

# Green Multicore Compiler



**Hironori Kasahara, Ph.D., IEEE Fellow, IPSJ Fellow**

**Senior Executive VP, Waseda University**

**IEEE Computer Society President 2018**

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

**1980 BS, 82 MS, 85 Ph.D. , Dept. EE, Waseda Univ.**  
**1985 Visiting Scholar: U. of California, Berkeley,**  
**1986 Assistant Prof., 1988 Associate Prof., 1989-90**  
**Research Scholar:U. of Illinois, Urbana-Champaign,**  
**Center for Supercomputing R&D, 1997 Prof., 2004**  
**Director, Advanced Multicore Research Institute,**  
**2017 member: the Engineering Academy of Japan**  
**and the Science Council of Japan**  
**2018 Nov. Senior Vice President, Waseda Univ.**

**1987 IFAC World Congress Young Author Prize**  
**1997 IPSJ Sakai Special Research Award**  
**2005 STARC Academia-Industry Research Award**  
**2008 LSI of the Year Second Prize**  
**2008 Intel AsiaAcademic Forum Best Research Award**  
**2010 IEEE CS Golden Core Member Award**  
**2014 Minister of Edu., Sci. & Tech. Research Prize**  
**2015 IPSJ Fellow, 2017 IEEE Fellow, Eta Kappa Nu**

**Reviewed Papers: 216, Invited Talks: 176, Granted**  
**Patents: 48 (Japan, US, GB, China), Articles in News**  
**Papers, Web News, Medias incl. TV etc.: 605**

**Committees in Societies and Government 258**  
**IEEE Computer Society : President 2018, Executive**  
**Committee(2017-19), BoG(2009-14), Strategic**  
**Planning Committee Chair 2017, Multicore STC Chair**  
**(2012-), Japan Chair(2005-07), IEEE TAB (2018)**  
**IPSJ Chair: HG for Magazine. & J. Edit, Sig. on ARC.**  
**【METI/NEDO】 Project Leaders: Multicore for**  
**Consumer Electronics, Advanced Parallelizing**  
**Compiler, Chair: Computer Strategy Committee**  
**【Cabinet Office】 CSTP Supercomputer Strategic ICT**  
**PT, Japan Prize Selection Committees, etc.**  
**【MEXT】 Info. Sci. & Tech. Committee,**  
**Supercomputers (Earth Simulator, HPCI Promo., Next**  
**Gen. Supercomputer K) Committees, etc.**

# IEEE Computer Society



**The first President from the outside of USA and Canada in 72 years history of IEEE CS**

**Bjarne Stroustrup: Morgan Stanley & Columbia Univ.**  
**2018 IEEE Computer Society Computer Pioneer Award**  
 IEEE COMPSAC2018 Keynote & Award Ceremony



July 26, 2018, Keynote, Hitotsubashi Hall



July 25, 2018 Award Ceremony Rihga Royal Hotel Tokyo

•84,000+ members



- 480 chapters
- 168 countries
- 31 technical committees & councils

# Bob Ramakrishna Rau Award Lunch in MICRO51 in Fukuoka Japan on Oct. 23, 2018.

**ACM/IEEE CS Micro51 with record high 706 Participants  
was operated by CS this year.**



**Rau Award Winner Dr. Ravi Nair,  
Rau Award Chair Dr. Kemal  
Ebcioglu & CS President Hironori  
Kasahara**



**General co-chairs Profs Koji Inoue & Mark Oskin with CS distinguished  
researchers and ACM SIGARCH CARE member in Banquet.**

# IEEE CS Awards Ceremonies with CS President 2018



**June BoG Award Dinner  
with CS Award Winners and  
their Families, Phoenix**



**Technical  
Achievement Award, in  
COMPSAC,  
Tokyo**



**Computer  
Pioneer Award  
to C++ Bjarne  
Stroustrup in  
COMPSAC,  
Tokyo**



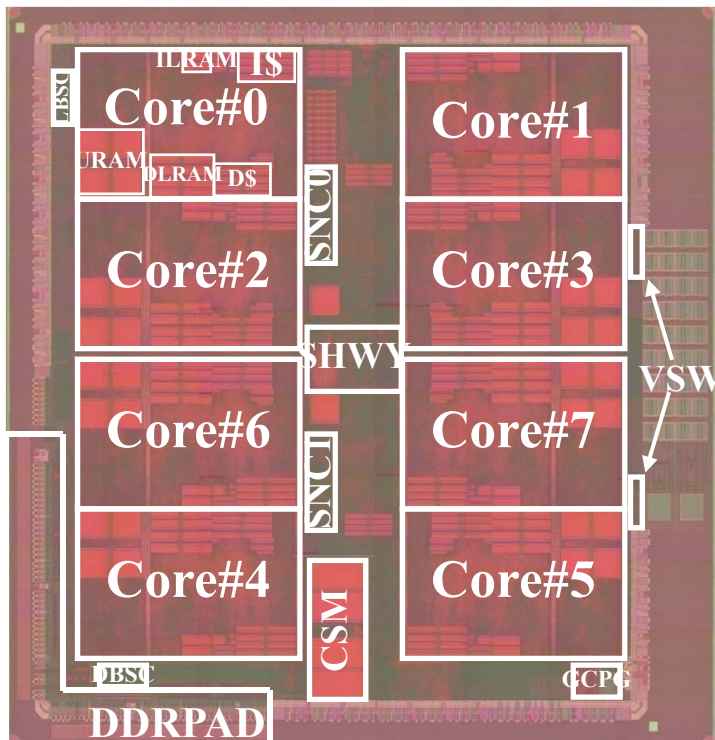
**B. Ramakrishna  
Rau Award in  
MICRO,  
Fukuoka**



**Award Ceremony in SC (Super Computing 2018 with 13 thousands participants), Dallas**

# 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  $\propto$  Frequency<sup>3</sup>

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>

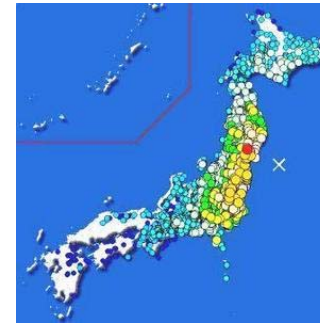
If 8cores are integrated on a chip,  
Power is still 1/8 and  
Performance becomes 2 times .

## Parallel Soft is important for scalable performance of multicore (LCPC2015)

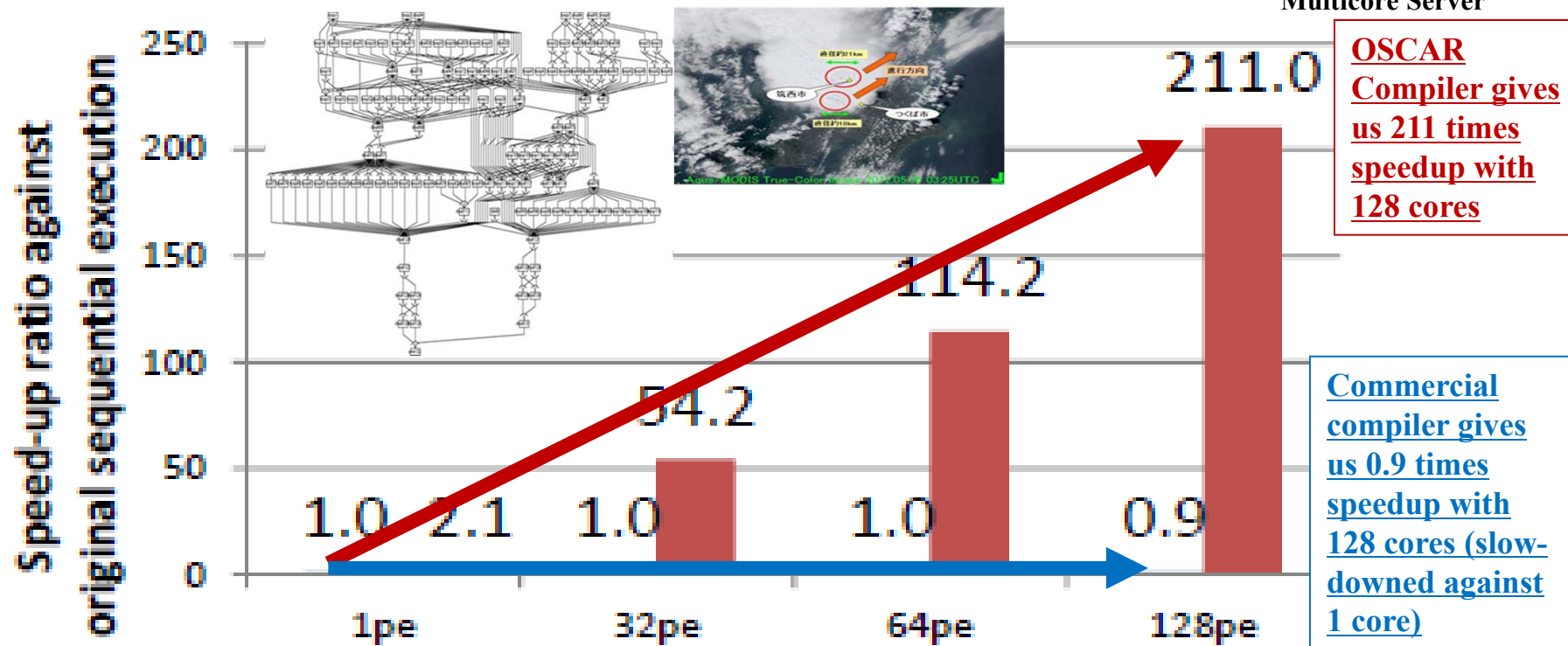
- Just more cores don't give us speedup
- Development cost and period of parallel software are getting a bottleneck of development of embedded systems, eg. IoT, Automobile

Earthquake wave propagation simulation GMS developed by National Research Institute for Earth Science and Disaster Resilience (NIED)

■ original (sun studio) ■ proposed method



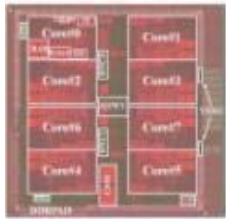
Fjitsu M9000 SPARC Multicore Server



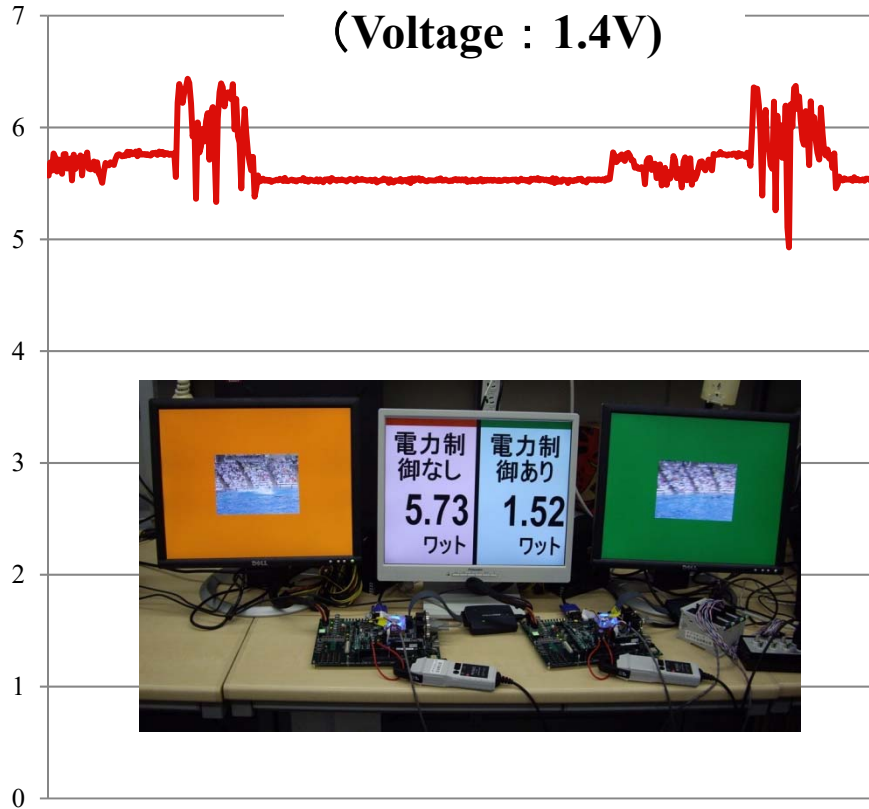
- Automatic parallelizing compiler available on the market gave us no speedup against execution time on 1 core on 64 cores
  - Execution time with 128 cores was slower than 1 core (0.9 times speedup)
- Advanced OSCAR parallelizing compiler gave us 211 times speedup with 128cores against execution time with 1 core using commercial compiler
  - OSCAR compiler gave us 2.1 times speedup on 1 core against commercial compiler by global cache optimization

# 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

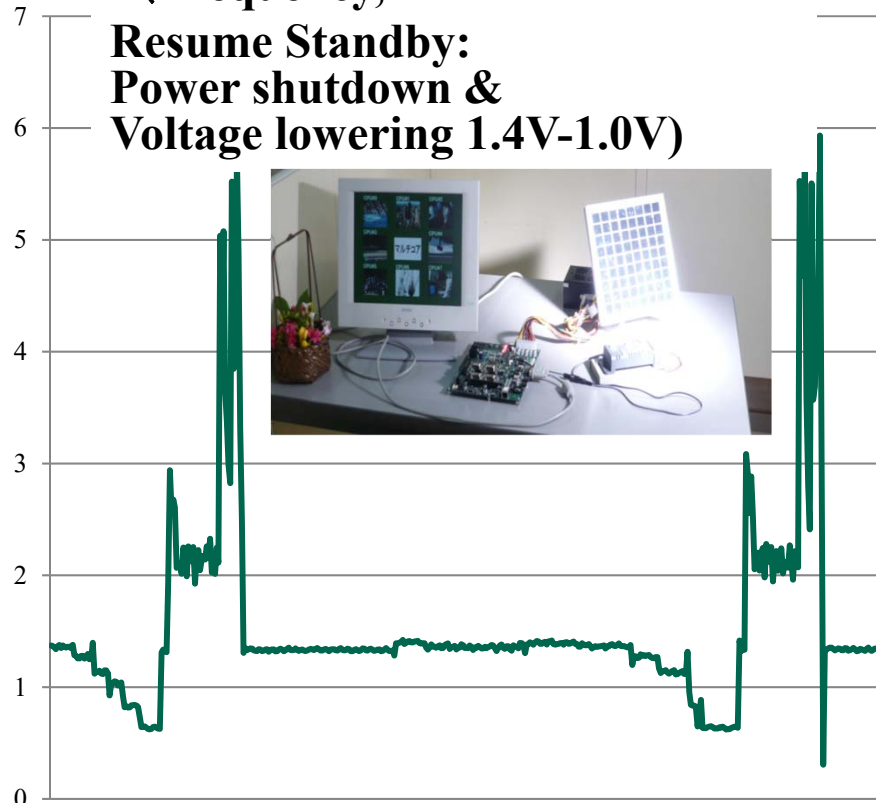


Without Power Control  
(Voltage : 1.4V)



Avg. Power  
5.73 [W]

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



Avg. Power  
1.52 [W]

73.5% Power Reduction



# OSCAR Parallelizing Compiler

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

**Multigrain Parallelization** (LCPC1991,2001,04)  
 coarse-grain parallelism among loops and subroutines (2000 on SMP), near fine grain parallelism among statements (1992) in addition to loop parallelism

## Data Localization

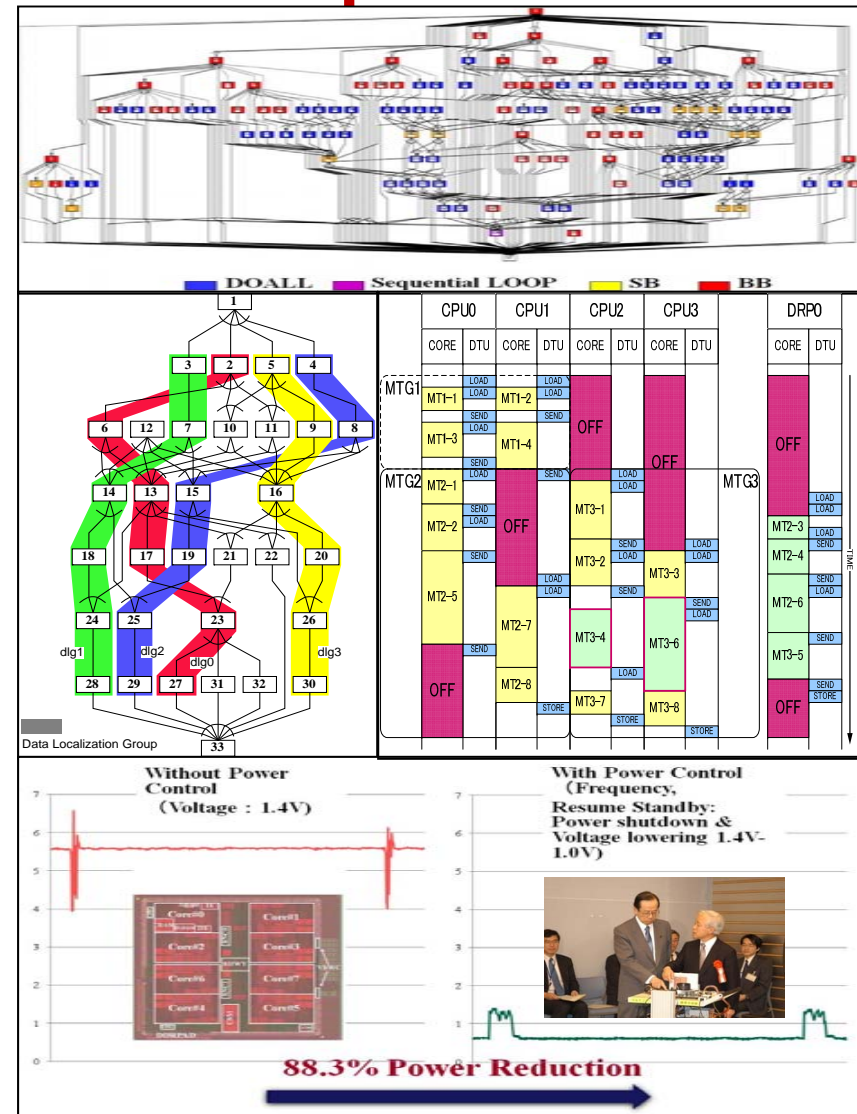
Automatic data management for distributed shared memory, cache and local memory (Local Memory 1995, 2016 on RP2, Cache2001,03)  
 Software Coherent Control (2017)

## Data Transfer Overlapping (2016 partially)

Data transfer overlapping using Data Transfer Controllers (DMAs)

## Power Reduction

(2005 for Multicore, 2011 Multi-processes, 2013 on ARM)  
 Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.

