

OSCAR Automatic Parallelizing and Power Reducing Multicore Compiler for Realtime Embedded to High Performance Computing

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



IEEE Computer Society President 2018

Professor, Dept. of Computer Science & Engineering

Director, Advanced Multicore Processor Research Institute

Waseda University, Tokyo, Japan

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., 1997, Waseda Univ., Now Dept. of Computer Sci. & Eng.
1989-90 Research Scholar: U. of Illinois, Urbana-Champaign, Center for Supercomputing R&D
2004 Director, Advanced Multicore Research Institute, 2017 member: the Engineering Academy of Japan and the Science Council of Japan

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, IEEE Eta Kappa Nu











Reviewed Papers: 216, Invited Talks: 155, Published Unexamined Patent Application:59 (Japan, US, GB, China Granted Patents: 30), Articles in News Papers, Web News, Medias incl. TV etc.: 578

Committees in Societies and Government 255

IEEE Computer Society President 2018, BoG(2009-14), Multicore STC Chair (2012-), Japan Chair (2005-07), IPSJ Chair: HG for Mag. & 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

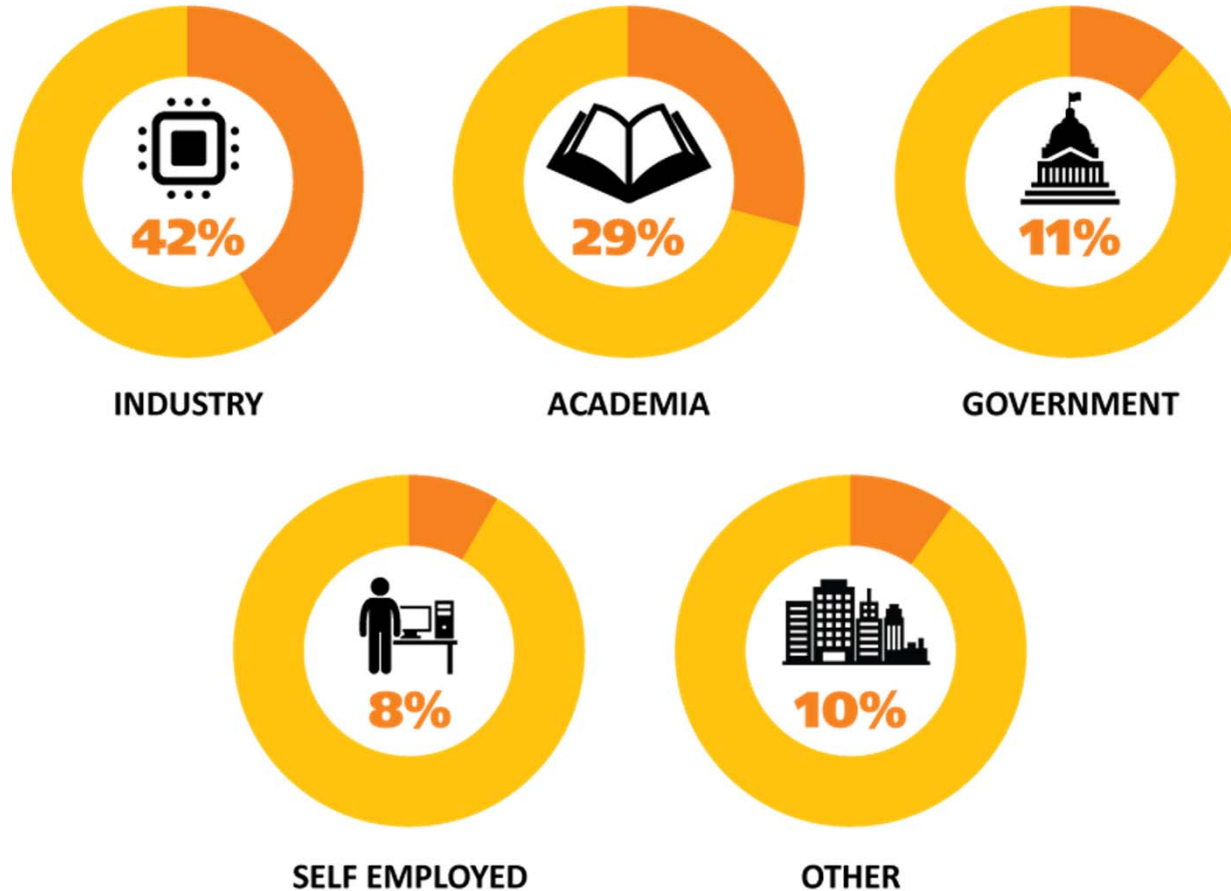
60,000+ members, volunteer-led organization,
200 technical conferences, 17 scholarly journals
and 13 magazines, awards program,
Digital Library with more than 550,000 articles and papers ,
400 local and regional chapters, 40 technical committees,

 Region 1 (Northeastern US)	 Region 6 (Western US)
 Region 2 (Eastern US)	 Region 7 (Canada)
 Region 3 (Southern US)	 Region 8 (Africa, Europe, Middle East)
 Region 4 (Central US)	 Region 9 (Latin America)
 Region 5 (Southwestern US)	 Region 10 (Asia and Pacific)

► IEEE-USA (Regions 1-6)



IEEE Computer Society Members



IEEE Computer Society BoG (Board of Governors) Feb.1, 2018



14-Mar-18

<https://www.computer.org/web/cshistory/officers-2018>

Past IEEE Computer Society Presidents

Chairs of the IRE Professional Group

on Electronic Computers

- 1951-53 Morton M. Astrahan
- 1953-54 John H. Howard
- 1954-55 Harry Larson
- 1955-56 Jean H. Felker
- 1956-57 Jerre D. Noe
- 1957-58 Werner Buchholz
- 1958-59 Willis H. Ware
- 1959-60 Richard O. Endres
- 1960-62 Arnold A. Cohen
- 1962-64 Walter L. Anderson

Chairs of the AIEE Committee on Large-Scale Computing Devices

- 1946-49 Charles Concordia
- 1949-51 John Grist Brainerd
- 1951-53 Walter H. MacWilliams
- 1953-55 Frank J. Maginniss
- 1955-57 Edwin L. Harder
- 1957-59 Morris Rubinoff
- 1959-61 Ruben A. Imm
- 1961-63 Claude A. Kagan
- 1963-64 Gerhard L. Hollander

Chairs & Presidents of the IEEE

Computer Society

- 1964-65 Keith Uncapher
- 1965-66 Richard I. Tanaka
- 1966-67 Samuel Levine
- 1968-69 Charles L. Hobbs
- 1970-71 Edward J. McCluskey
- 1972-73 Albert S. Hoagland
- 1974-75 Stephen S. Yau
- 1976 Dick B. Simmons
- 1977-78 Merlin G. Smith
- 1979-80 Tse-Yun Feng
- 1981 Richard E. Merwin
- 1982-83 Oscar N. Garcia
- 1984-85 Martha Sloan
- 1986-87 Roy L. Russo
- 1988 Edward A. Parrish
- 1989 Kenneth A. Anderson
- 1990 Helen M. Wood
- 1991 Duncan H. Lawrie
- 1992 Bruce D. Shriver
- 1993 James H. Aylor
- 1994 Laurel V. Kaleda
- 1995 Ronald G. Hoelzeman
- 1996 Mario R. Barbacci
- 1997 Barry W. Johnson
- 1998 Doris L. Carver
- 1999 Leonard L. Tripp
- 2000 Guylaine M. Pollock
- 2001 Benjamin W. Wah
- 2002 Willis K. King
- 2003 Stephen Diamond
- 2004 Carl K. Chang
- 2005 Gerald L. Engel
- 2006 Deborah M. Cooper
- 2007 Michael R. Williams
- 2008 Rangachar Kasturi
- 2009 Susan K. (Kathy) Land,
- 2010 James D. Isaak
- 2011 Sorel Reisman
- 2012 John W. Walz
- 2013 David Alan Grier
- 2014 Dejan S. Milojevic
- 2015 Thomas M. Conte
- 2016 Roger U. Fujii
- 2017 Jean-Luc Gaudiot
- 2018 Hironori Kasahara

IPSJ/IEEE-CS Young Computer Researcher Award

For members of the IPSJ and the IEEE-CS

The First Award Ceremony:

COMPSAC2018, July 23-27, NII, Tokyo

<https://ieeecompsac.computer.org/2018/>



Bjarne Stroustrup
2018 Computer
Society Computer
Pioneer Award
Columbia University

6

14-Mar-18



Masaru Kitsuregawa
Director General of
NII, Past President of
IPSJ



Margaret Martonosi
2018 Computer
Society Technical
Achievement
Princeton University



Dejan Milojevic
CS President 2014
HP Labs
CS 2022 Report



Choose Your Content Bundle

Preferred Plus



[More Info](#)

Training & Development



[More Info](#)

Research



[More Info](#)

Basic



[More Info](#)

Student



[More Info](#)

IEEE Member: Add Computer Society to your existing IEEE membership. IEEE Membership is an additional charge.

Affiliate Member: Join Computer Society only as a Computer Society affiliate. IEEE benefits are not included.

Student Member: Students must join IEEE when joining Computer Society. IEEE Student members can add Computer Society.

IEEE Member: \$30
Affiliate Member: \$78

[JOIN NOW](#)

IEEE Member: \$28
Affiliate Member: \$71

[JOIN NOW](#)

IEEE Member: \$28
Affiliate Member: \$71

[JOIN NOW](#)

IEEE Member: \$20
Affiliate Member: \$63

[JOIN NOW](#)

IEEE Student Member: \$4
New Student Member: \$20

[JOIN NOW](#)

Computer magazine (12 digital issues)*



ComputingEdge (12 issues)



Members-only discounts on conferences and events



Members-only webinars



Unlimited access to Computing Now, computer.org, and the new mobile-ready myCS



Local chapter membership



Skillsoft Skillchoice™ Complete with 67,000+ books, videos, courses, practice exams and mentorship resources



Books24x7 on-demand access to 15,000 technical and business resources



Two complimentary Computer Society [magazines](#) subscriptions



myComputer mobile app

30 tokens

30 tokens

30 tokens

Computer Society Digital Library

12 FREE downloads

Member pricing

12 FREE downloads

Member pricing

Included

Training webinars

3 FREE webinars

3 Free webinars

Member pricing

Member pricing

Member pricing

Priority registration to Computer Society events



Right to vote and hold office



One-time 20% Computer Society online store discount



* Print publications are available for an additional fee. See IEEE catalog for details.

Multicore Video Series

- **Automatic Parallelization: David Padua**
- **Autoparallelization for GPUs: Wen-mei Hwu**
- **Dependences and Dependence Analysis: Utpal Banerjee**
- **Dynamic Parallelization: Rudolf Eigenmann**
- **Instruction Level Parallelization: Alexandru Nicolau**
- **Multigrain Parallelization and Power Reduction:
Hironori Kasahara**
- **The Polyhedral Model: Paul Feautrier**
- **Vector Computation: David Kuck (Computer Pioneer)**
- **Vectorization: P. Sadayappan**
- **Vectorization/Parallelization in the IBM Compiler:
Yaoqing Gao**
- **Vectorization/Parallelization in the Intel Compiler:
Peng Tu**
- **Roundtable Discussion by all presenters**

Self-Paced Learning:

Approximate time = 12 hours

- PDH: 12.0
- CEU: 1.2

Full Series Price:

- IEEE CS Member: \$195
- Nonmember: \$1,000

Individual Videos:

- IEEE CS Member: \$30
- Nonmember: \$125

See individual videos below.

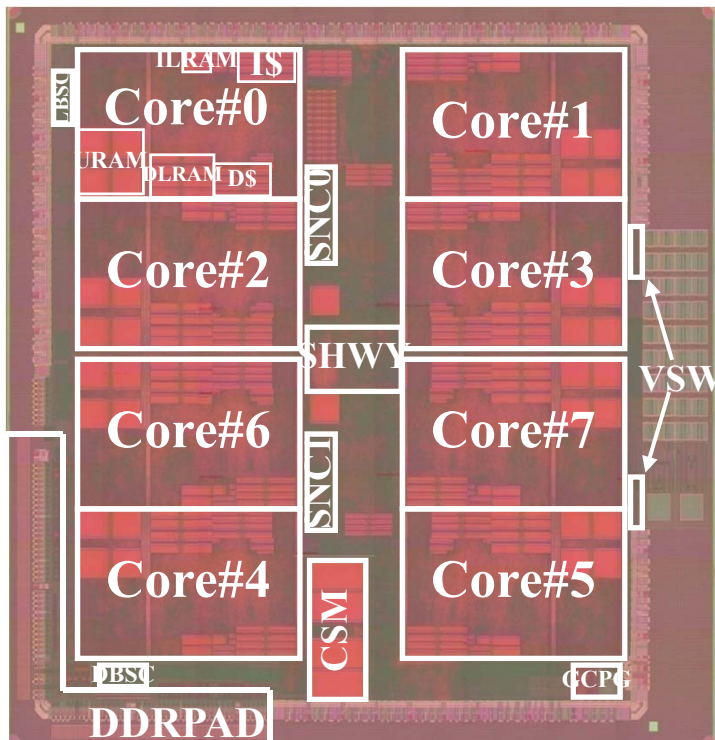
For questions, please contact
certification@computer.org.

Toward 2018

1. Refining content and services to further improve the satisfaction of CS members;
2. Considering an incentive for volunteers to further accelerate CS activities and promptly provide technical benefits for people around the globe;
To express appreciation to volunteers:
CS Point (Mileage) System: Annual & Life Time Honor, Premier Seating, Premier Registration, Distinguished Reviewer, etc
3. Offering more attractive services for practitioners in industry;
4. Providing the world's best educational content and historical treasures for future generations, which only the CS can create with our pioneering researchers (for example, the Multicore Compiler Video Series found at www.computer.org/web/education/multicore-video-series);
5. Thinking about sustainable membership fees while considering the diversity of economic situations within the 10 regions;
6. Cooperating with other IEEE societies and sister societies in a timely and efficient manner;
7. Intelligibly introducing the latest computer-related technologies to younger generations, including children, so that they can realize their technological dreams.

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”

Power \propto Frequency * Voltage²
(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>

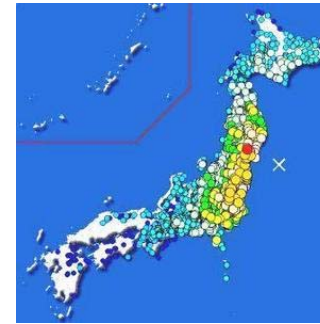
If 8cores are integrated on a chip,
Power is still 1/8 and
Performance becomes 2 times .

