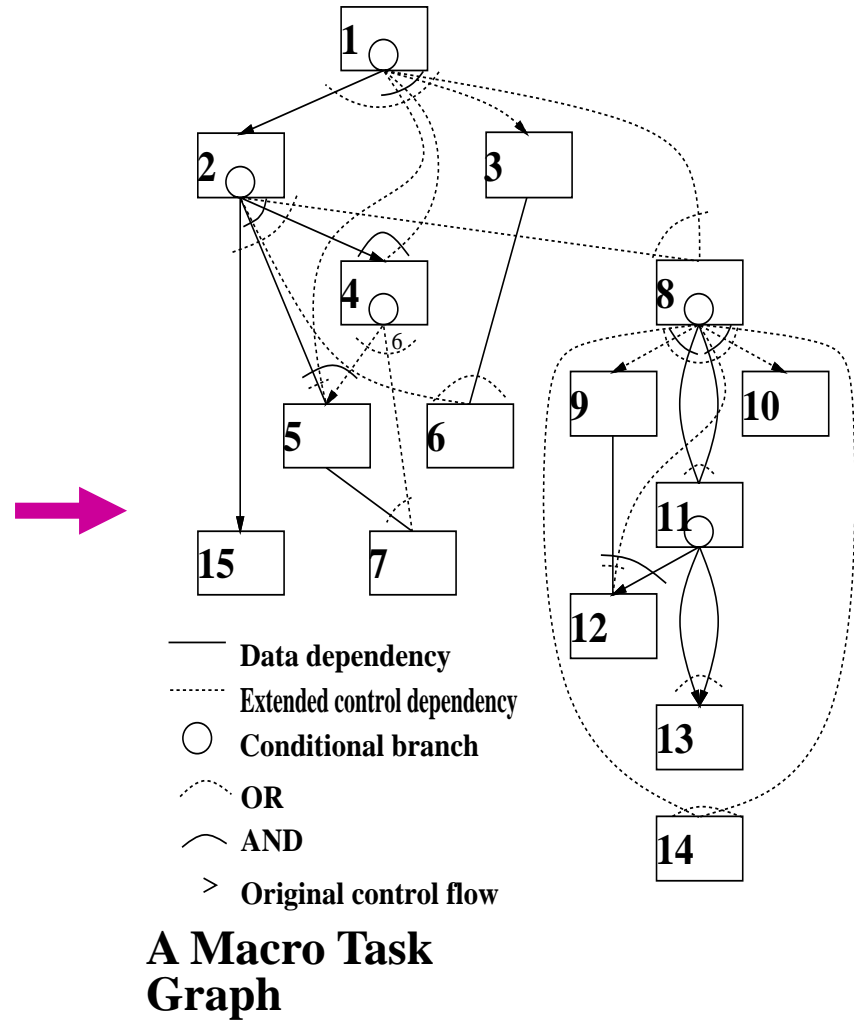
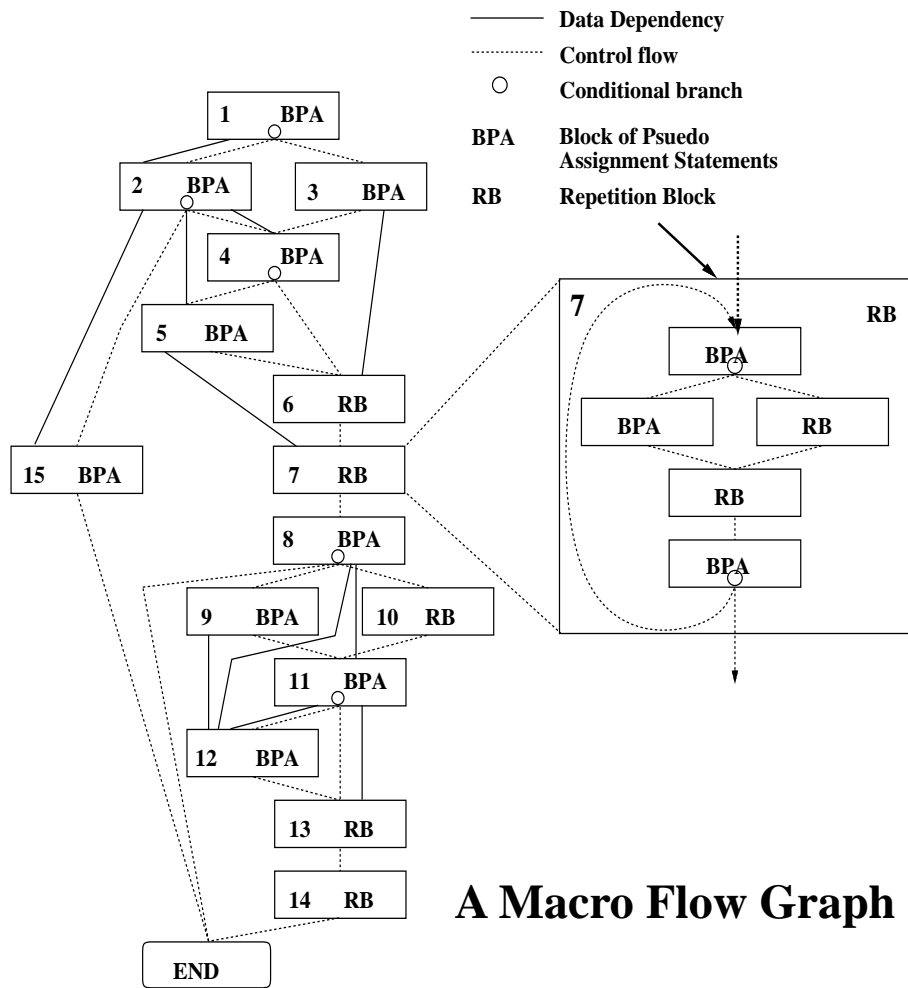
Generation of Coarse Grain Tasks

■ Macro-tasks (MTs)

- **Block of Pseudo Assignments (BPA): Basic Block (BB)**
- **Repetition Block (RB) : natural loop**
- **Subroutine Block (SB): subroutine**



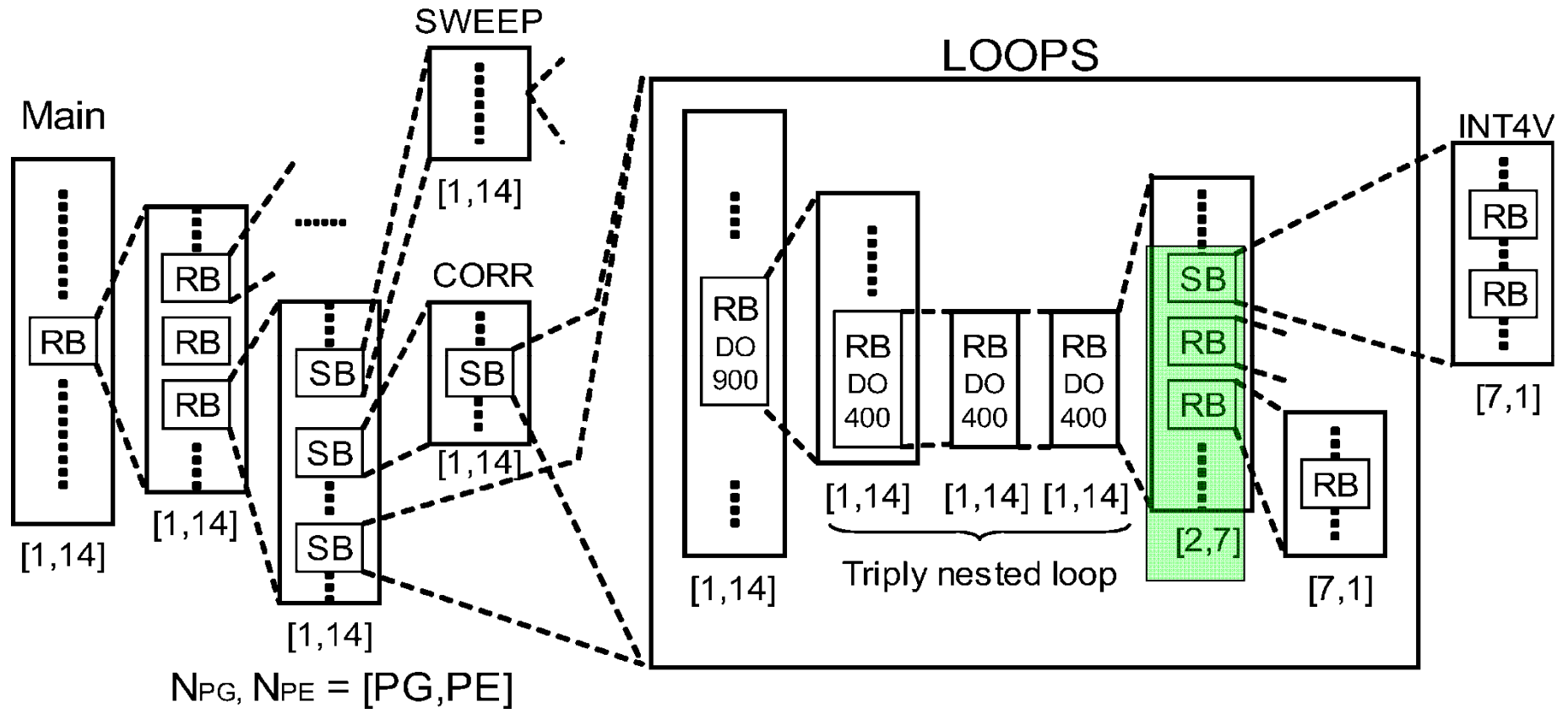
Earliest Executable Condition Analysis for Coarse Grain Tasks (Macro-tasks)



Automatic processor assignment in 103.su2cor

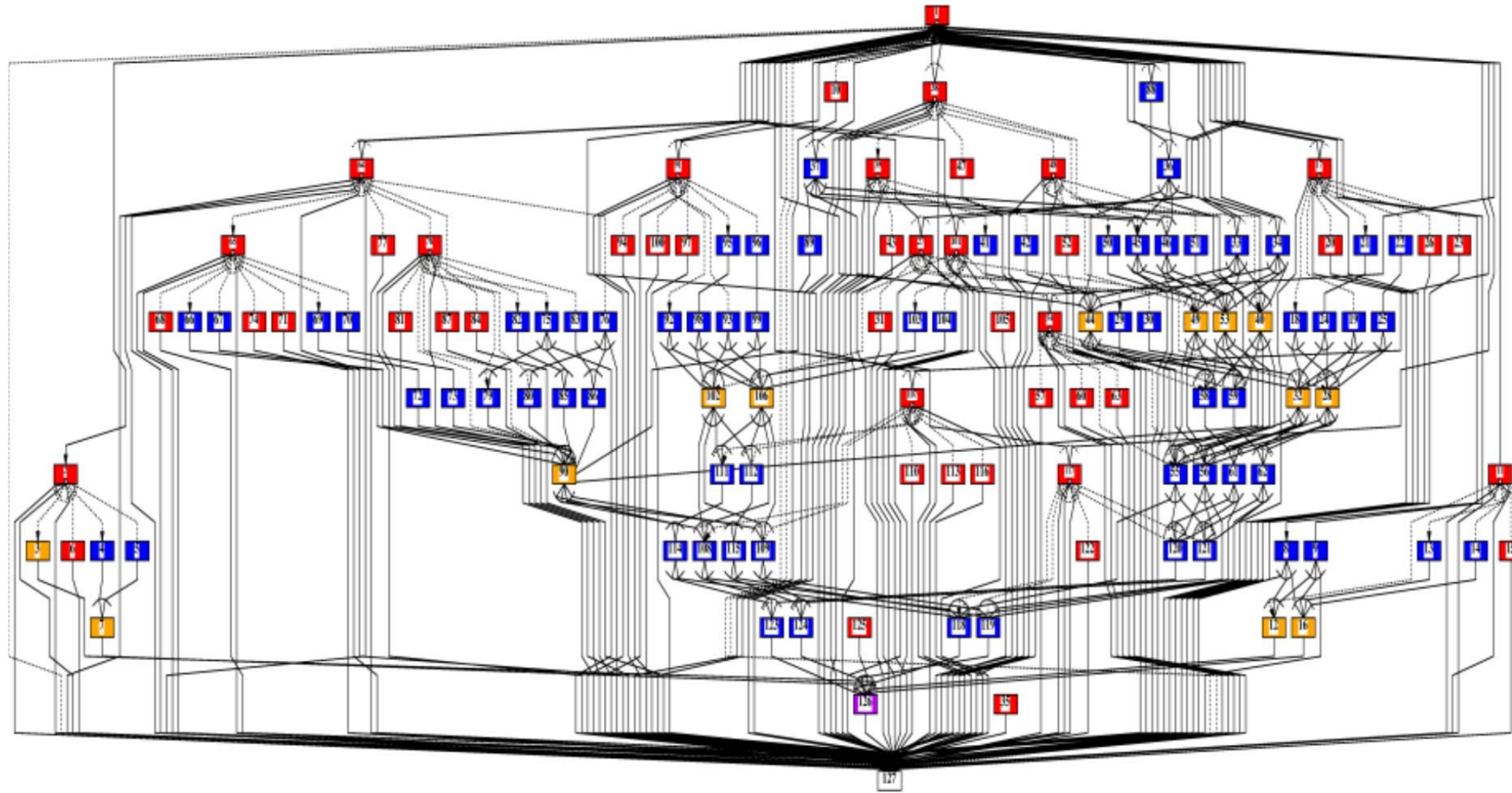
- Using 14 processors

Coarse grain parallelization within DO400



MTG of Su2cor-LOOPS-DO400

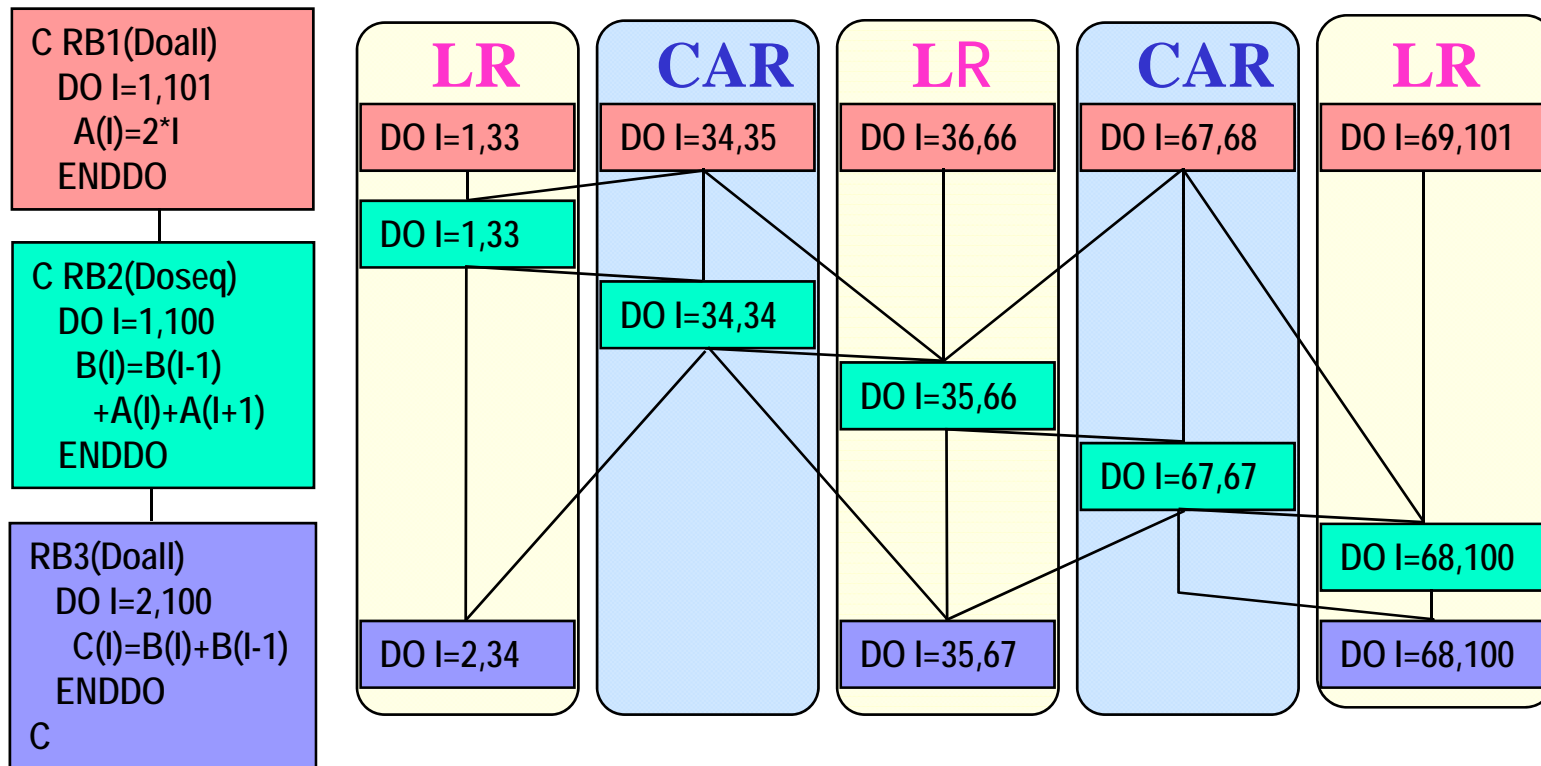
- Coarse grain parallelism $\text{PARA_ALD} = 4.3$