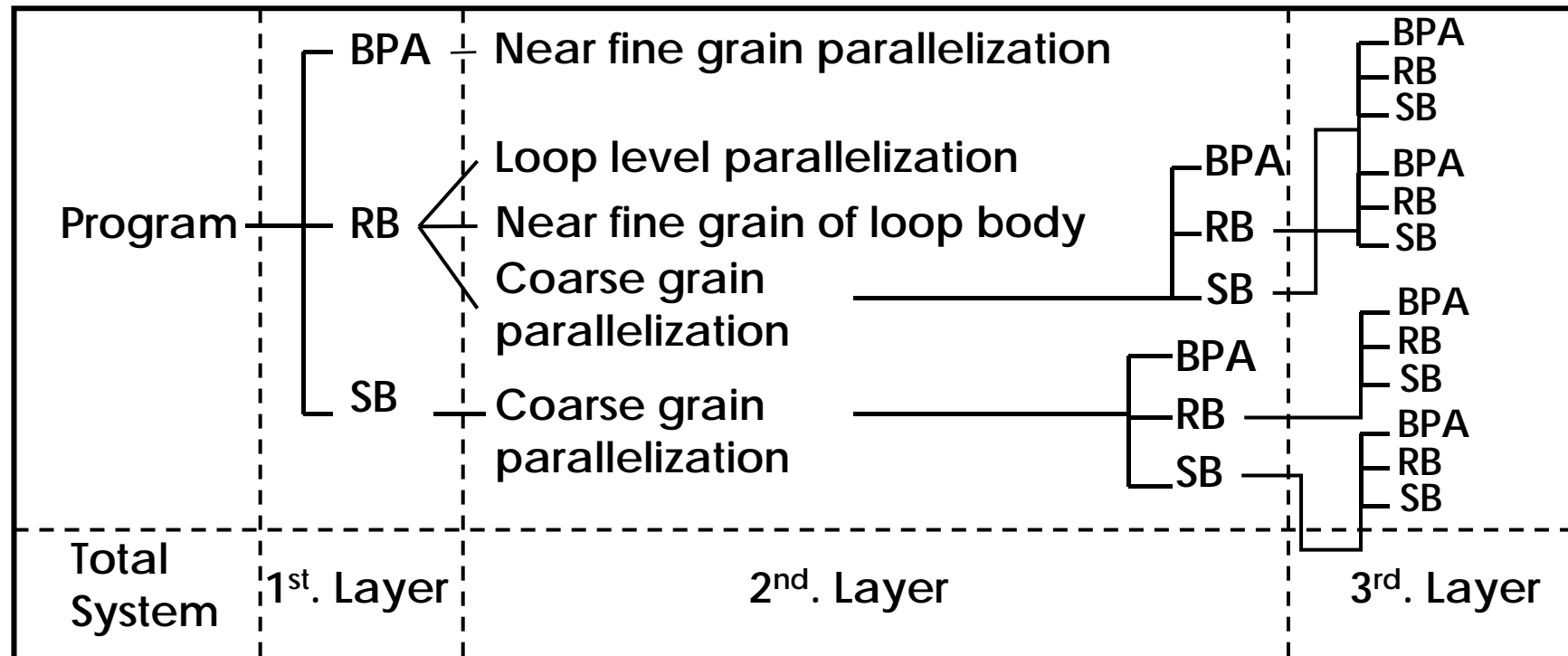




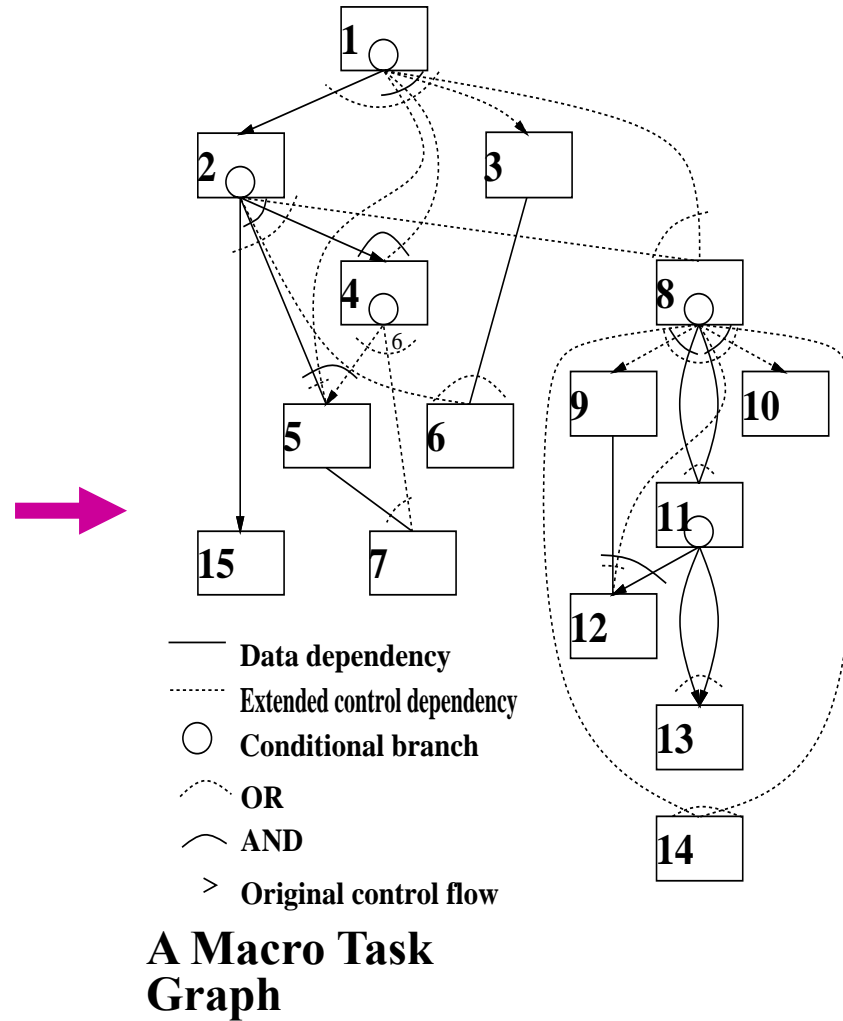
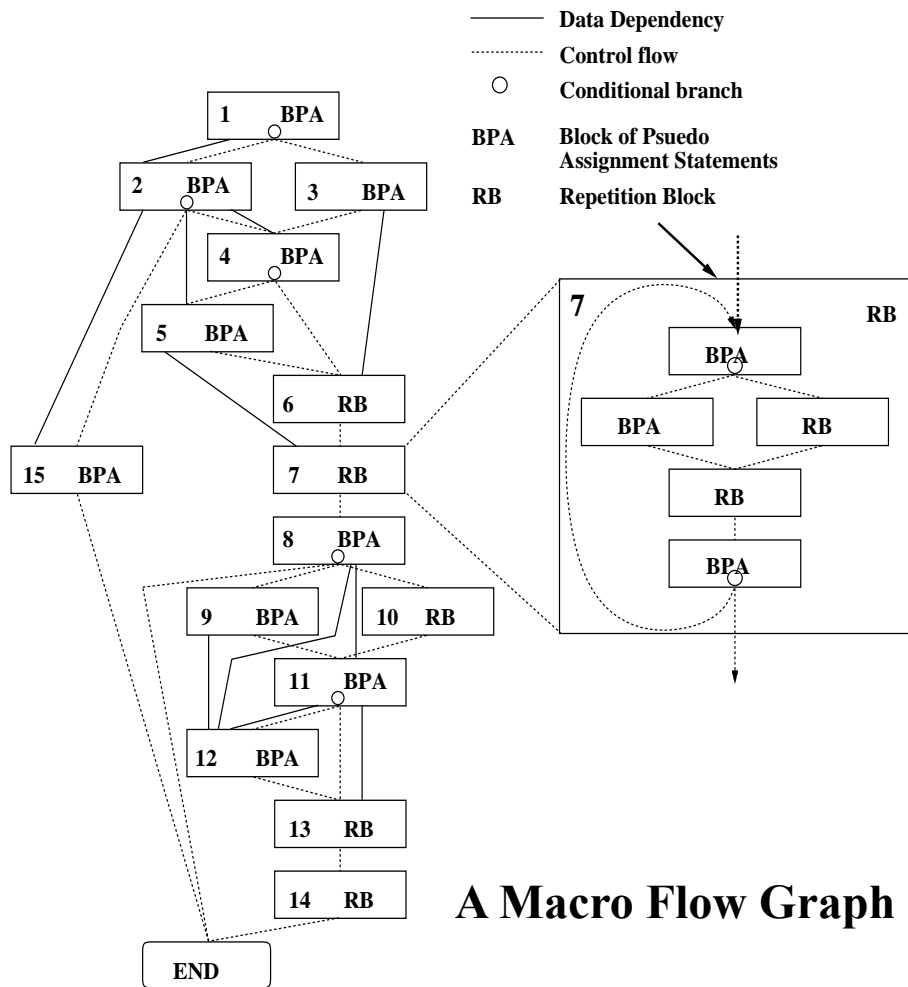
# Generation of Coarse Grain Tasks

## ■ 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)



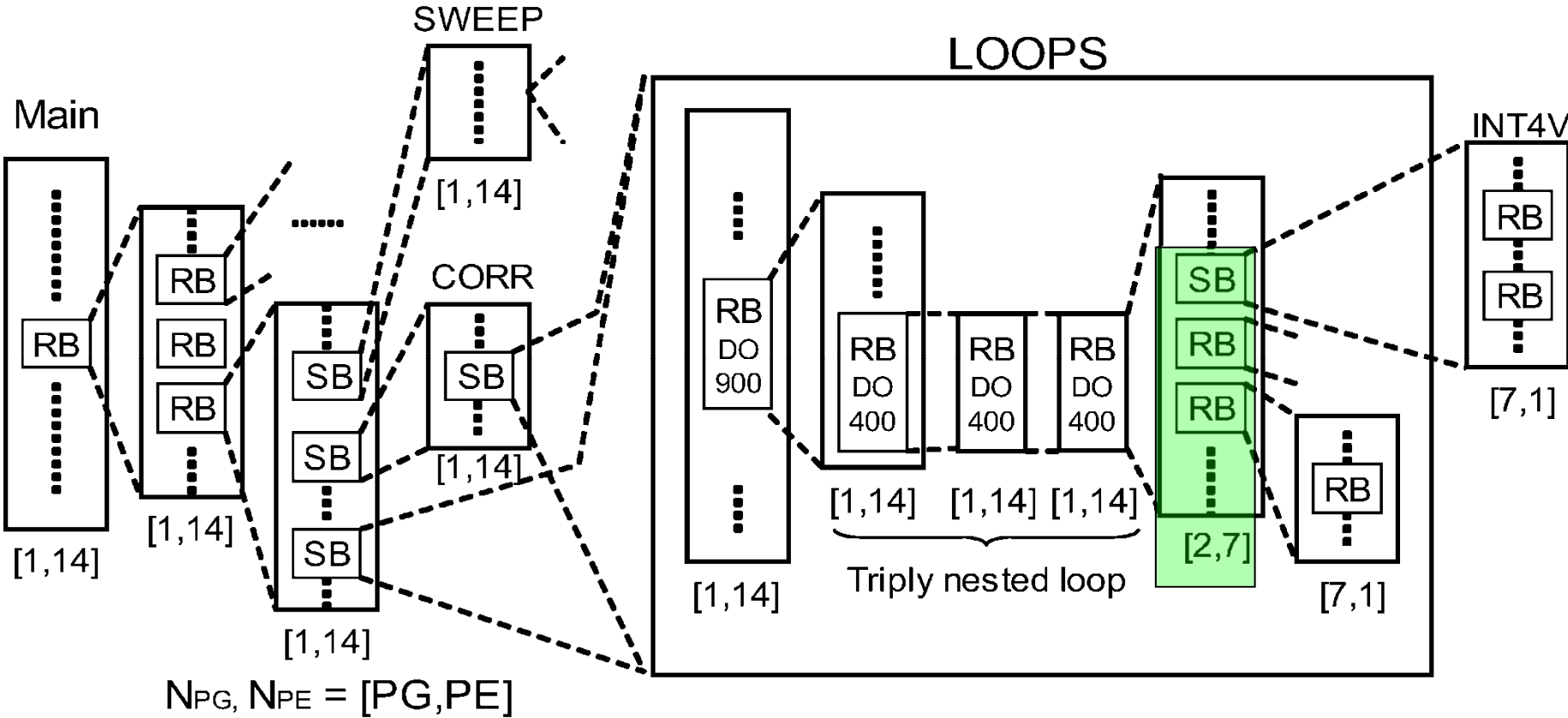
### Earliest Executable Conditions

Macrotask No.	Earliest Executable Condition
1	
2	1 2
3	(1) 3
4	2 4 OR (1) 3
5	(4) 5 AND [ 2 4 OR (1) 3 ]
6	3 OR (2) 4
7	5 OR (4) 6
8	(2) 4 OR (1) 3
9	(8) 9
10	(8) 10
11	8 9 OR 8 10
12	11 12 AND [ 9 OR (8) 10 ]
13	11 13 OR 11 12
14	(8) 9 OR (8) 10
15	2 15