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

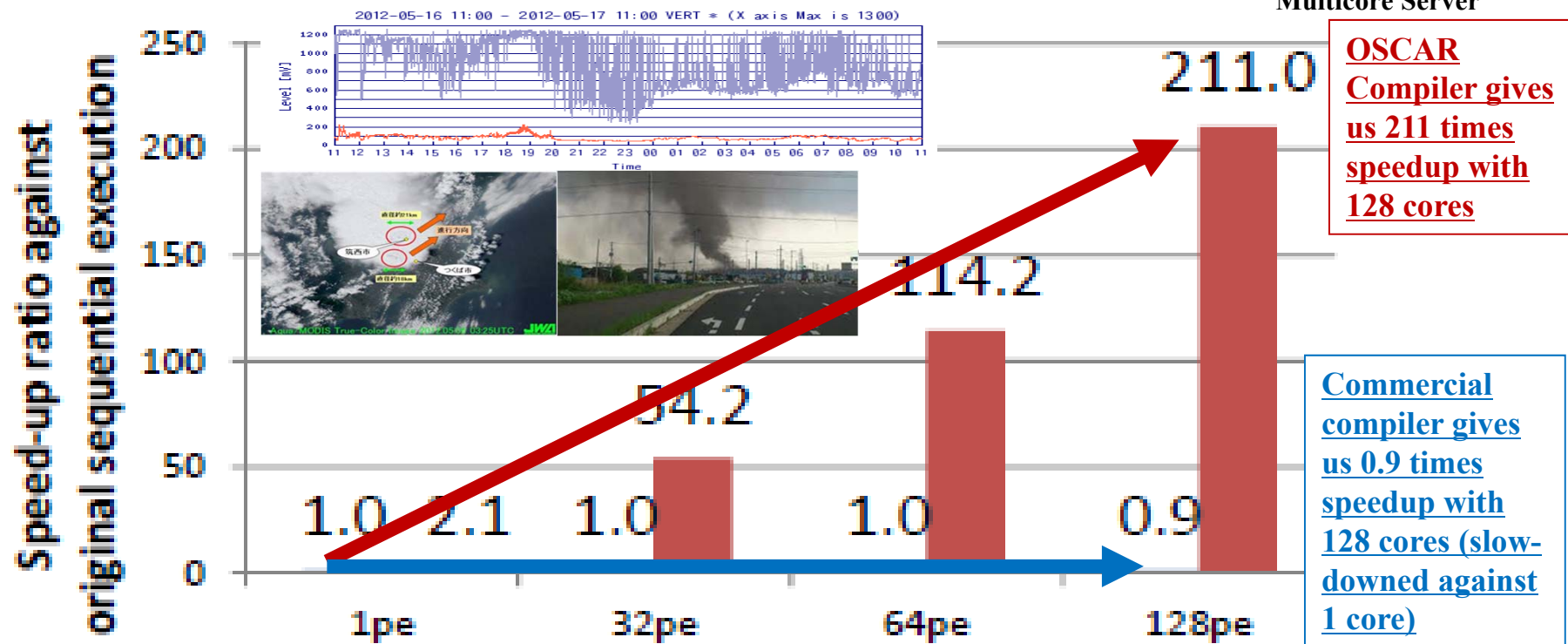
- Just more cores don't give us speedup
- Development cost and period of parallel software are getting a bottleneck of development of embedded systems, eg. IoT, Automobile

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

■ original (sun studio) ■ proposed method



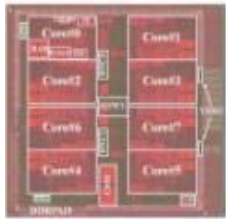
Fujitsu M9000 SPARC Multicore Server



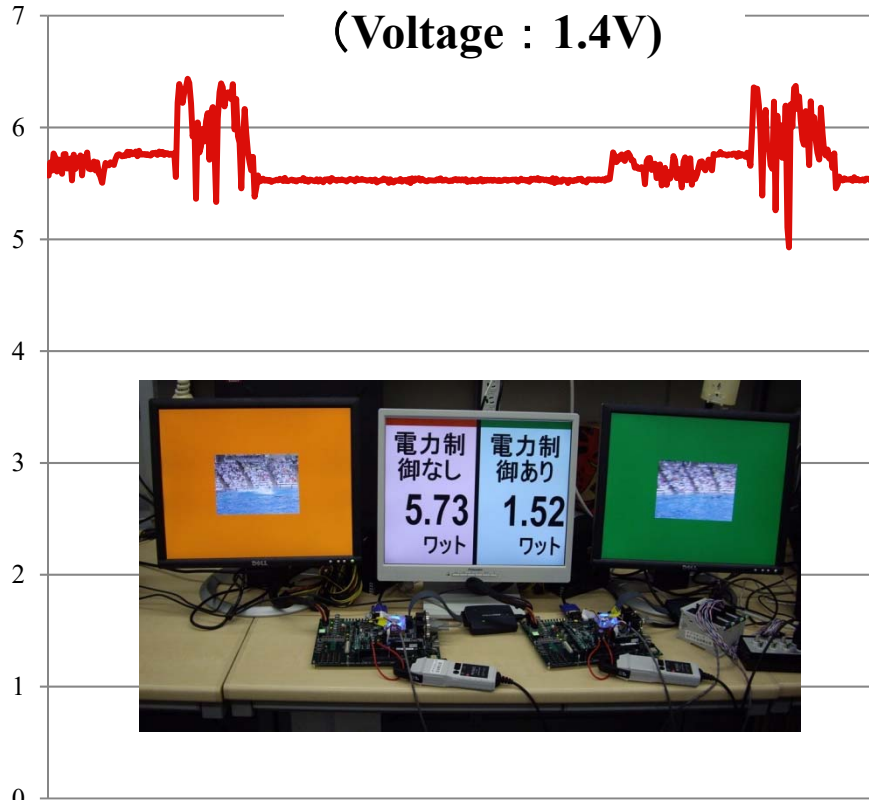
- Automatic parallelizing compiler available on the market gave us no speedup against execution time on 1 core on 64 cores
 - Execution time with 128 cores was slower than 1 core (0.9 times speedup)
- Advanced OSCAR parallelizing compiler gave us 211 times speedup with 128cores against execution time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

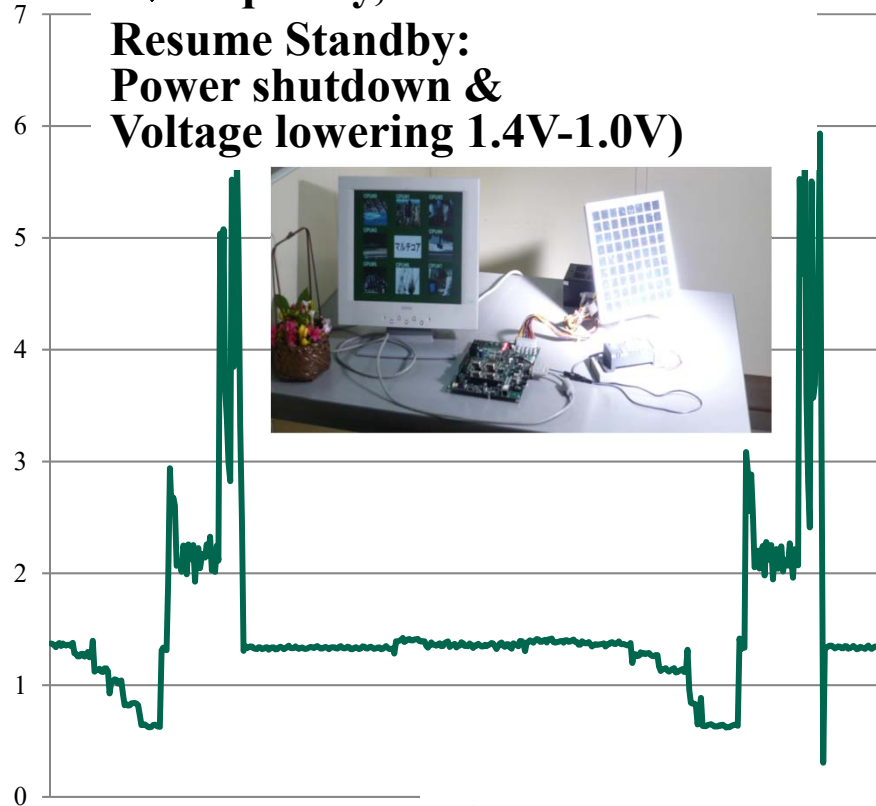
MPEG2 Decoding with 8 CPU cores



Without Power Control
(Voltage : 1.4V)



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]

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

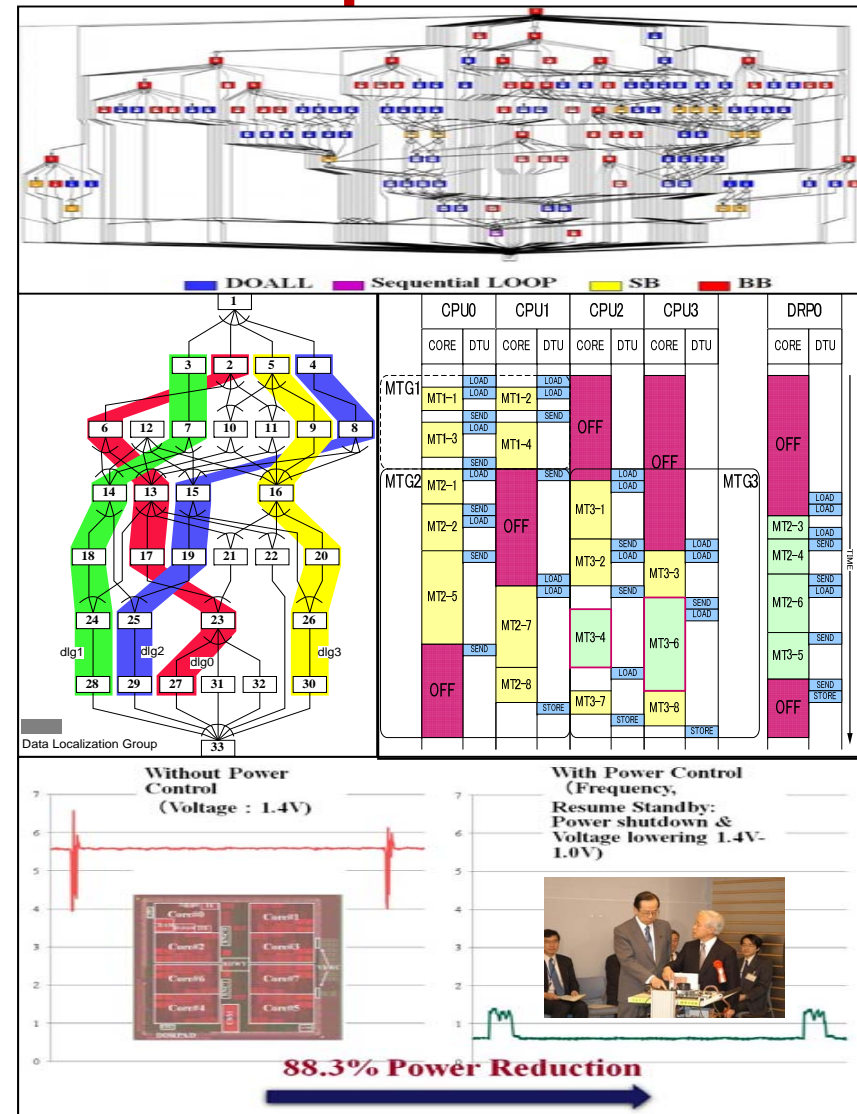
Automatic data management for distributed shared memory, cache and local memory (Local Memory 1995, 2016 on RP2, Cache2001,03)
Software Coherent Control (2017)

Data Transfer Overlapping (2016 partially)

Data transfer overlapping using Data Transfer Controllers (DMAs)

Power Reduction

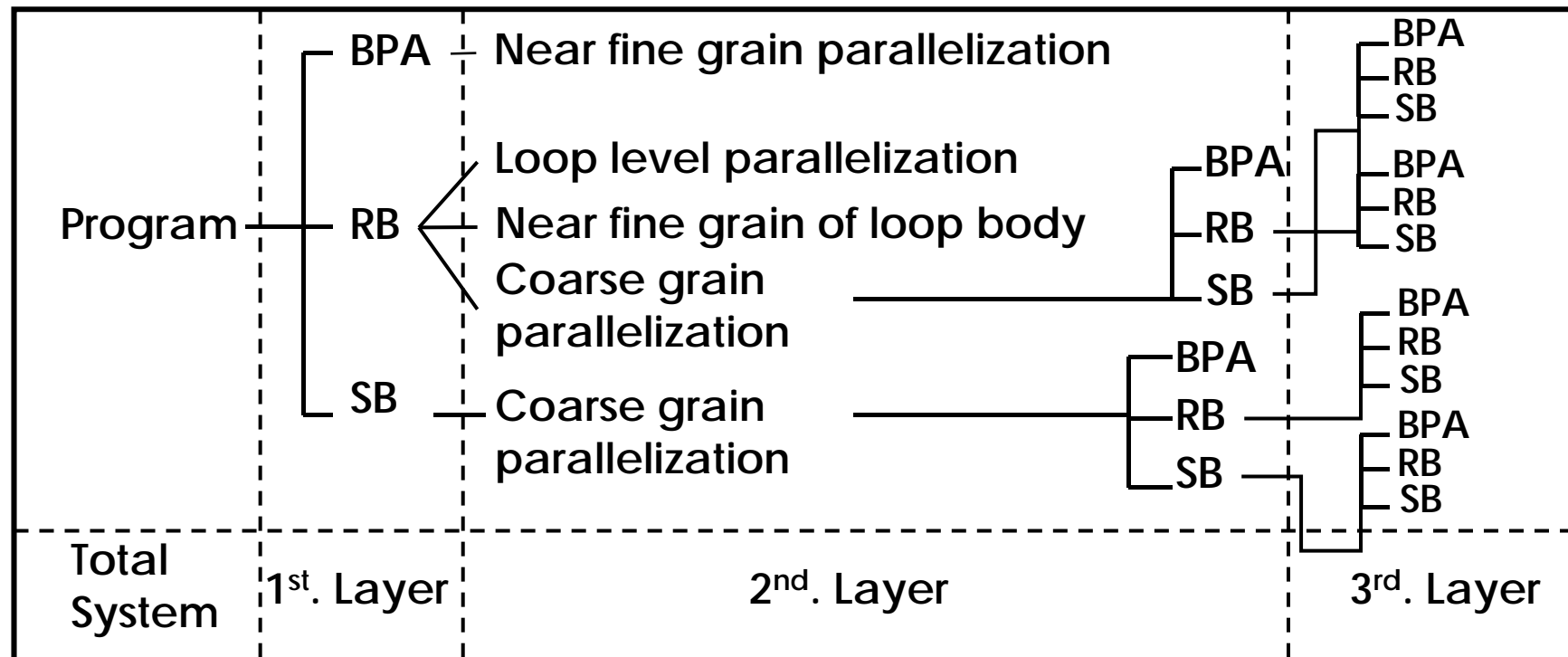
(2005 for Multicore, 2011 Multi-processes, 2013 on ARM)
Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.



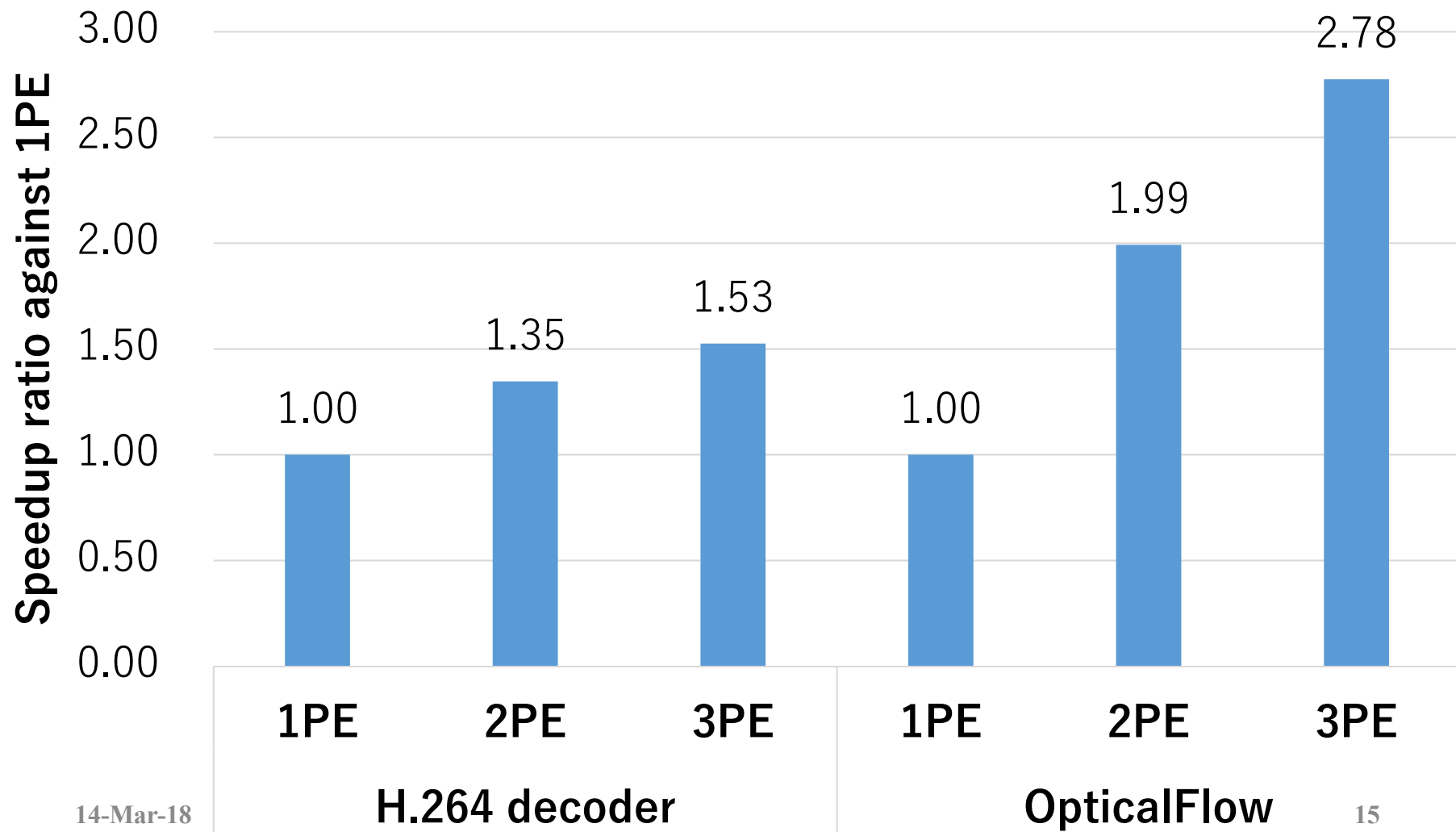
Generation of Coarse Grain Tasks

■ Macro-tasks (MTs)