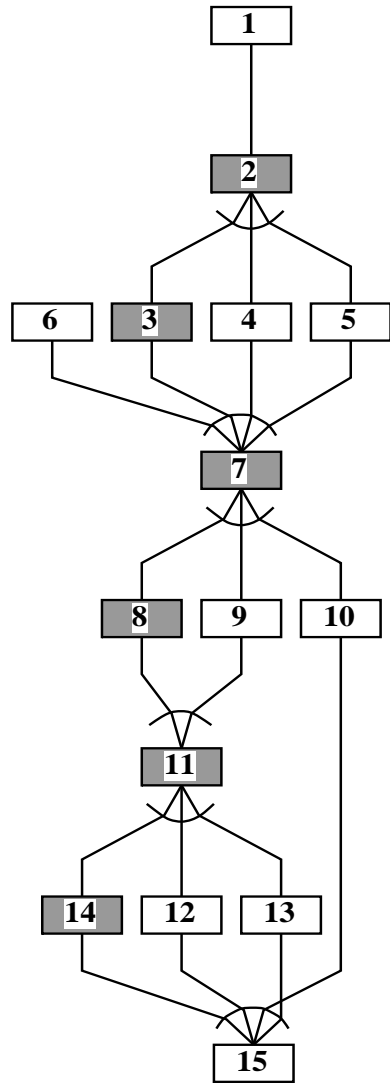
■ DOALL ■ Sequential LOOP ■ SB ■ BB

Data-Localization: Loop Aligned Decomposition

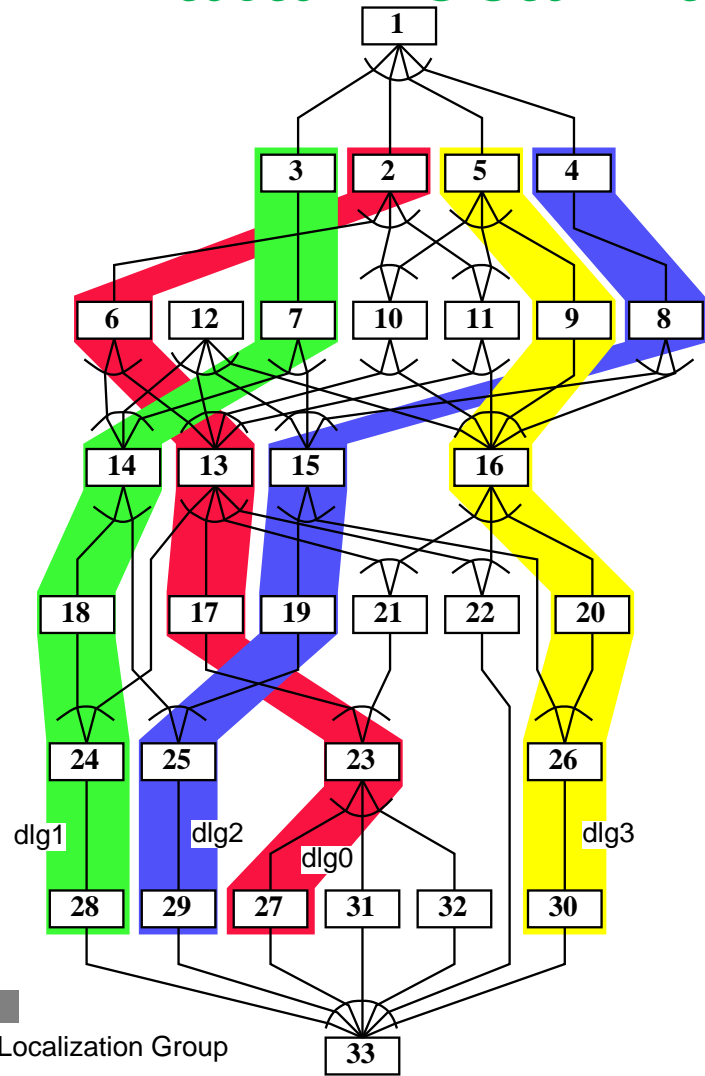
- Decompose multiple loop (Doall and Seq) into **CARs** and **LRs** considering inter-loop data dependence.
 - Most data in **LR** can be passed through LM.
 - LR: Localizable Region, CAR: Commonly Accessed Region**



Data Localization



MTG



MTG after Division

■ Data Localization Group

PE0	PE1
12	1
2	3
6	7
4	14
8	18
15	5
19	9
25	11
29	10
13	16
17	20
22	26
21	30
23	24
27	28
	32
	31

A schedule for two processors

An Example of Data Localization for Spec95 Swim

```

DO 200 J=1,N
DO 200 I=1,M
  UNEW(I+1,J) = UOLD(I+1,J)+
1  TDTS8*(Z(I+1,J+1)+Z(I+1,J))*(CV(I+1,J+1)+CV(I,J+1)+CV(I,J)
2  +CV(I+1,J))-TDTSDX*(H(I+1,J)-H(I,J))
  VNEW(I,J+1) = VOLD(I,J+1)-TDTSD8*(Z(I+1,J+1)+Z(I,J+1))
1  *(CU(I+1,J+1)+CU(I,J+1)+CU(I,J)+CU(I+1,J))
2  -TDTSDY*(H(I,J+1)-H(I,J))
  PNEW(I,J) = POLD(I,J)-TDTSDX*(CU(I+1,J)-CU(I,J))
1  -TDTSDY*(CV(I,J+1)-CV(I,J))
200 CONTINUE
  
```

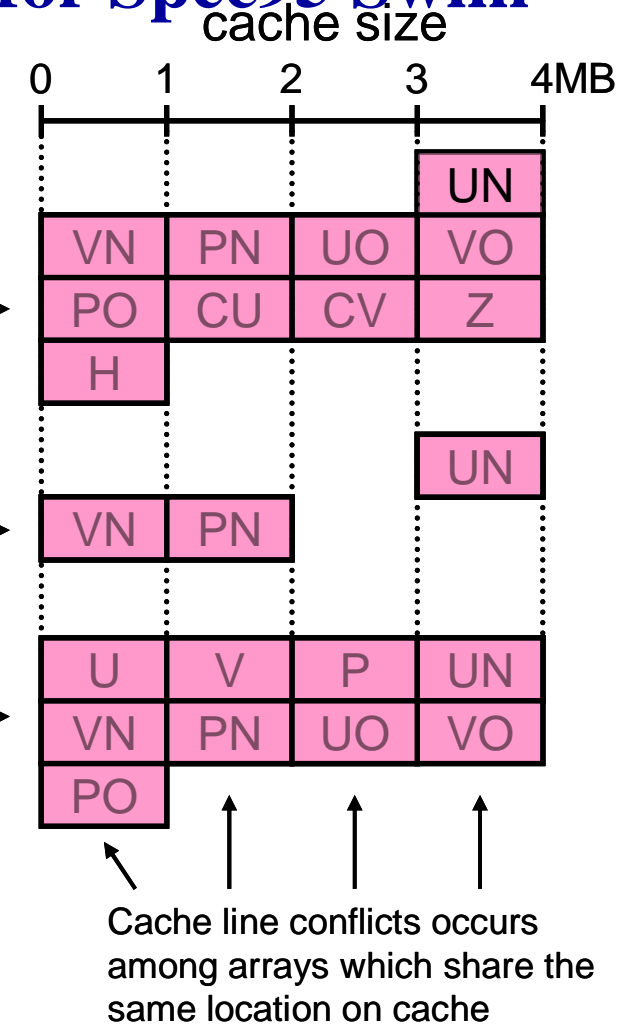
```

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
  
```

```

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
  
```

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



(b) 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

after padding

PARAMETER (N1=513, N2=513)

PARAMETER (N1=513, N2=544)

COMMON U(N1,N2), V(N1,N2), P(N1,N2),

COMMON U(N1,N2), V(N1,N2), P(N1,N2),

* UNEW(N1,N2), VNEW(N1,N2),

* UNEW(N1,N2), VNEW(N1,N2),

1 PNEW(N1,N2), UOLD(N1,N2),

1 PNEW(N1,N2), UOLD(N1,N2),

* VOLD(N1,N2), POLD(N1,N2),

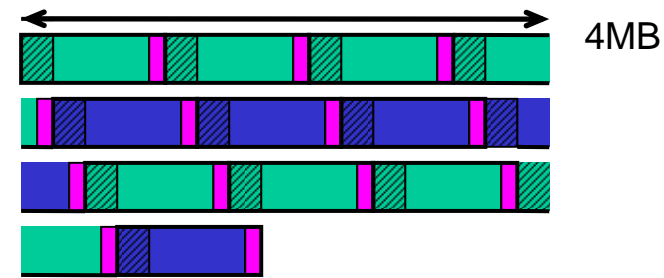
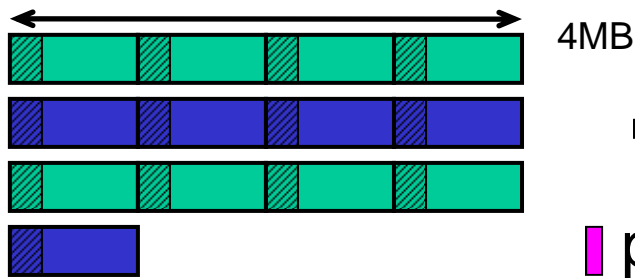
* VOLD(N1,N2), POLD(N1,N2),

2 CU(N1,N2), CV(N1,N2),

2 CU(N1,N2), CV(N1,N2),

* Z(N1,N2), H(N1,N2)

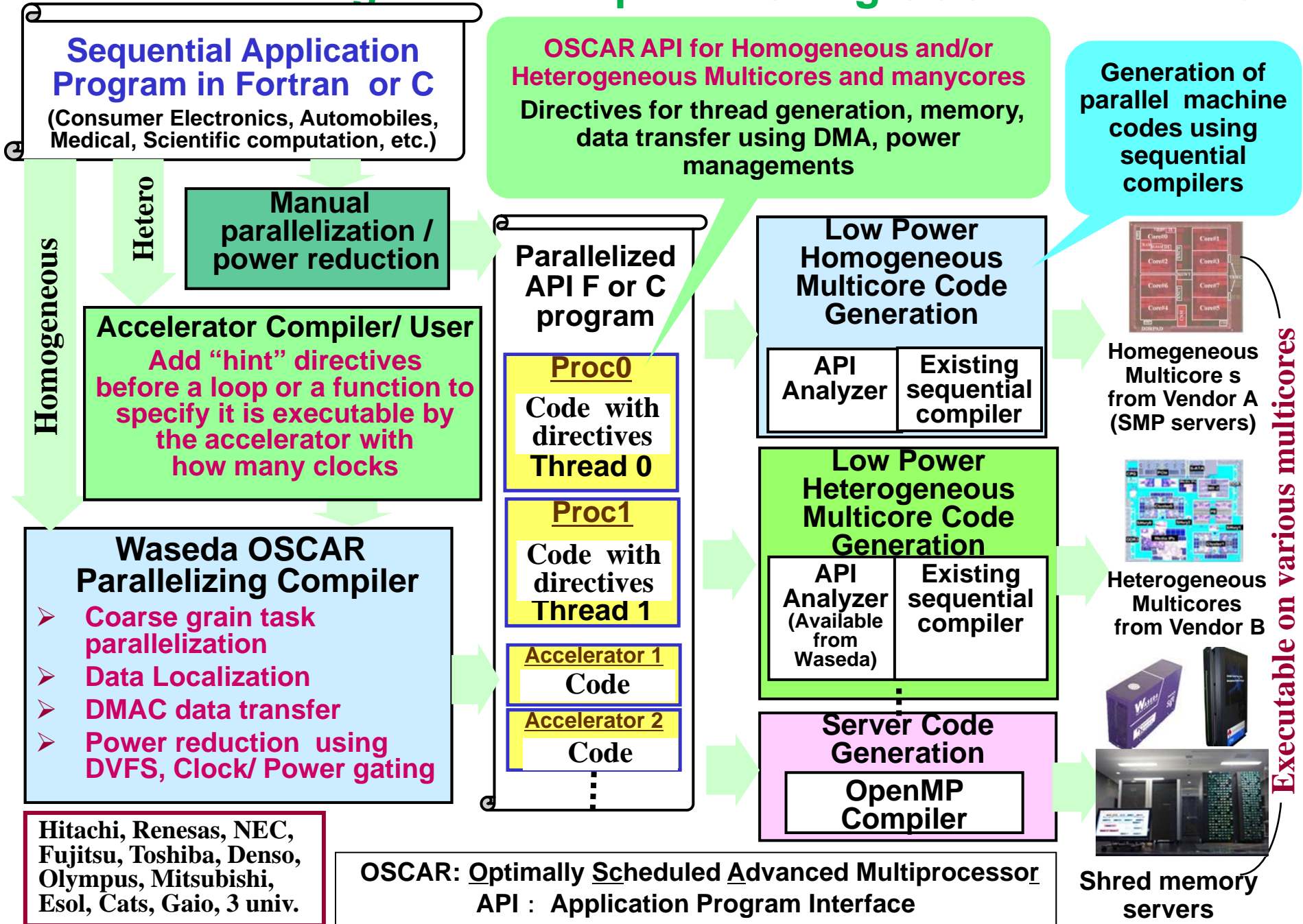
* Z(N1,N2), H(N1,N2)



padding

Box: Access range of DLG0

Multicore Program Development Using OSCAR API V2.0

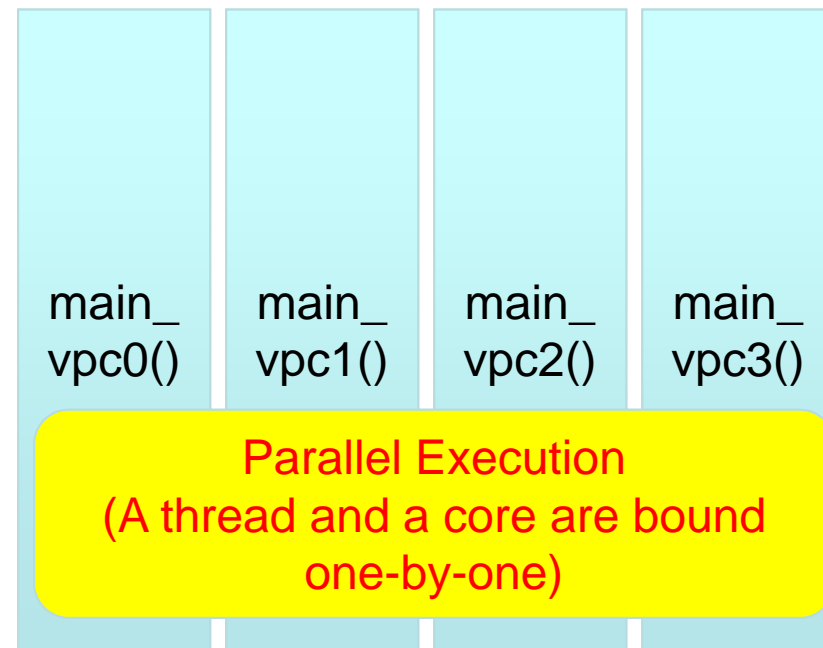
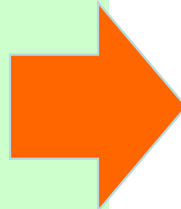


Parallel Execution

- **Start of parallel execution**
 - **#pragma omp parallel sections (C)**
 - **!\$omp parallel sections (Fortran)**
- **Specifying critical section**
 - **#pragma omp critical (C)**
 - **!\$omp critical (Fortran)**
- **Enforcing an order of the memory operations**
 - **#pragma omp flush (C)**
 - **!\$omp flush (Fortran)**
- **These are from OpenMP.**

Thread Execution Model

```
#pragma omp parallel sections  
{  
#pragma omp section  
main_vpc0();  
#pragma omp section  
main_vpc1();  
#pragma omp section  
main_vpc2();  
#pragma omp section  
main_vpc3();  
}
```



VPC: Virtual Processor Core

Memory Mapping

- **Placing variables on an onchip centralized shared memory (onchipCSM)**
 - `#pragma oscar onchipshared (C)`
 - `!$oscar onchipshared (Fortran)`
- **Placing variables on a local data memory (LDM)**
 - `#pragma omp threadprivate (C)`
 - `!$omp threadprivate (Fortran)`
 - This directive is an extension to OpenMP
- **Placing variables on a distributed shared memory (DSM)**
 - `#pragma oscar distributedshared (C)`
 - `!$oscar distributedshared (Fortran)`

Data Transfer

- Specifying **data transfer lists**
 - `#pragma oscar dma_transfer` (C)
 - `!$oscar dma_transfer` (Fortran)
 - Containing following parameter directives
- Specifying **a contiguous data transfer**
 - `#pragma oscar dma_contiguous_parameter` (C)
 - `!$oscar dma_contiguous_parameter` (Fortran)
- Specifying **a stride data transfer**
 - `#pragma oscar dma_stride_parameter`
 - `!$oscar dma_stride_parameter`
 - This can be used for scatter/gather data transfer
- **Data transfer synchronization**
 - `#pragma oscsar dma_flag_check`
 - `!$oscar dma_flag_check`