# Automatic processor assignment in 103.su2cor

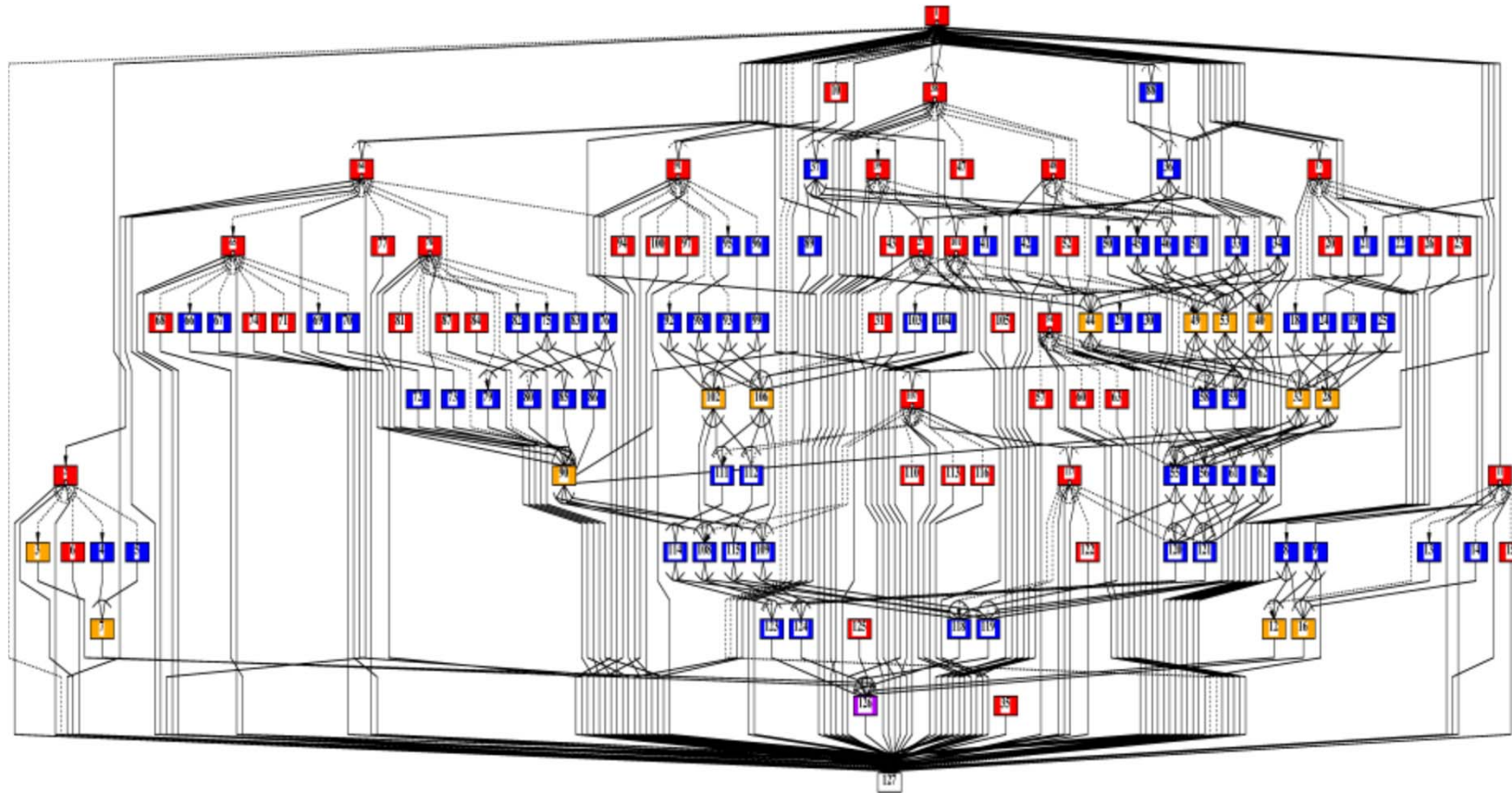
- Using 14 processors

Coarse grain parallelization within DO400



# MTG of Su2cor-LOOPS-DO400

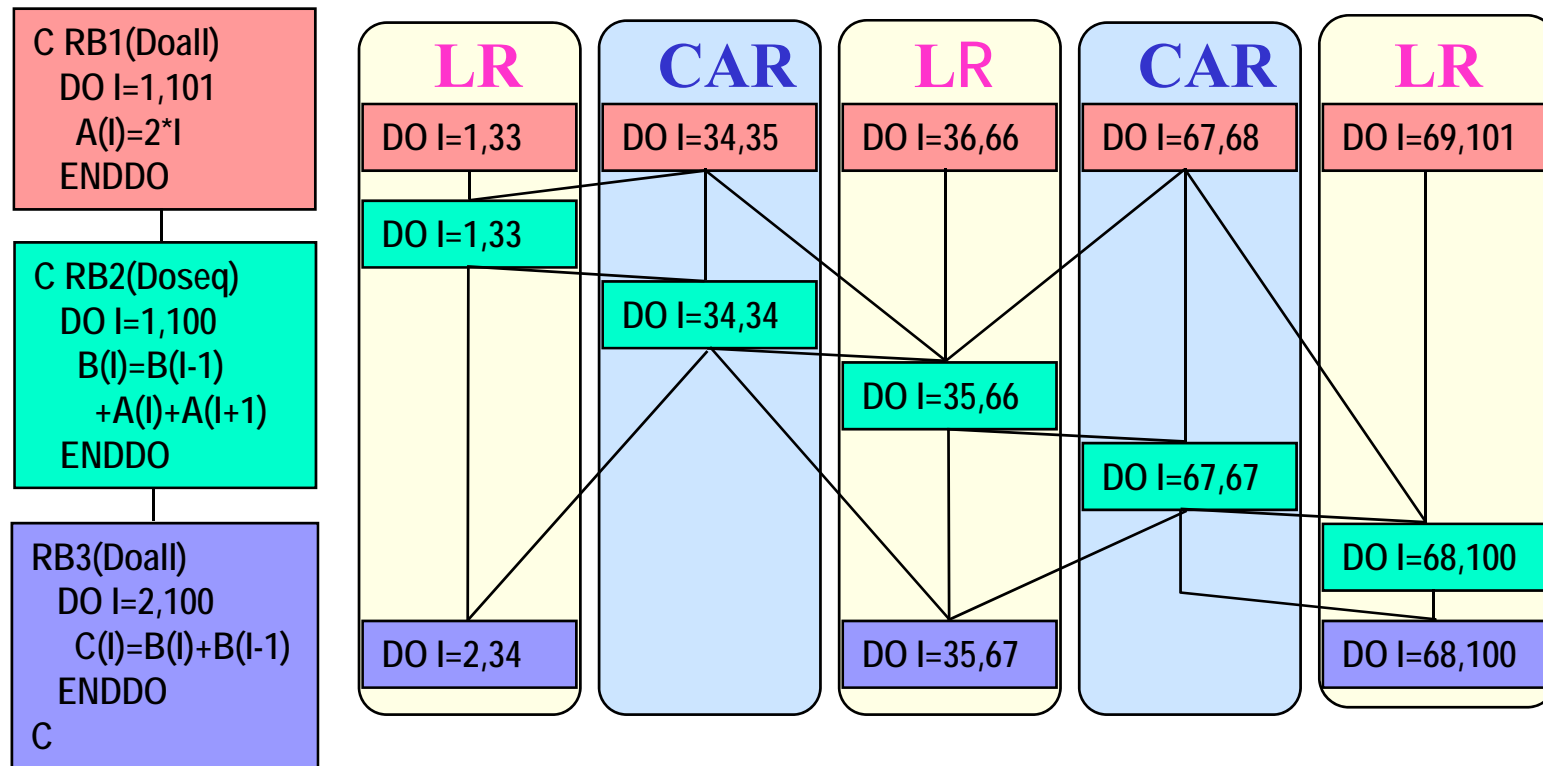
- **Coarse grain parallelism PARA\_ALD = 4.3**



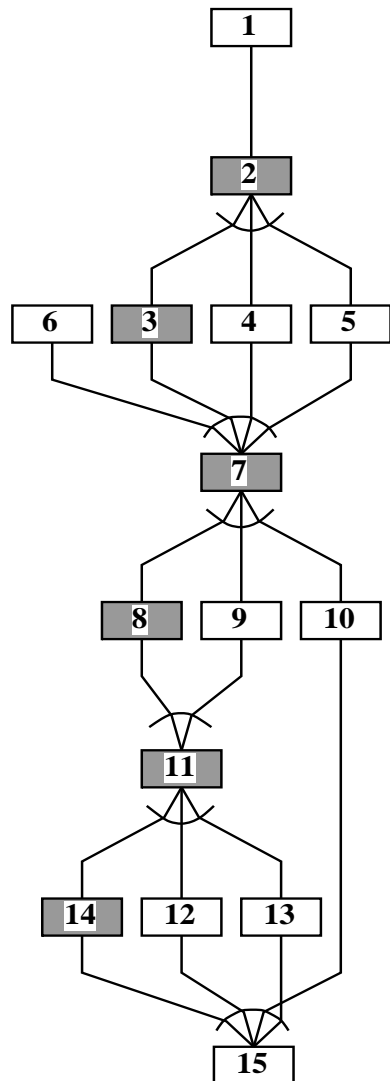
■ DOALL   ■ Sequential LOOP   ■ SB   ■ BB

# Data-Localization: Loop Aligned Decomposition

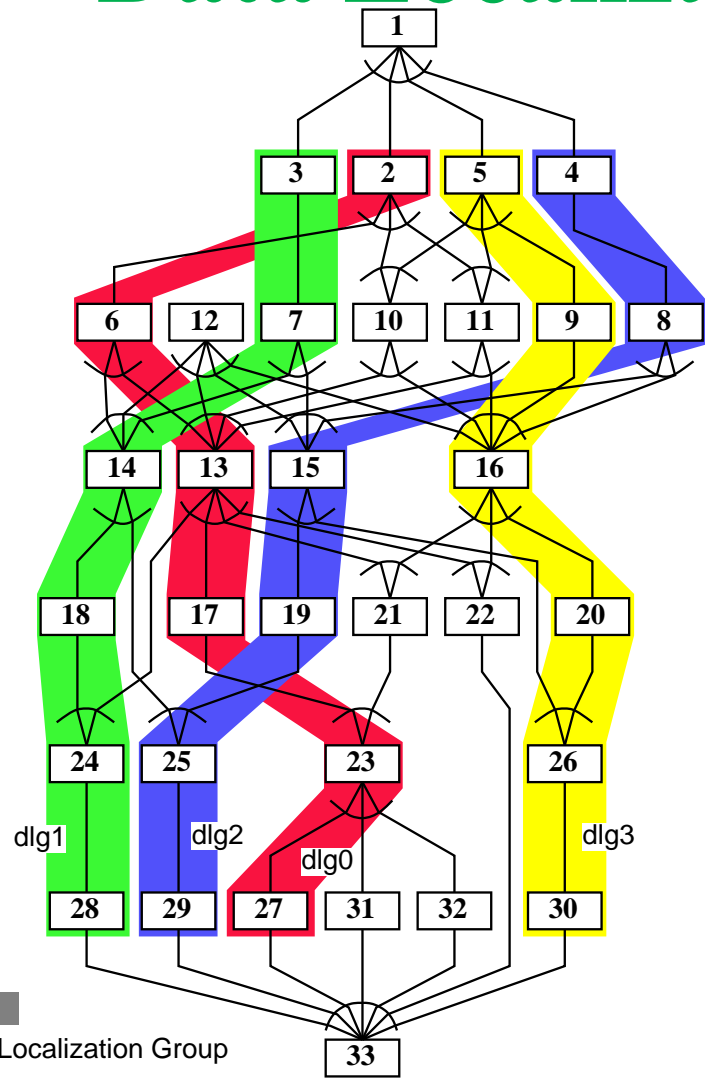
- Decompose multiple loop (Doall and Seq) into **CARs** and **LRs** considering inter-loop data dependence.
  - Most data in **LR** can be passed through LM.
  - LR: Localizable Region, CAR: Commonly Accessed Region**



# Data Localization



MTG



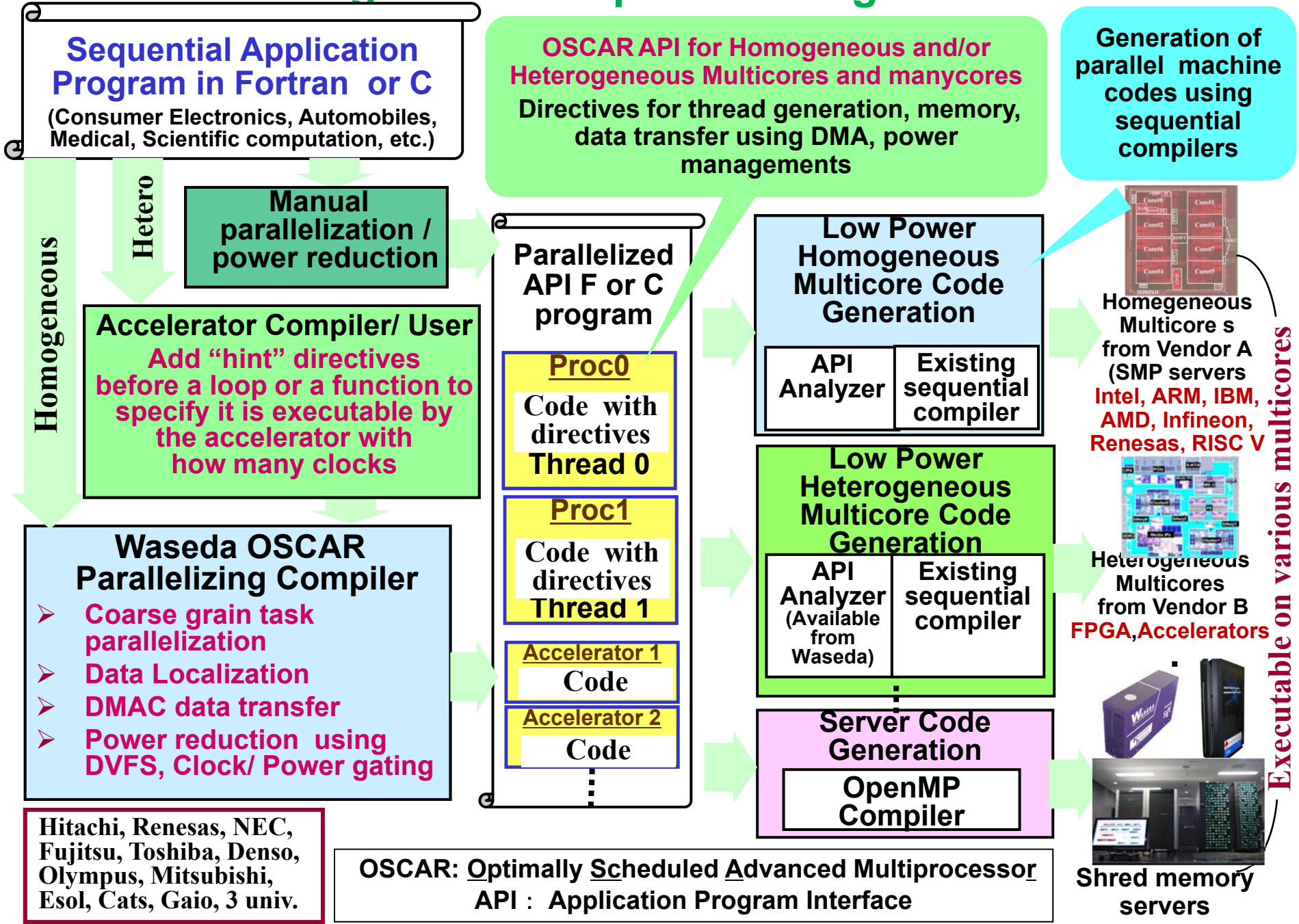
MTG after Division

■ 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

# Multicore Program Development Using OSCAR API V2.0

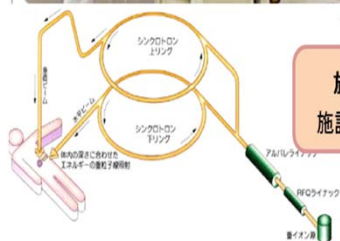




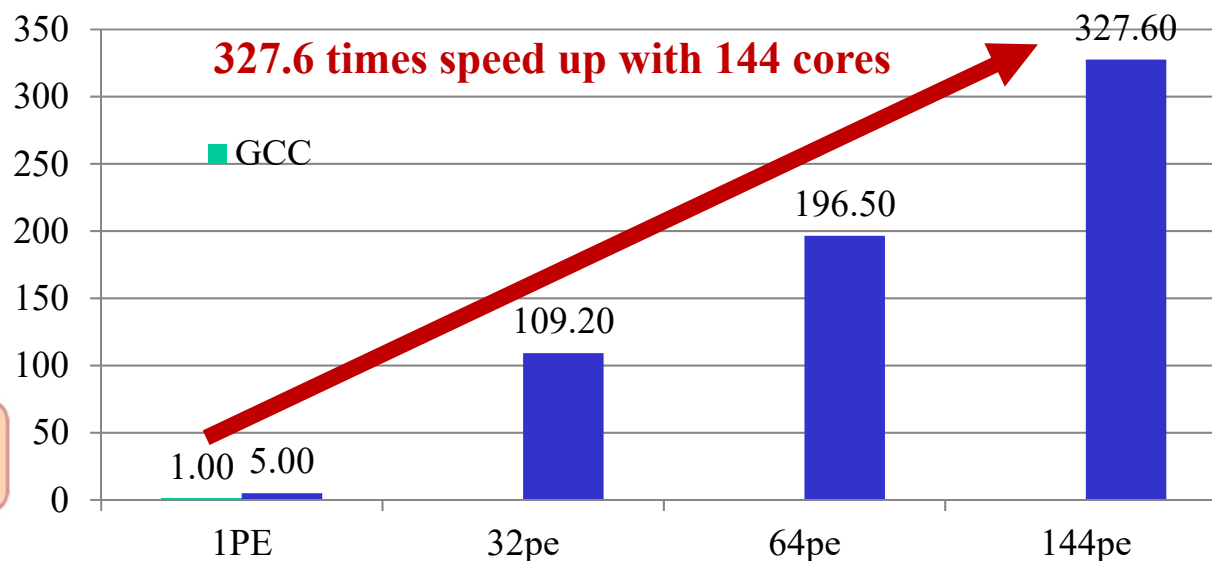
# Performance on Multicore Server for Latest Cancer Treatment Using Heavy Particle (Proton, Carbon Ion)

327 times speedup on 144 cores

Hitachi 144cores SMP Blade Server BS500:  
Xeon E7-8890 V3(2.5GHz 18core/chip) x8 chip

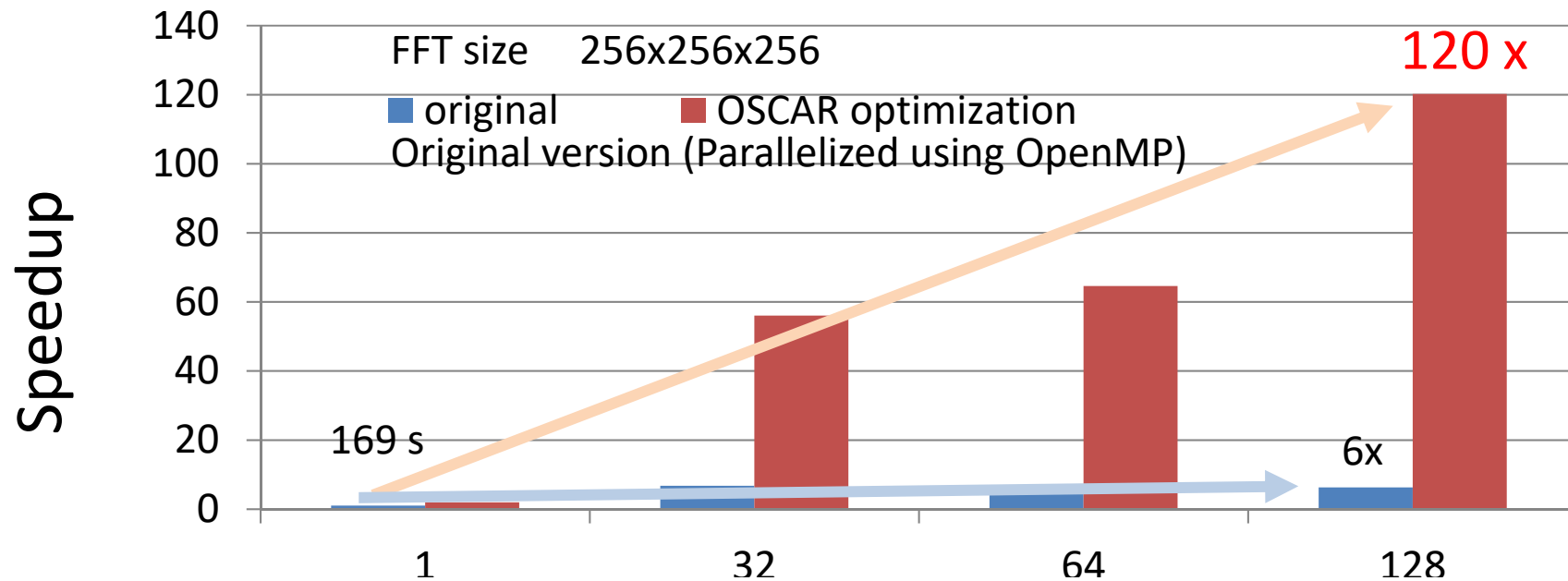


放射線医学研究所  
施設の費用: 120億円



- Original **sequential execution time 2948 sec (50 minutes)** using GCC was reduced to **9 sec with 144 cores** (327.6 times speedup)
- Reduction of treatment cost and reservation waiting period is expected

# Parallelization of 3D-FFT for New Magnetic Material Computation on Hitachi SR16000 Power7 CC-Numa Server



## OSCAR optimization

- reducing number of data transpose with interchange, code motion and loop fusion