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



Speedup ratio for H.264 and Optical Flow on ARM Cortex-A9 Android 3 cores by OSCAR Automatic Parallelization

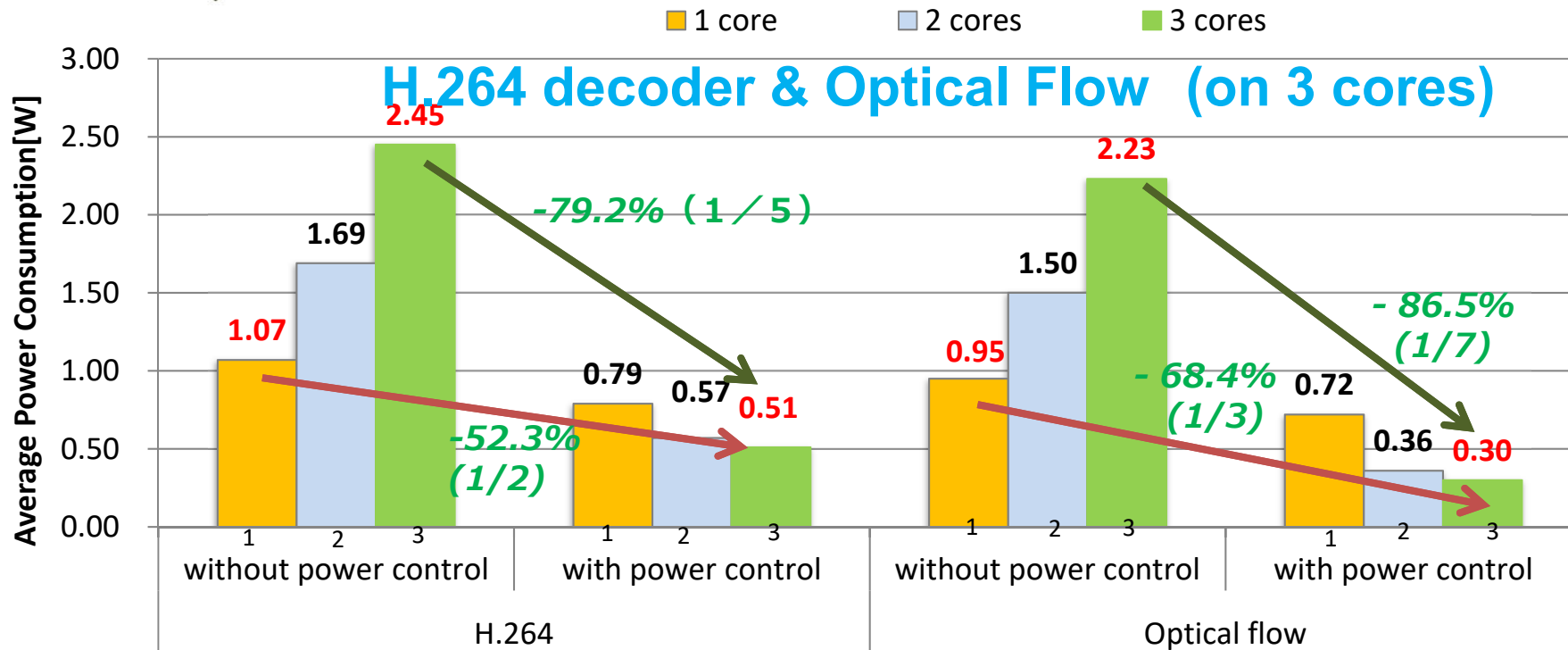
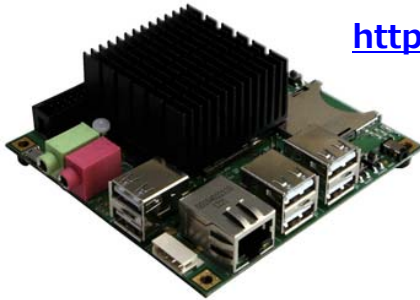


Automatic Power Reduction on ARM CortexA9 with Android

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

ODROID X2

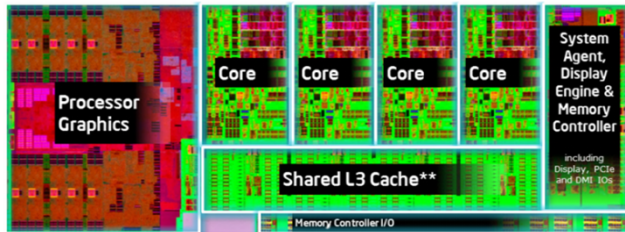
Samsung Exynos4412 Prime, ARM Cortex-A9 Quad core
1.7GHz~0.2GHz, used by Samsung's Galaxy S3



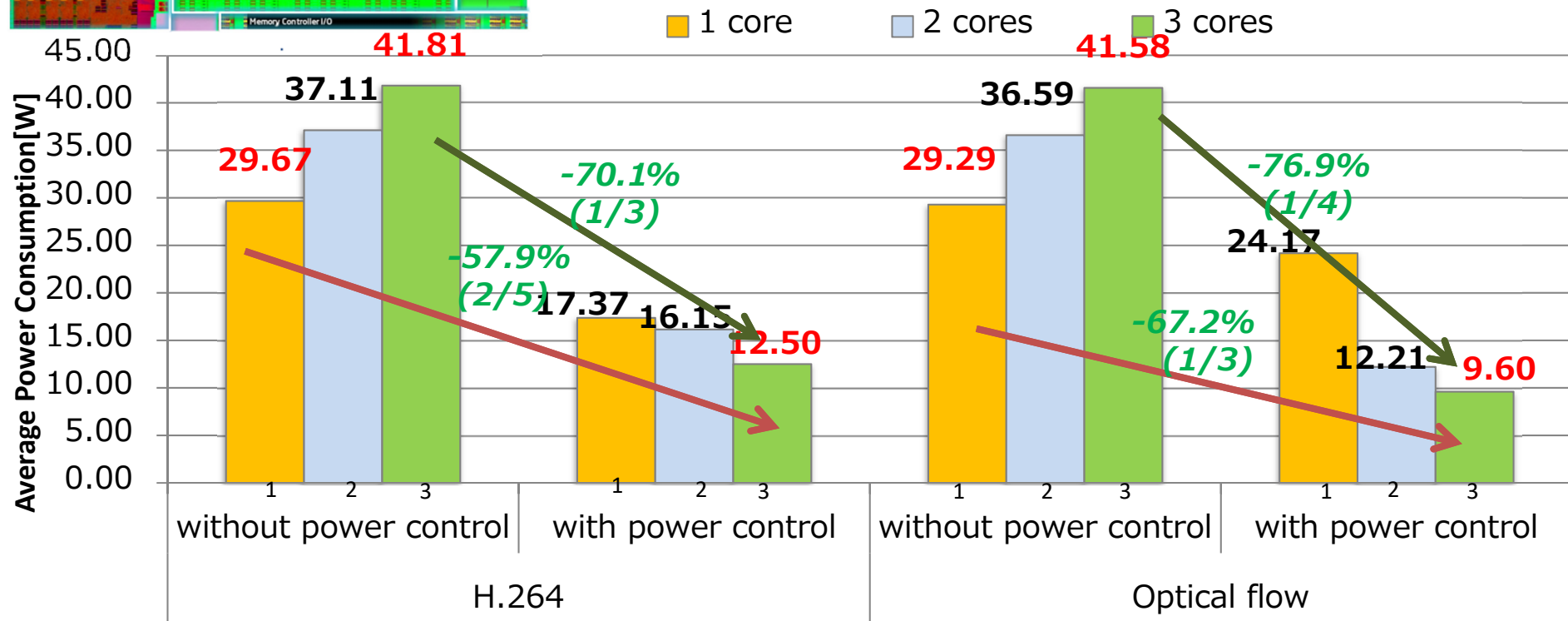
Power for 3cores was reduced to **1/5~1/7** against without software power control
Power for 3cores was reduced to **1/2~1/3** against ordinary 1core execution

Automatic Power Reuction on Intel Haswell

H.264 decoder & Optical Flow (3cores)



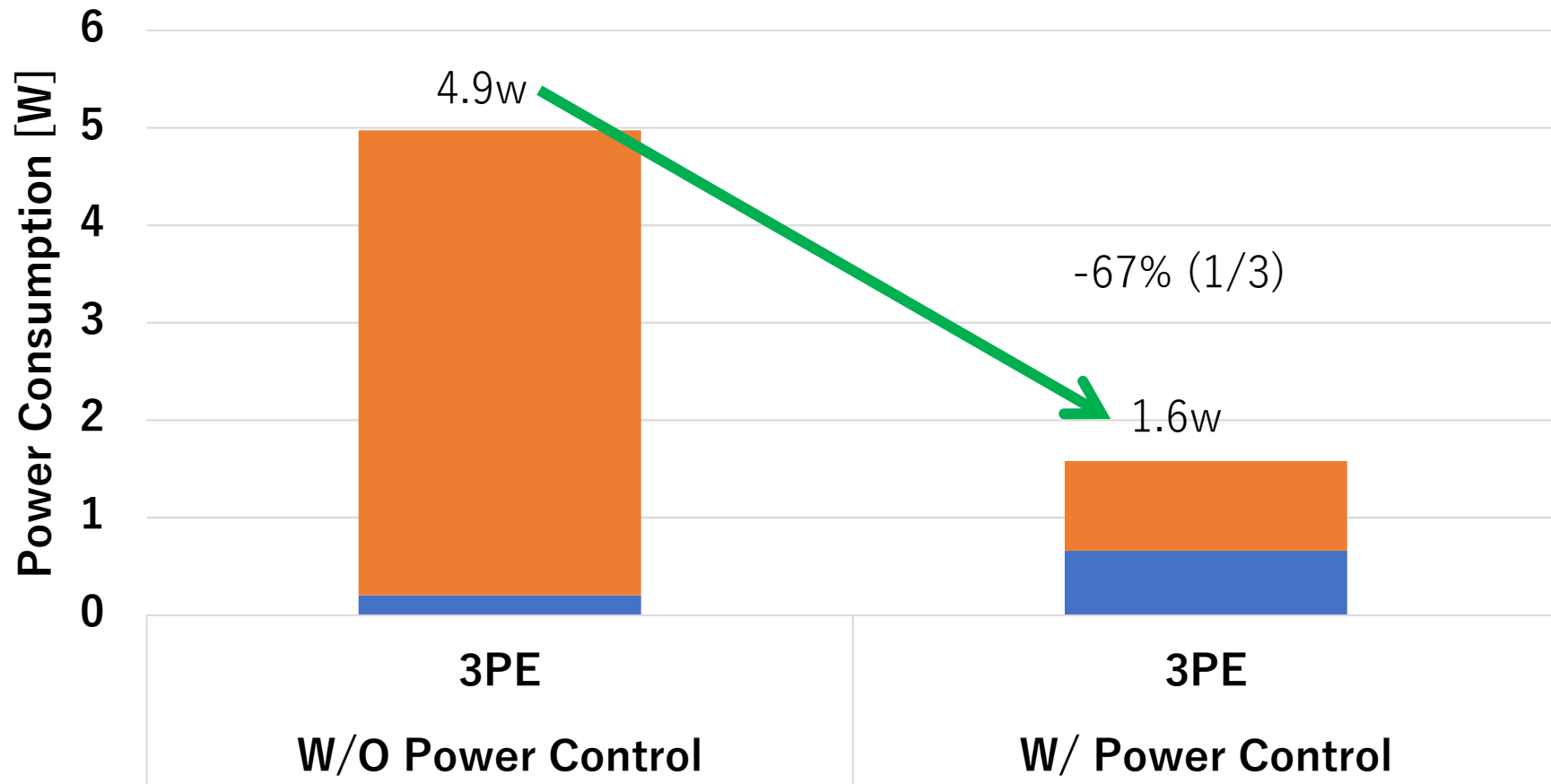
H81M-A, Intel Core i7 4770k
Quad core, 3.5GHz~0.8GHz



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**

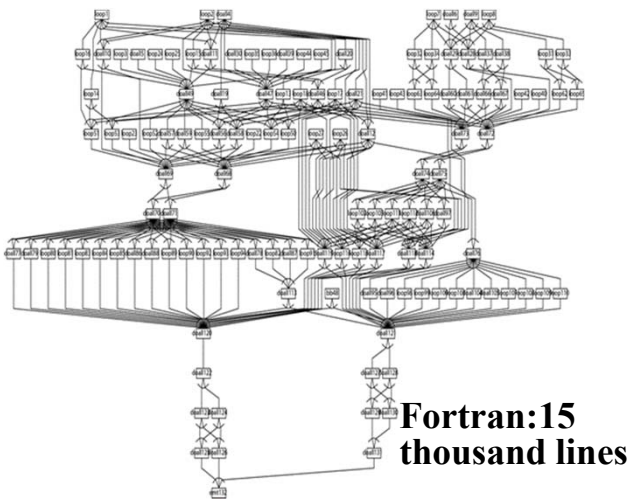
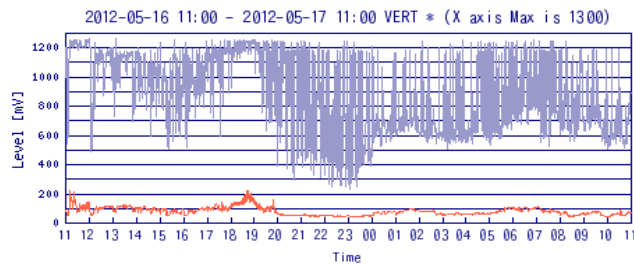
■ Cortex-A7 ■ Cortex-A15

- **Samsung Exynos 5422 Processor**

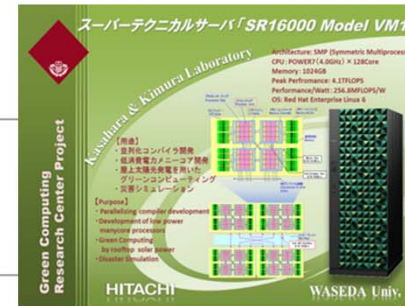
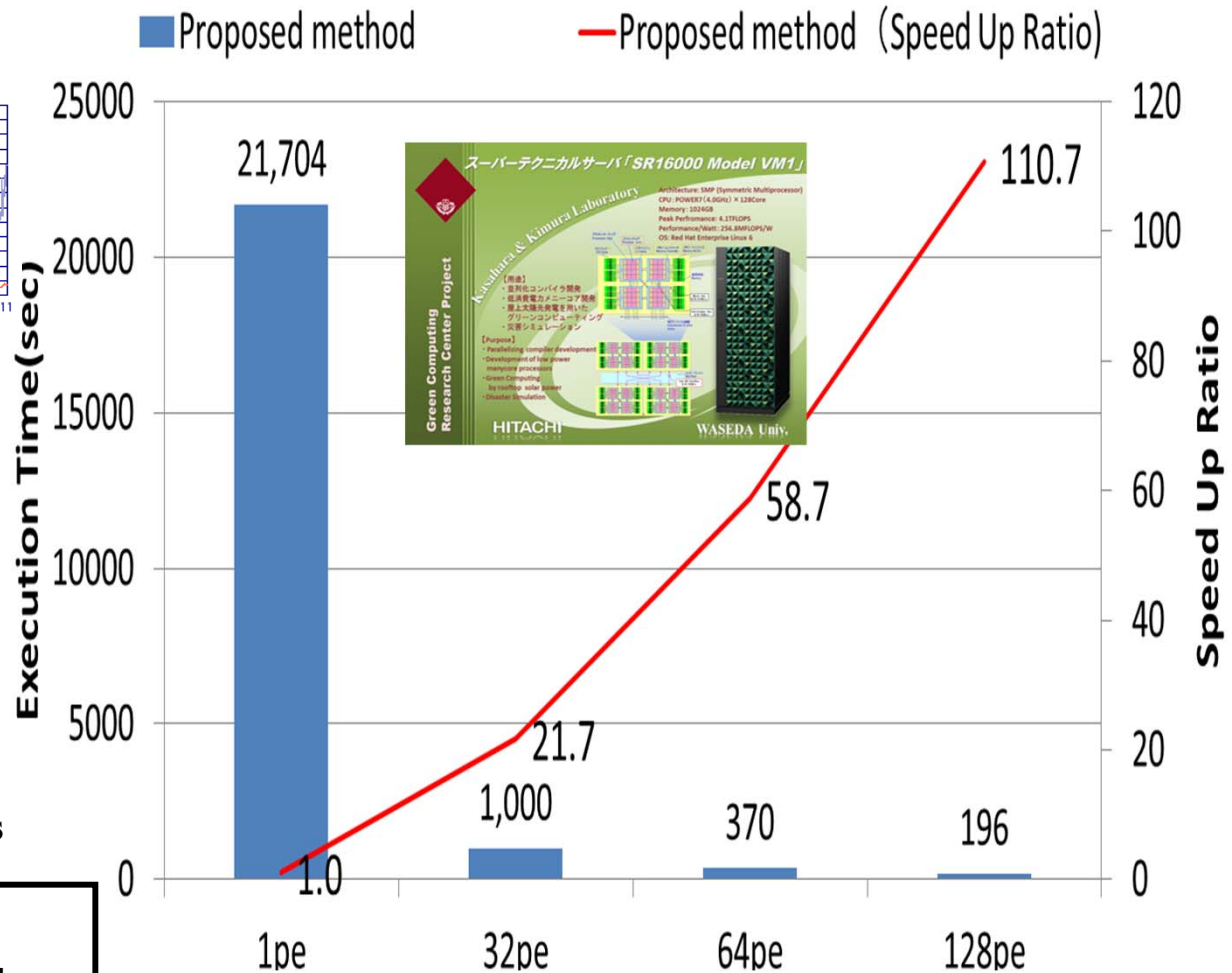
- 4x Cortex-A15 2.0GHz, 4x Cortex-A7 1.4GHz big.LITTLE Architecture
- 2GB LPDDR3 RAM
- Frequency can be changed by each cluster unit