Power Control

- Making a module into specifying frequency and voltage state

- `#pragma oscar fvcontrol (C)`

- `!$oscar fvcontrol (Fortran)`

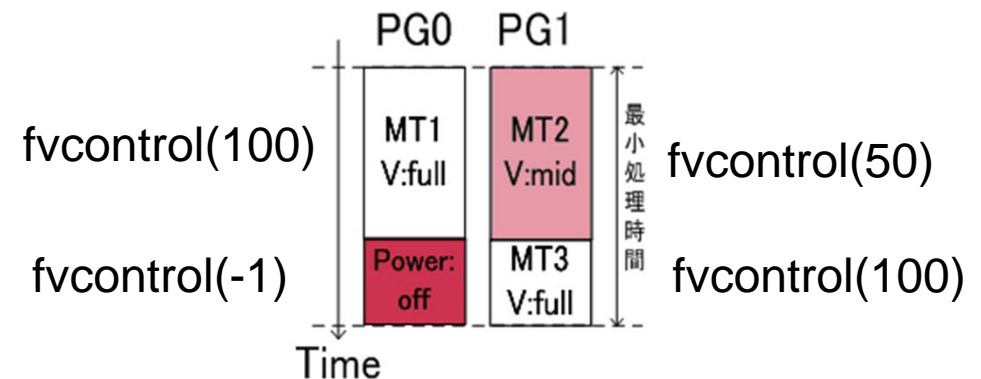
- state examples

- 100: max frequency

- 50: half frequency

- 0: clock off

- -1: power off



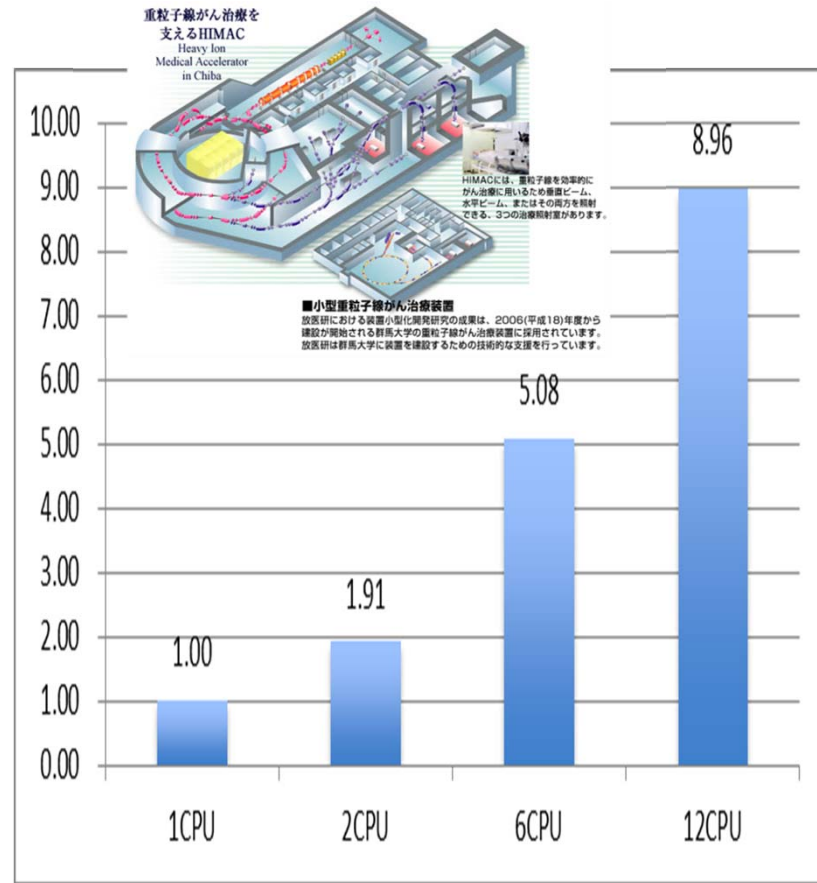
- Getting a frequency and voltage state of a module

- `#pragma oscar get_fvstatus (C)`

- `!$oscar get_fvstatus (Fortran)`

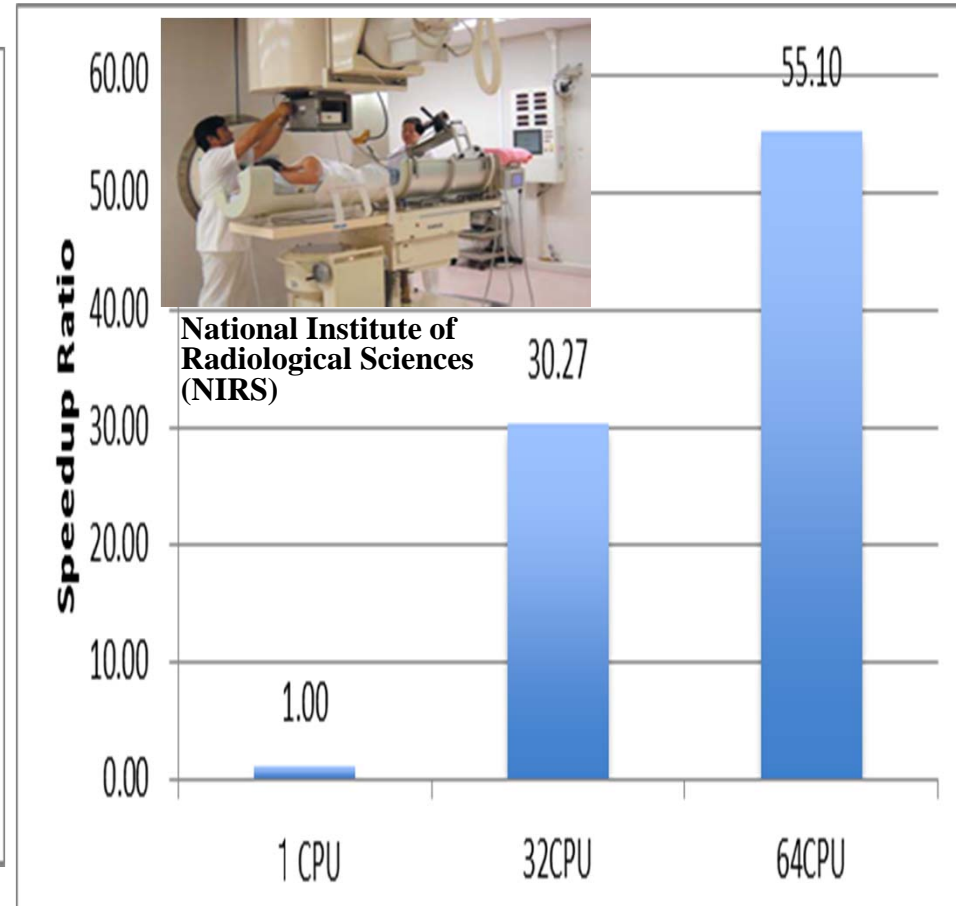
Cancer Treatment Carbon Ion Radiotherapy

(Previous best was 2.5 times speedup on 16 processors with hand optimization)



8.9times speedup by 12 processors

Intel Xeon X5670 2.93GHz 12 core SMP (Hitachi HA8000)

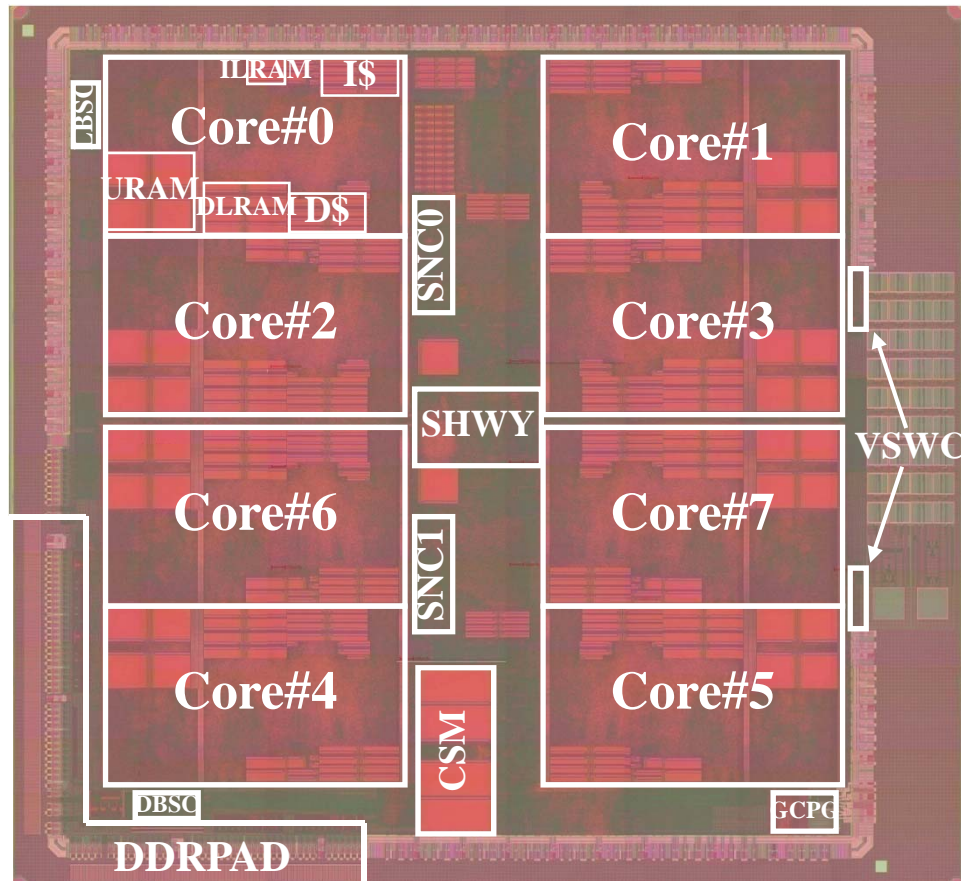


55 times speedup by 64 processors

IBM Power 7 64 core SMP (Hitachi SR16000)

Renesas-Hitachi-Waseda Low Power 8 core RP2

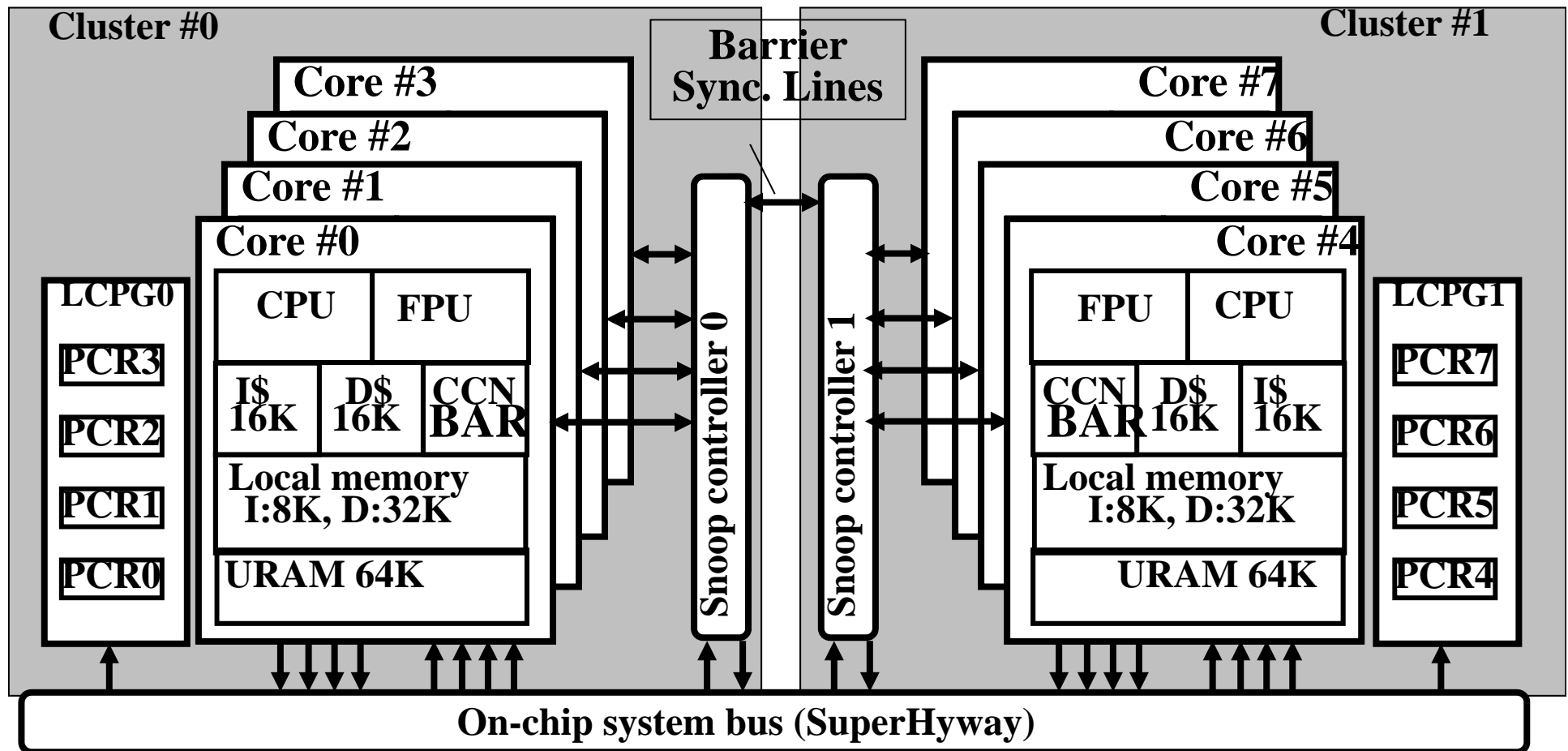
Developed in 2007 in METI/NEDO project



Process Technology	90nm, 8-layer, triple-Vth, CMOS
Chip Size	104.8mm ² (10.61mm x 9.88mm)
CPU Core Size	6.6mm ² (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”

8 Core RP2 Chip Block Diagram



LCPG: Local clock pulse generator
PCR: Power Control Register
CCN/BAR: Cache controller/Barrier Register
URAM: User RAM (Distributed Shared Memory)

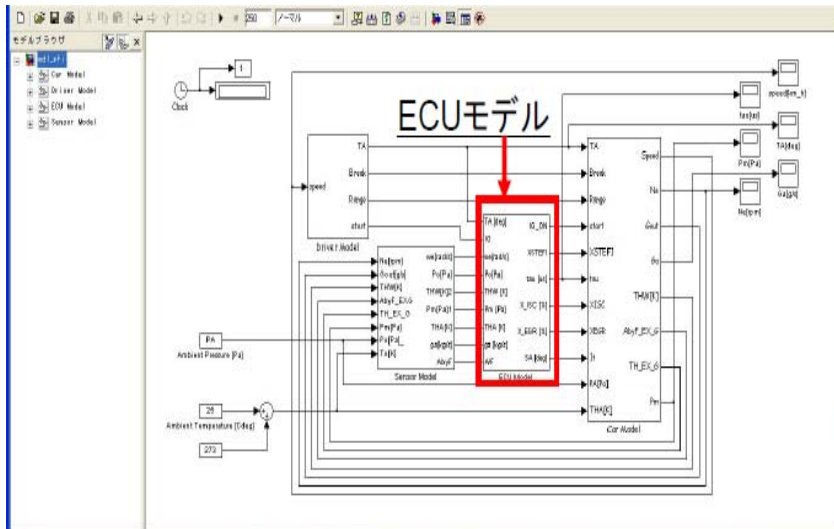
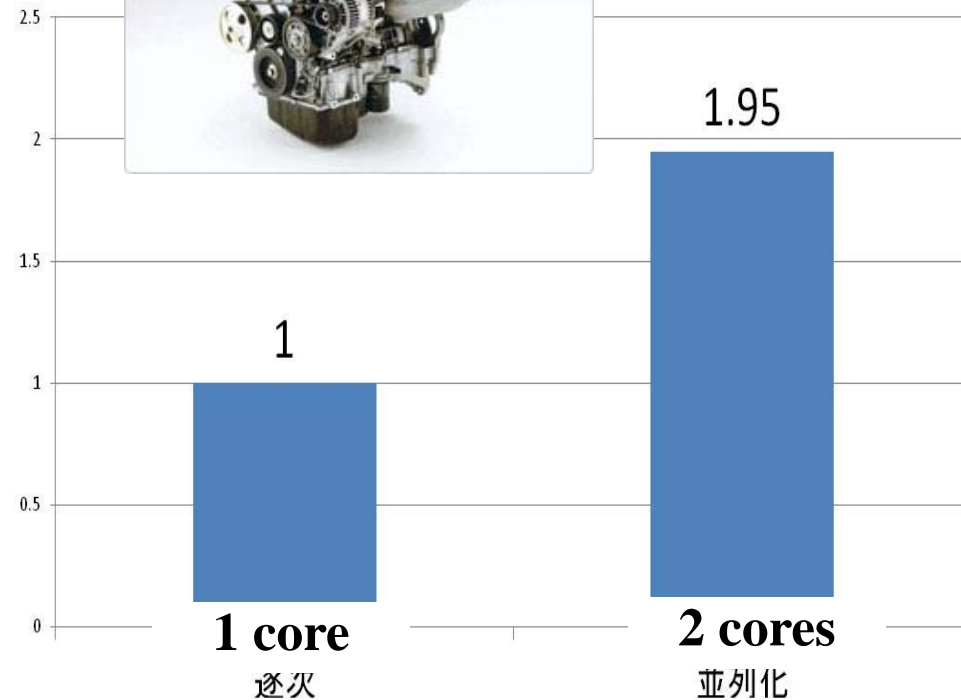