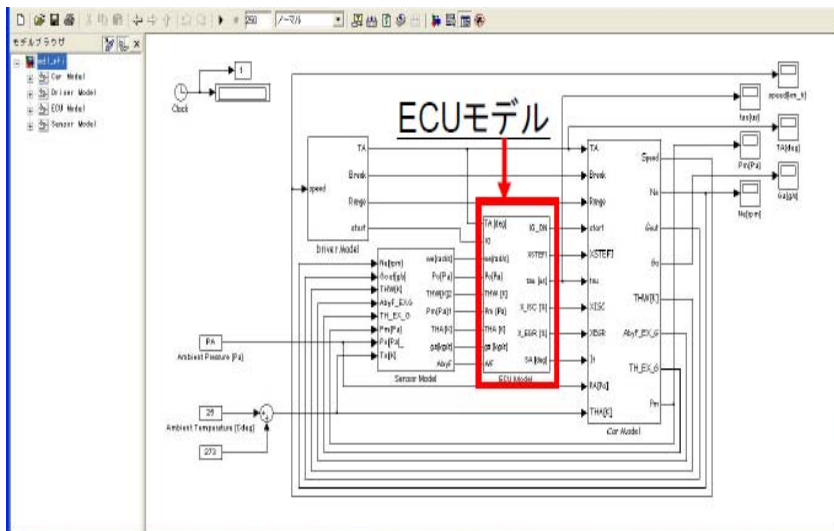
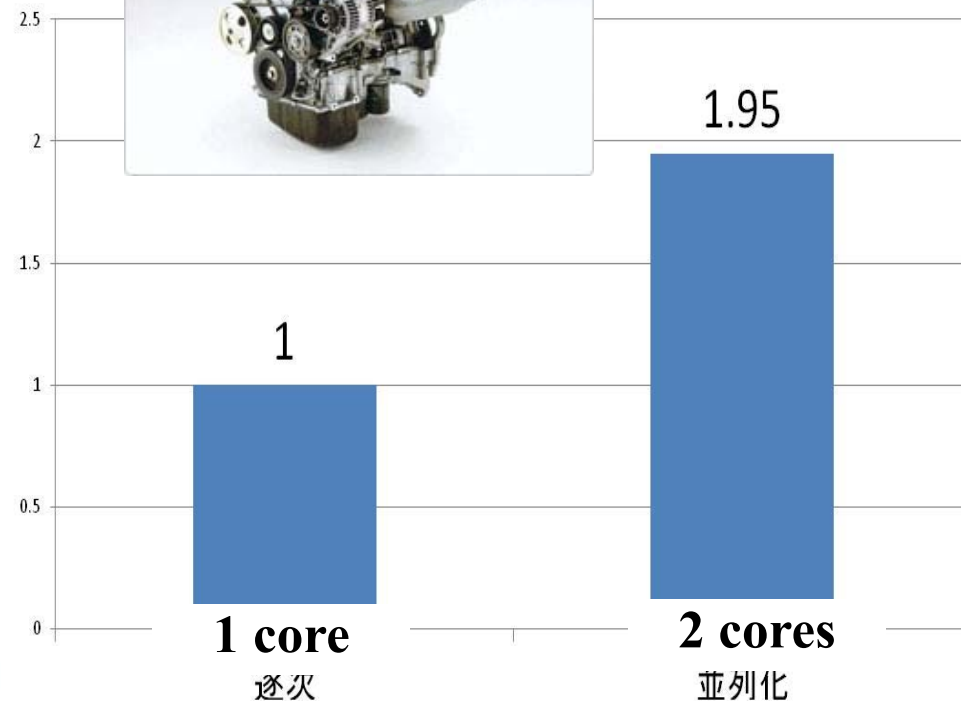


# Engine Control by multicore with Denso

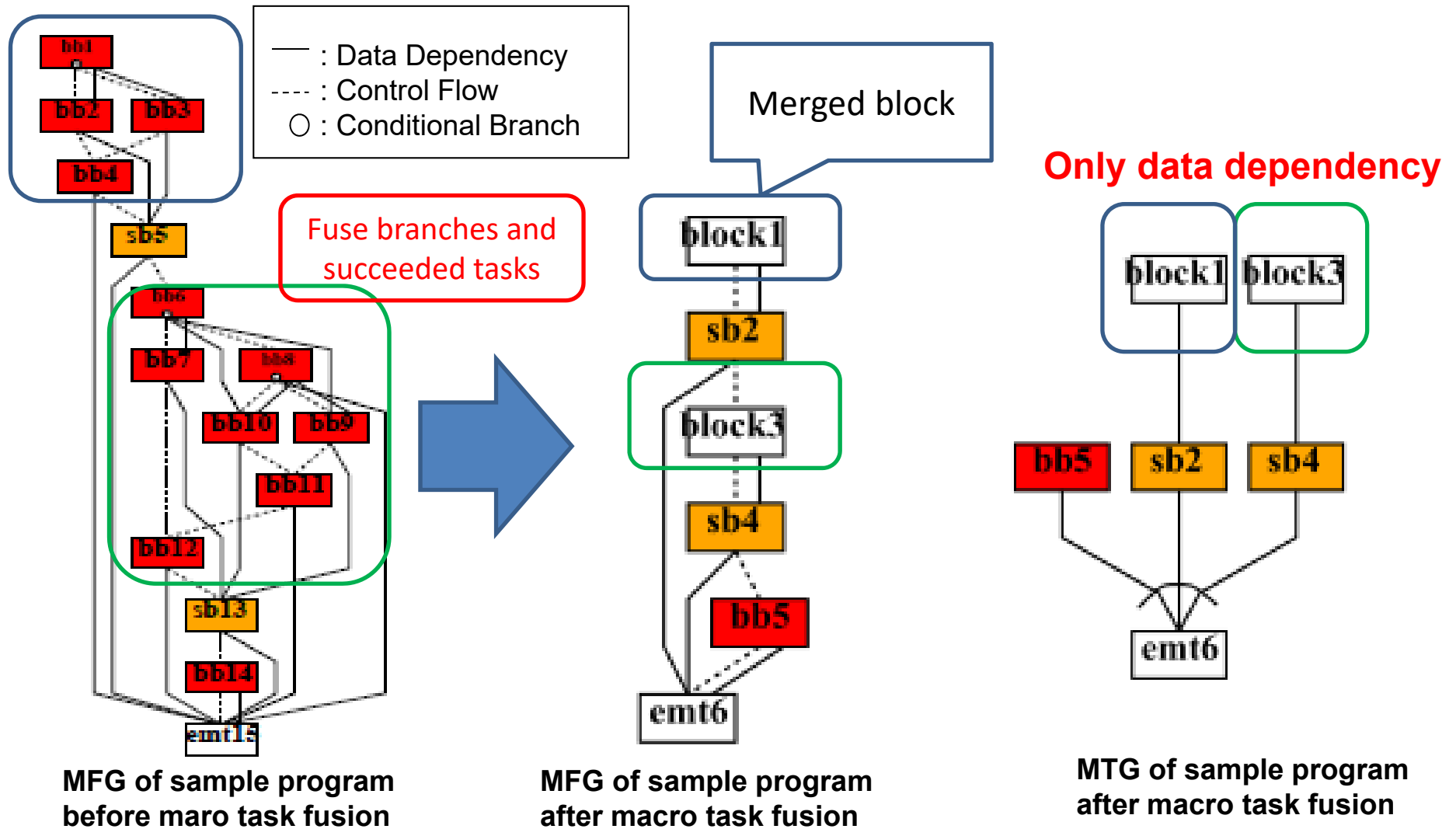
Though so far parallel processing of the engine control on multicore has been very difficult, Denso and Waseda succeeded 1.95 times speedup on 2core V850 multicore processor.



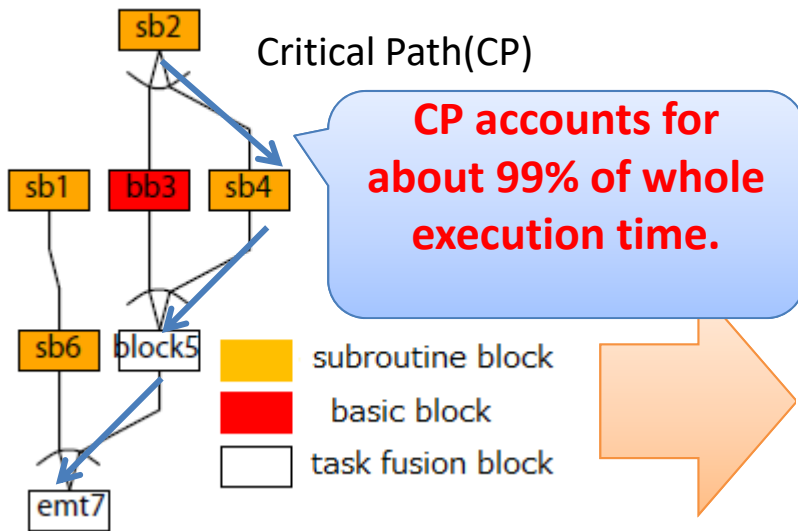
- Hard real-time automobile engine control by multicore using local memories
- Millions of lines C codes consisting conditional branches and basic blocks



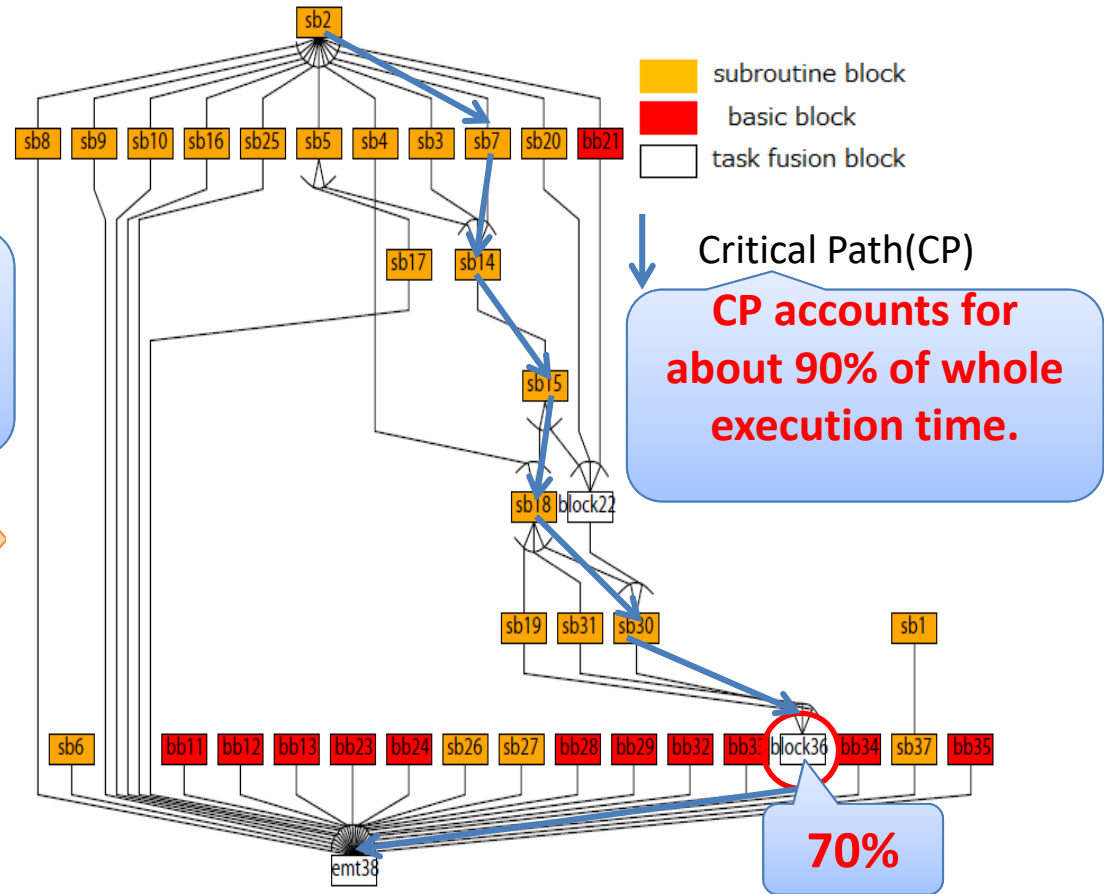
# Macro Task Fusion for Static Task Scheduling



# MTG of Crankshaft Program Using Inline Expansion



MTG of crankshaft program before restructuring



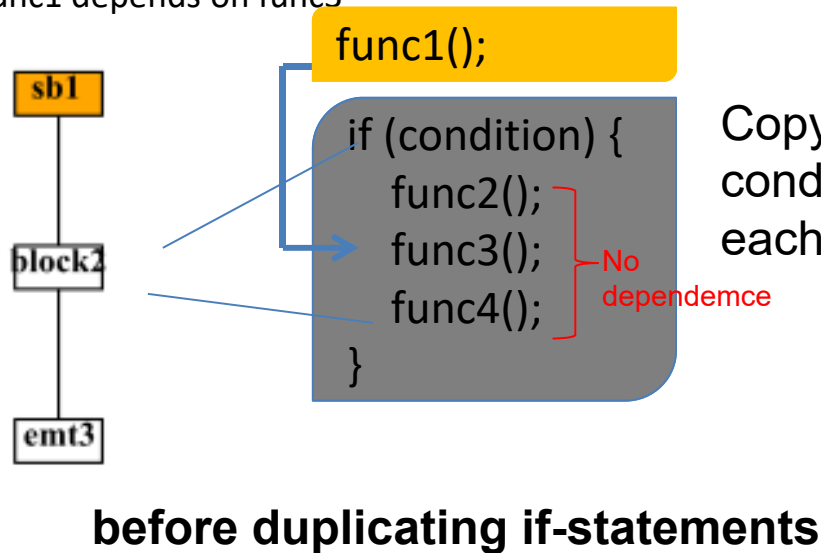
**Not enough coarse grain parallelism yet!**

## 3.2 Restructuring: Duplicating If-statements

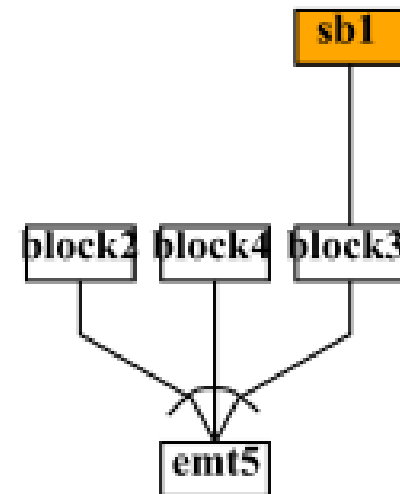
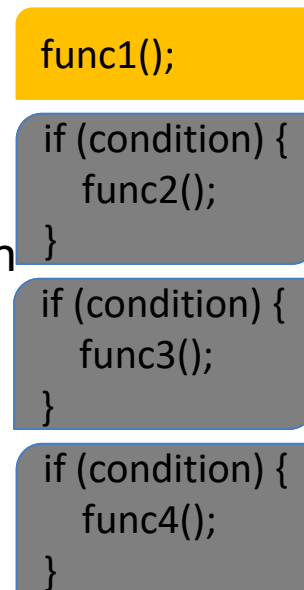
- Duplicating if-statements is effective
  - To increase coarse grain parallelism
- Duplicates fused tasks having inner parallelism

**Improves coarse grain parallelism**

Func1 depends on func3

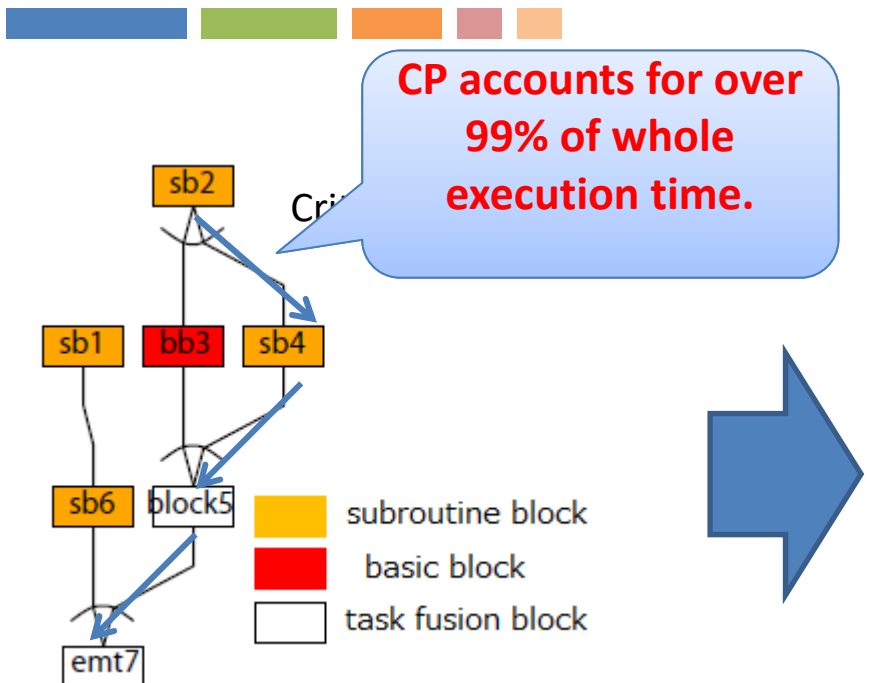


Copying if-condition for each function



**after duplicating if-statements**

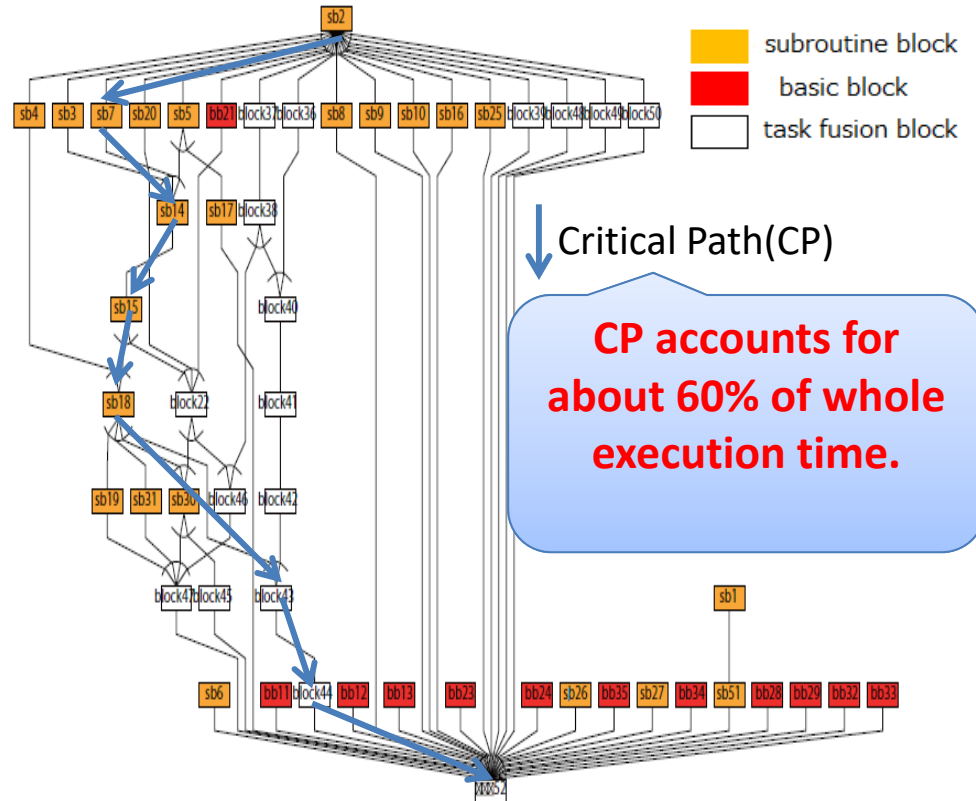
# MTG of Crankshaft Program Using Inline Expansion and Duplicating If-statements



CP accounts for over 99% of whole execution time.