110 Times Speedup against the Sequential Processing for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000

(Power7 Based 128 Core Linux SMP) [\(LCPC2015\)](#)



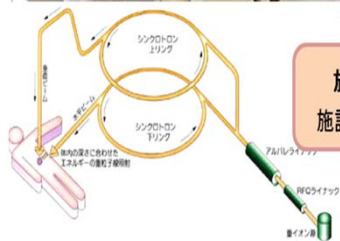
First touch for distributed shared memory and cache optimization over loops are important for scalable speedup



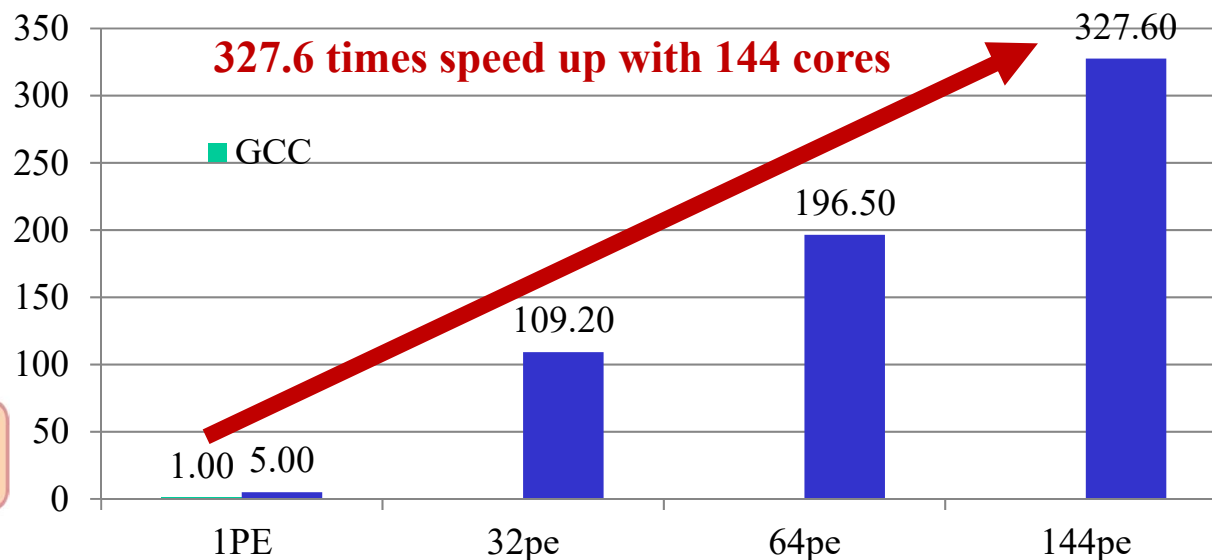
Performance on Multicore Server for Latest Cancer Treatment Using Heavy Particle (Proton, Carbon Ion)

327 times speedup on 144 cores

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

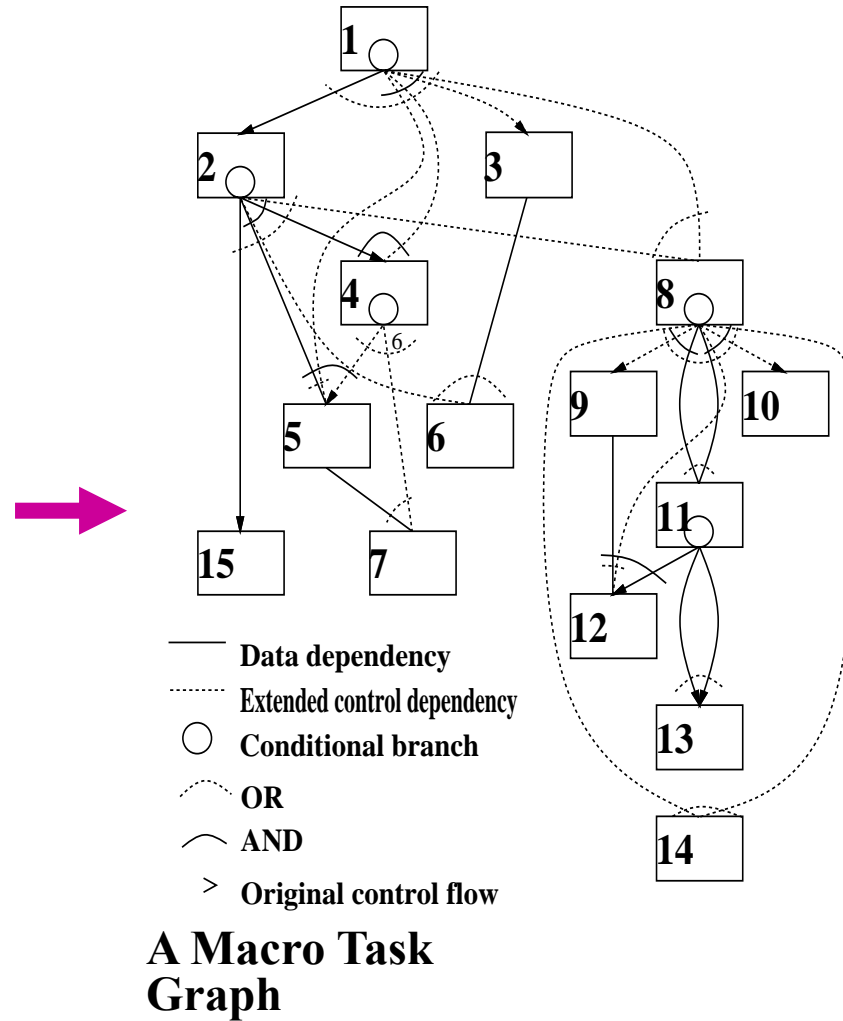
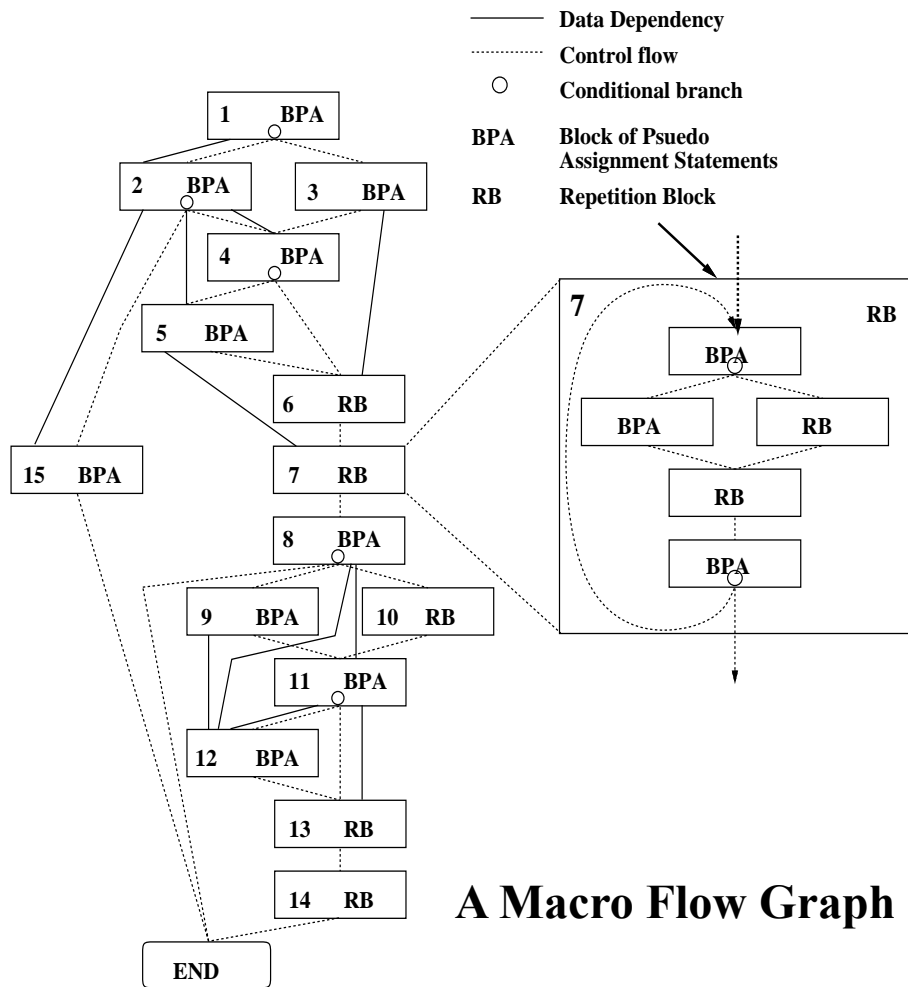


放射線医学研究所
施設の費用: 120億円

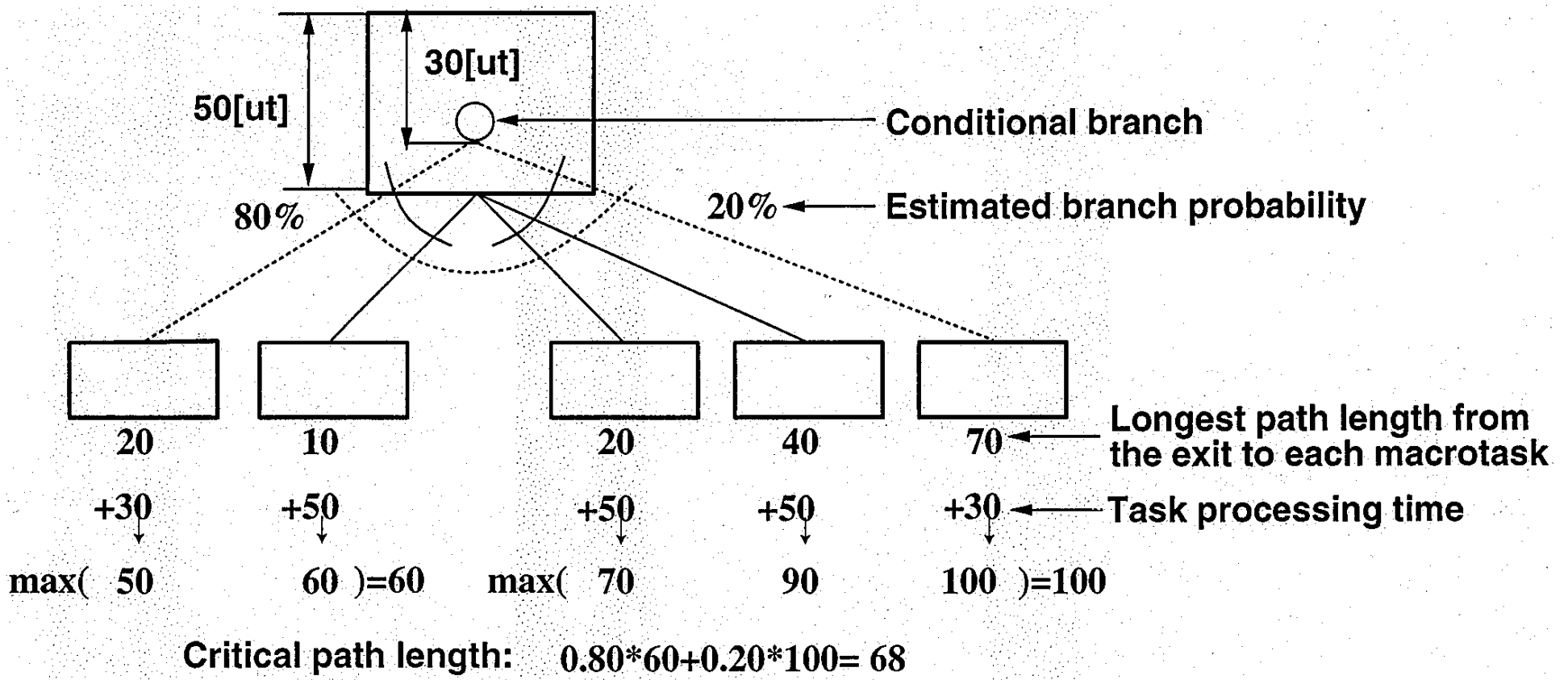


- Original sequential execution time 2948 sec (50 minutes) using GCC was reduced to 9 sec with 144 cores (327.6 times speedup)
- Reduction of treatment cost and reservation waiting period is expected