Engine Control by multicore with Denso

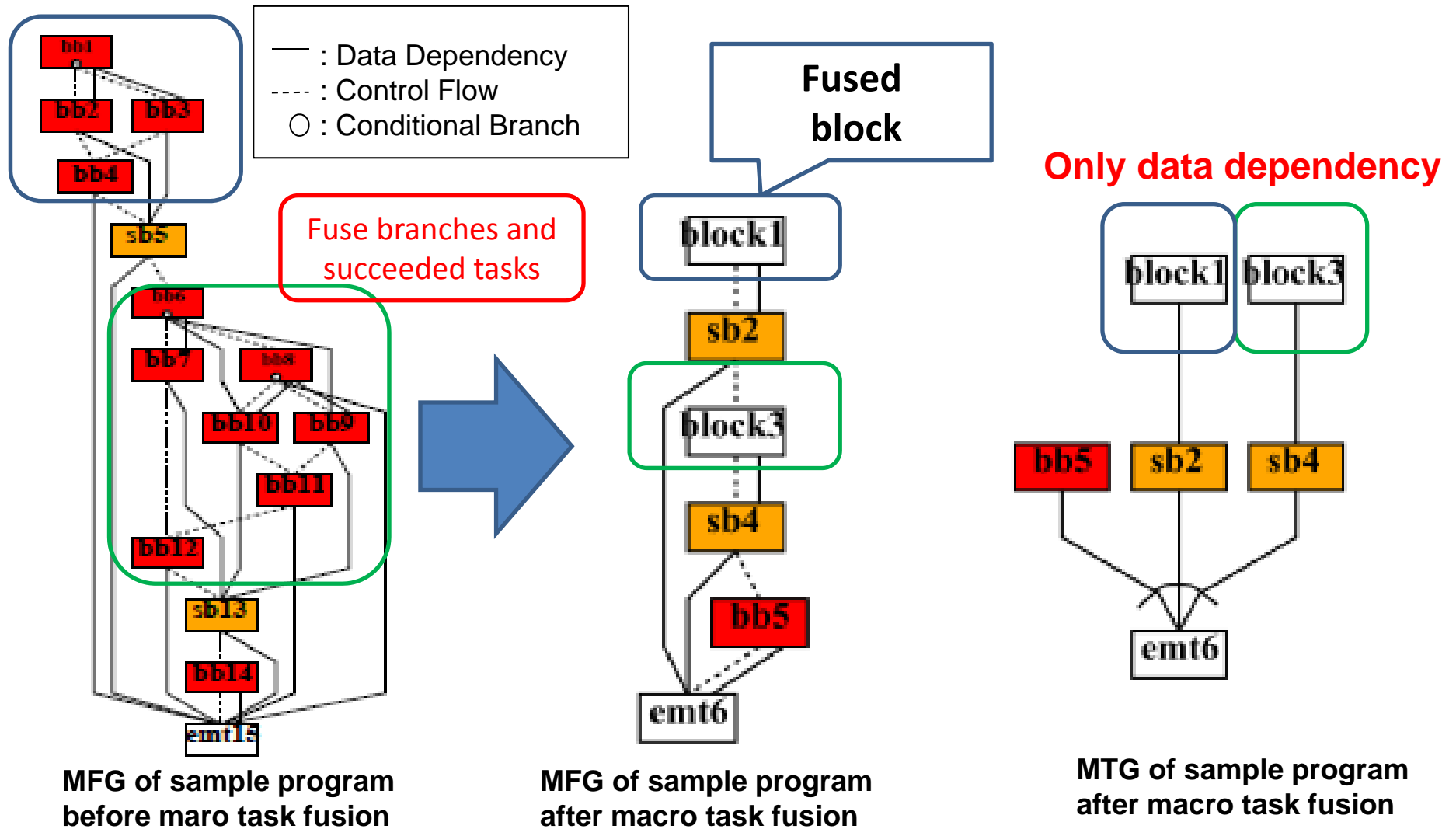
Though so far parallel processing of the engine control on multicore has been very difficult, Denso and Waseda succeeded 1.95 times speedup on 2core V850 multicore processor.



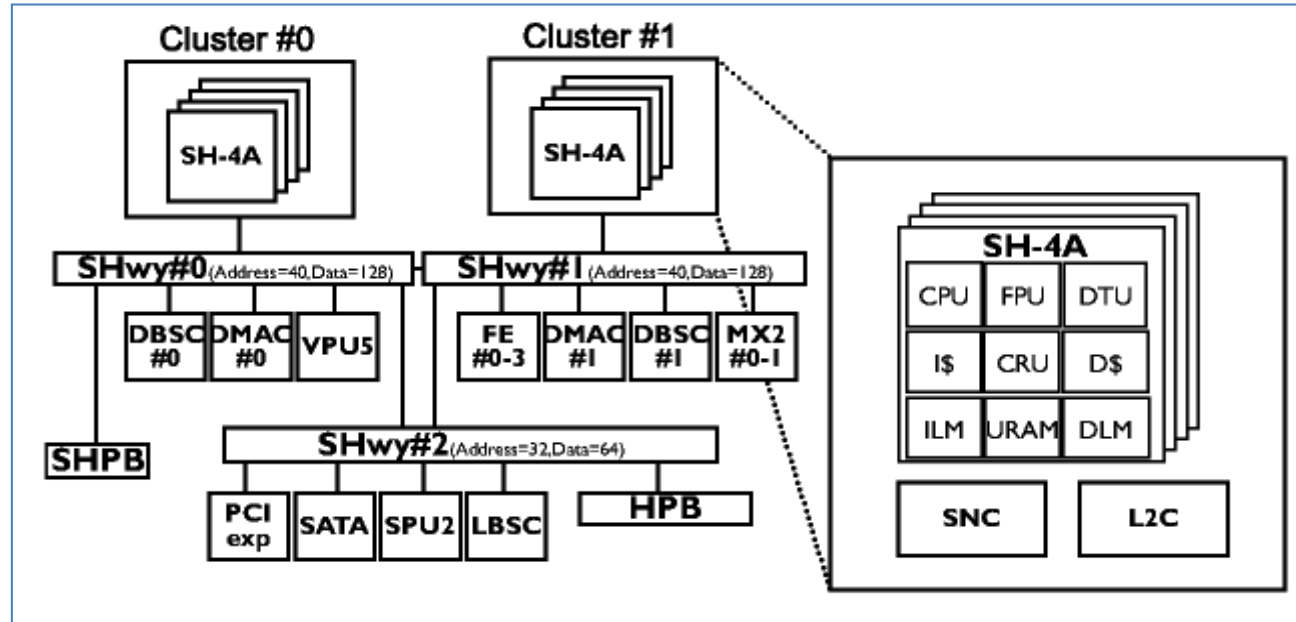
Hard real-time automobile engine control by multicore



Macro Task Fusion for Static Task Scheduling

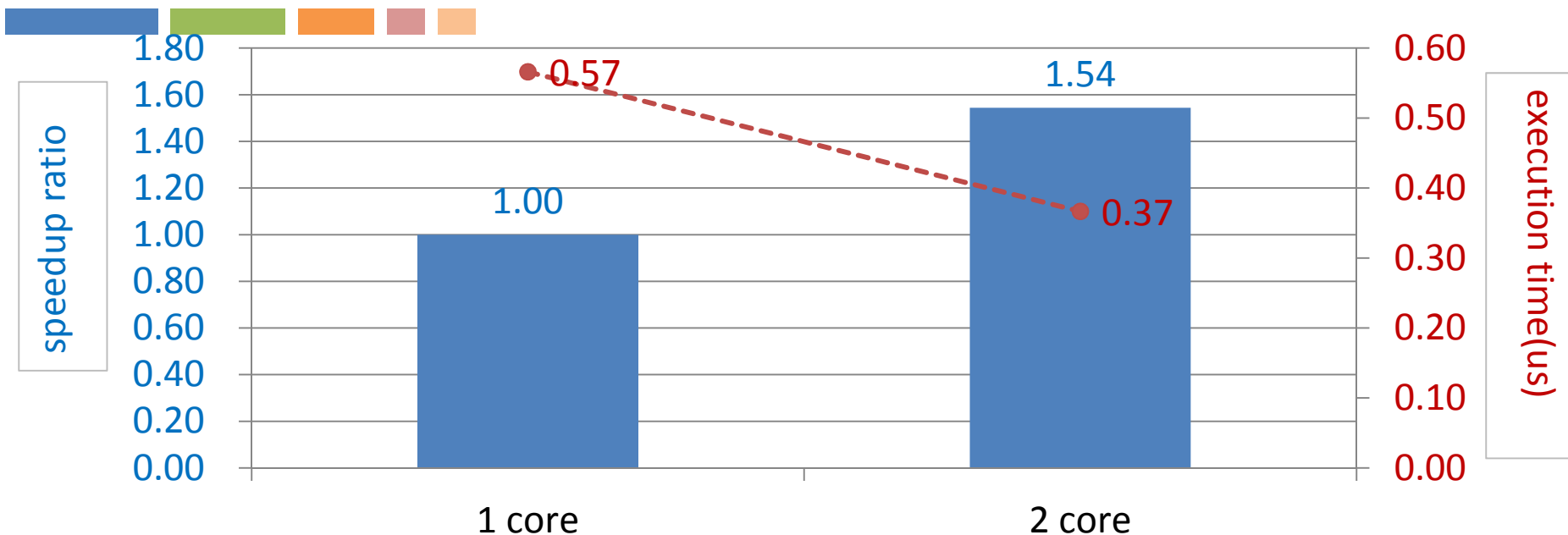


Evaluation Environment : Embedded Multi-core Processor RPX



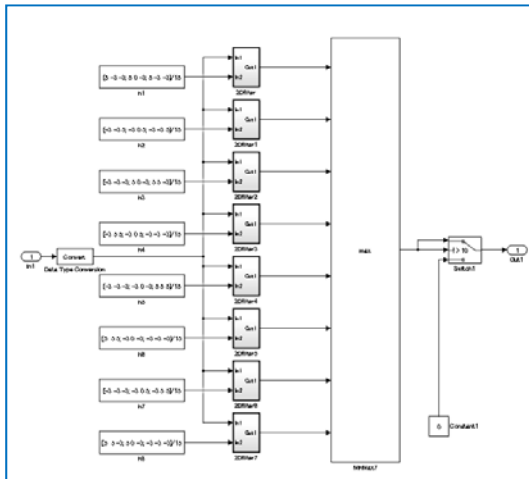
- SH-4A 648MHz * 8
 - As a first step, we use just two SH-4A cores because target dual-core processors are currently under design for next-generation automobiles

Evaluation of Crankshaft Program with Multi-core Processors



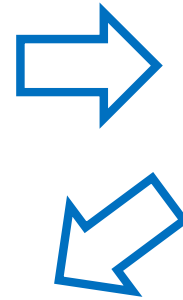
- Attain 1.54 times speedup on RPX
 - There are no loops, but only many conditional branches and small basic blocks and difficult to parallelize this program
- This result shows possibility of multi-core processor for engine control programs

OSCAR Compile Flow for Simulink Applications



Simulink model

Generate C code
using Embedded Coder



```

/* Model step function */
void VesselExtraction_step(void)
{
    int32_T i;
    real_T u0;

    /* DataTypeConversion: '<S1>/Data Type Conversion' incorporates:
     * Inport: '<Root>/In1'
     */
    for (i = 0; i < 16384; i++) {
        VesselExtraction_B.DataTypeConversion[i] = VesselExtraction_U.In1[i];
    }

    /* End of DataTypeConversion: '<S1>/Data Type Conversion' */

    /* Outputs for Atomic SubSystem: '<S1>/2Dfilter' */

    /* Constant: '<S1>/h1' */
    VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
        VesselExtraction_P.h1_Value, &VesselExtraction_B.Dfilter,
        (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter);

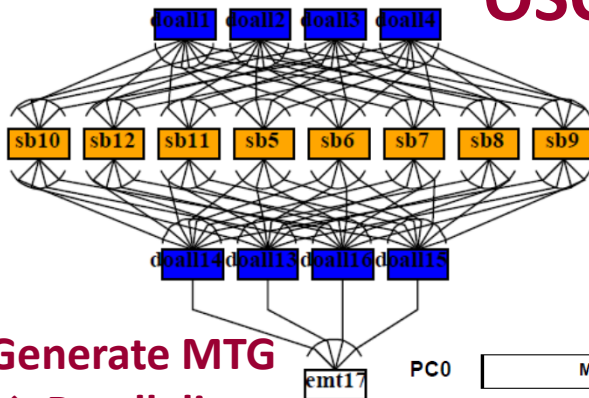
    /* End of Outputs for SubSystem: '<S1>/2Dfilter1' */

    /* Outputs for Atomic SubSystem: '<S1>/2Dfilter1' */

    /* Constant: '<S1>/h2' */
    VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
        VesselExtraction_P.h2_Value, &VesselExtraction_B.Dfilter1,
        (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter1);
}
    
```

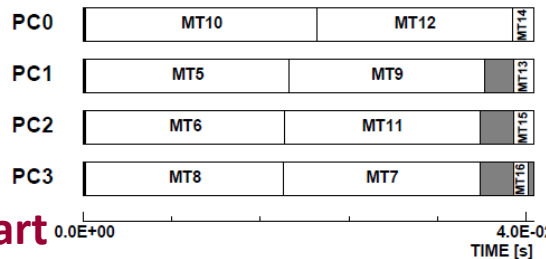
C code

OSCAR Compiler



(1) Generate MTG
→ Parallelism

(2) Generate gantt chart
→ Scheduling in a multicore



```

void VesselExtraction_step ( )
{
    int thr1 ;
    int thr2 ;
    int thr3 ;

    void thread_function_001 ( void )
    {
        VesselExtraction_step_PE1 ( ) ;
    }

    oscar_thread_create ( & thr1 ,
        thread_function_001 , (void*)1 ) ;
    oscar_thread_create ( & thr2 ,
        thread_function_002 , (void*)2 ) ;
    oscar_thread_create ( & thr3 ,
        thread_function_003 , (void*)3 ) ;

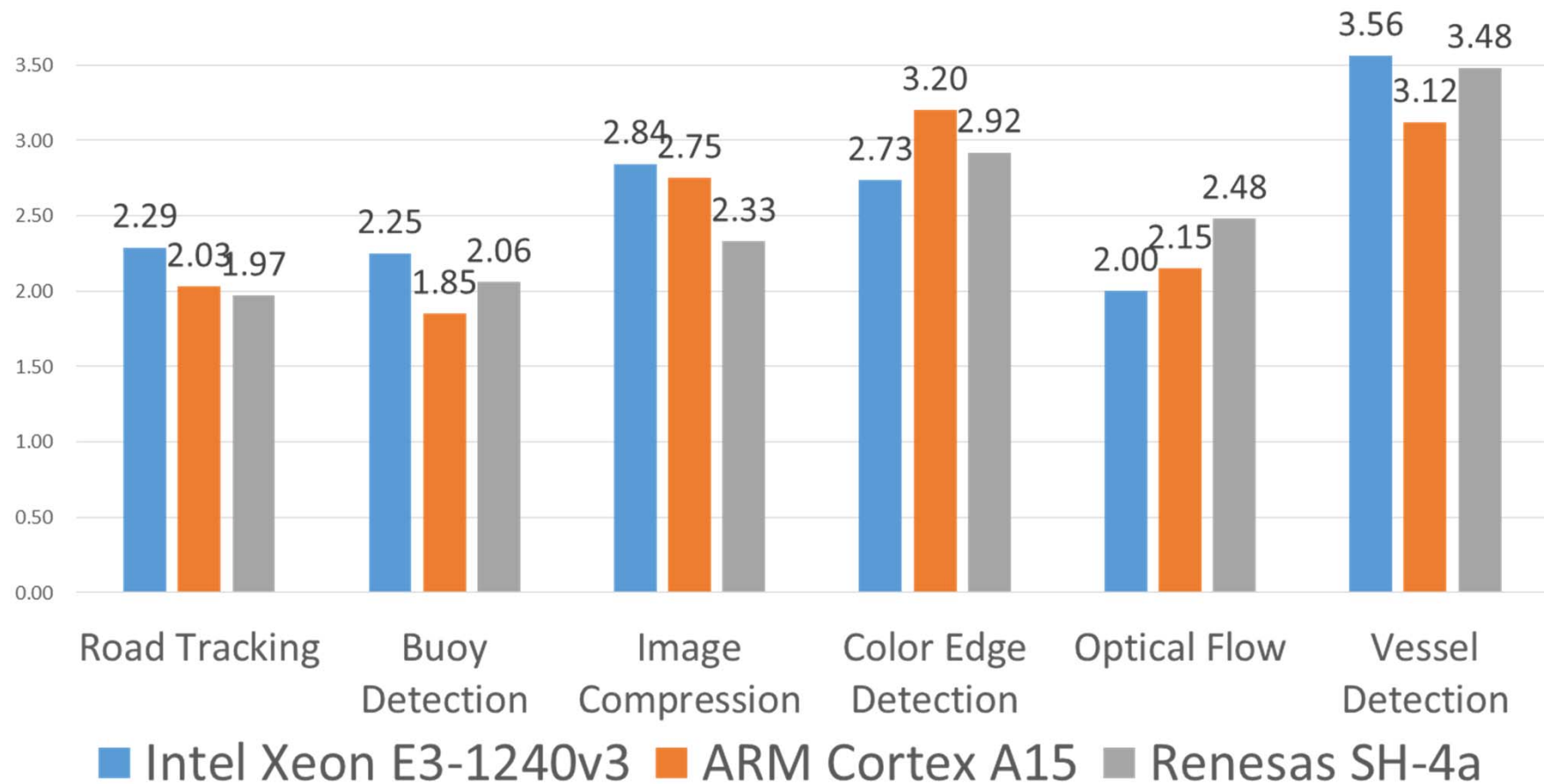
    VesselExtraction_step_PEO ( ) ;

    oscar_thread_join ( thr1 ) ;
    oscar_thread_join ( thr2 ) ;
    oscar_thread_join ( thr3 ) ;
}
    
```