MTG of crankshaft program before restructuring

- Succeed to reduce CP
  - 99% -> 60%

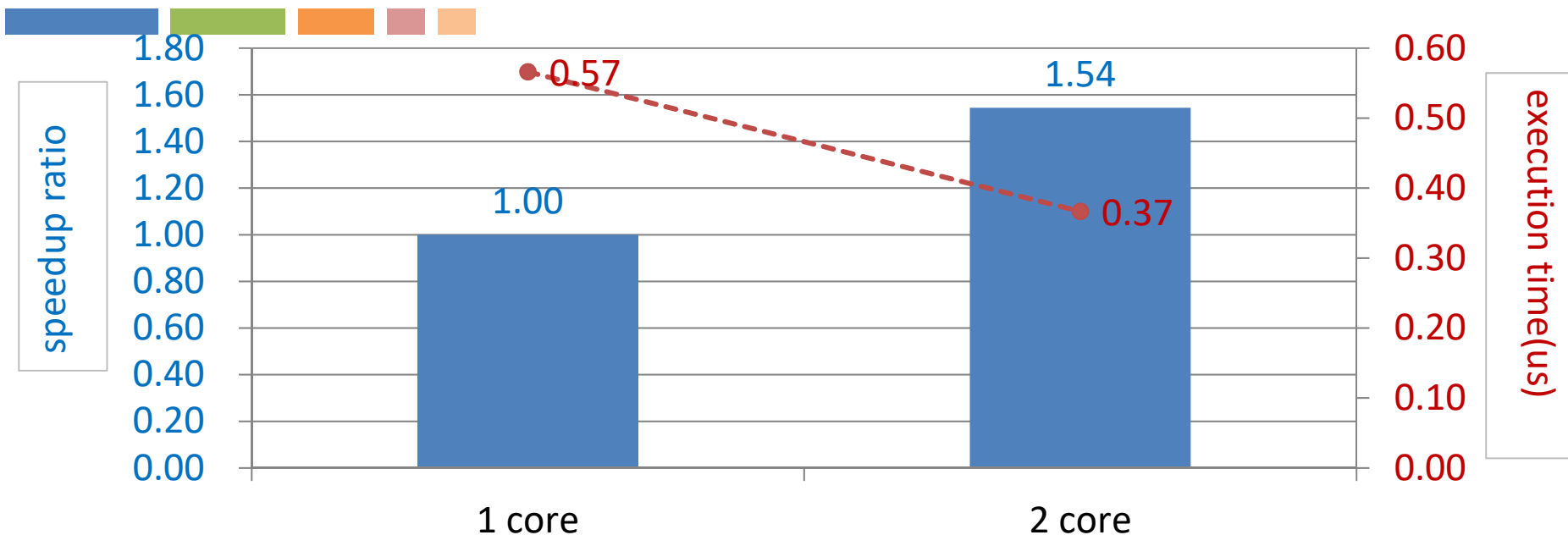


CP accounts for about 60% of whole execution time.

MTG of crankshaft program after restructuring

**Successfully increased coarse grain parallelism**

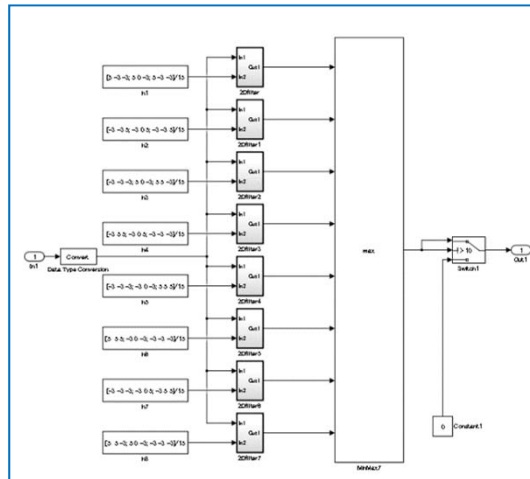
# Evaluation of Crankshaft Program with Multi-core Processors



- 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

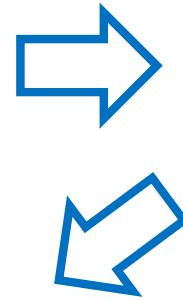


# OSCAR Compile Flow for Simulink Applications



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:
     * Import: '<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' */

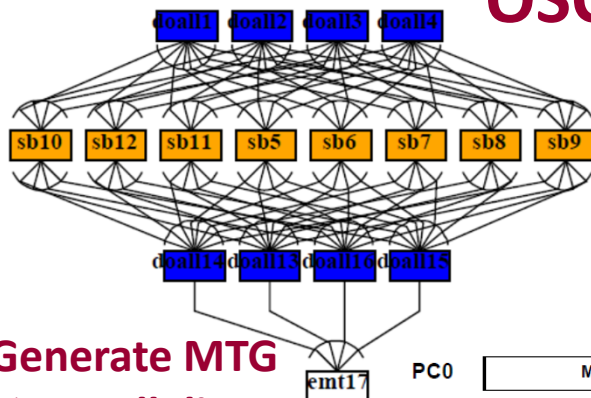
    /* Constant: '<S1>/h1' */
    VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
        VesselExtraction_P.h1_Value, &VesselExtraction_B.Dfilter,
        (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter);

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

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

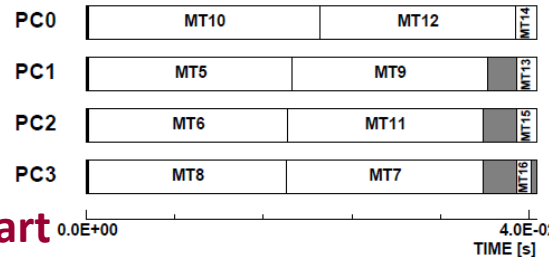
    /* Constant: '<S1>/h2' */
    VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
        VesselExtraction_P.h2_Value, &VesselExtraction_B.Dfilter1,
        (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter1);
}
    
```

C code



(1) Generate MTG  
→ Parallelism

(2) Generate gantt chart  
→ Scheduling in a multicore



## OSCAR Compiler

```

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

    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 );

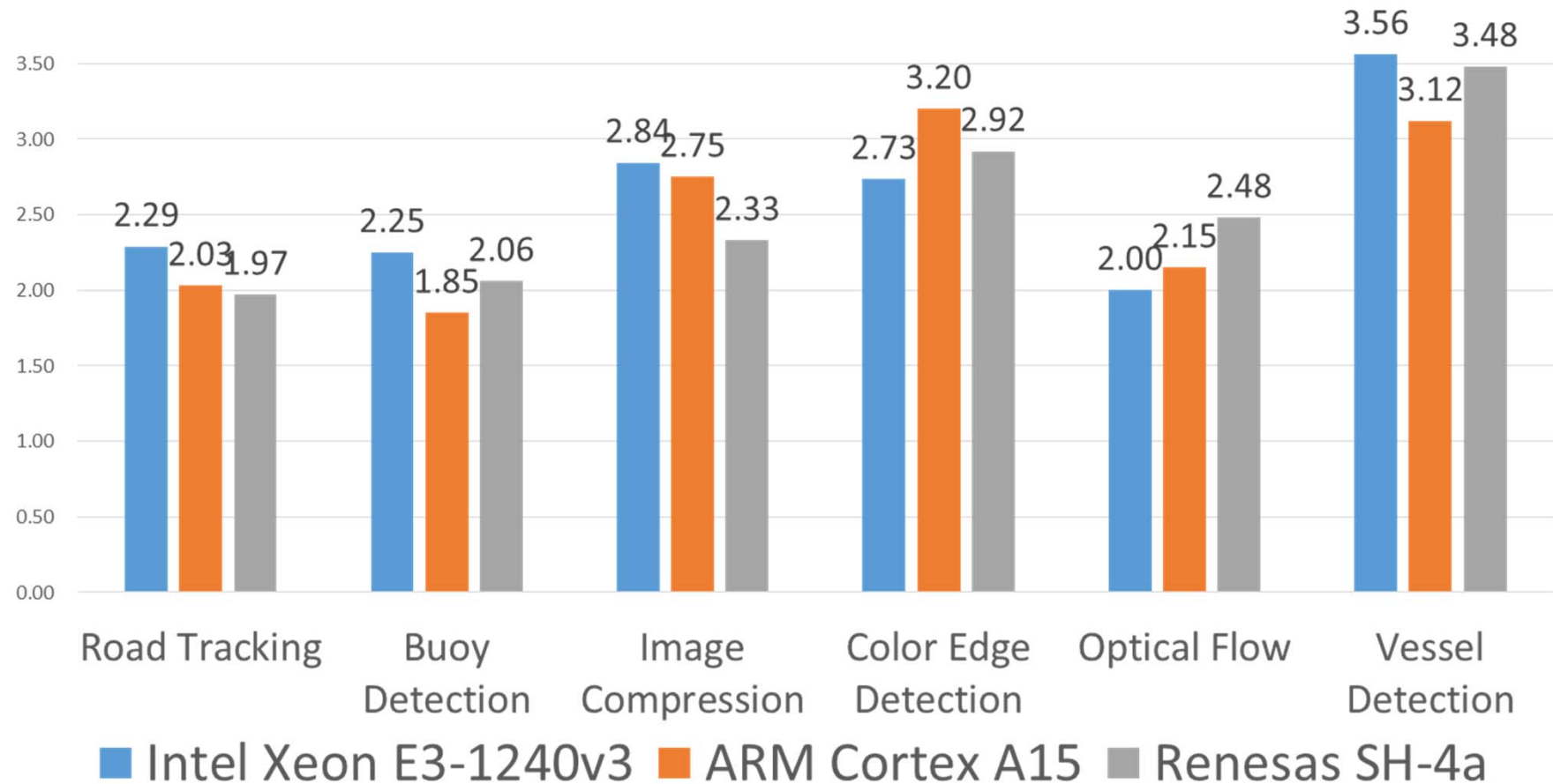
    VesselExtraction_step_PEO ( ) ;

    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-/>

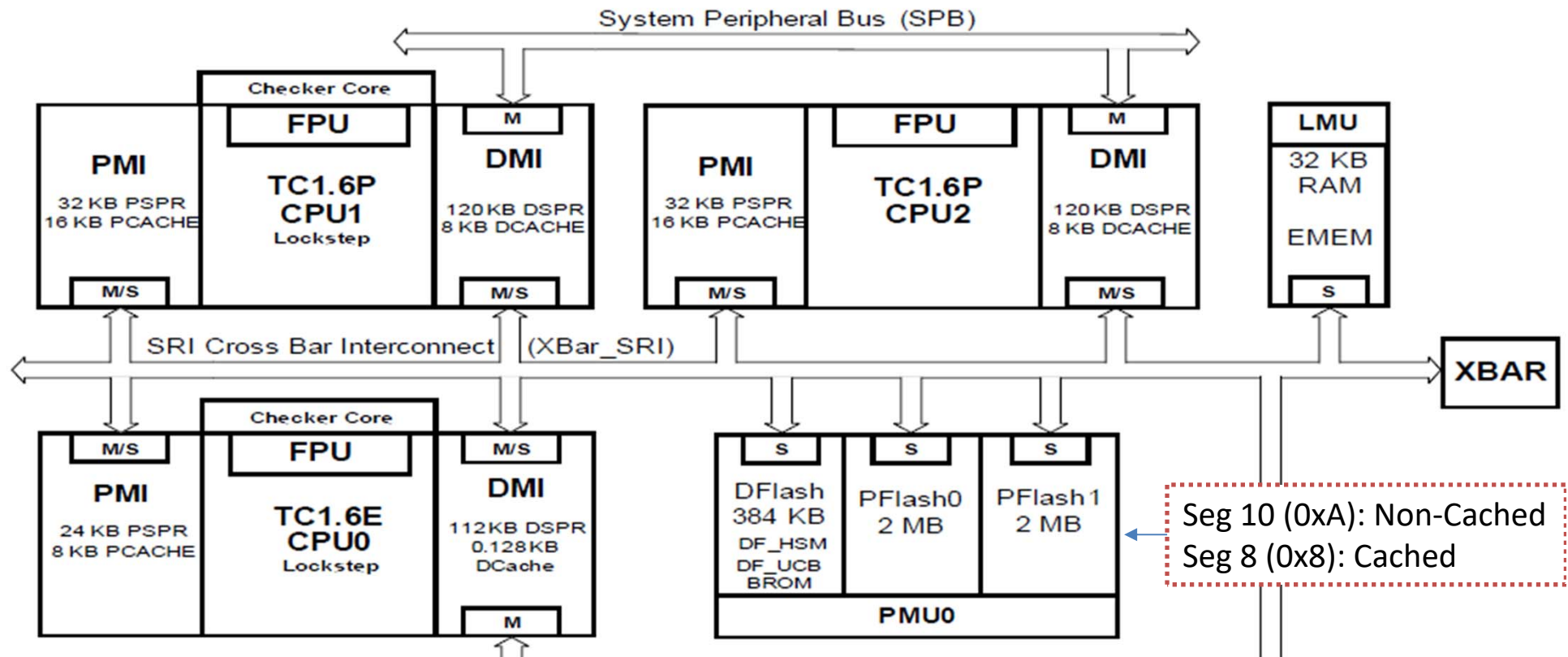
Vessel Detection : <http://www.mathworks.co.jp/matlabcentral/fileexchange/24990-retinal-blood-vessel-extraction/>

# Infineon AURIX TC277

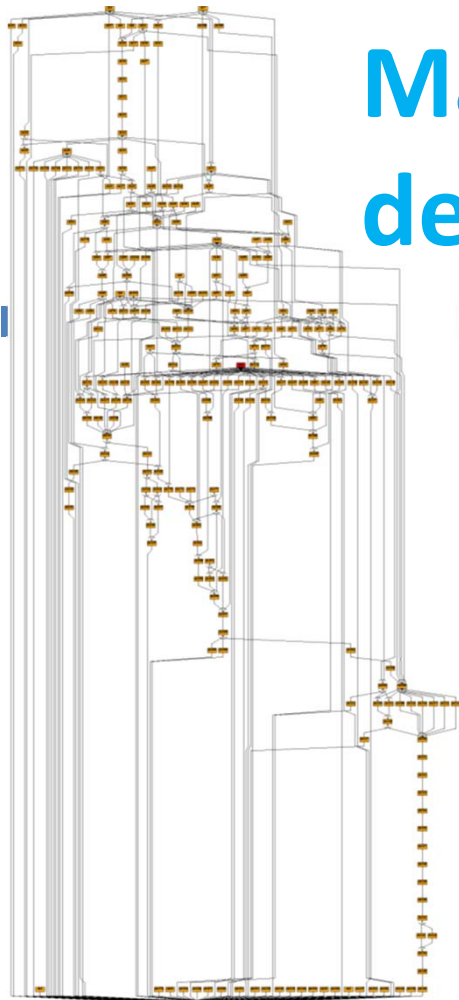


## Abbreviations :

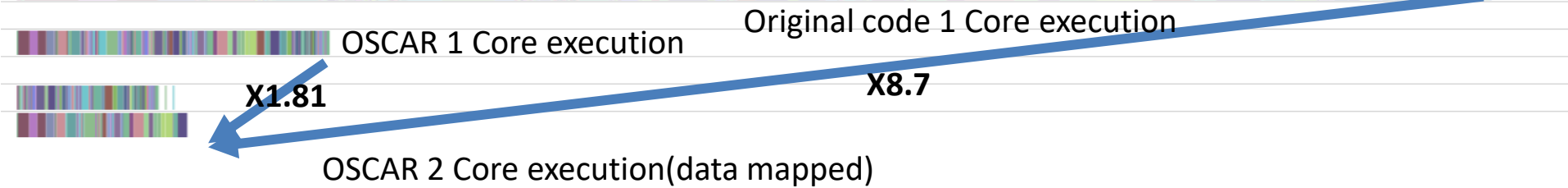
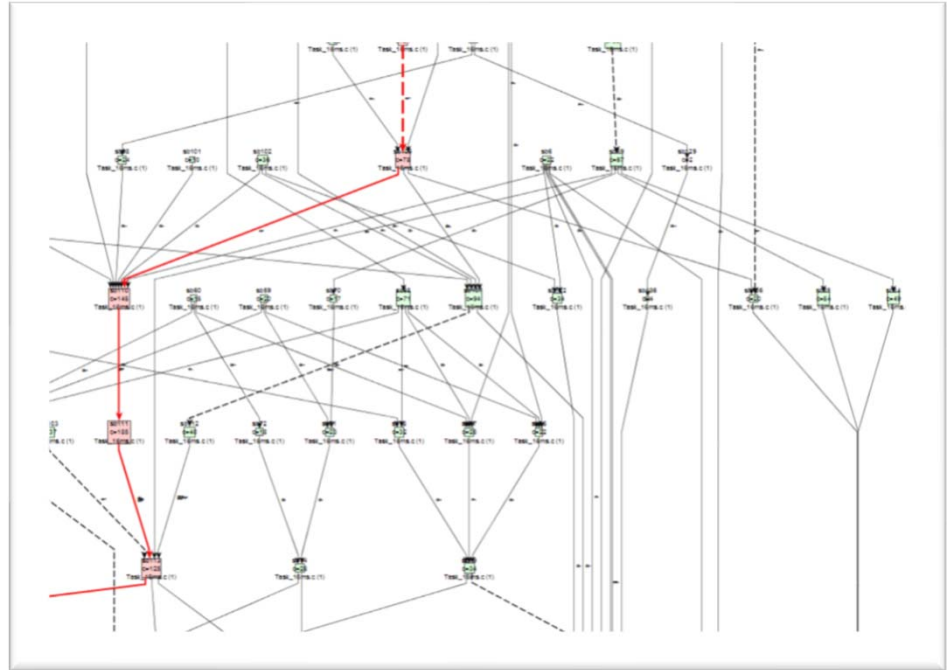
PCACHE:	Program Cache
DCACHE:	Data Cache
DSPR:	Data Scratch-Pad RAM
PSPR:	Program Scratch-Pad RAM
BROM:	Boot ROM
PFlash:	Program Flash
DFlash:	Data Flash (EEPROM)
<b>S</b> :	SRI Slave Interface
<b>M</b> :	SRI Master Interface