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



PRIORITY DETERMINATION IN DYNAMIC CP METHOD



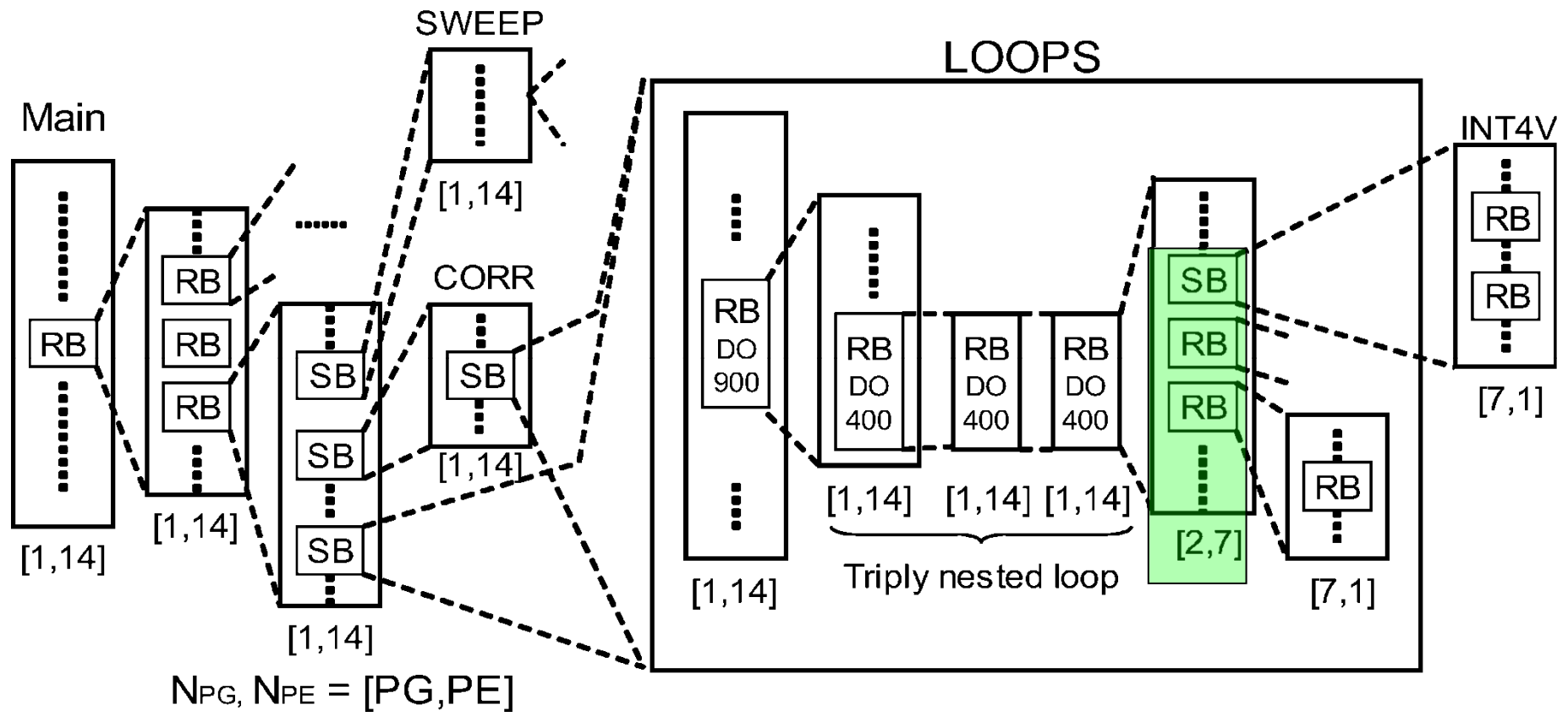
Earliest Executable Conditions

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

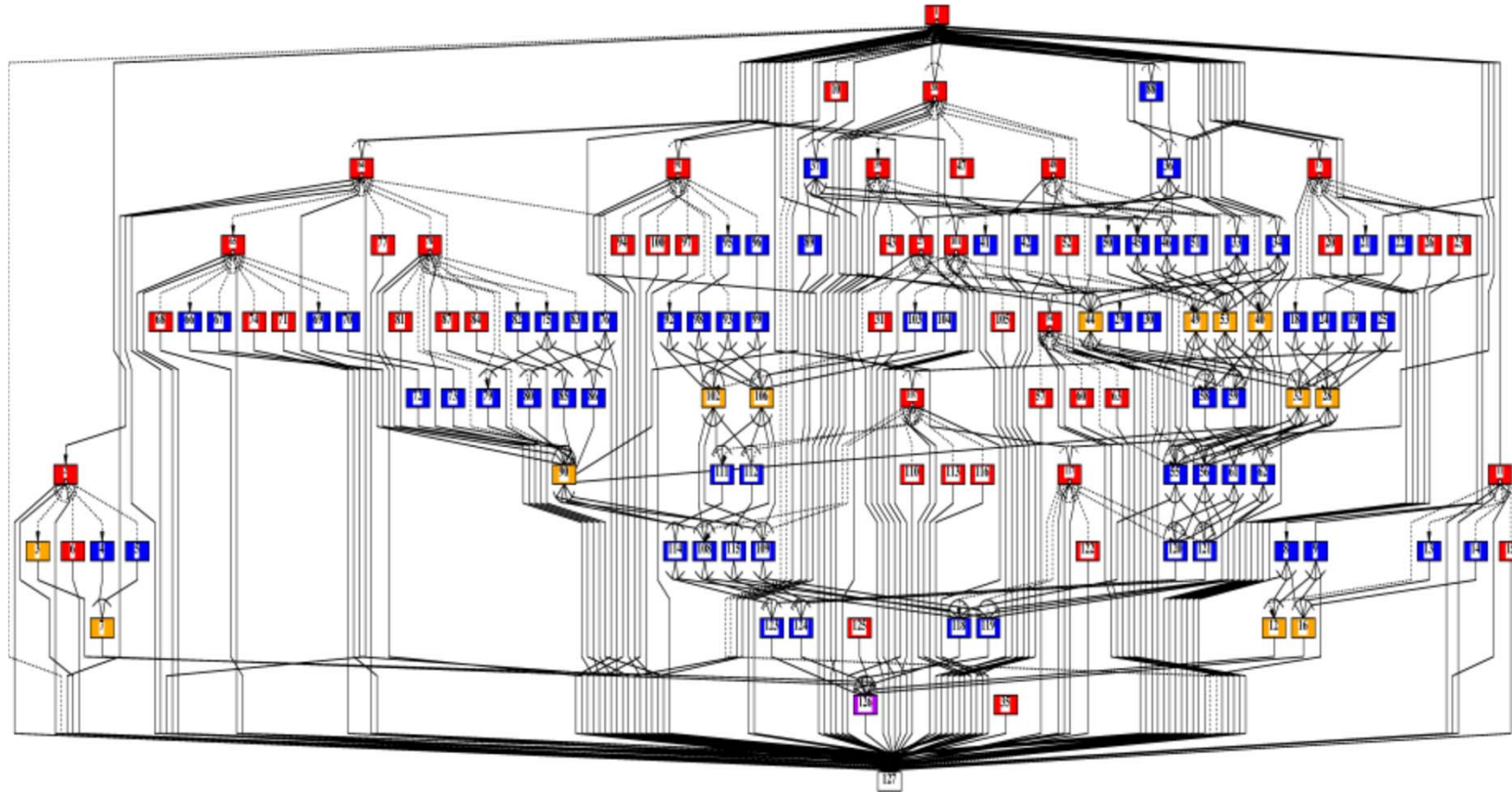
- Using 14 processors

Coarse grain parallelization within DO400



MTG of Su2cor-LOOPS-DO400

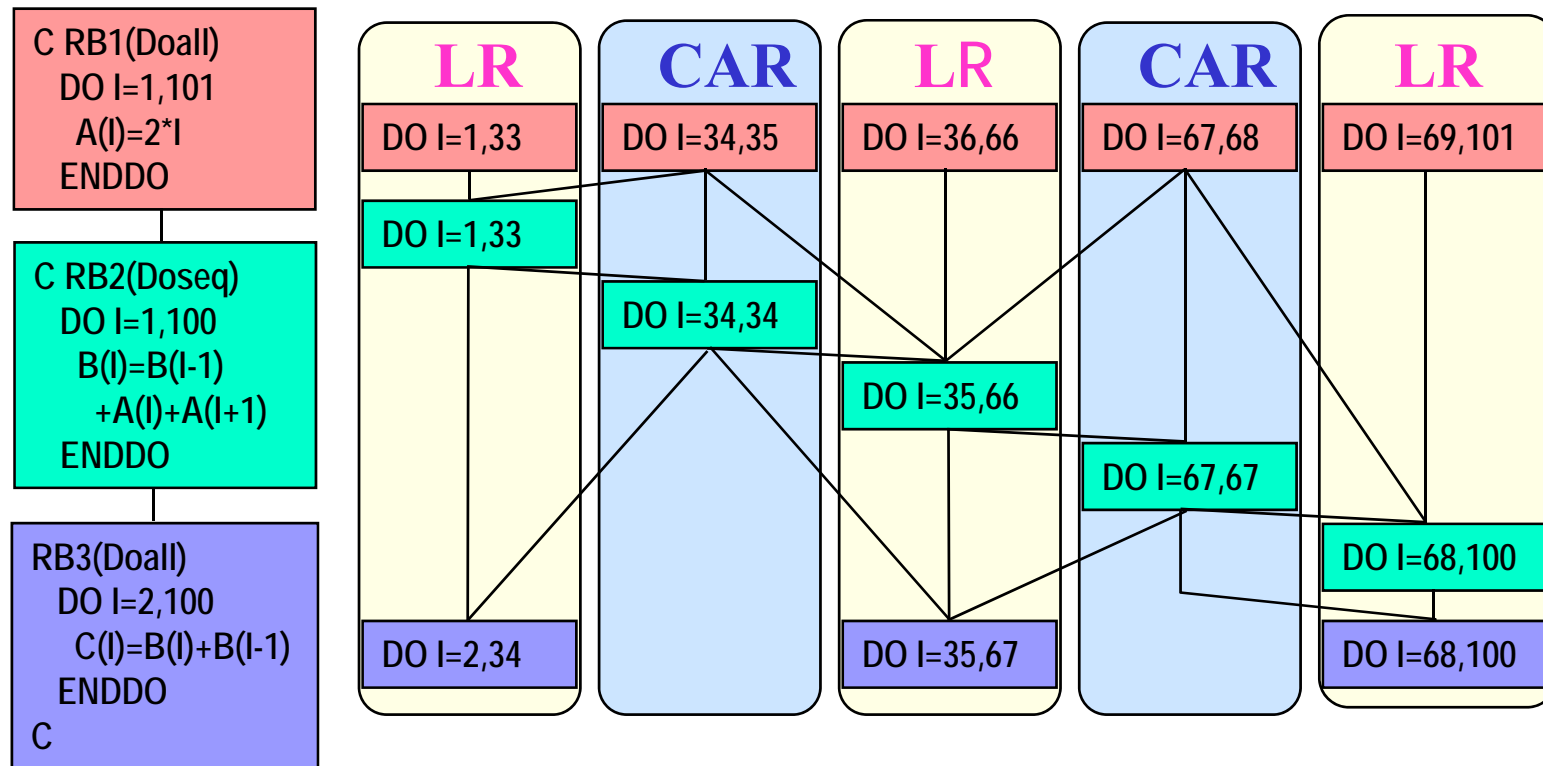
- Coarse grain parallelism **PARA_ALD = 4.3**



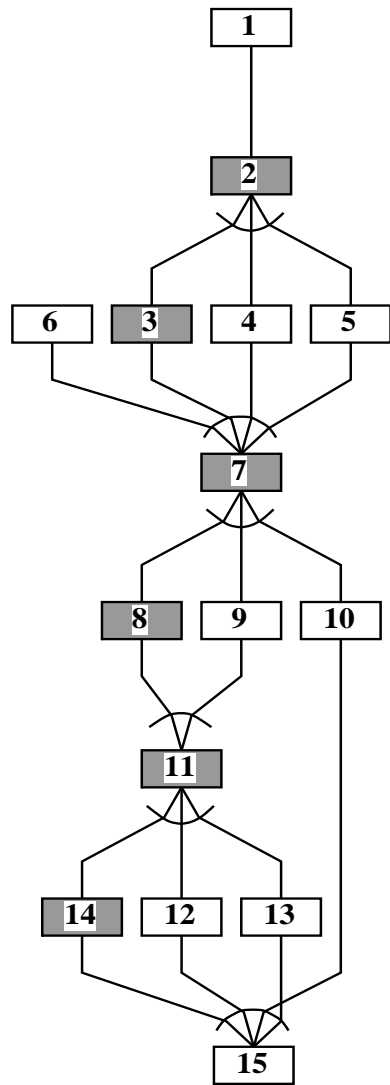
■ DOALL ■ Sequential LOOP ■ SB ■ BB

Data-Localization: Loop Aligned Decomposition

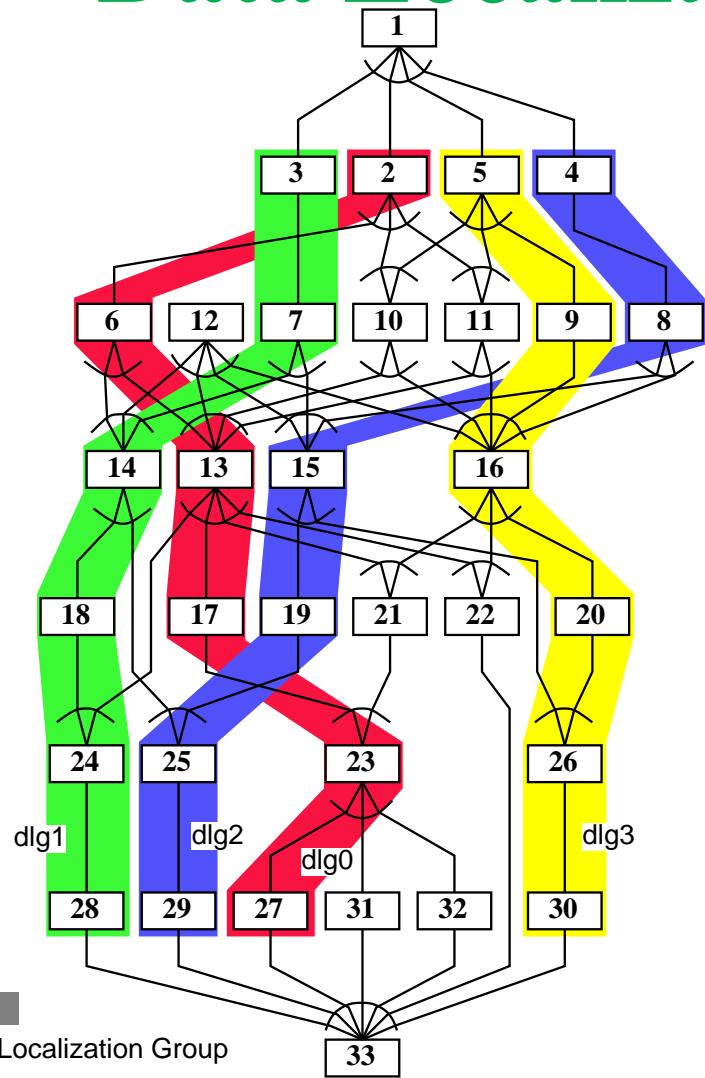
- Decompose multiple loop (Doall and Seq) into **CARs** and **LRs** considering inter-loop data dependence.
 - Most data in **LR** can be passed through LM.
 - LR: Localizable Region, CAR: Commonly Accessed Region**



Data Localization



MTG



■ Data Localization Group

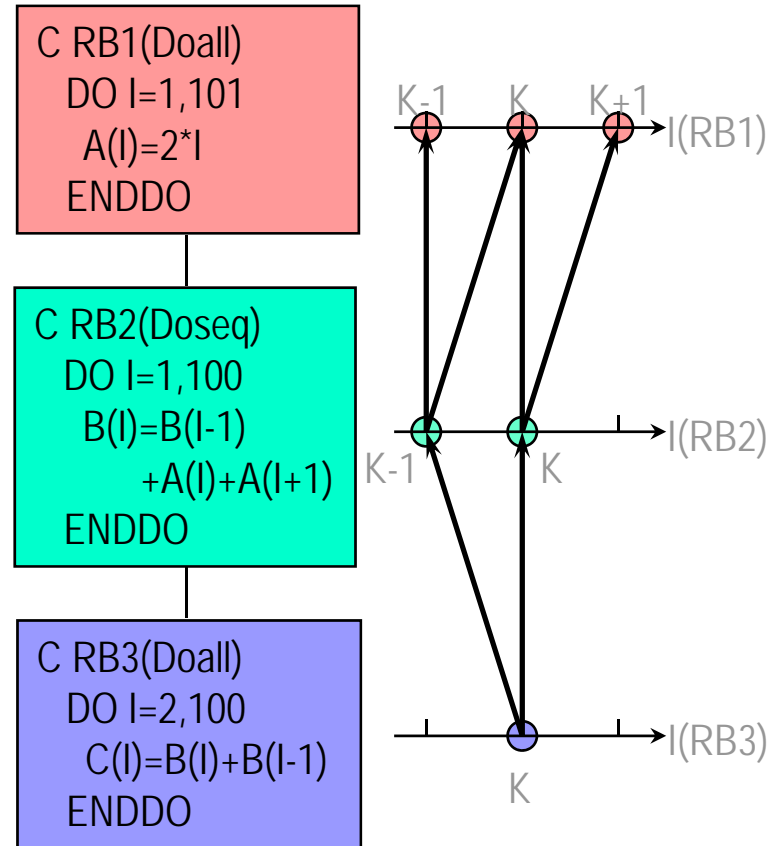
MTG after Division

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

Inter-loop data dependence analysis in TLG

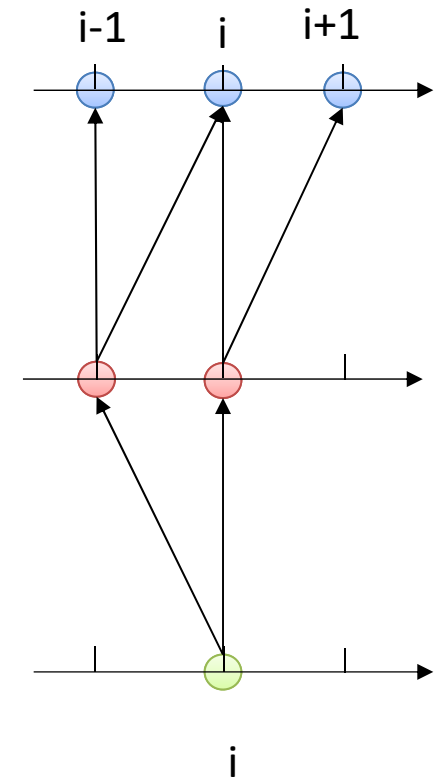
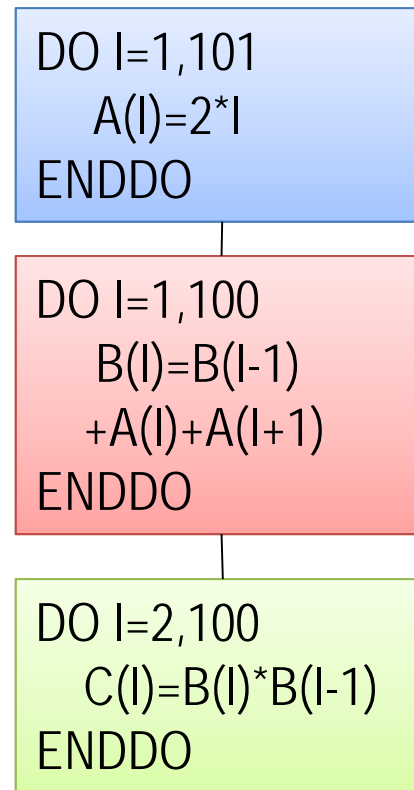
- Define exit-RB in TLG as Standard-Loop
- Find iterations on which a iteration of Standard-Loop is data dependent
 - e.g. K_{th} of RB3 is data-dep on $K-1_{th}, K_{th}$ of RB2, on $K-1_{th}, K_{th}, K+1_{th}$ of RB1