(3) Generate parallelized C code
using the OSCAR API
→ Multiplatform execution
(Intel, ARM and SH etc)

Speedups of MATLAB/Simulink Image Processing on Various 4core Multicores

(Intel Xeon, ARM Cortex A15 and Renesas SH4A)



Road Tracking, Image Compression : <http://www.mathworks.co.jp/jp/help/vision/examples>

Buoy Detection : <http://www.mathworks.co.jp/matlabcentral/fileexchange/44706-buoy-detection-using-simulink>

Color Edge Detection : <http://www.mathworks.co.jp/matlabcentral/fileexchange/28114-fast-edges-of-a-color-image--actual-color--not-converting-to-grayscale-/>

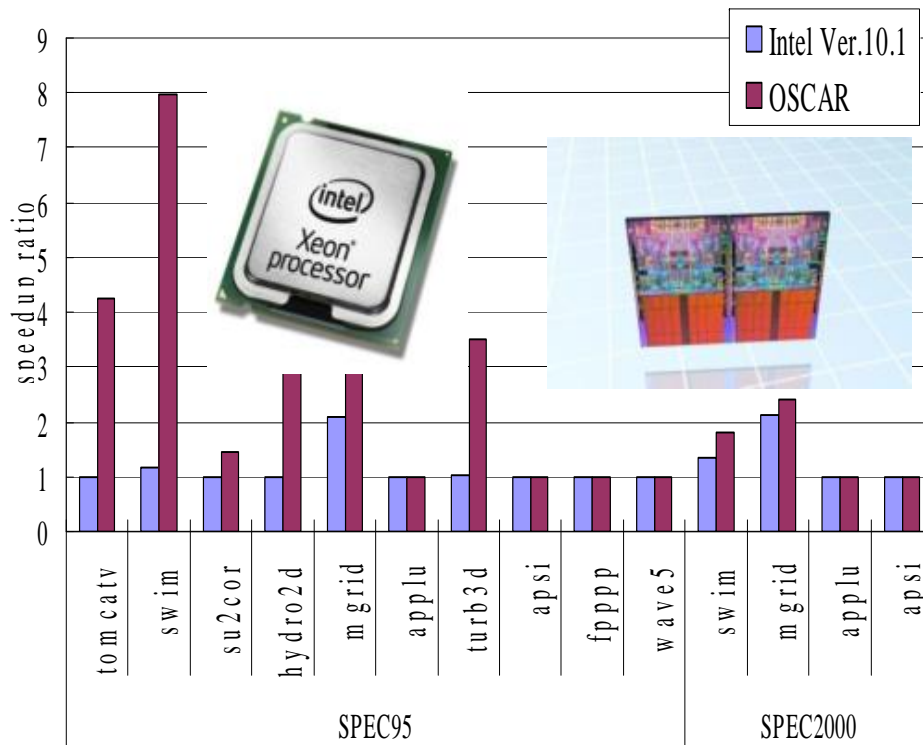
Vessel Detection : <http://www.mathworks.co.jp/matlabcentral/fileexchange/24990-retinal-blood-vessel-extraction/>

OSCAR Compiler Accelerates Various Multicores' Performance Several Times Including Intel and IBM

On Intel Quad-core Xeon

On Intel 4core Multicore, Compared with Intel Compiler

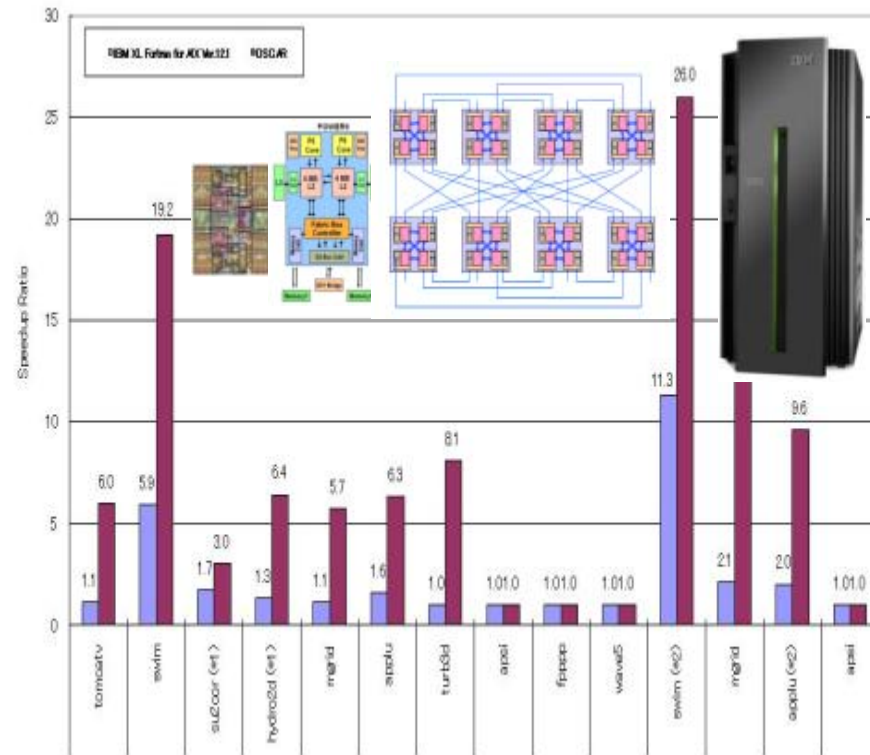
2.1 times Speedup



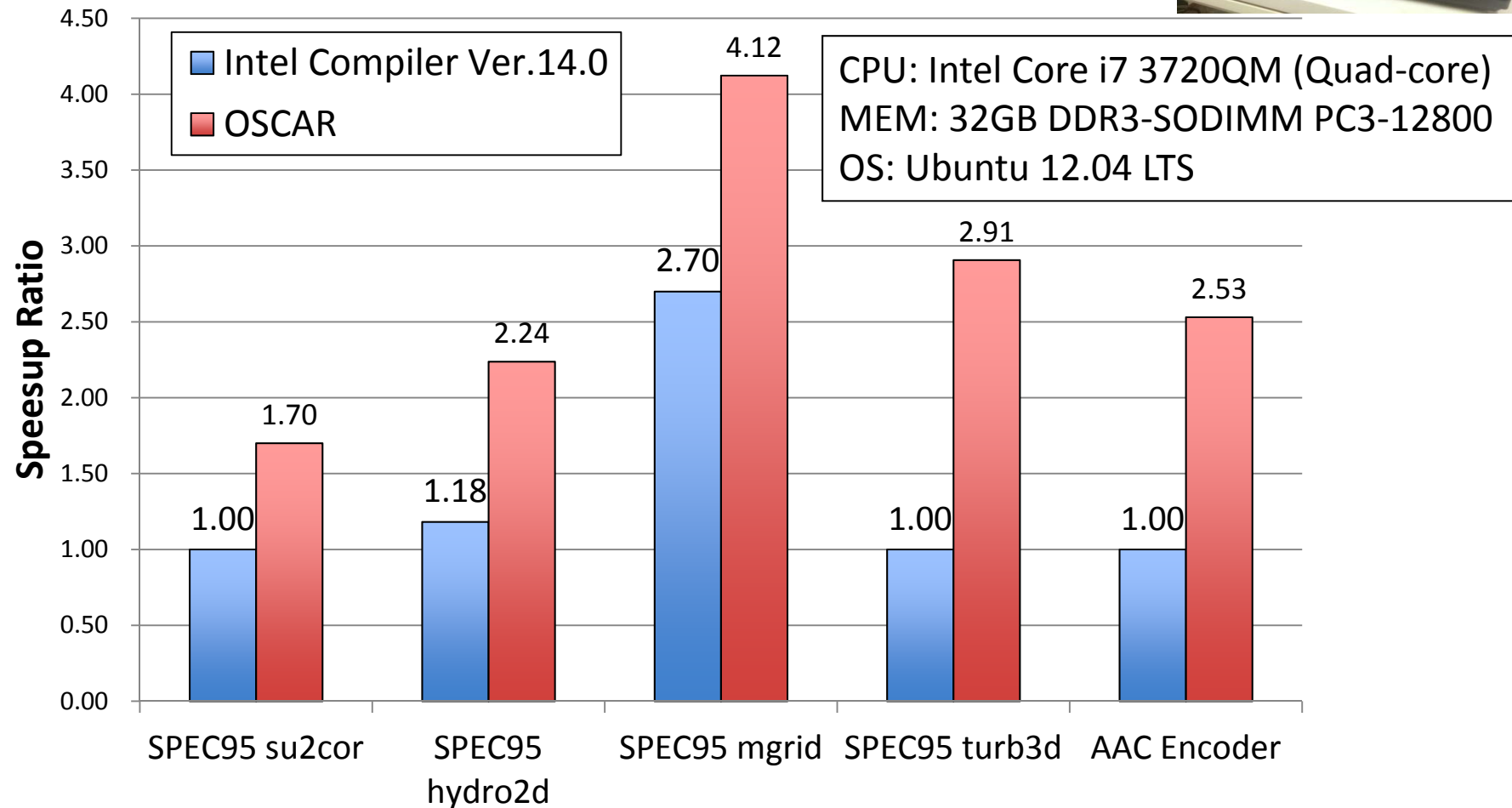
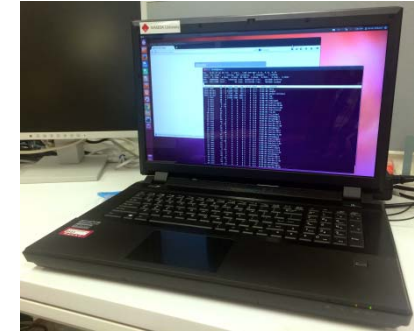
On IBM Power Servers, such as Power 4, 5, **6(4.2GHz)**, 7 and later

On IBM P6 595 Server, Compared with IBM Compiler

3.3 times Speedup



Performance of OSCAR Compiler on Intel Core i7 Notebook PC



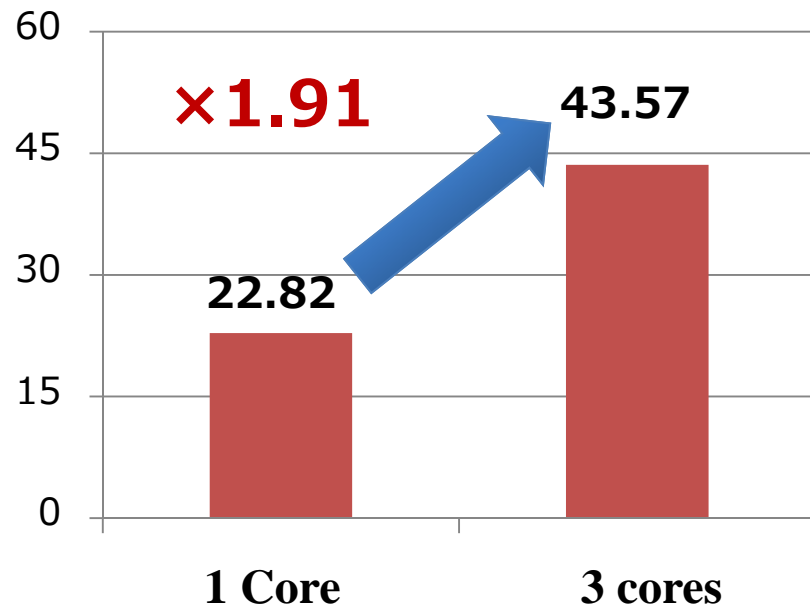
- OSCAR Compiler accelerate Intel Compiler about **2.0 times** on average

Parallelization of 2D Rendering Engine SKIA on 3 cores of Google NEXUS7

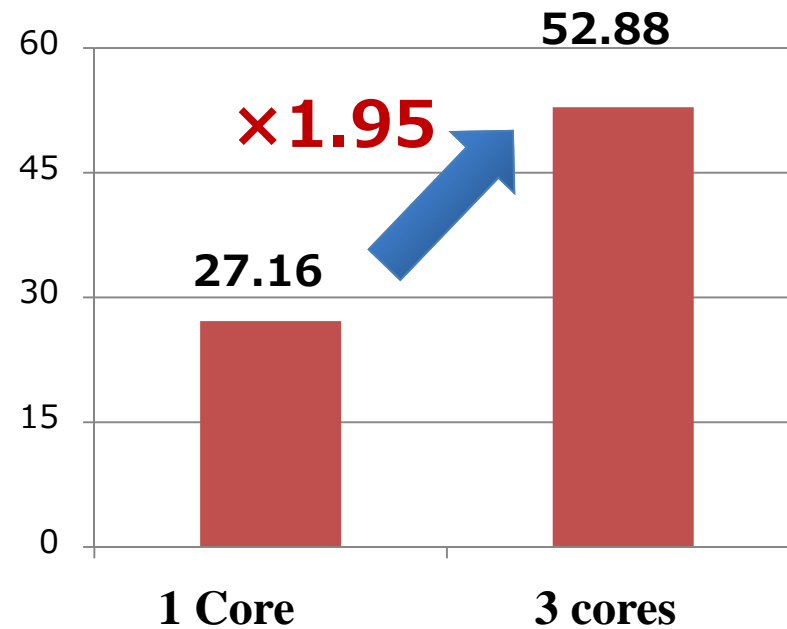
http://www.youtube.com/channel/UCS43INYEIkC8i_KIgfZYQBQ



DrawRect :FPS

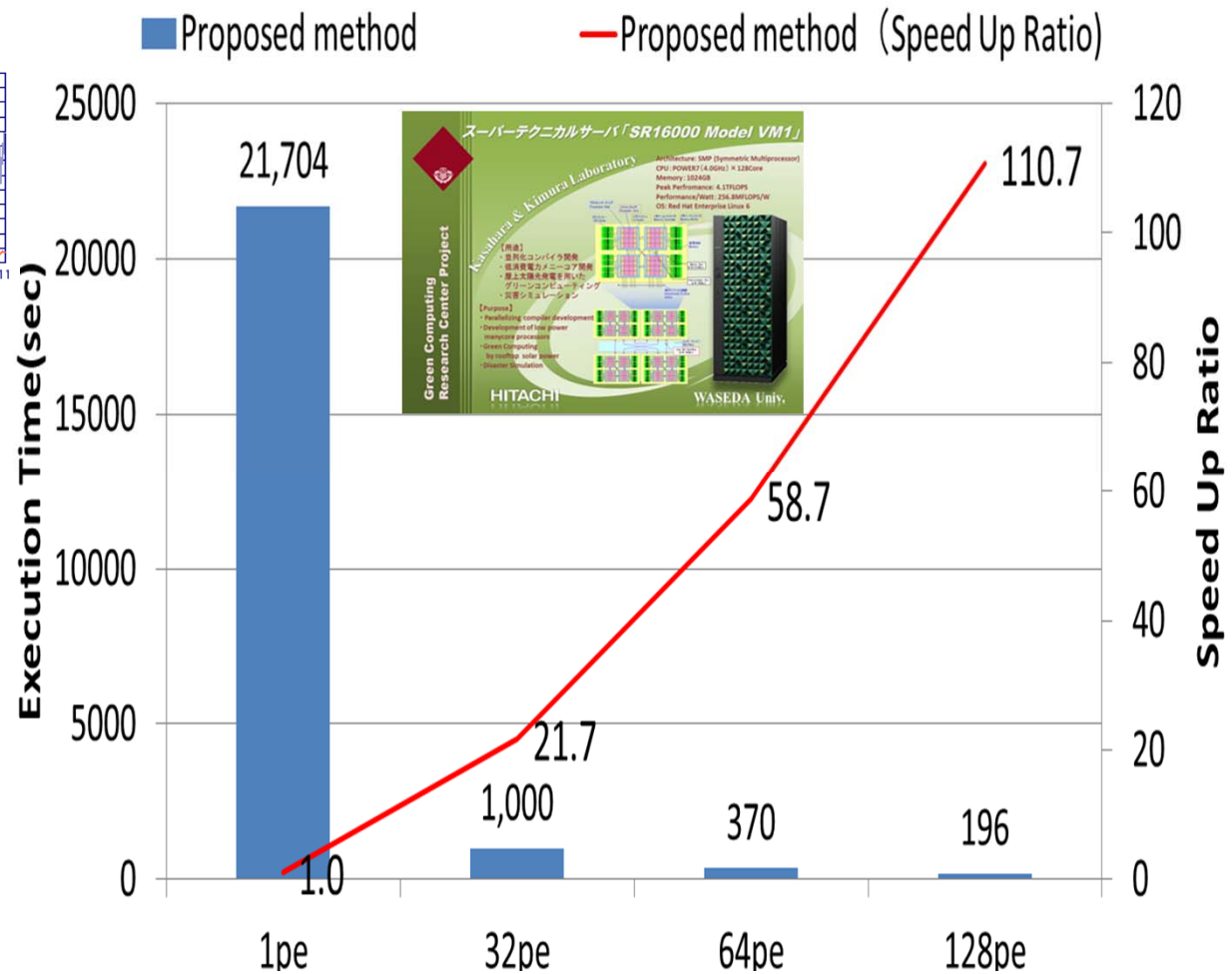
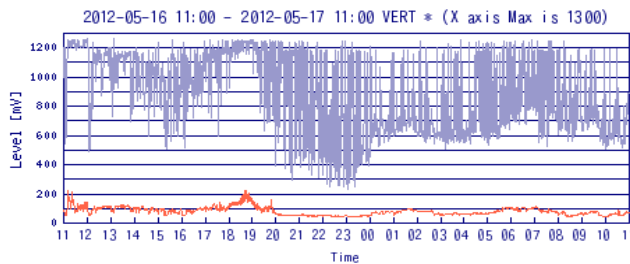