# Macrotask Graph, Dependence details and schedules

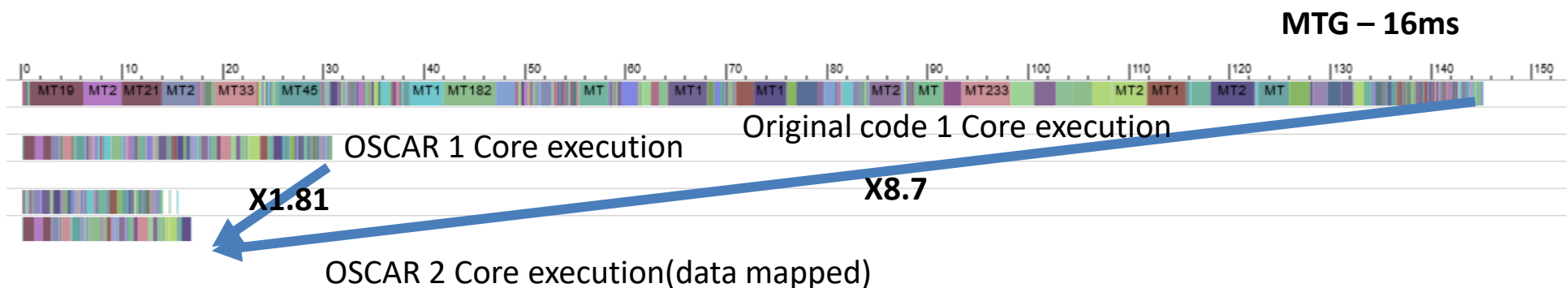


MTG – task\_16ms

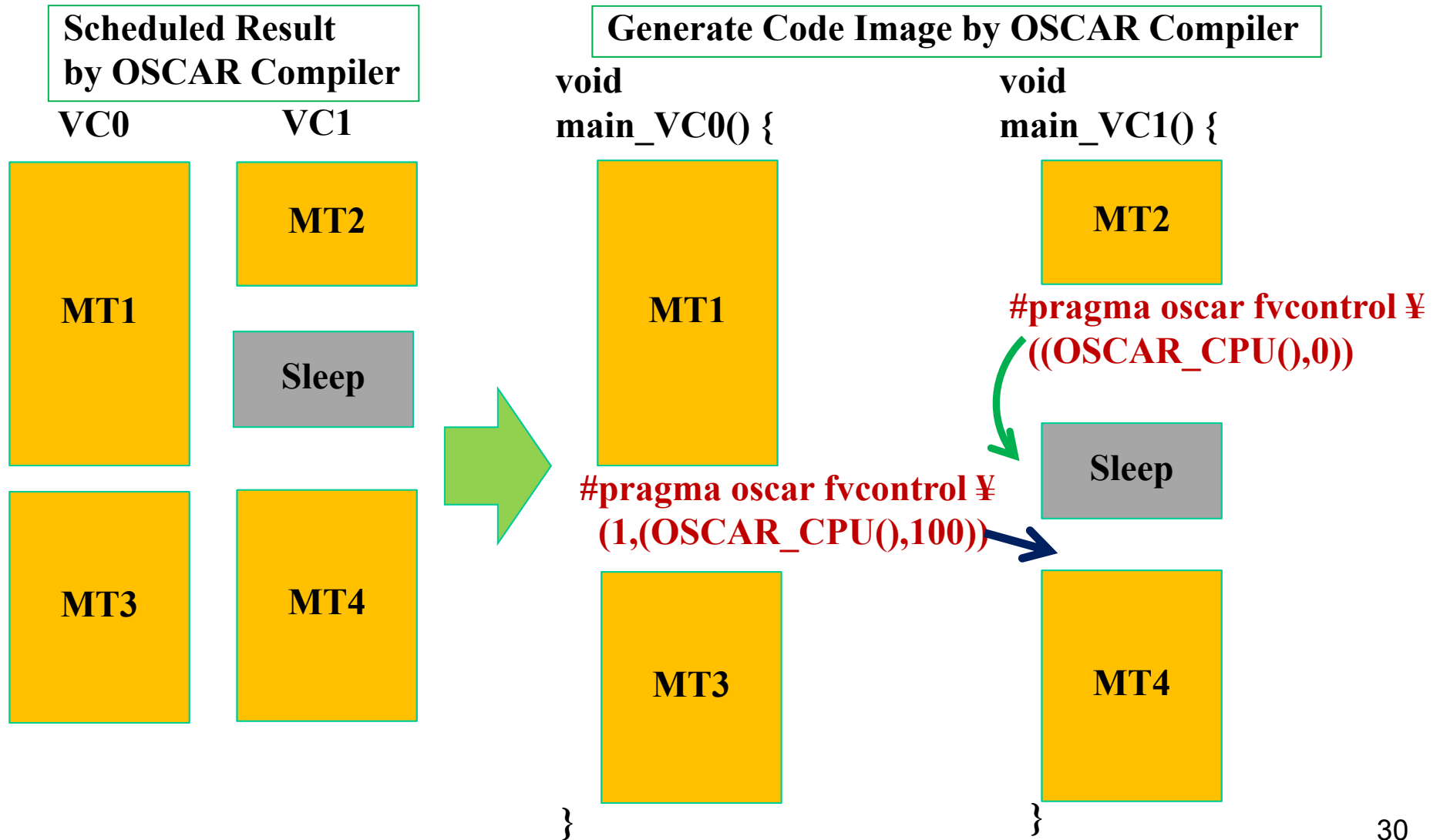


# Automatic Parallelization of an Engine Control C Program with 400 thousands lines on AUTOSAR on 2 cores of Infineon AURIX TC277

- **Original sequential** execution time on 1 core: **145500** cycles
- **Sequential execution time by OSCAR** on 1 core: **29700** cycles
  - **4.9 times speedup on 1 core** against original execution by OSCAR Compilers automatic data allocation for local scratch pad memory, flush memory modules
- **2 core execution by OSCAR** Compiler: **16400** cycles
  - **1.81 times speedup with 2 core** against **1 core execution with OSCAR Compiler**
  - **8.7 times speedup against original sequential execution.**



# Low-Power Optimization with OSCAR API

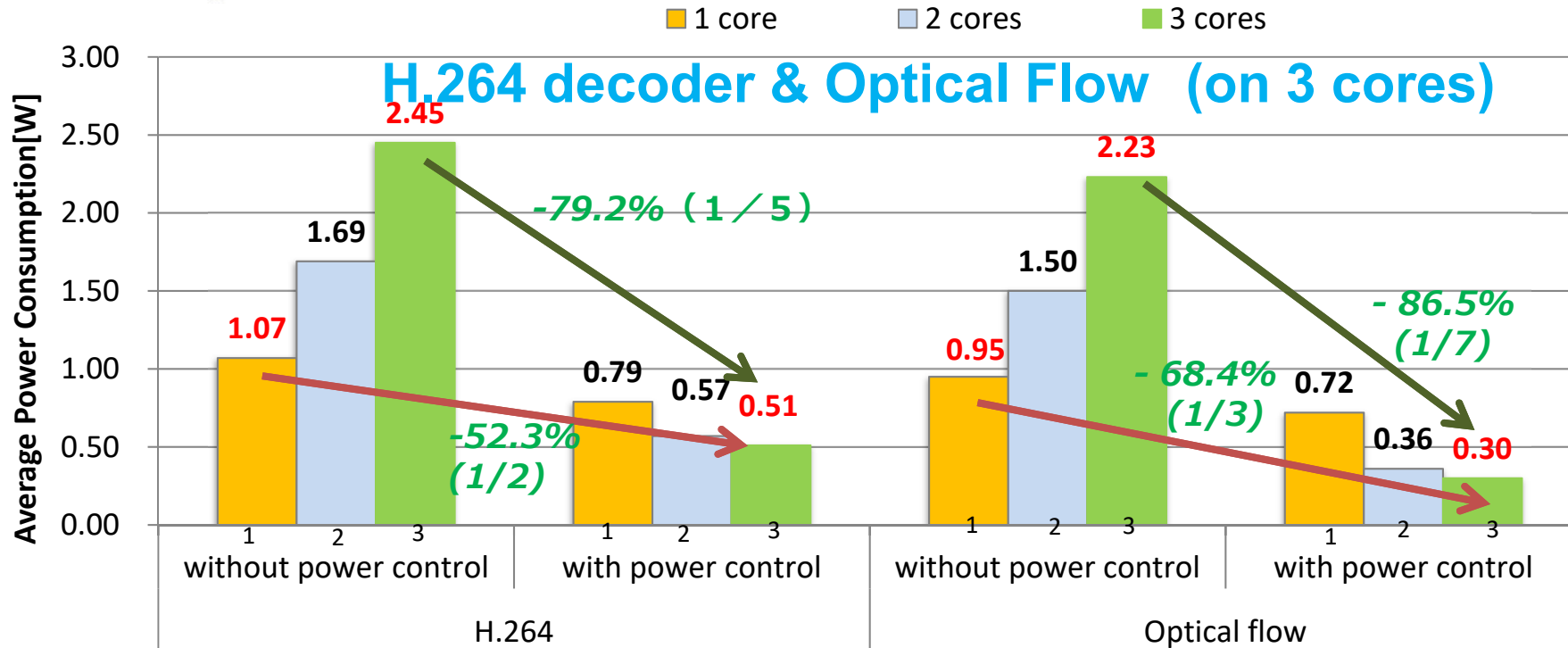
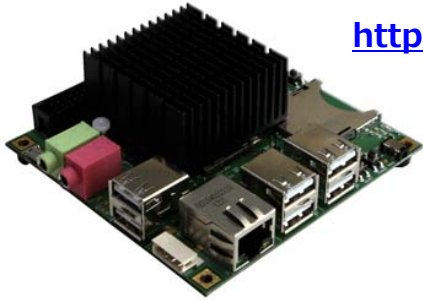


# Automatic Power Reduction on ARM CortexA9 with Android

[http://www.youtube.com/channel/UCS43INYEIkC8i\\_KIgfZYQBQ](http://www.youtube.com/channel/UCS43INYEIkC8i_KIgfZYQBQ)

ODROID X2

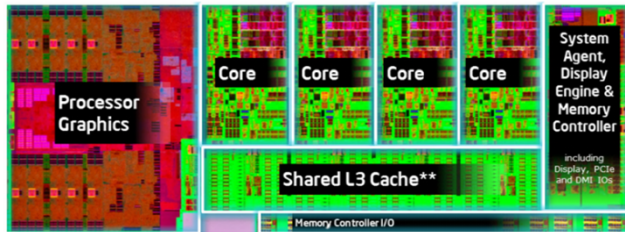
Samsung Exynos4412 Prime, ARM Cortex-A9 Quad core  
1.7GHz~0.2GHz, used by Samsung's Galaxy S3



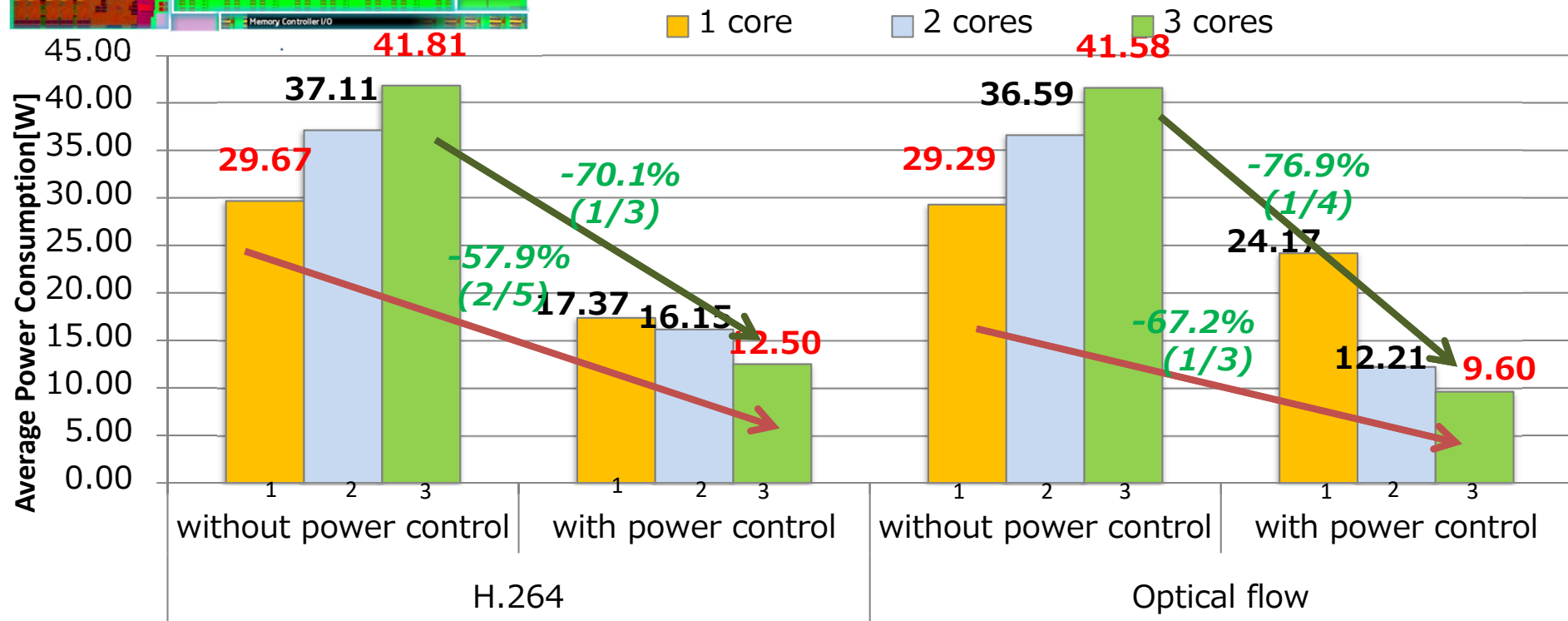
Power for 3cores was reduced to **1/5~1/7** against without software power control  
Power for 3cores was reduced to **1/2~1/3** against ordinary 1core execution

# Automatic Power Reuction on Intel Haswell

## H.264 decoder & Optical Flow (3cores)



H81M-A, Intel Core i7 4770k  
Quad core, 3.5GHz~0.8GHz

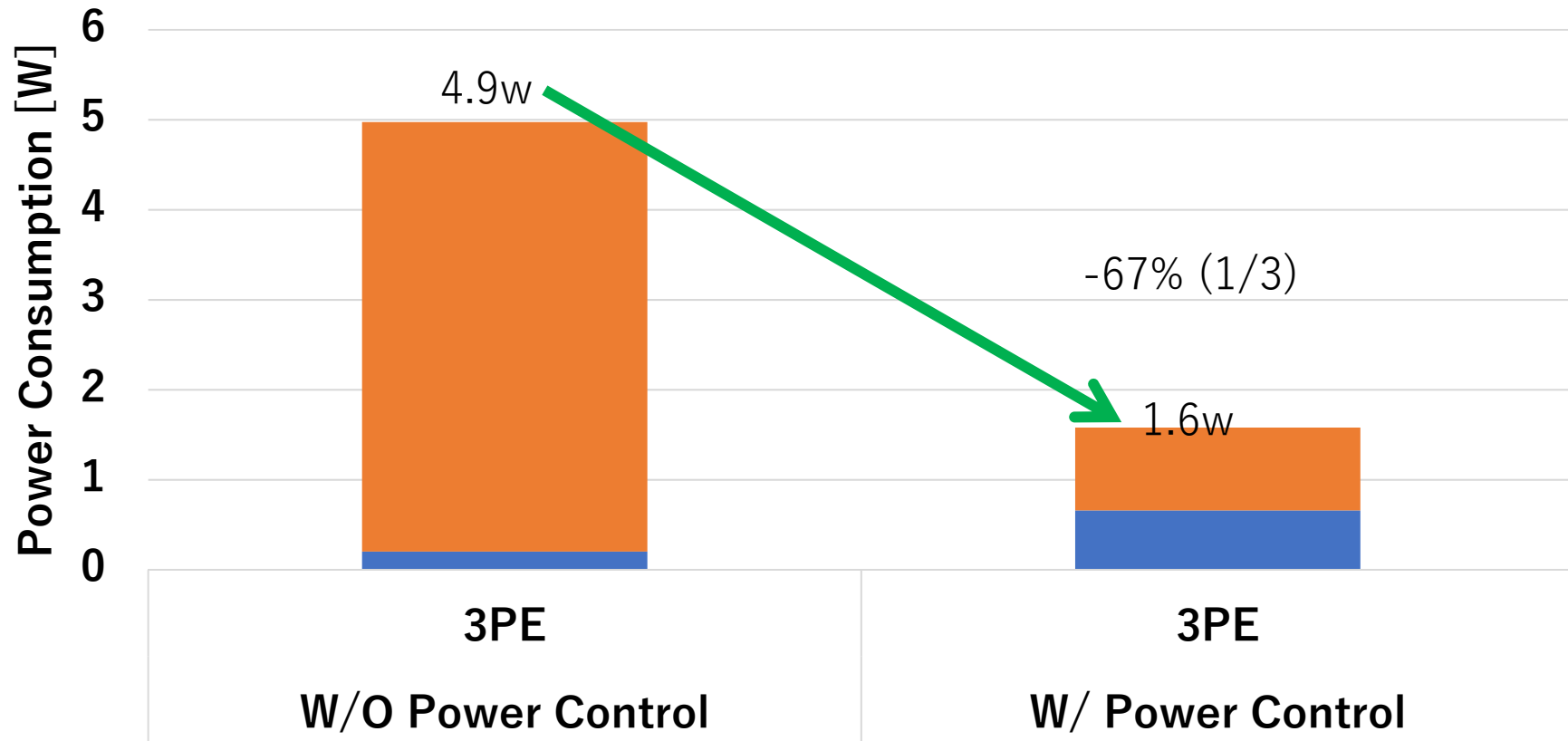


Power for 3cores was reduced to **1/3~1/4** against without software power control