Example of TLG

Target Loop Group Creation and Inter-Loop Dependence Analysis

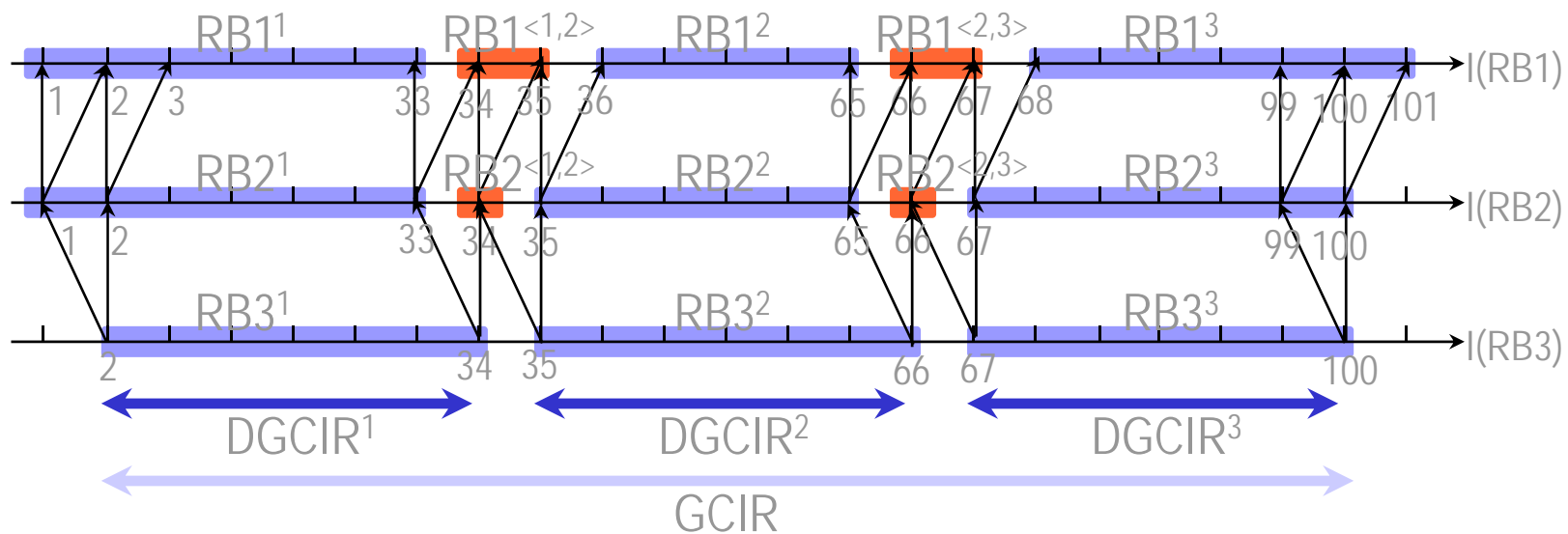
- Target Loop Groups
 - grouped loops that access the same array
 - baseline loop chosen for each group
 - the largest estimated time loop
- Inter-Loop Dependency Analysis
 - data dependencies between loops within the TLGs
 - detects relevant iterations of those loops that have dependence with the iterations of the baseline loop



Inter-Loop dependence

Decomposition of RBs in TLG

- Decompose GCIR into $DGCIR^p (1 \leq p \leq n)$
 - n: (multiple) num of PCs, DGCIR: Decomposed GCIR
- Generate CAR on which $DGCIR^p \& DGCIR^{p+1}$ are data-dep.
- Generate LR on which $DGCIR^p$ is data-dep.

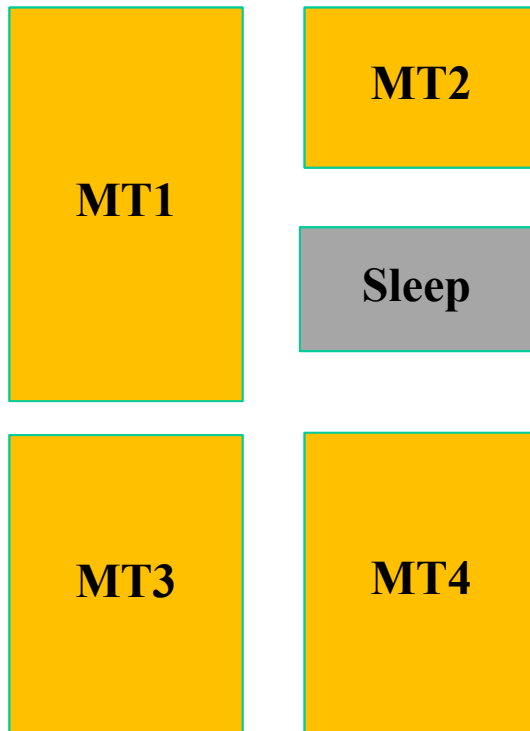


Low-Power Optimization with OSCAR API

Scheduled Result
by OSCAR Compiler

VC0

VC1



Generate Code Image by OSCAR Compiler

```
void  
main_VC0() {
```



```
#pragma oscar fvcontrol ¥  
(1,(OSCAR_CPU(),100))
```

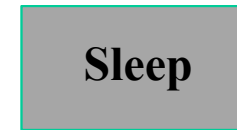


```
}
```

```
void  
main_VC1() {
```

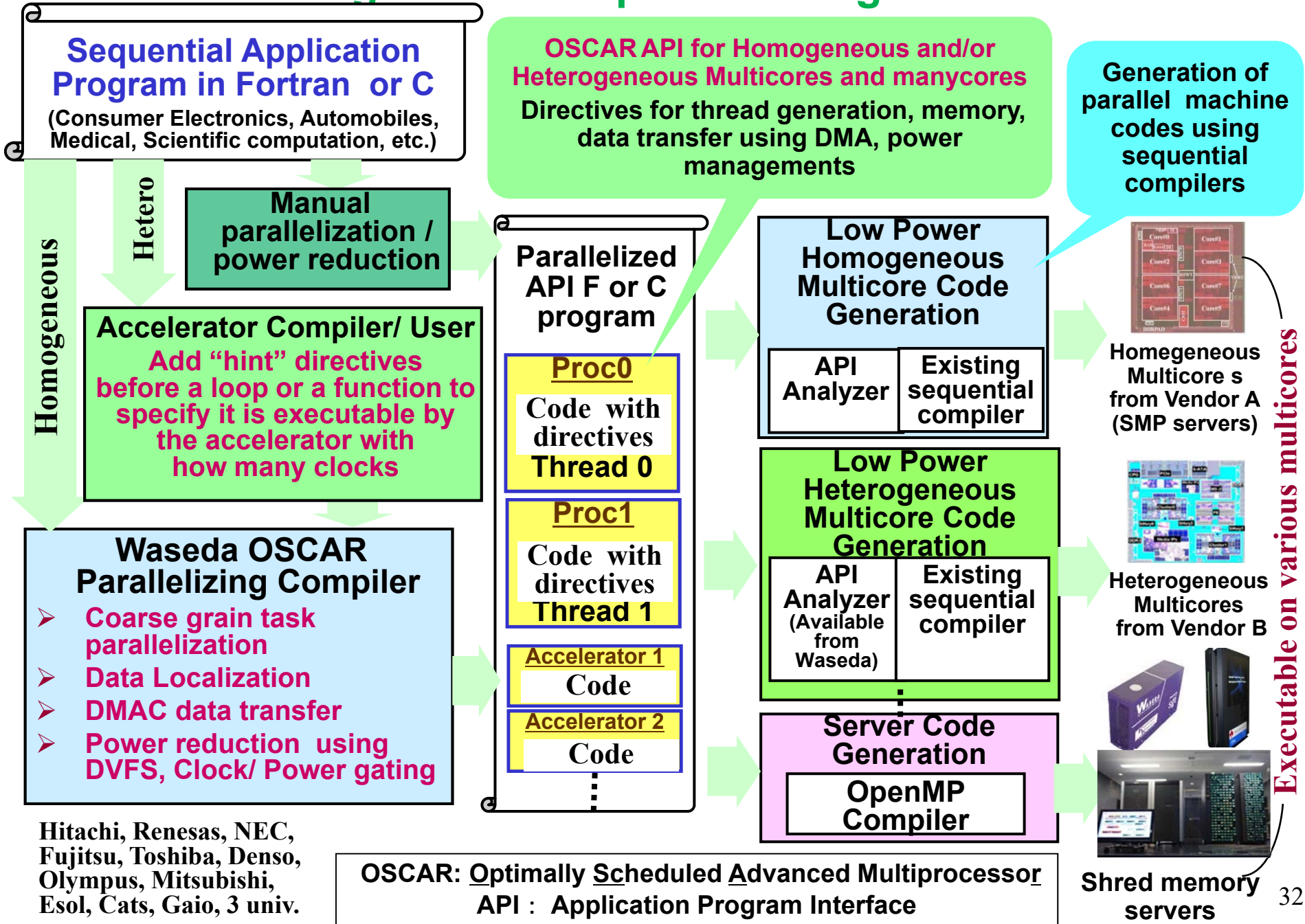


```
#pragma oscar fvcontrol ¥  
((OSCAR_CPU(),0))
```



```
}
```

Multicore Program Development Using OSCAR API V2.0



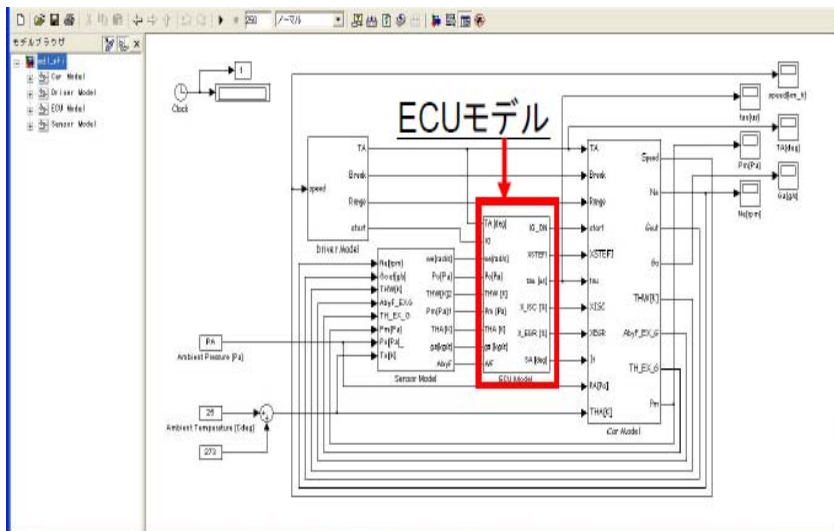
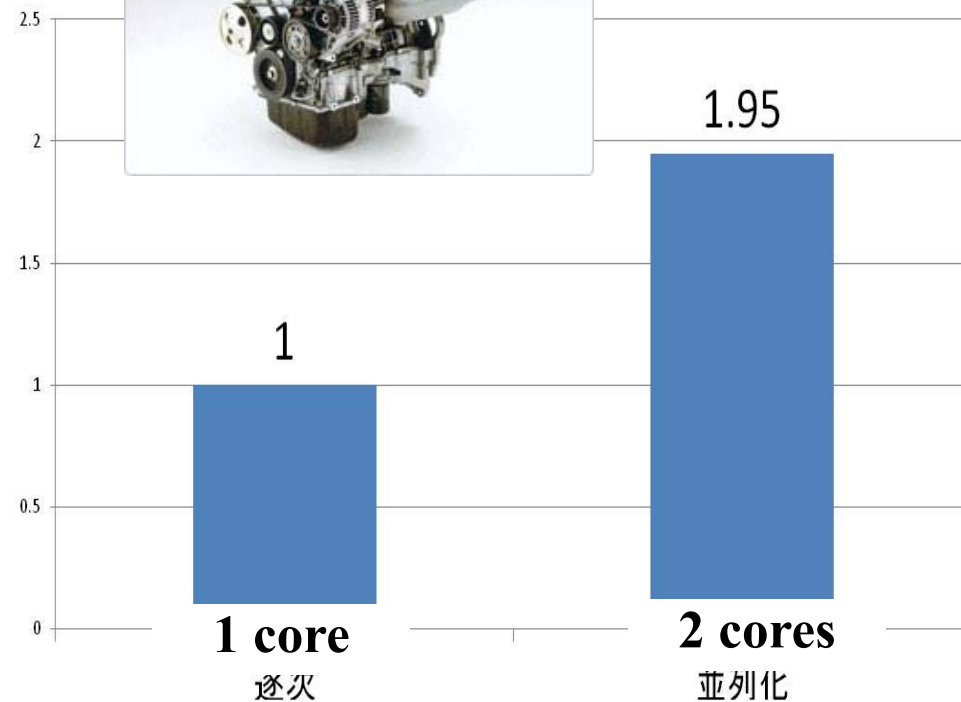


Engine Control by multicore with Denso

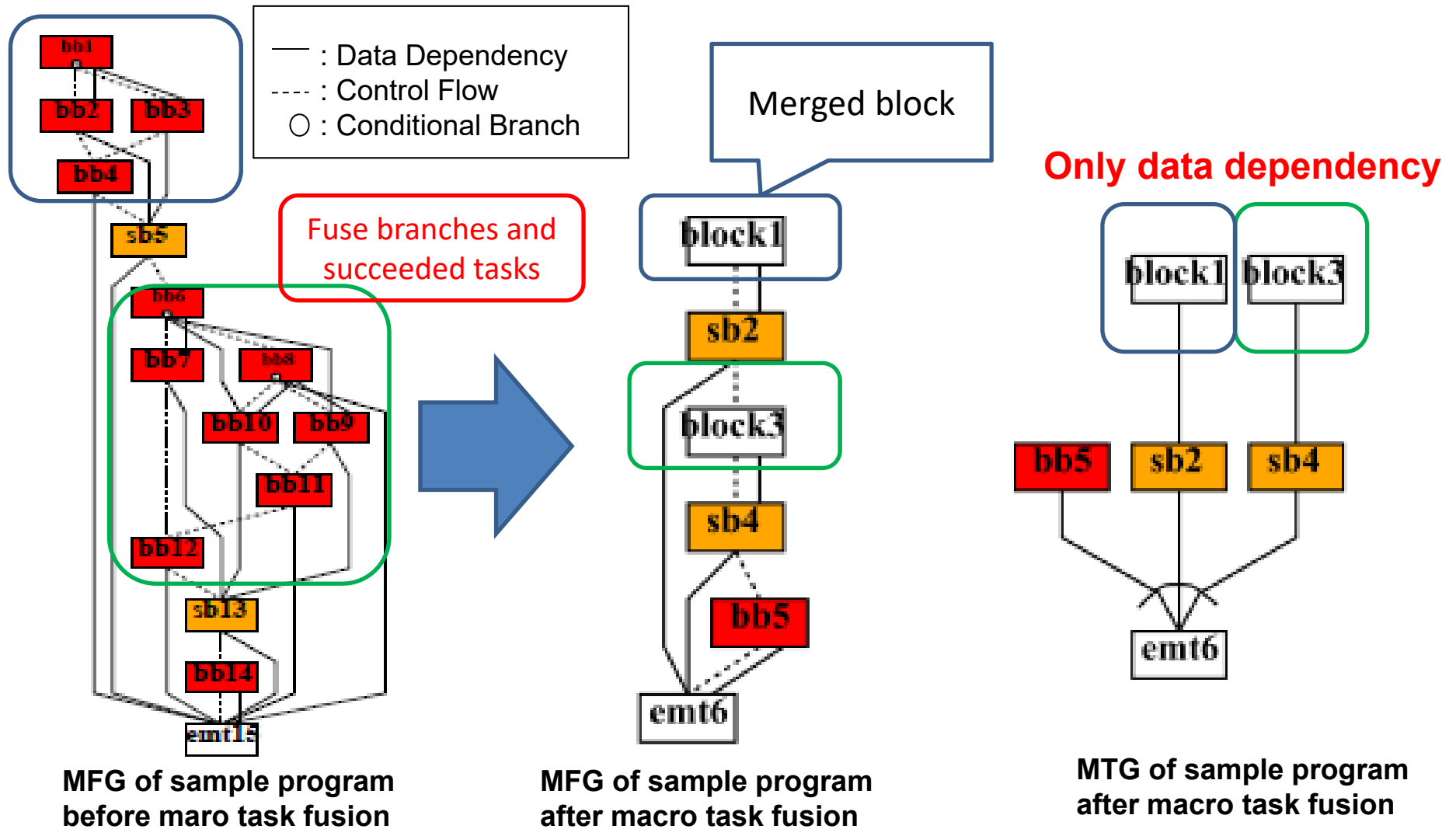
Though so far parallel processing of the engine control on multicore has been very difficult, Denso and Waseda succeeded 1.95 times speedup on 2core V850 multicore processor.