DrawImage : FPS



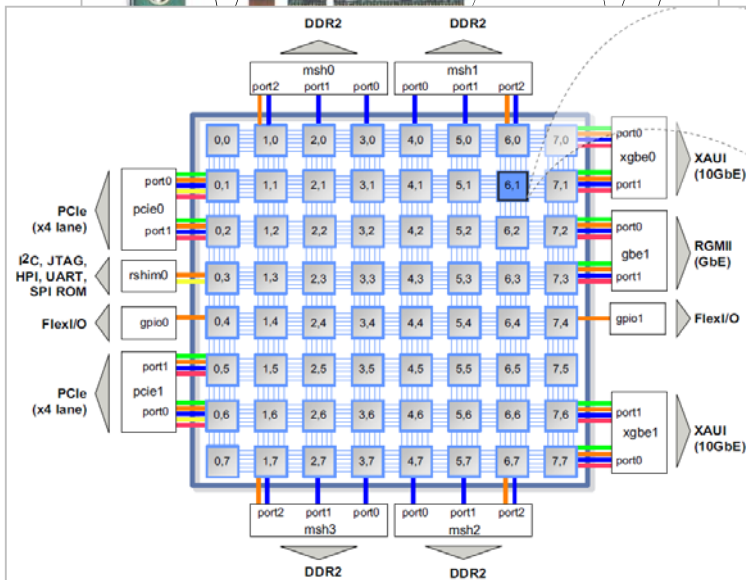
On Nexus7, 3 core parallelization gave us
for DrawRect **1.91** speedup
for DrawImage **1.95** speedup

110 Times Speedup against the Sequential Processing for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000 (Power7 Based 128 Core Linux SMP)

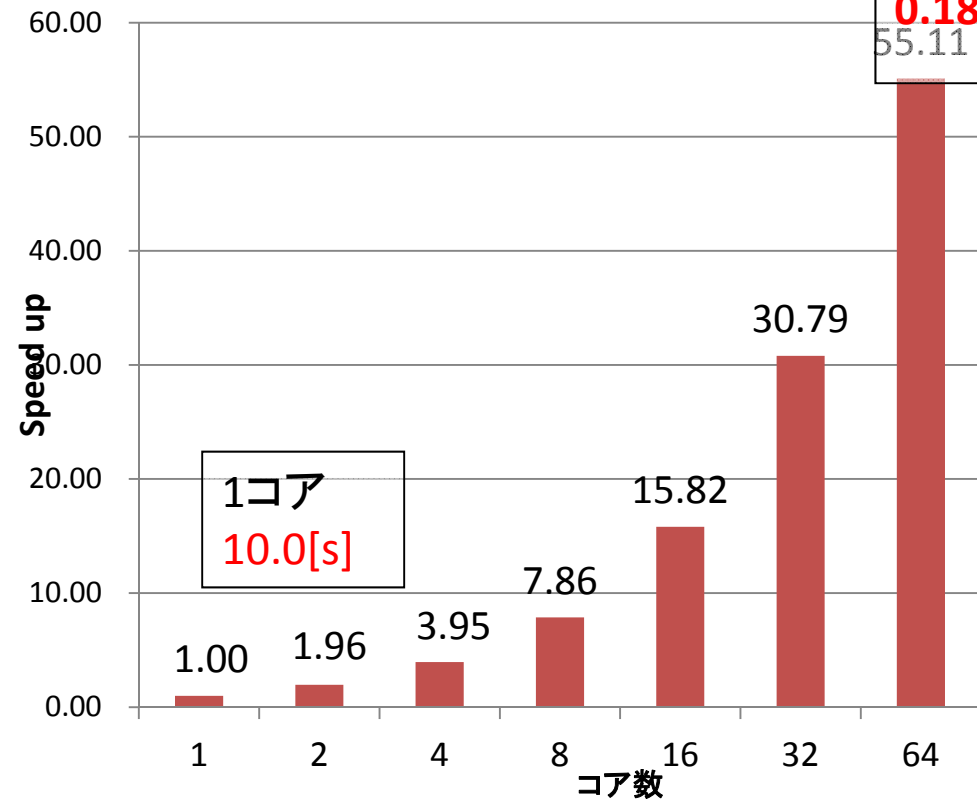


Automatic Parallelization of Still Image Encoding Using JPEG-XR for the Next Generation Cameras and Drinkable Inner Camera

TILEPro64

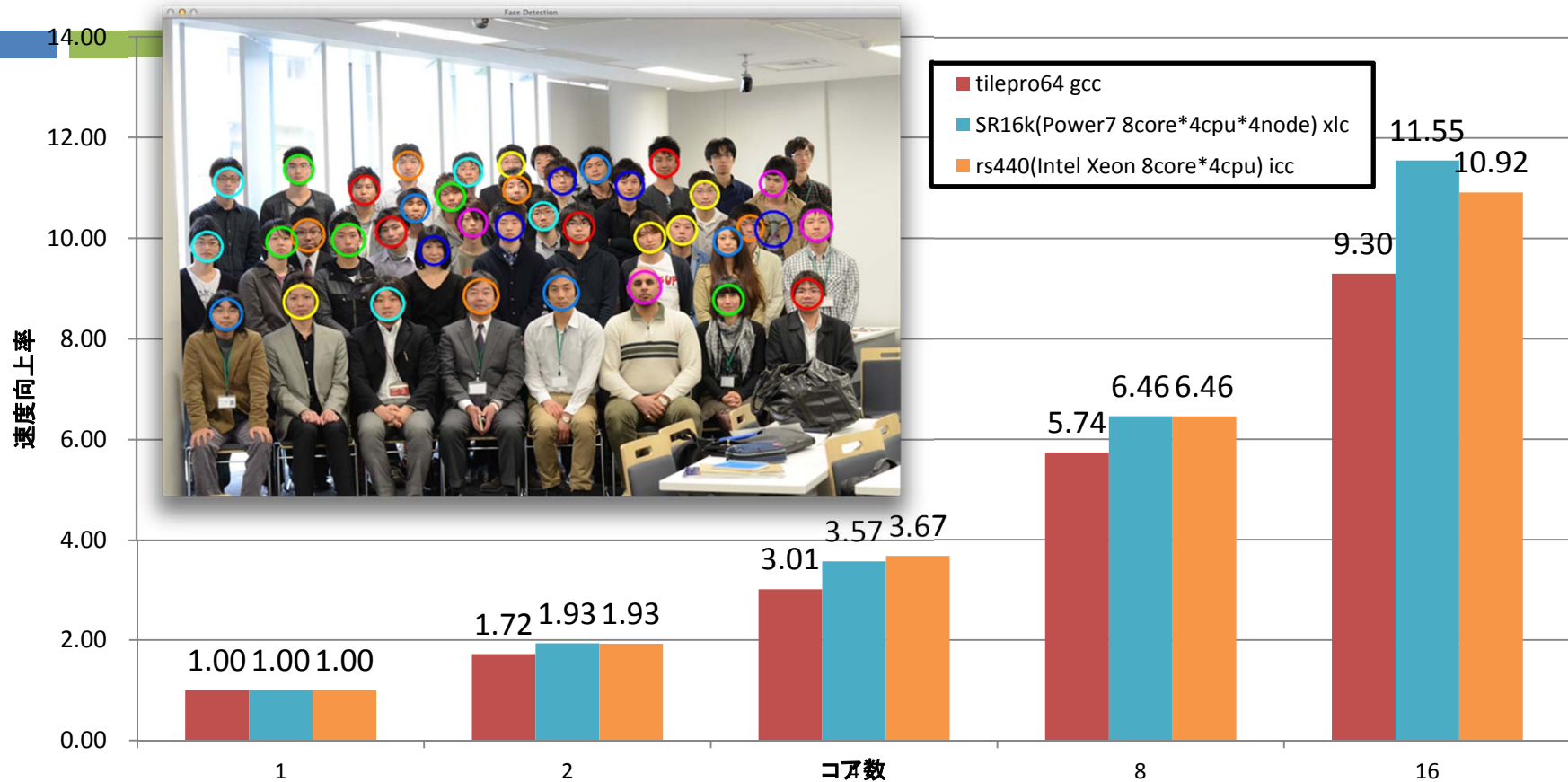


Speed-ups on TILEPro64 Manycore



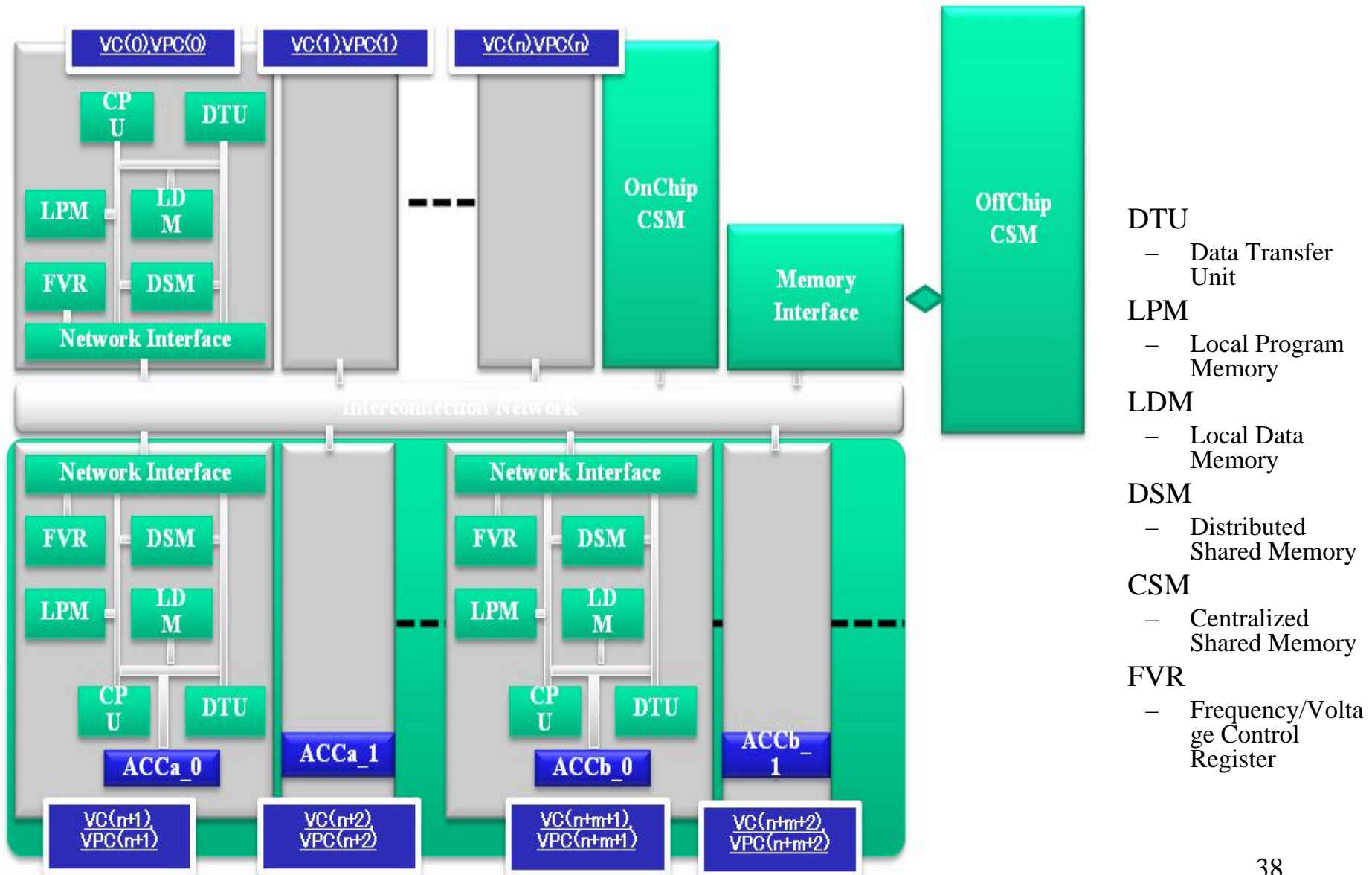
55 times speedup with 64 cores against 1 core

Parallel Processing of Face Detection on Manycore, Highend and PC Server



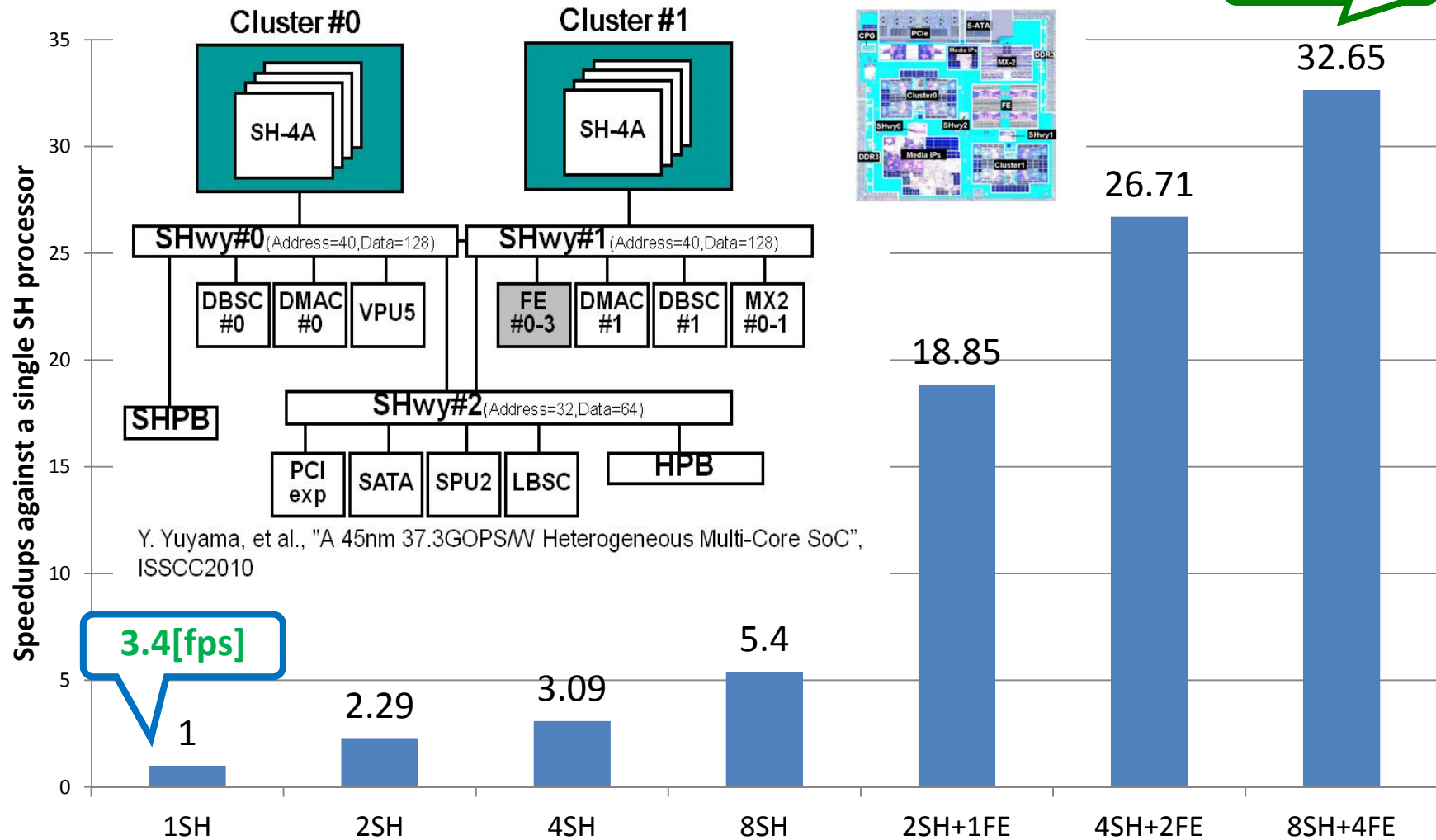
□ OSCAR compiler gives us **11.55 times** speedup for 16 cores against 1 core on SR16000 Power7 highend server.