Power for 3cores was reduced to **2/5~1/3** against ordinary 1core execution



# Automatic Power Reduction of OpenCV Face Detection on big.LITTLE ARM Processor



- **ODROID-XU3**

■ Cortex-A7 ■ Cortex-A15

- **Samsung Exynos 5422 Processor**

- 4x Cortex-A15 2.0GHz, 4x Cortex-A7 1.4GHz big.LITTLE Architecture
- 2GB LPDDR3 RAM
- Frequency can be changed by each cluster unit

# OSCAR API Ver. 2.0 for Homogeneous (LCPC2009) /Heterogeneous (LCPC2010) Multicores and Manycores

API for Parallel Processing on Various Multicores, Power Management, Hardware and Software Cache Control, and Local Memory Management

Manual Download: <http://www.kasahara.cs.waseda.ac.jp/api/regist.php?lang=en&ver=2.1>

## List of Directives (22 directives)

- ▶ Parallel Execution API
  - ▶ **parallel sections (\*)**
  - ▶ **flush (\*)**
  - ▶ **critical (\*)**
  - ▶ execution
- ▶ Memoay Mapping API
  - ▶ **threadprivate (\*)**
  - ▶ distributedshared
  - ▶ onchipshared
- ▶ Synchronization API
  - ▶ groupbarrier
- ▶ Data Transfer API
  - ▶ dma\_transfer
  - ▶ dma\_contiguous\_parameter
  - ▶ dma\_stride\_parameter
  - ▶ dma\_flag\_check
  - ▶ dma\_flag\_send

(\* from OpenMP)

- ▶ Power Control API
  - ▶ fvcontrol
  - ▶ get\_fvstatus
- ▶ Timer API
  - ▶ get\_current\_time
- ▶ Accelerator
  - ▶ accelerator\_task\_entry
- ▶ Cache Control
  - ▶ cache\_writeback
  - ▶ cache\_selfinvalidate
  - ▶ complete\_memop
  - ▶ noncacheable
  - ▶ aligncache

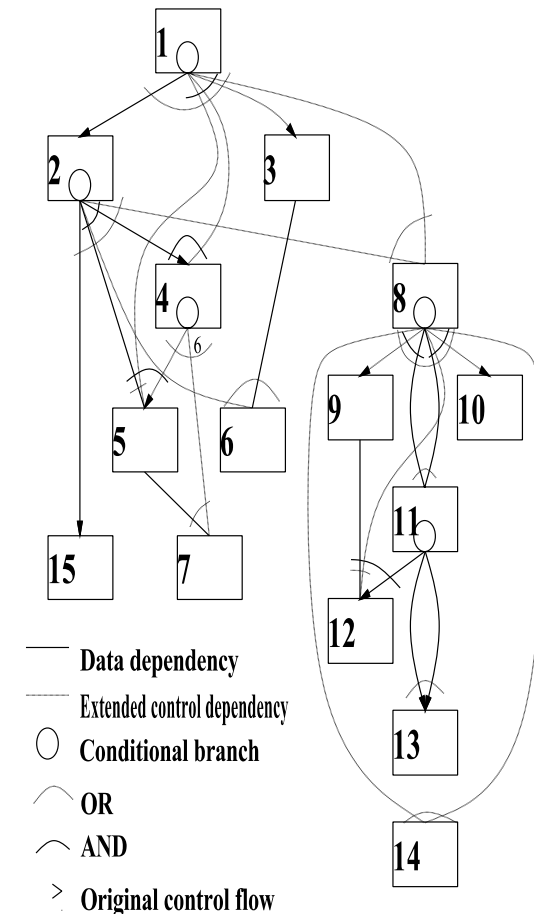
2 hint directives for OSCAR compiler

- accelerator\_task
- oscar\_comment

from V2.0

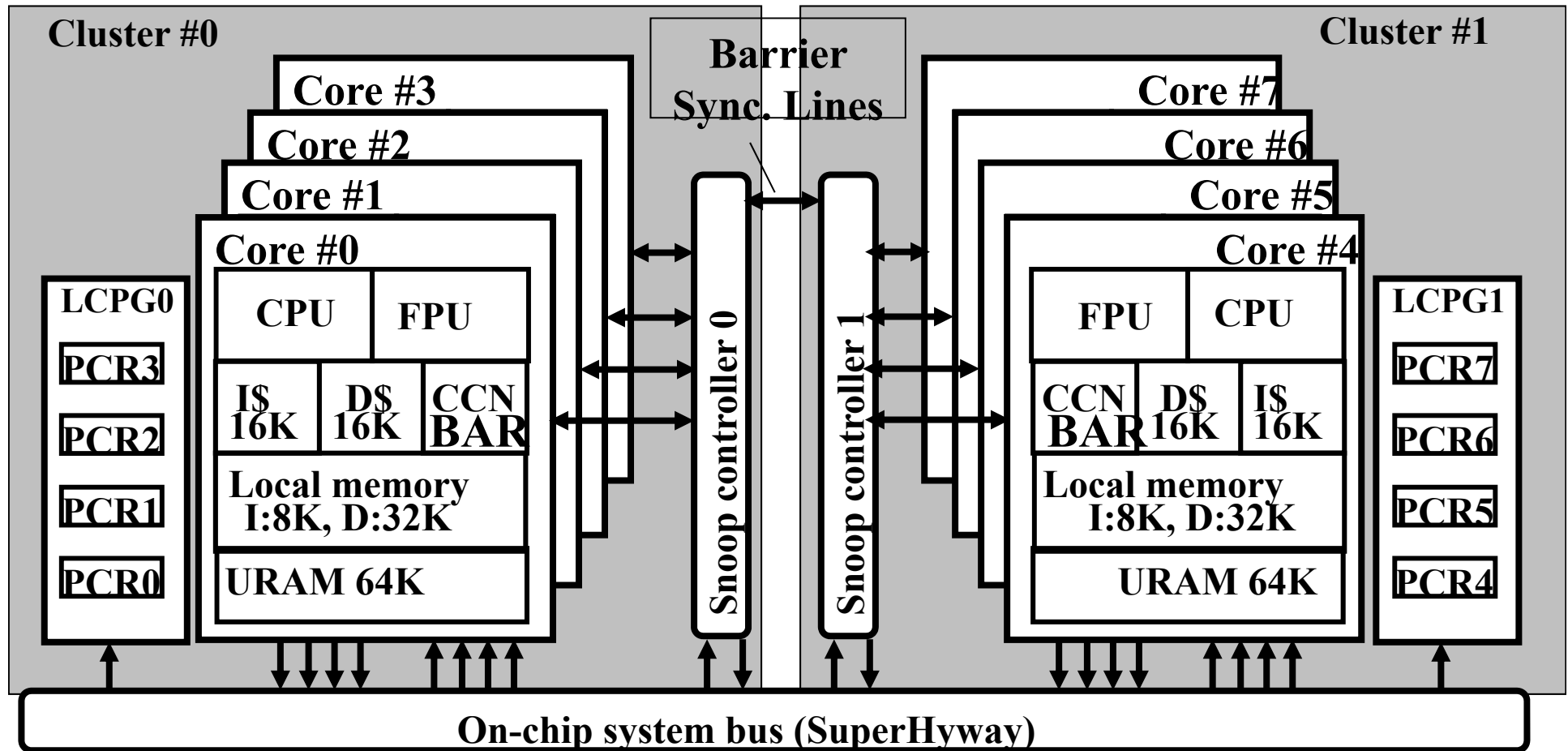
# Software Coherence Control Method on OSCAR Parallelizing Compiler

- Coarse grain task parallelization with **earliest condition analysis** (control and data dependency analysis to detect parallelism among coarse grain tasks).
- OSCAR compiler automatically controls coherence using following simple program restructuring methods:
  - To cope with stale data problems:
    - ◆ **Data synchronization by compilers**
  - To cope with false sharing problem:
    - ◆ **Data Alignment**
    - ◆ **Array Padding**
    - ◆ **Non-cacheable Buffer**



**MTG generated by  
earliest executable  
condition analysis**

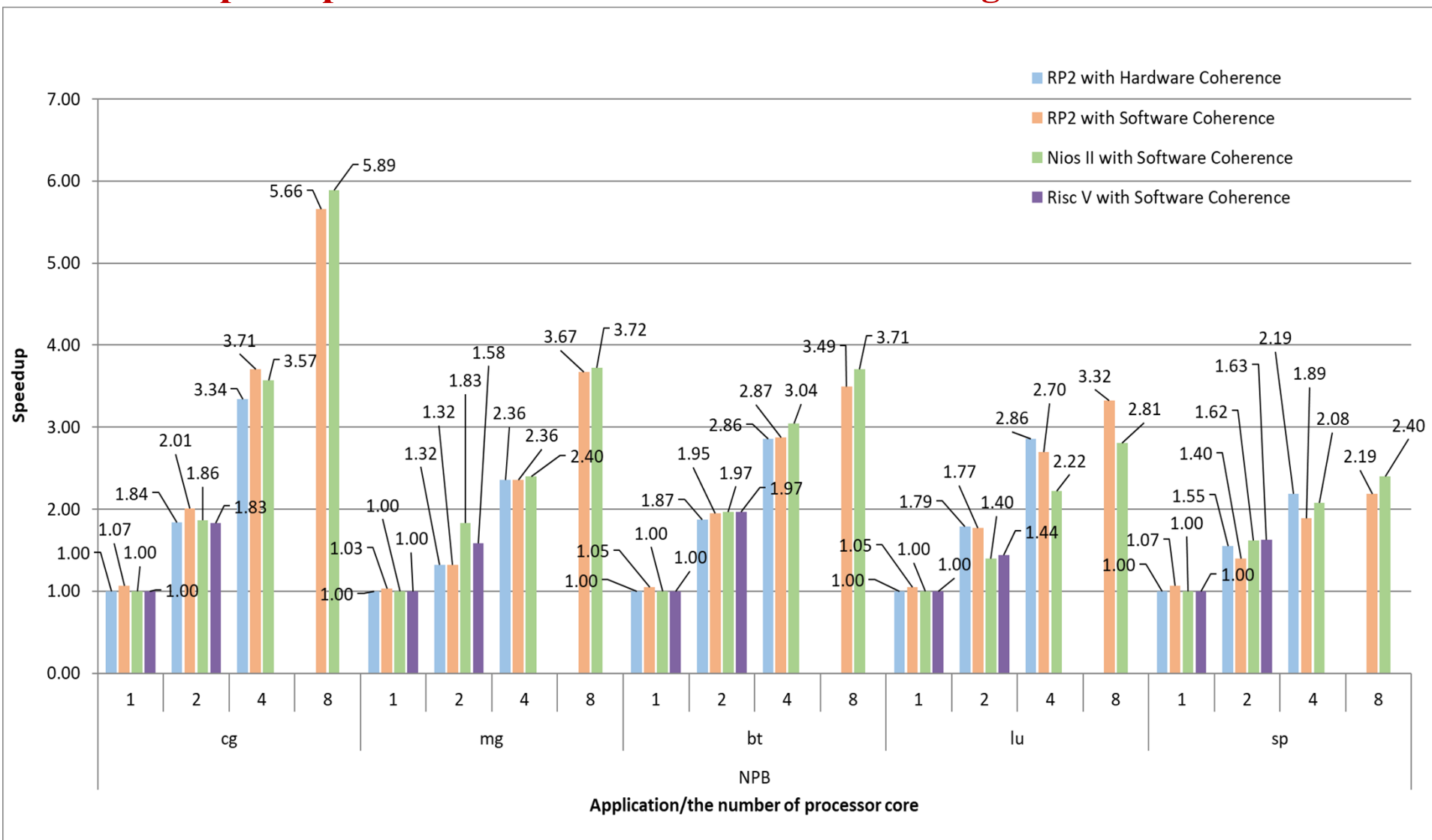
# 8 Core RP2 Chip Block Diagram



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

# OSCAR Software Cache Coherent Control for NIOS and RISC-V cores on FPGA

1.86x Speedups for NIOS and 1.83x for RISC-V using 2 cores for NPB CG

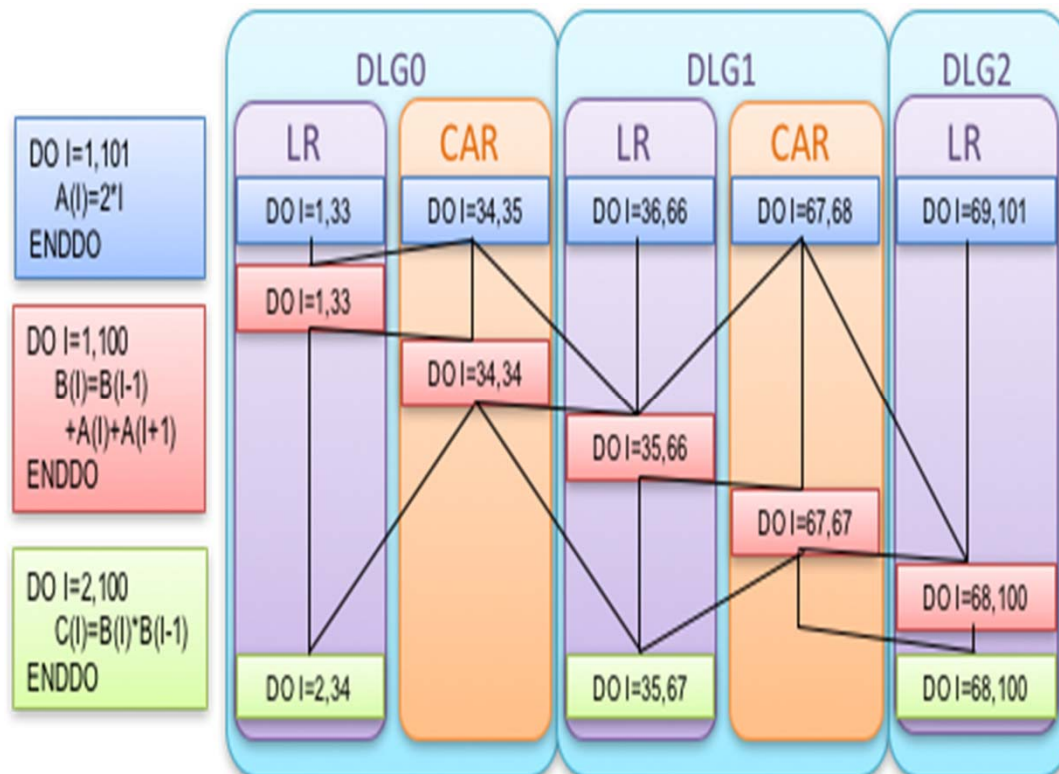


# Automatic Local Memory Management

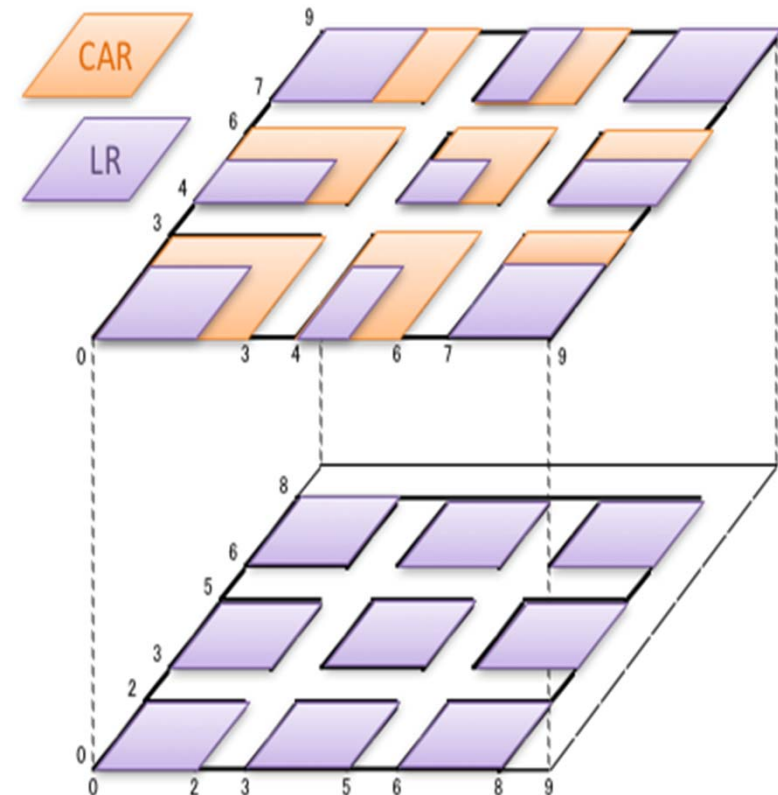
## Data Localization: Loop Aligned Decomposition

- Decomposed loop into LRs and CARs
  - LR (Localizable Region): Data can be passed through LDM
  - CAR (Commonly Accessed Region): Data transfers are required among processors