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



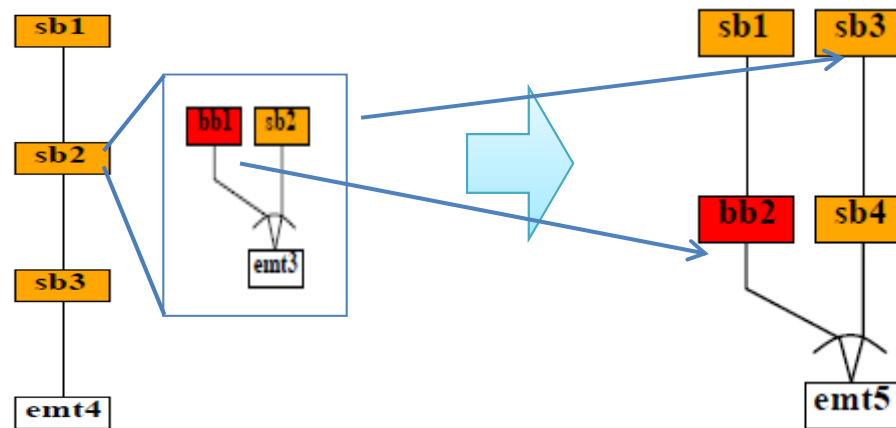
Macro Task Fusion for Static Task Scheduling



3.1 Restructuring : Inline Expansion

- Inline expansion is effective
 - To increase coarse grain parallelism
- Expands functions having inner parallelism

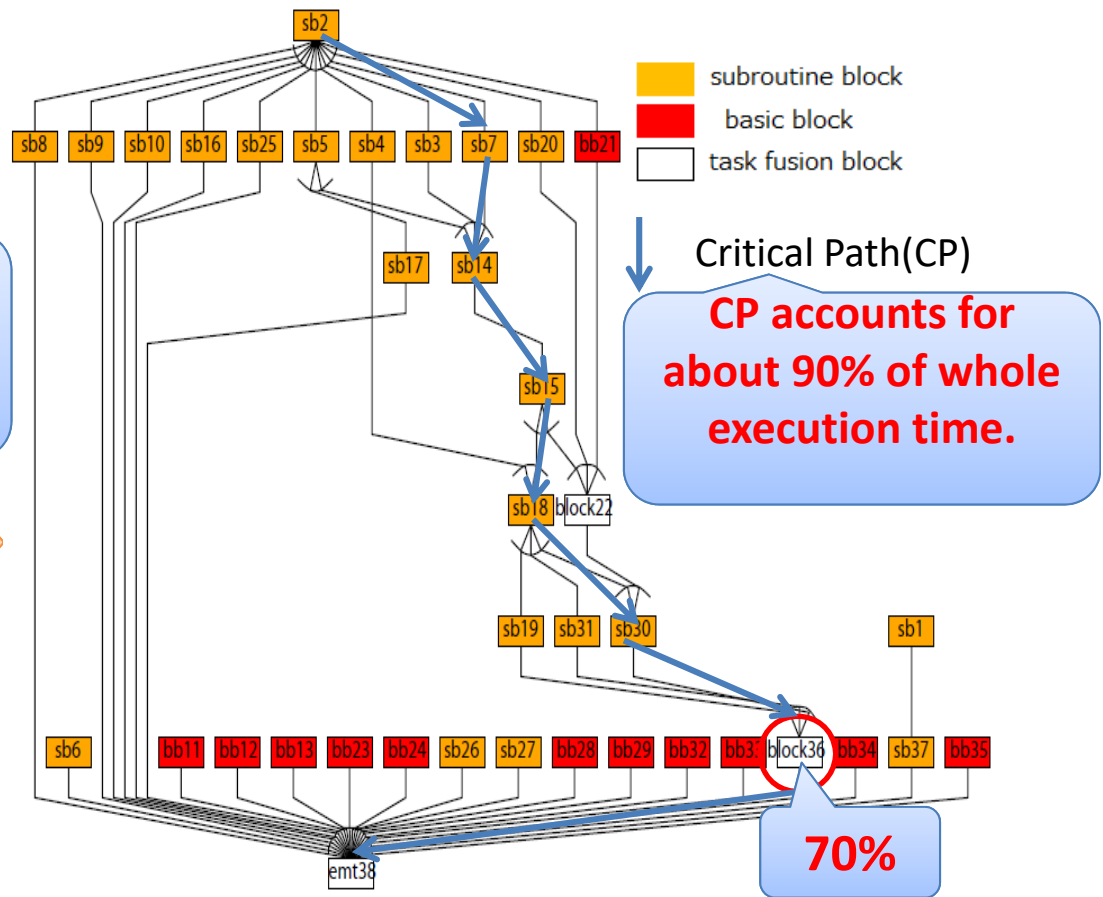
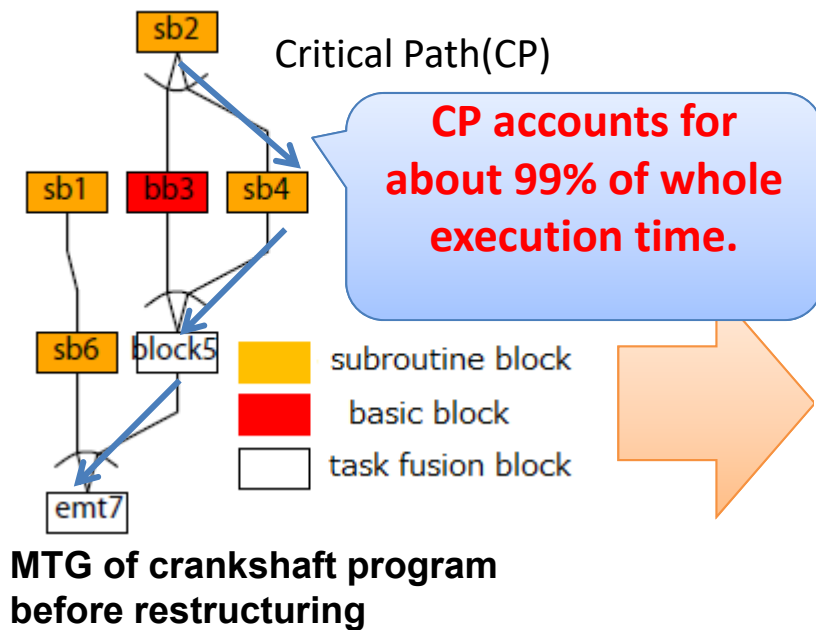
Improves coarse grain parallelism



MTG before inline expansion

MTG after inline expansion

MTG of Crankshaft Program Using Inline Expansion



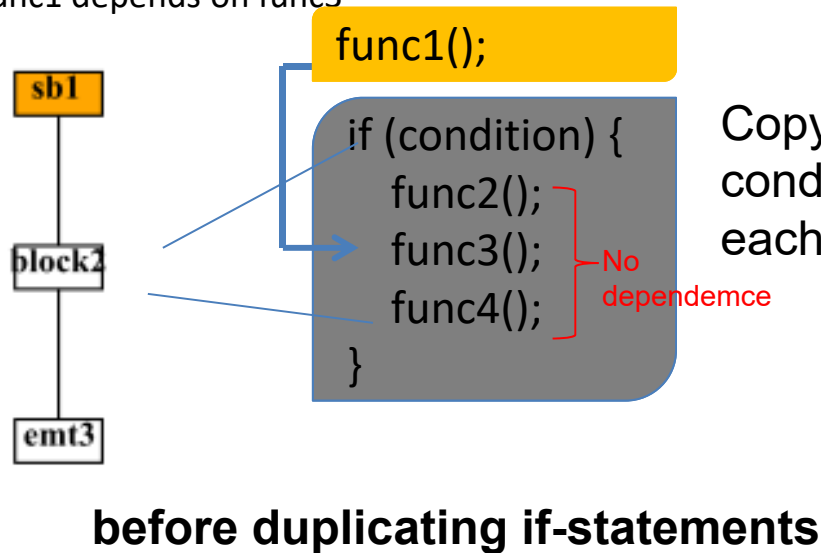
Not enough coarse grain parallelism yet!

3.2 Restructuring: Duplicating If-statements

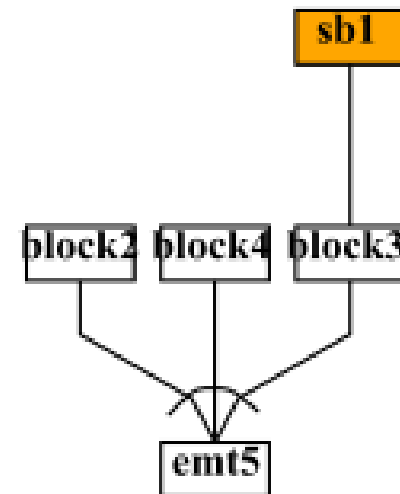
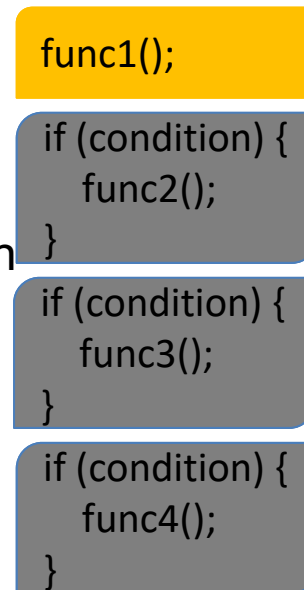
- Duplicating if-statements is effective
 - To increase coarse grain parallelism
- Duplicates fused tasks having inner parallelism