OSCAR Heterogeneous Multicore



33 Times Speedup Using OSCAR Compiler and OSCAR API on RP-X (Optical Flow with a hand-tuned library)

111[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)

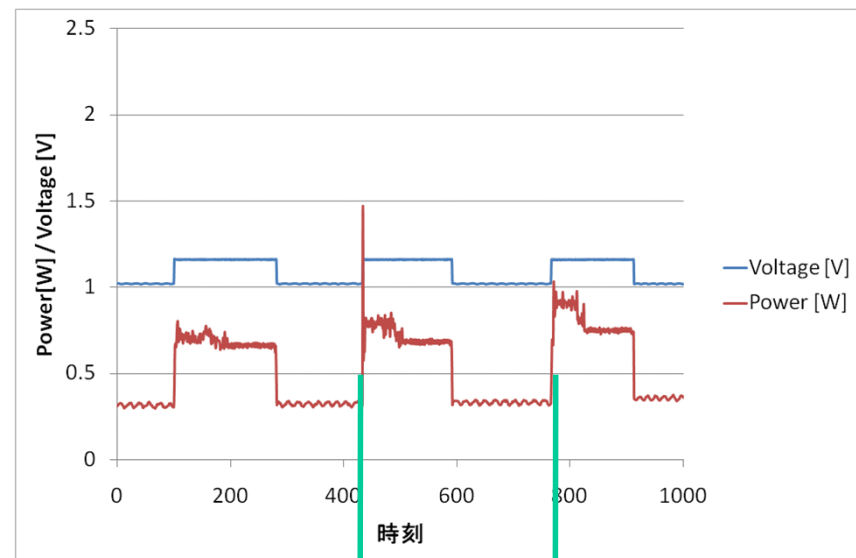
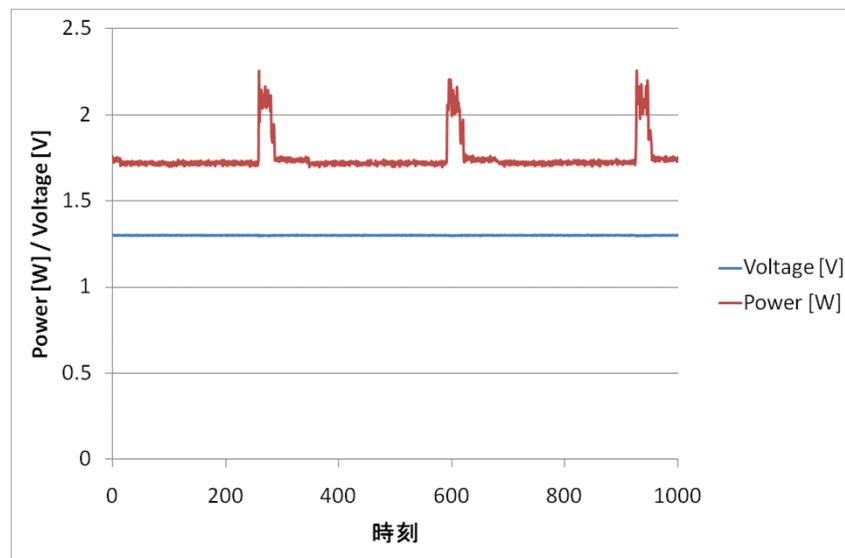
Without Power Reduction

With Power Reduction by OSCAR Compiler
70% of power reduction

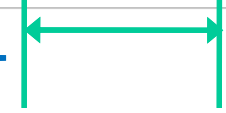
Average: 1.76[W]



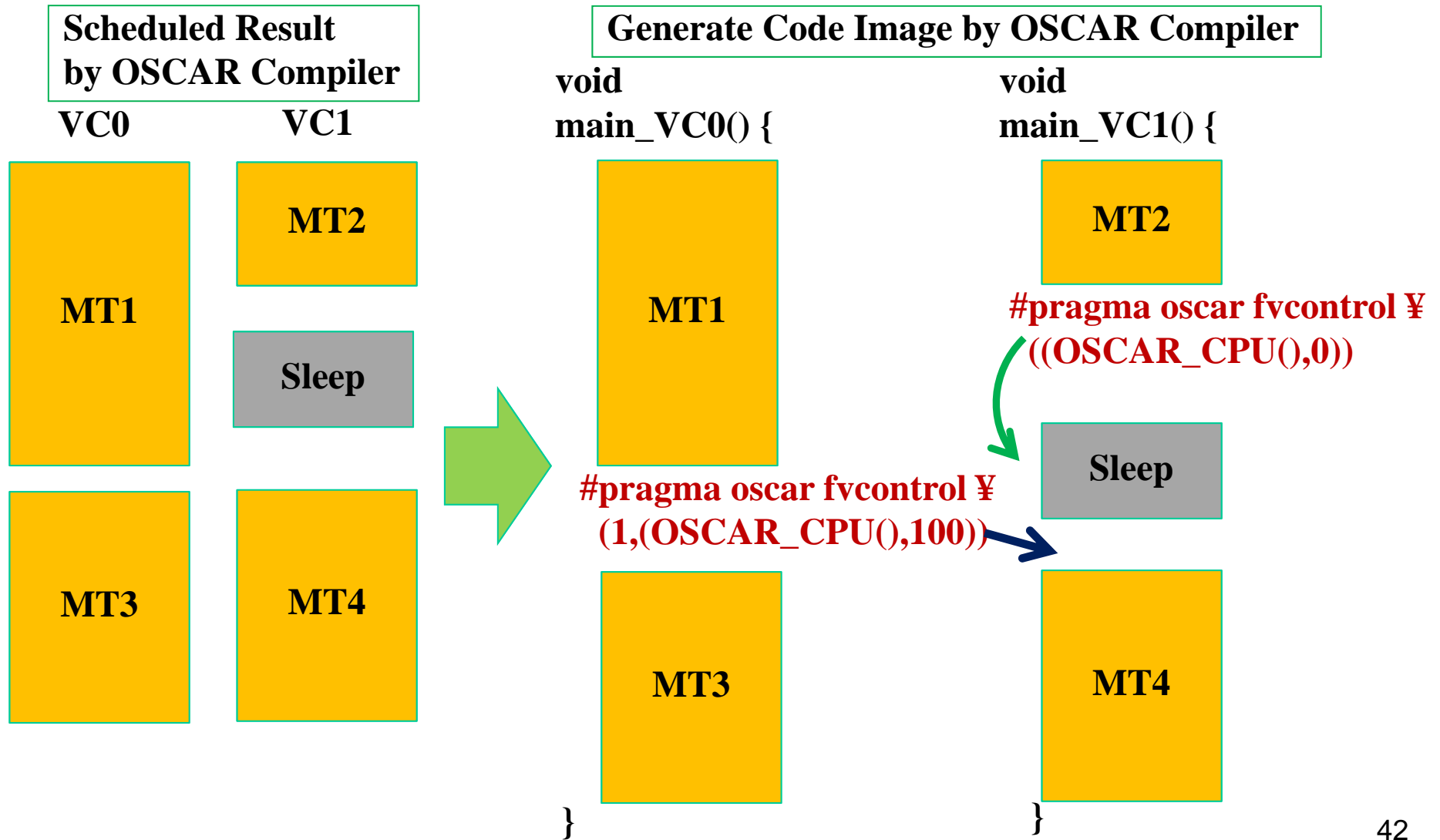
Average: 0.54[W]



**1cycle : 33[ms]
→30[fps]**



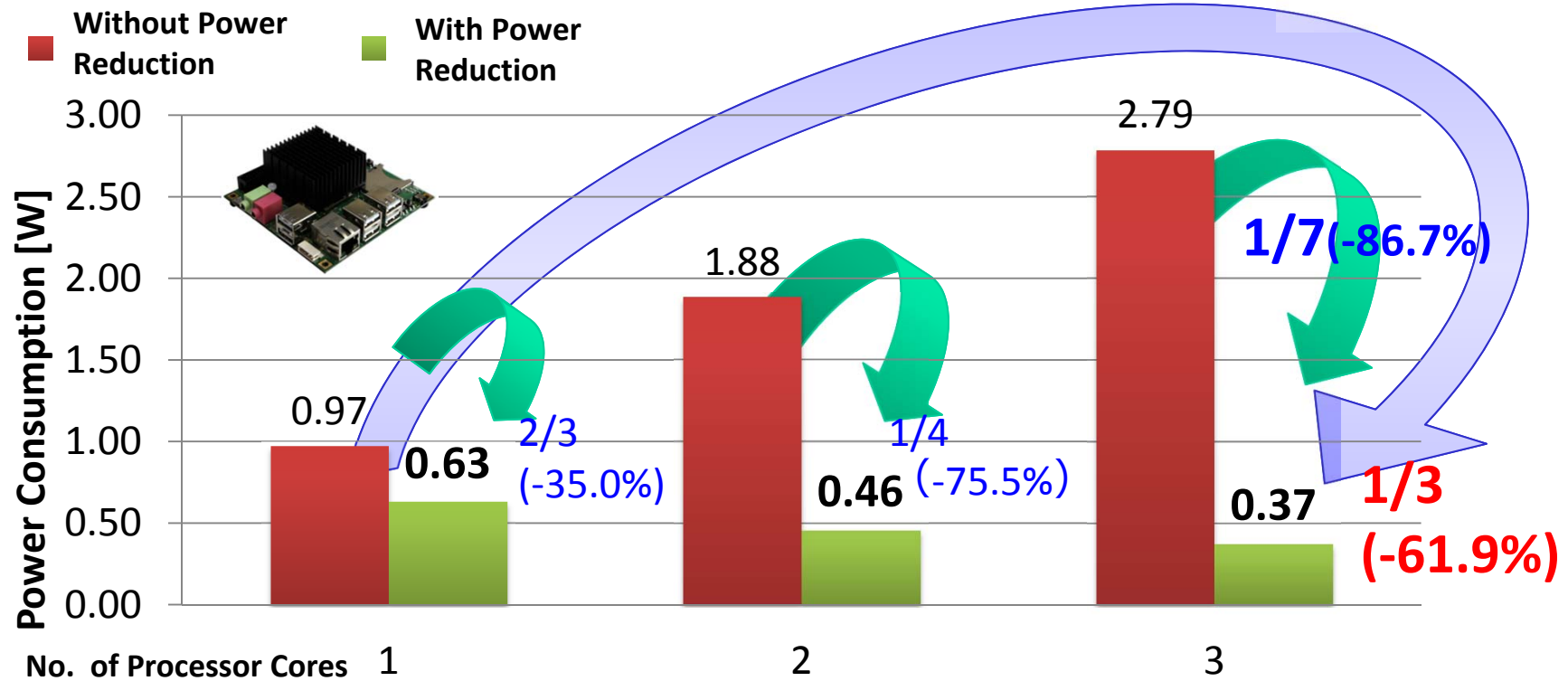
Low-Power Optimization with OSCAR API



Automatic Power Reduction for MPEG2 Decode on Android Multicore

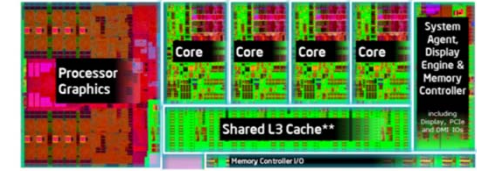
ODROID X2 ARM Cortex-A9 4 cores

http://www.youtube.com/channel/UCS43INYEIkC8i_KIgFZYQBQ



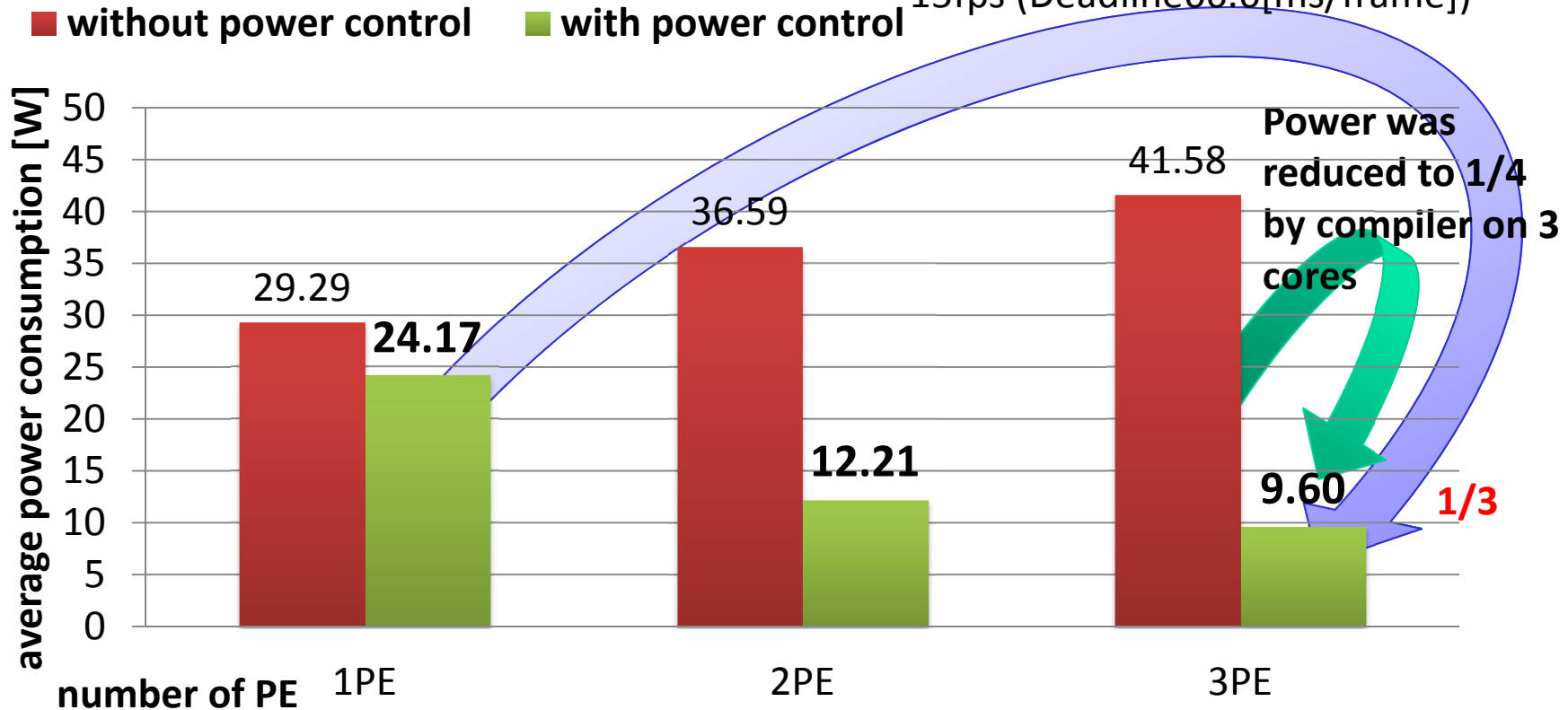
- On 3 cores, Automatic Power Reduction control successfully reduced power to 1/7 against without Power Reduction control.
- 3 cores with the compiler power reduction control reduced power to 1/3 against ordinary 1 core execution.

Power Reduction on Intel Haswell for Real-time Optical Flow



Intel CPU Core i7 4770K

For HD 720p(1280x720) moving pictures
15fps (Deadline 66.6[ms/frame])



Power was reduced to **1/4 (9.6W)** by the compiler power optimization **on the same 3 cores (41.6W)**.

Power with 3 core was reduced to **1/3 (9.6W)** against **1 core (29.3W)**.

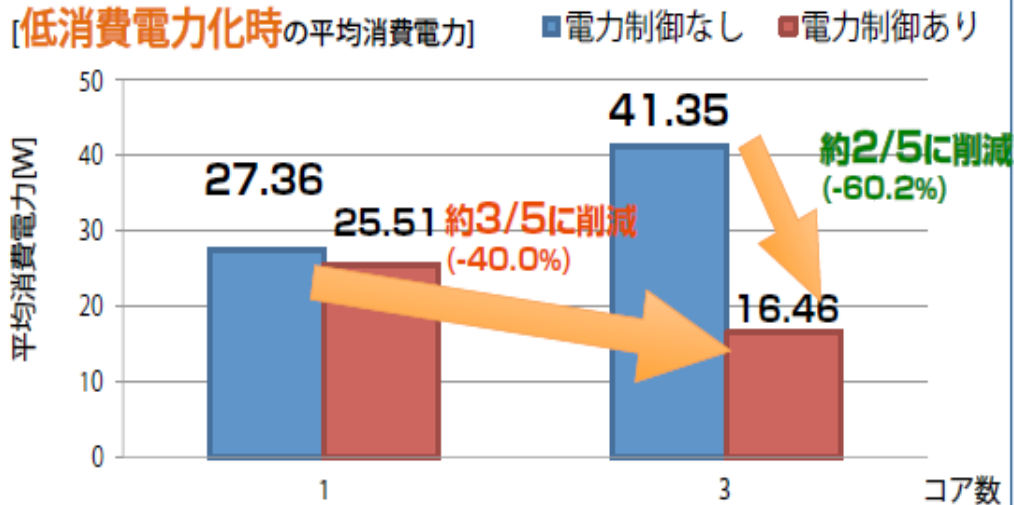


WASEDA UNIVERSITY

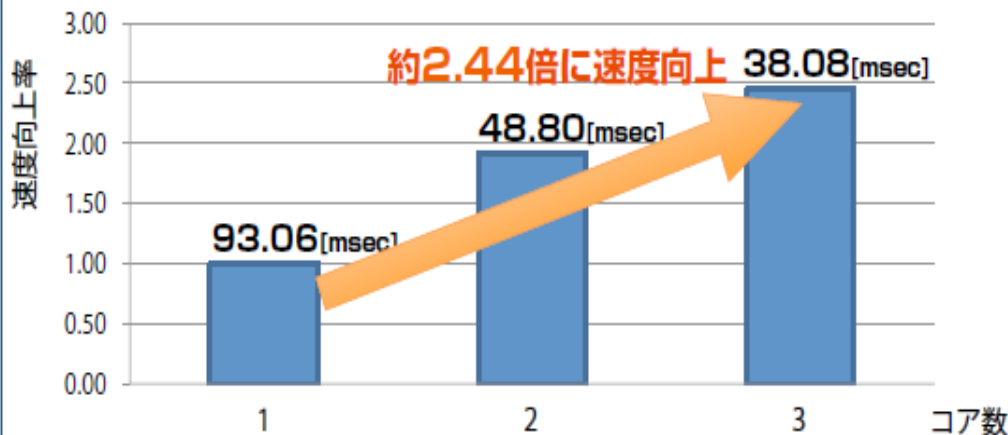
OSCARコンパイラによるHaswellマルチコア上での 自動低消費電力化(Intel 4コア) - 消費電力を2/5に削減 -