### Single dimension Decomposition

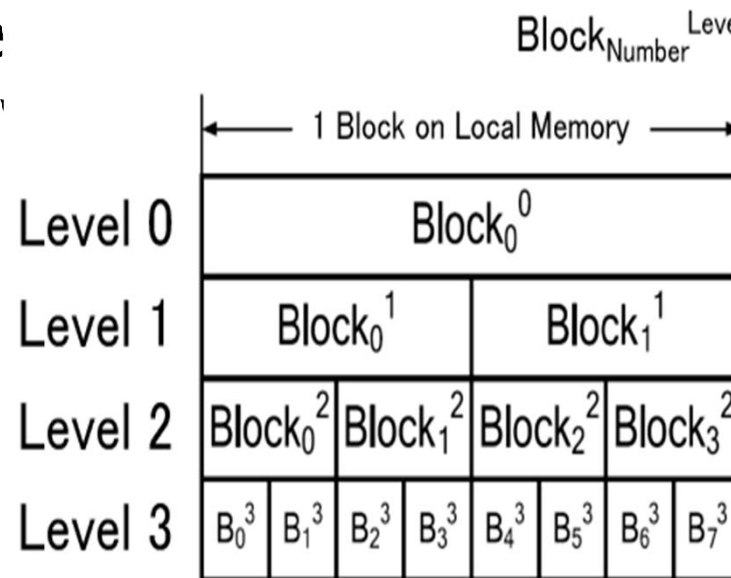


### Multi-dimension Decomposition



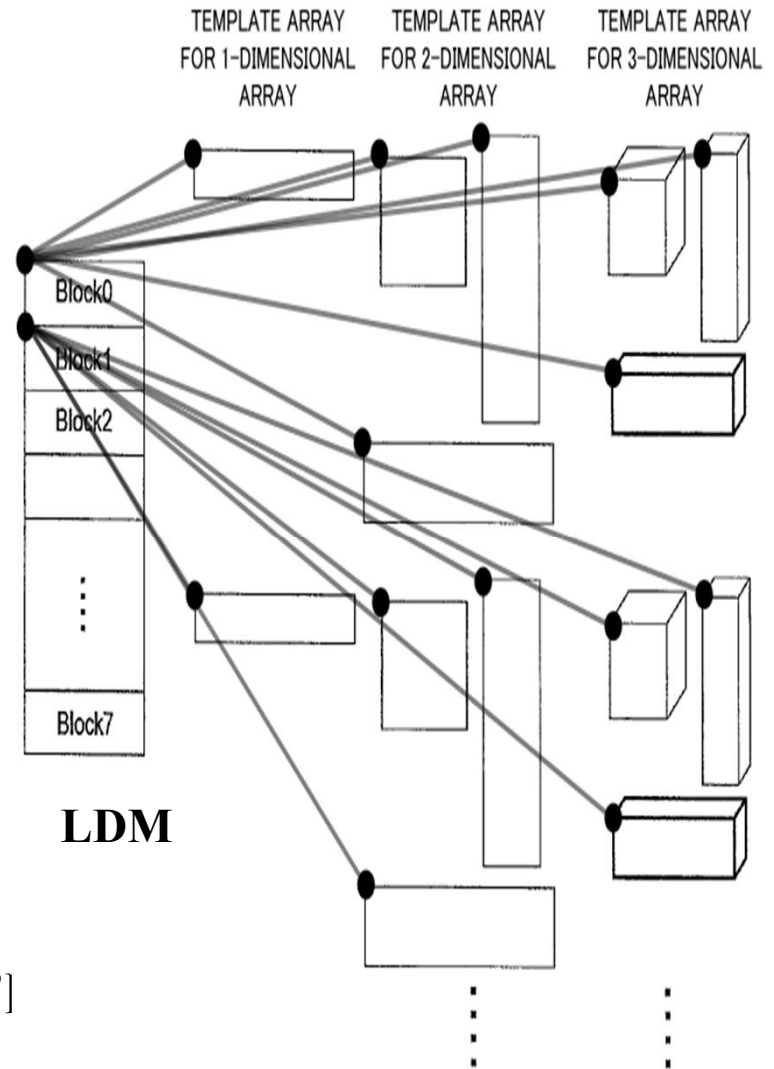
# Adjustable Blocks

- Handling a suitable block size for each application
  - different from a fixed block size in cache
  - each block can be divided into smaller blocks with integer and scalar



# Multi-dimensional Template Arrays for Improving Readability

- a mapping technique for arrays with varying dimensions
  - each block on LDM corresponds to multiple empty arrays with varying dimensions
  - these arrays have an additional dimension to store the corresponding block number
    - $TA[Block\#][\ ]$  for single dimension
    - $TA[Block\#][\ ][\ ]$  for double dimension
    - $TA[Block\#][\ ][\ ][\ ]$  for triple dimension
    - ...
- LDM are represented as a one dimensional array
  - without Template Arrays, multi-dimensional arrays have complex index calculations
    - $A[i][j][k] \rightarrow TA[offset + i' * L + j' * M + k']$
  - Template Arrays provide readability
    - $A[i][j][k] \rightarrow TA[Block\#][i'][j'][k']$





# Block Replacement Policy

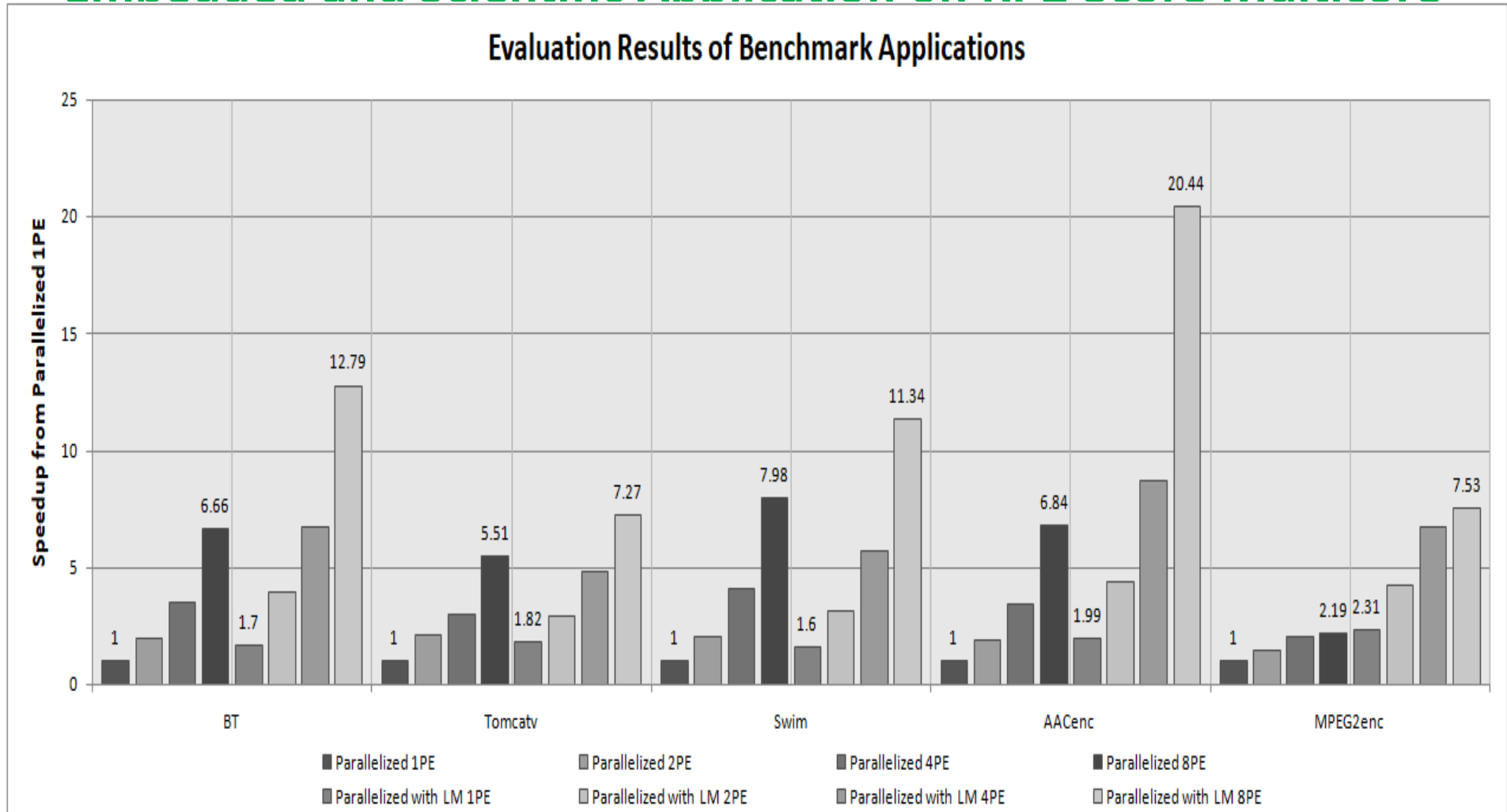
## □ Compiler Control Memory block Replacement

- using live, dead and reuse information of each variable from the scheduled result
- different from LRU in cache that does not use data dependence information

## □ Block Eviction Priority Policy

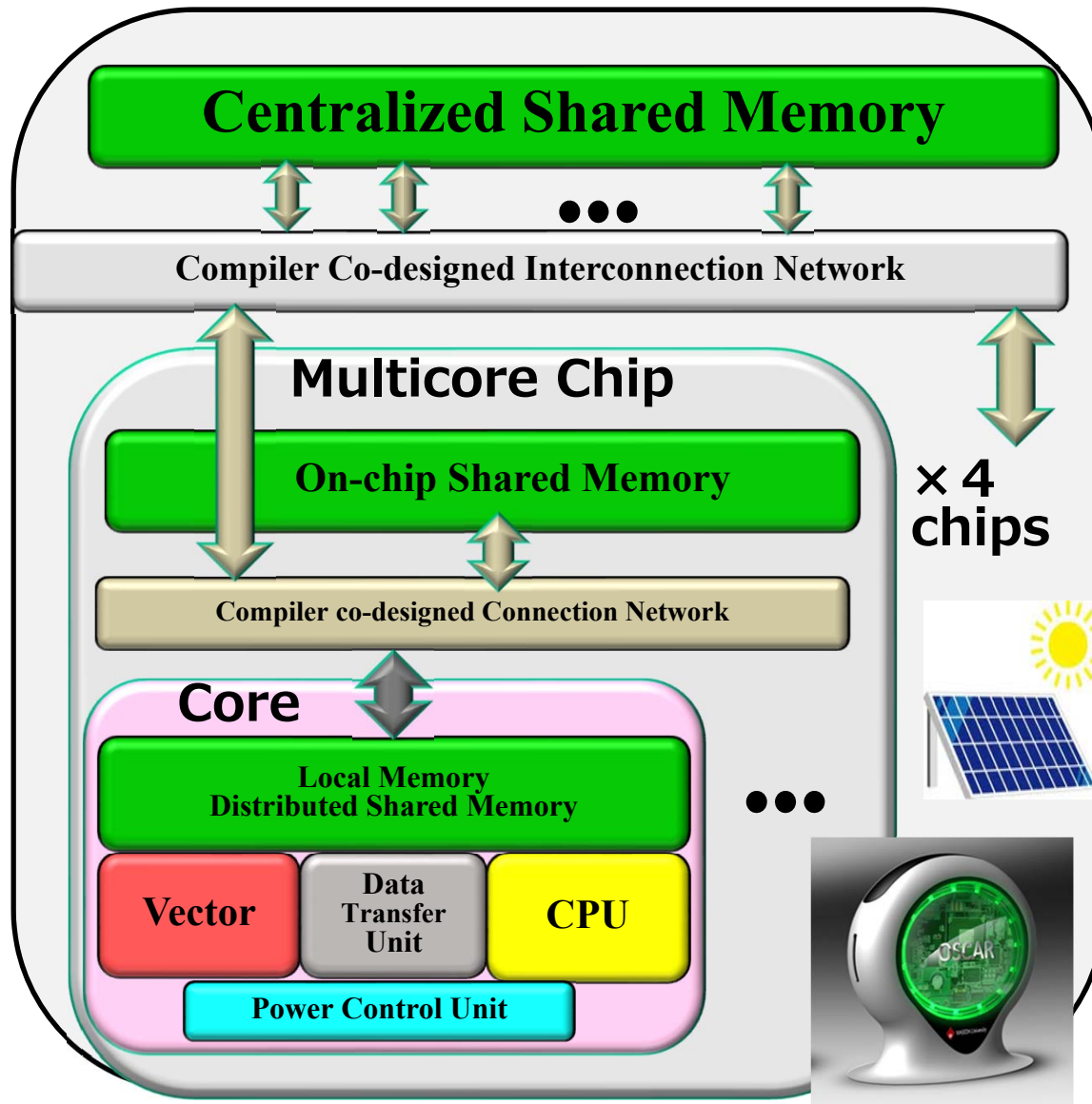
1. (Dead) Variables that will not be accessed later in the program
2. Variables that are accessed only by other processor cores
3. Variables that will be later accessed by the current processor core
4. Variables that will immediately be accessed by the current processor core

# Speedups by OSCAR Automatic Local Memory Management compared to Executions Utilizing Centralized Shared Memory on Embedded and Scientific Application on RP2 8core Multicore



**Maximum of 20.44 times speedup on 8 cores using local memory against sequential execution using off-chip shared memory**

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



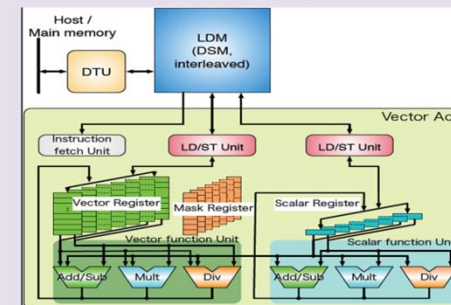
Target:

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

## Vector Accelerator

### Features

- Attachable for any CPUs (Intel, ARM, IBM)
- Data driven initiation by sync flags



### Function Units [tentative]

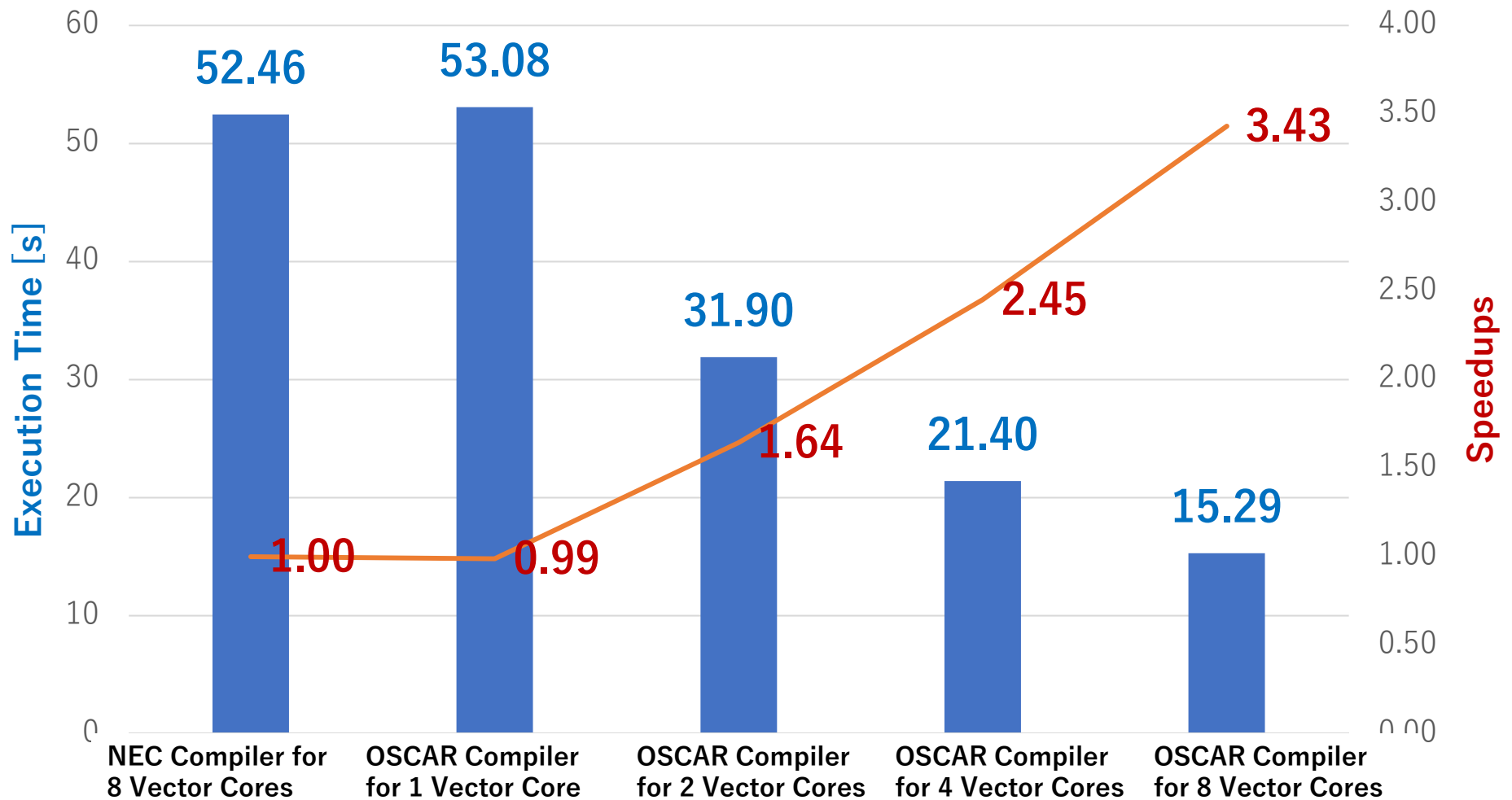
- Vector Function Unit
  - 8 double precision ops/clock
  - 64 characters ops/clock
  - Variable vector register length
  - Chaining LD/ST & Vector pipes
- Scalar Function Unit

### Registers[tentative]

- Vector Register 256Bytes/entry, 32entry
- Scalar Register 8Bytes/entry
- Floating Point Register 8Bytes/entry
- Mask Register 32Bytes/entry

# Speedups by OSCAR Automatic Vectorization on NEC SX-Aurora TSUBASA compared with NEC Compiler

3.43x Speedup using 8 Vector Cores



## 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 **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, 110x speedup for “Earthquake Wave Propagation Simulation” on 128 cores of IBM Power 7 against 1 core, 55x speedup for “Carbon Ion Radiotherapy Cancer Treatment” on 64cores IBM Power7, 1.95x for “Automobile Engine Control” on Renesas 2 cores using SH4A or V850, 55x for “JPEG-XR Encoding for Capsule Inner Cameras” on Tileria 64 cores Tile64 manycore, 1.83 x speedup for 2 core RISC V for CG on FPGA.**
- **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.****
- **Local memory management for automobiles and software coherent control for RISC V have been patented and already realized by OSCAR compiler.**
- **OSCAR compiler will be available from Web sites from OSCAR Technology.**