- OSCAR Compiler
- Intel Haswell
- 低消費電力化

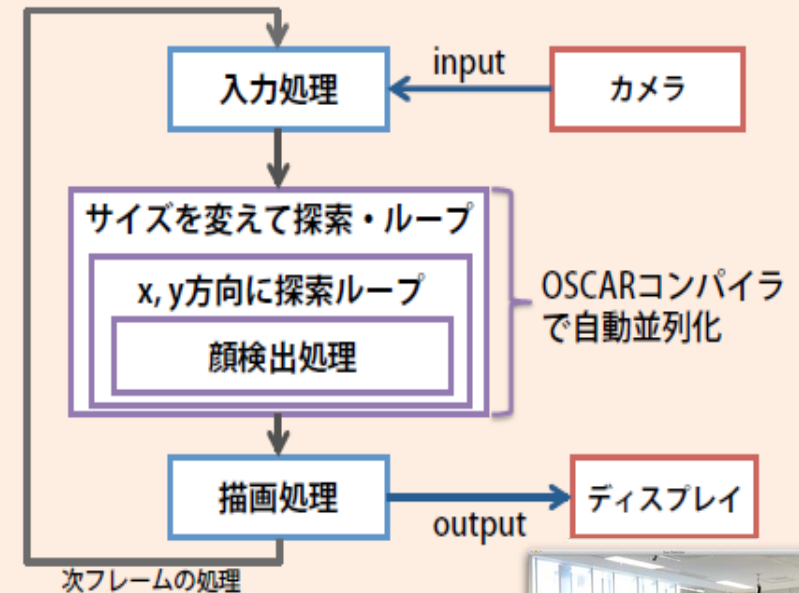
Intel Haswell 4コア上での顔認識プログラム 並列化



[最速実行処理時の速度向上率]

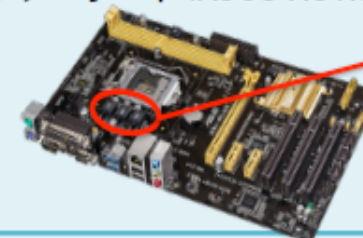


顔認識プログラムの並列処理



Intel Haswell 4コアの電力測定

CPU : Intel Core i7 4770k
コア数 : 4
周波数 : 3.5GHz~0.8GHz
マザーボード : ASUS H81M-A

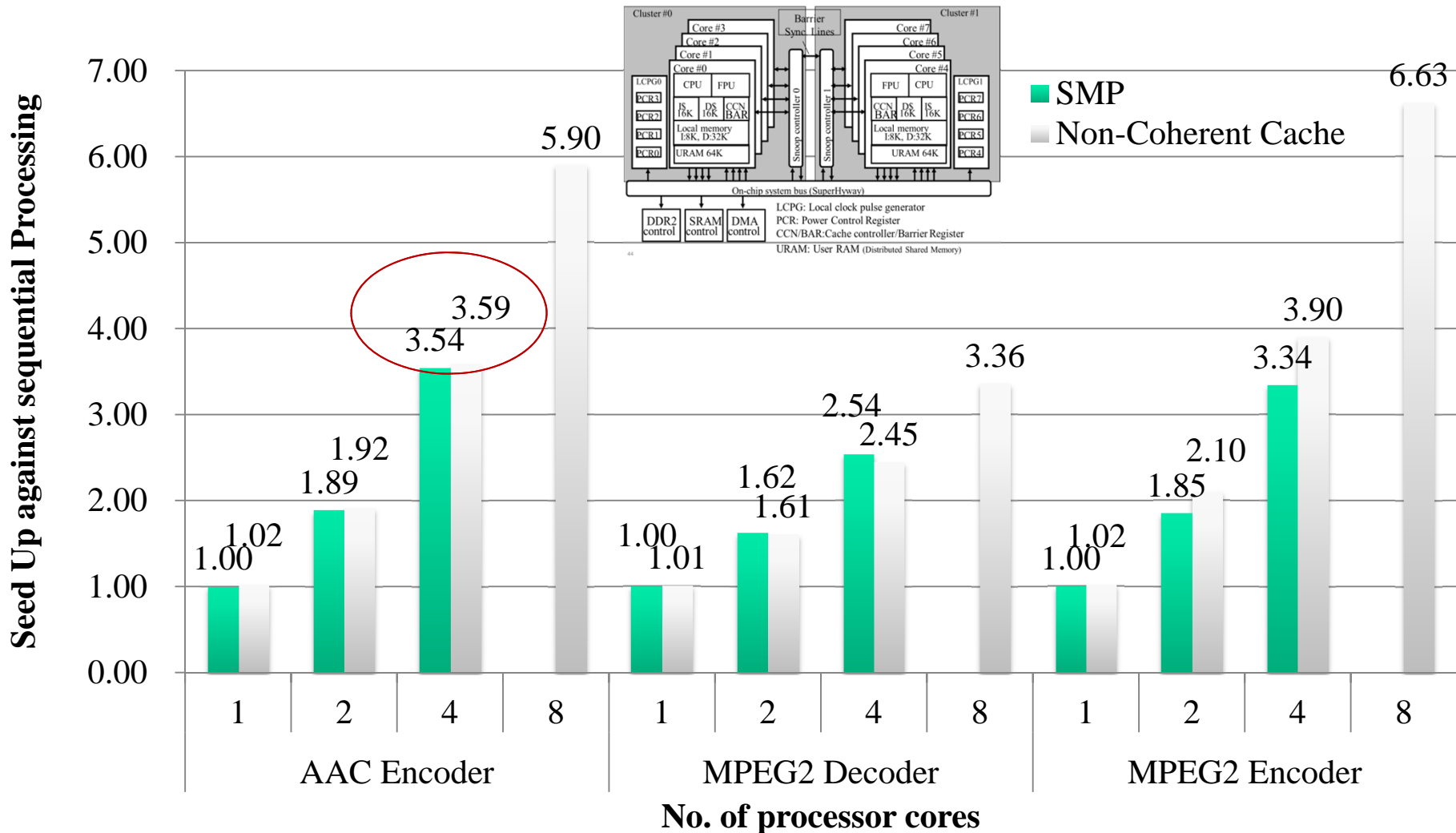


PMICとCPU間に
電力測定回路を作成



Performance of OSCAR Compiler Software Coherence Control

- **Faster or Equal Processing Performance up to 4cores with hardware coherent mechanism on RP2.**
- **Software Coherence gives us correct execution without hardware coherence mechanism on 8 cores.**



OSCAR Technology

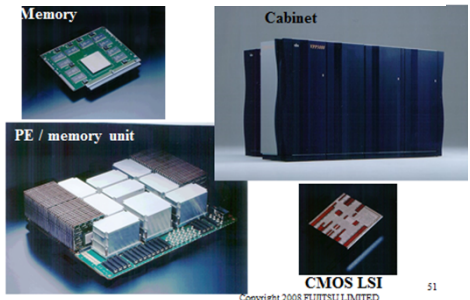
Started up on Feb.28, 2013:

Licensing the all patents and OSCAR compiler from Waseda Univ.



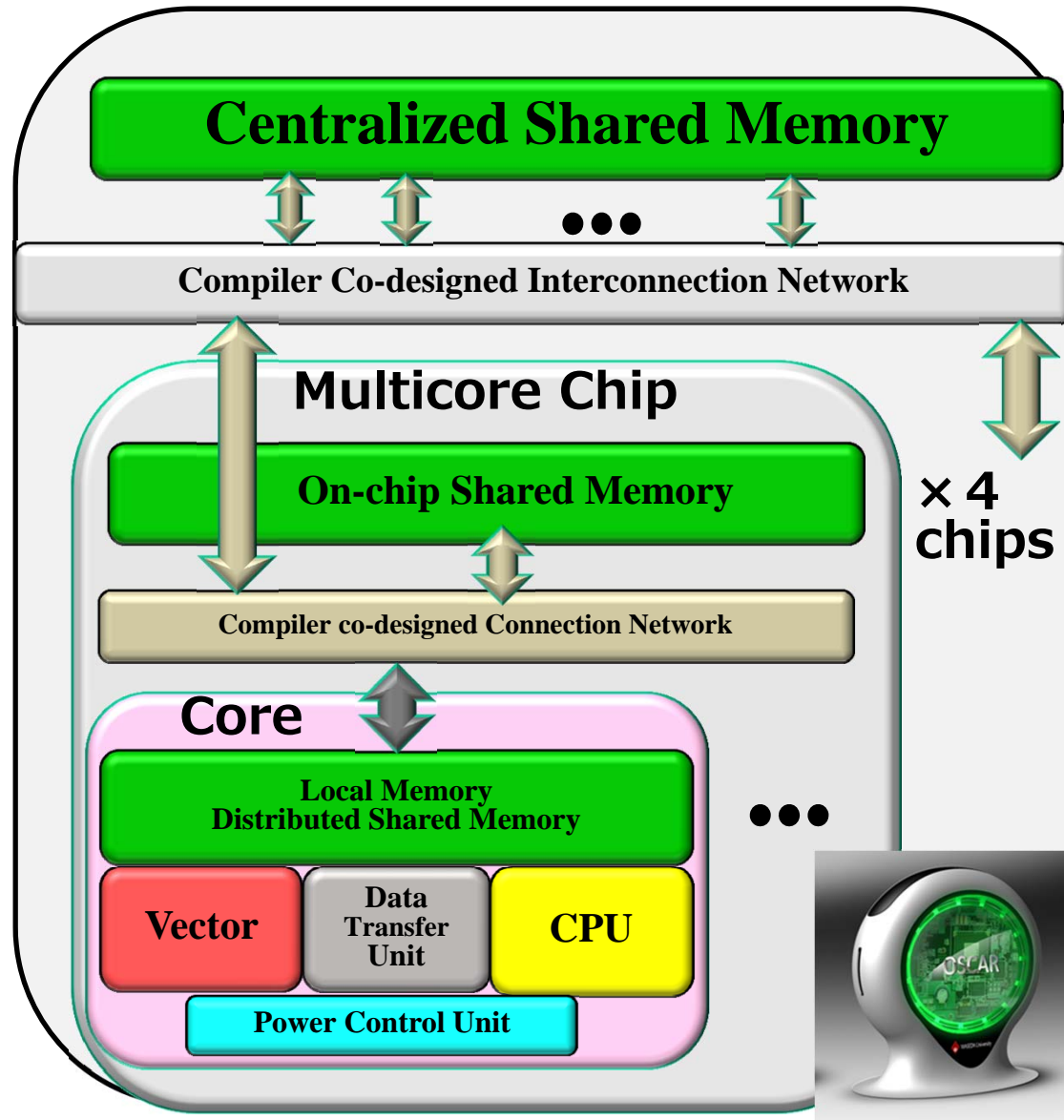
- CEO:** Dr. T. Ono (Ex- CEO of First Section-listed Company, VP of National Univ., Invited Prof. of Waseda U.)
- Executives:** Mr. T. Ito (Visiting Prof. Tokyo Agricult. and Eng. U.)
Prof. K. Shirai (Ex-President of Waseda U Chairman of Japanese Open Univ.)
- CTO:** Mr. M. Takamura (Ex-Fellow Fujitsu Lab., Fujitsu VPP500, 5000 & NWT Development Leader)
Mr. K. Ashida(Ex-VP Sumitomo Trading, Ashida Consult. CEO, A leader of Business World
- Auditor:** Dr. S. Matsuda (Prof. Emeritus Waseda U. Ex-President Ventures and Entrepreneurs Society)
- Advisors:** Dr. T. Sato (Patent Attorney, Ex-President of Patent Attorneys Assoc., Gov. IP Committee)
Ms. K. Ishiguro (Lawyer, Supreme Court Trainer)
Mr. A. Fukuda (Leader of Alumni Assoc.)
Prof. K. Kimura (Waseda Univ.)
Prof. H. Kasahara (Waseda Univ.)

Fujitsu VPP5000



OSCAR TECHNOLOGY CORPORATION

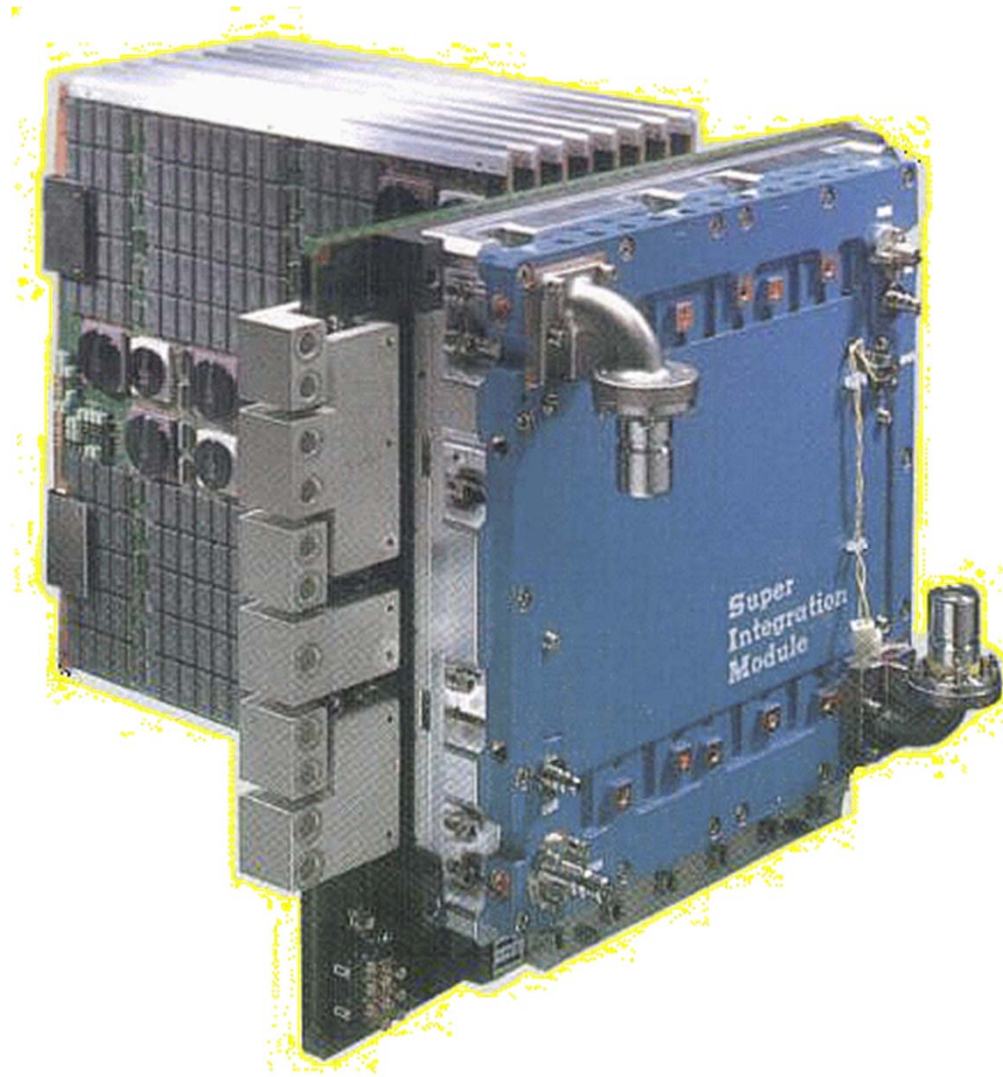
OSCAR Vector Multicore and Compiler for Embedded to Servers with OSCAR Technology



Target:

- **Solar Powered with compiler power reduction.**
- **Fully automatic parallelization and vectorization including local memory management and data transfer.**

Fujitsu VPP500/NWT: PE Unit



Summary

- Waseda University Green Computing Systems R&D Center supported by METI has been researching **on low-power high performance Green Multicore** hardware, software and application with **government and industry including Hitachi, Fujitsu, NEC, Renesas, Denso, Toyota, Olympus and OSCAR Technology.**
- OSCAR Automatic Parallelizing and Power Reducing Compiler has succeeded speedup and/or power reduction of scientific applications including **“Earthquake Wave Propagation”**, medical applications including **“Cancer Treatment Using Carbon Ion”**, and **“Drinkable Inner Camera”**, industry application including **“Automobile Engine Control”**, **“Smartphone”**, and **“Wireless communication Base Band Processing”** on **various multicores from** different vendors including **Intel, ARM, IBM, AMD, Qualcomm, Freescale, Renesas and Fujitsu.**
- In automatic parallelization, **110 times speedup** for **“Earthquake Wave Propagation Simulation”** on **128 cores of IBM Power 7** against **1 core**, **55 times speedup** for **“Carbon Ion Radiotherapy Cancer Treatment”** on **64cores IBM Power7**, **1.95 times** for **“Automobile Engine Control”** on **Renesas 2 cores using SH4A or V850**, **55 times** for **“JPEG-XR Encoding for Capsule Inner Cameras”** on **Tilera 64 cores Tile64 manycore.**
 - The compiler will be available on market from OSCAR Technology.
- In automatic power reduction, **consumed powers** for real-time multi-media applications like **Human face detection, H.264, mpeg2 and optical flow** were **reduced to 1/2 or 1/3 using 3 cores** of **ARM Cortex A9 and Intel Haswell** and **1/4 using Renesas SH4A 8 cores against ordinary single core execution.**