Improves coarse grain parallelism

Func1 depends on func3

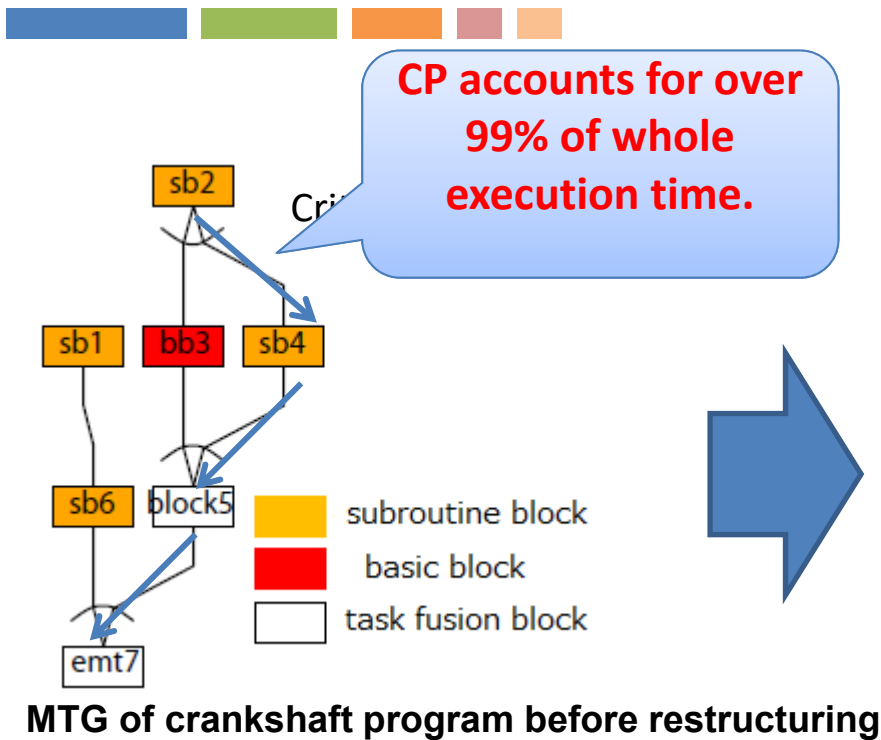


Copying if-condition for each function

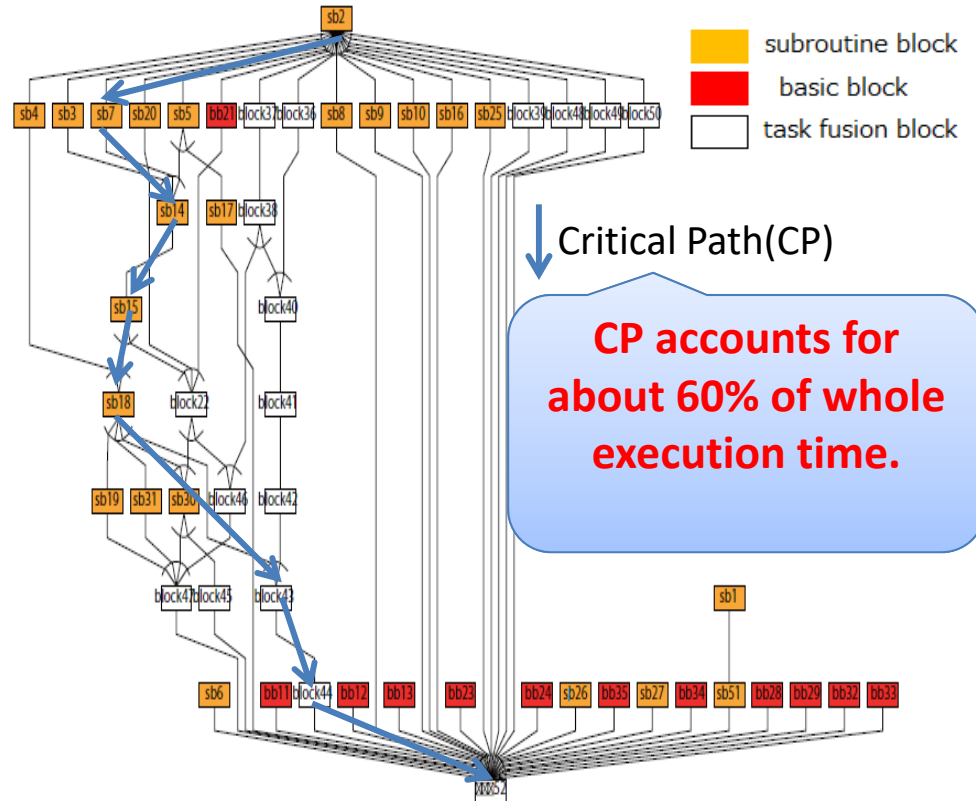


after duplicating if-statements

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

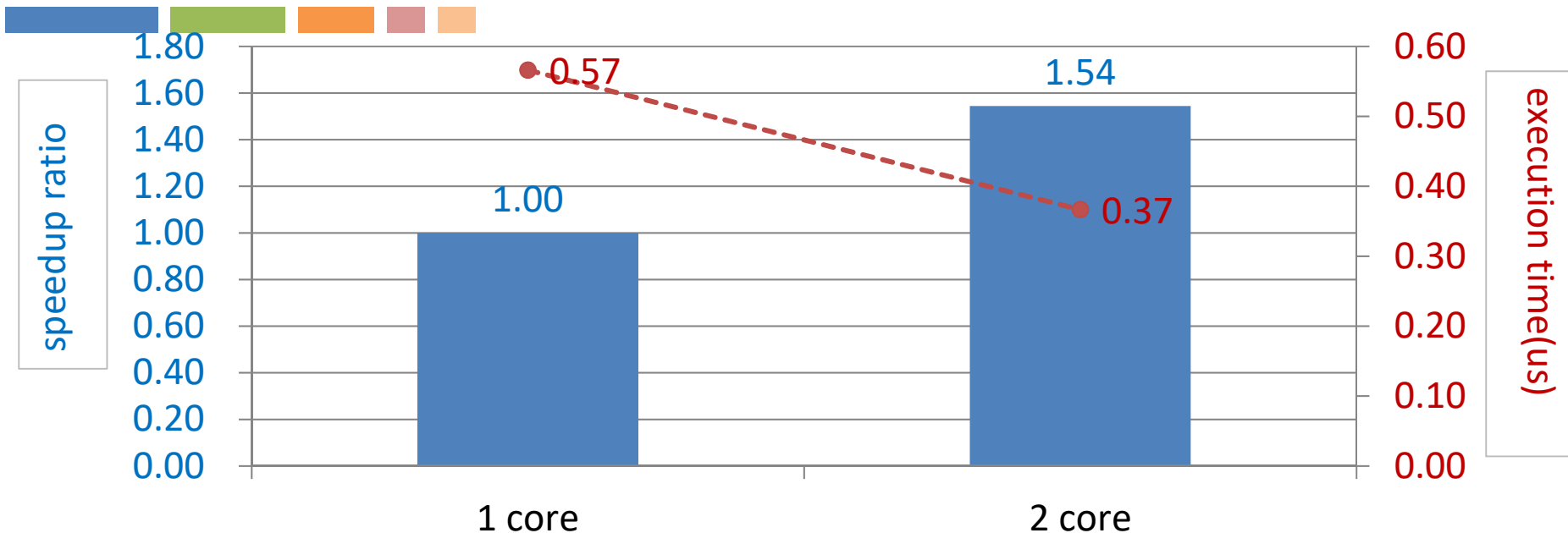


- Succeed to reduce CP
 - 99% -> 60%



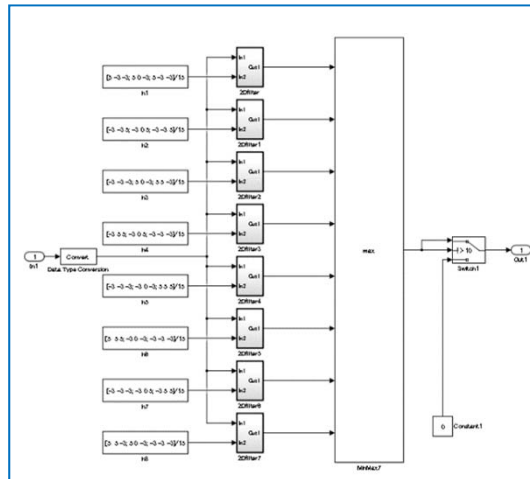
Successfully increased coarse grain parallelism

Evaluation of Crankshaft Program with Multi-core Processors



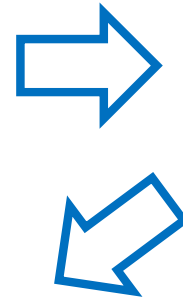
- Attain 1.54 times speedup on RPX
 - There are no loops, but only many conditional branches and small basic blocks and difficult to parallelize this program
- This result shows possibility of multi-core processor for engine control programs

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' */

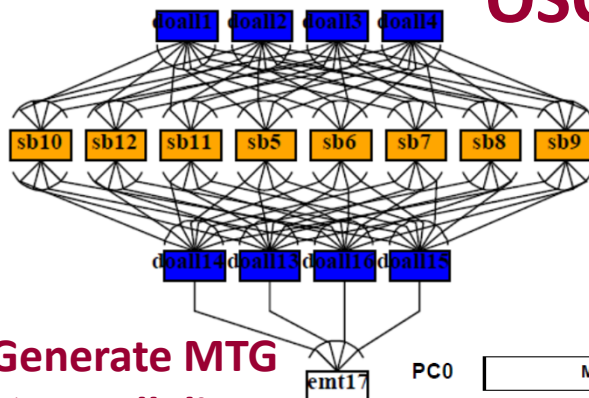
    /* Constant: '<S1>/h1' */
    VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
        VesselExtraction_P.h1_Value, &VesselExtraction_B.Dfilter,
        (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter);

    /* 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);
}
    
```

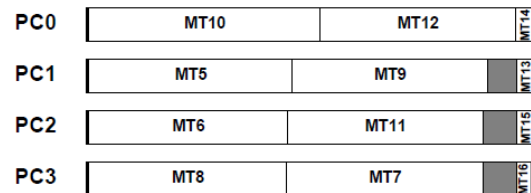
C code



OSCAR Compiler

(1) Generate MTG
→ Parallelism

(2) Generate gantt chart
→ Scheduling in a multicore



0.0E+00 4.0E-02
TIME [s]



```

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 ) ;

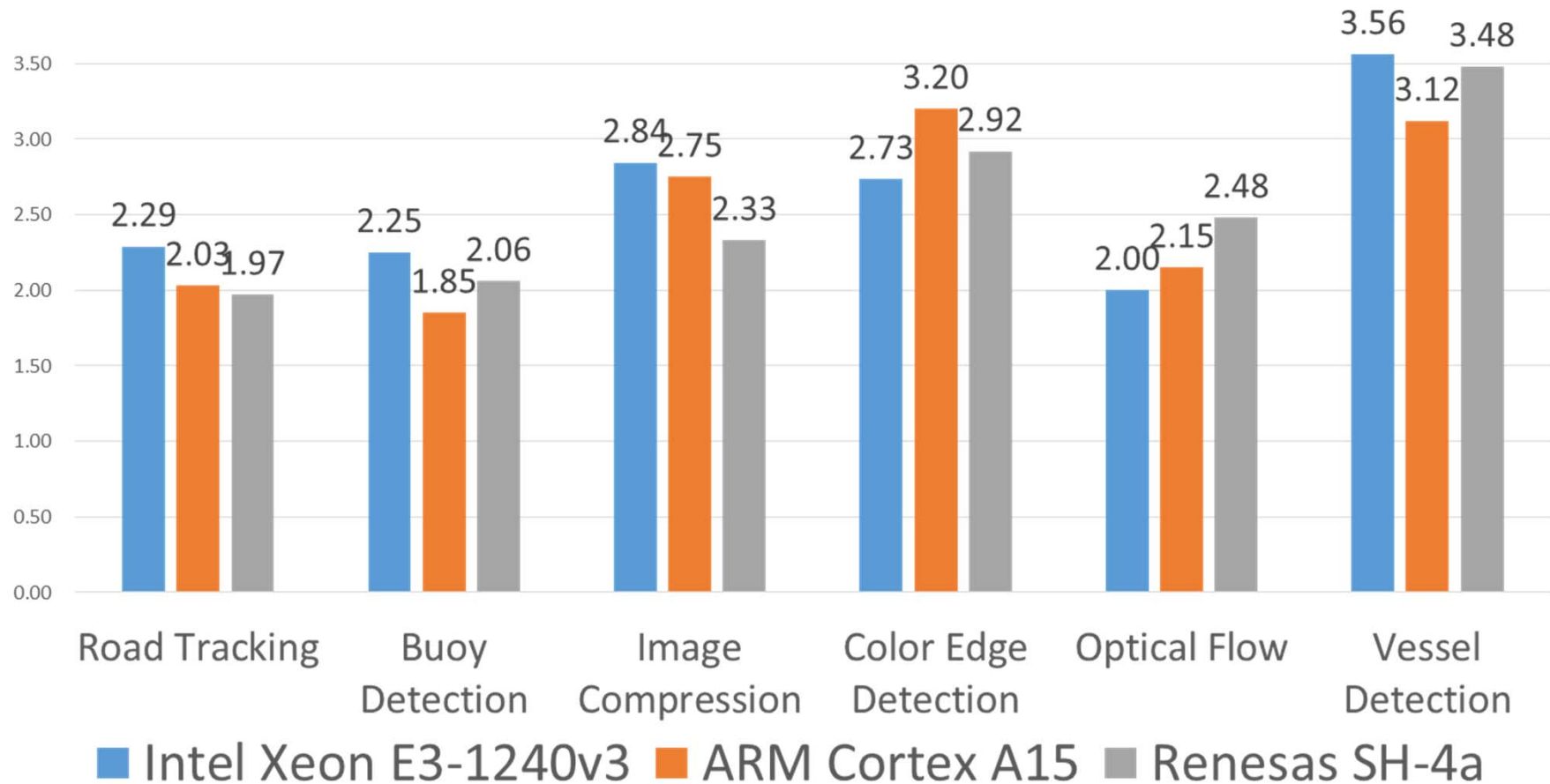
    VesselExtraction_step_PEO ( ) ;

    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)



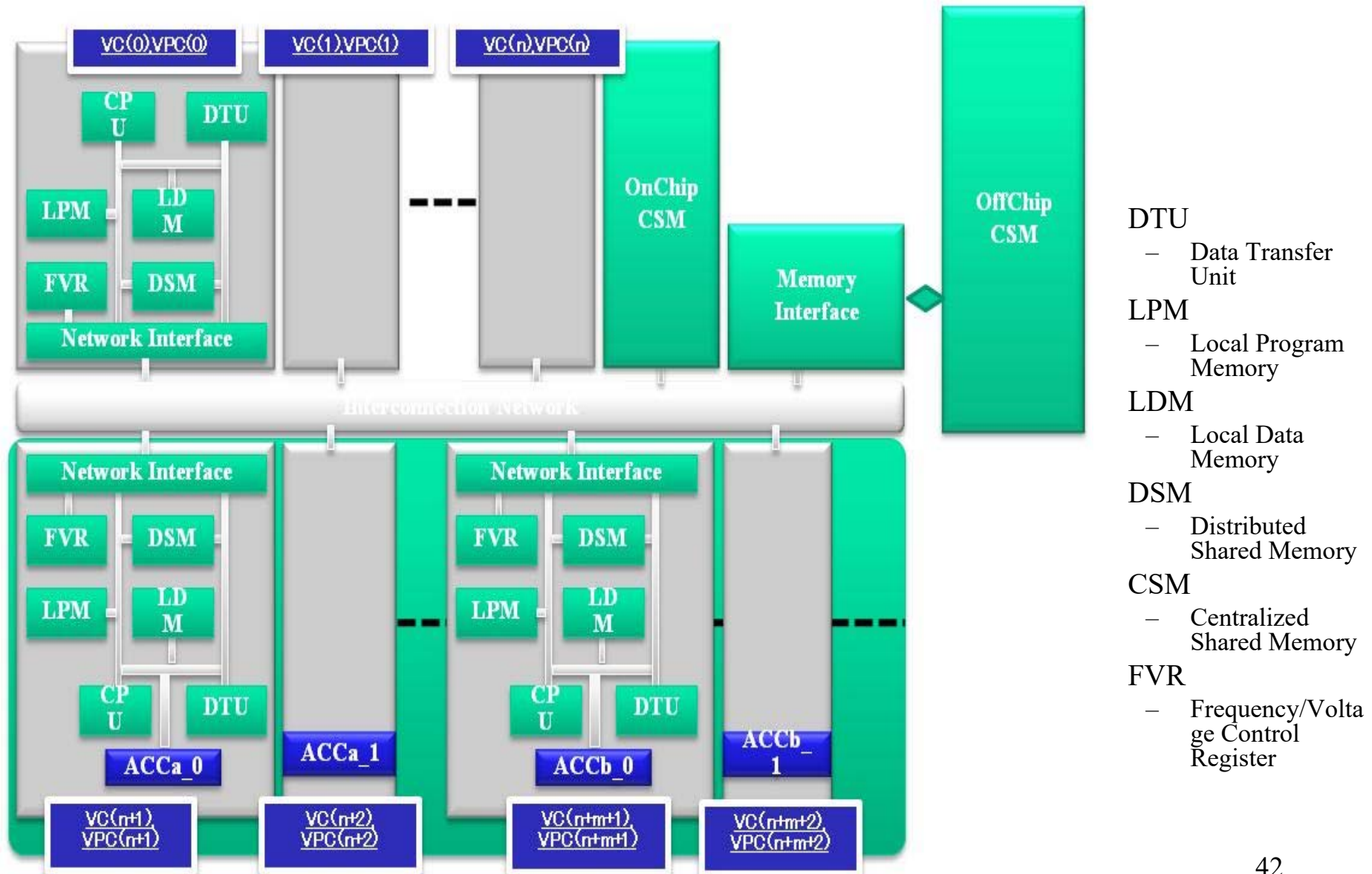
Road Tracking, Image Compression : <http://www.mathworks.co.jp/jp/help/vision/examples>

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

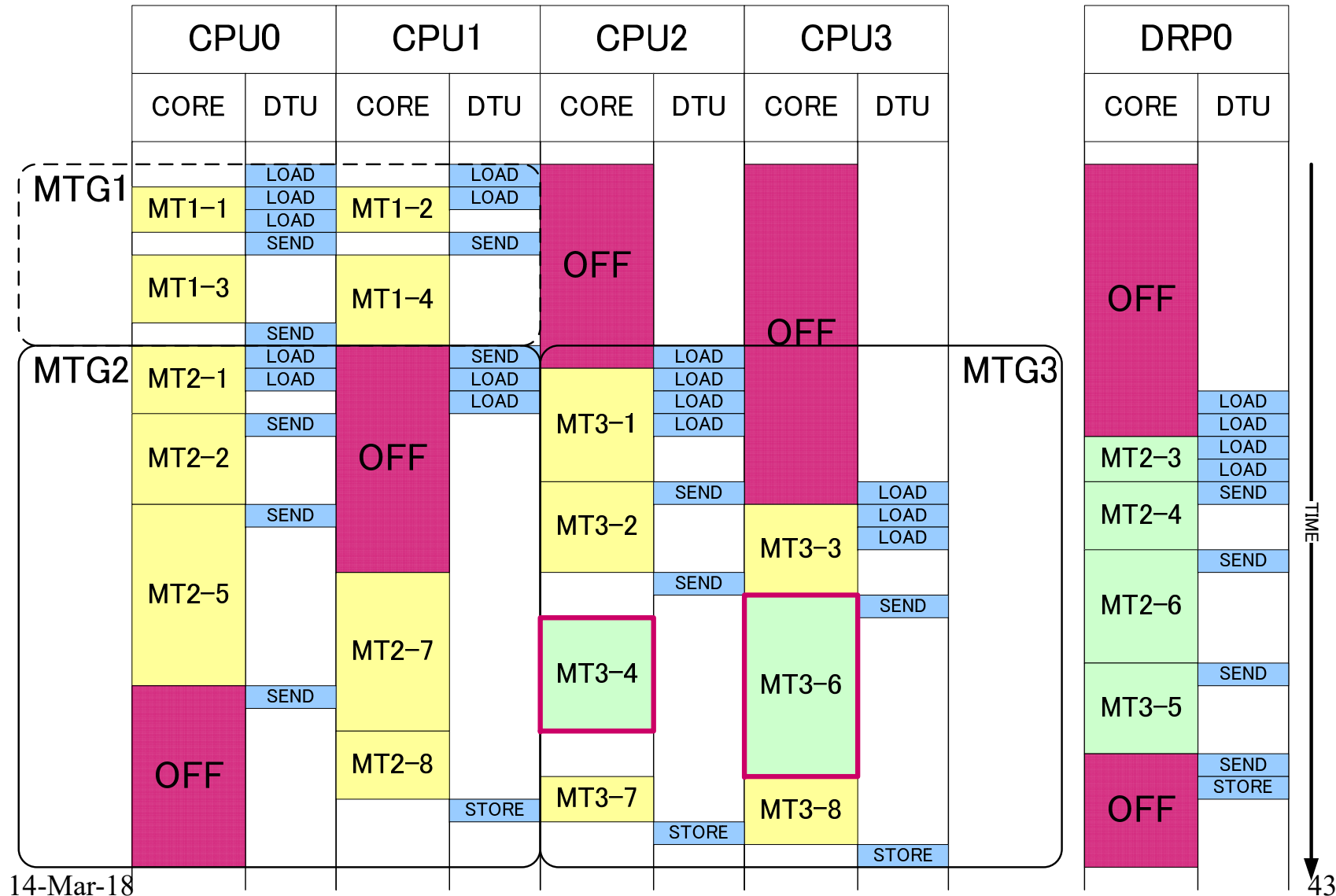
Color Edge Detection : <http://www.mathworks.co.jp/matlabcentral/fileexchange/28114-fast-edges-of-a-color-image--actual-color--not-converting-to-grayscale-/>

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

OSCAR Heterogeneous Multicore

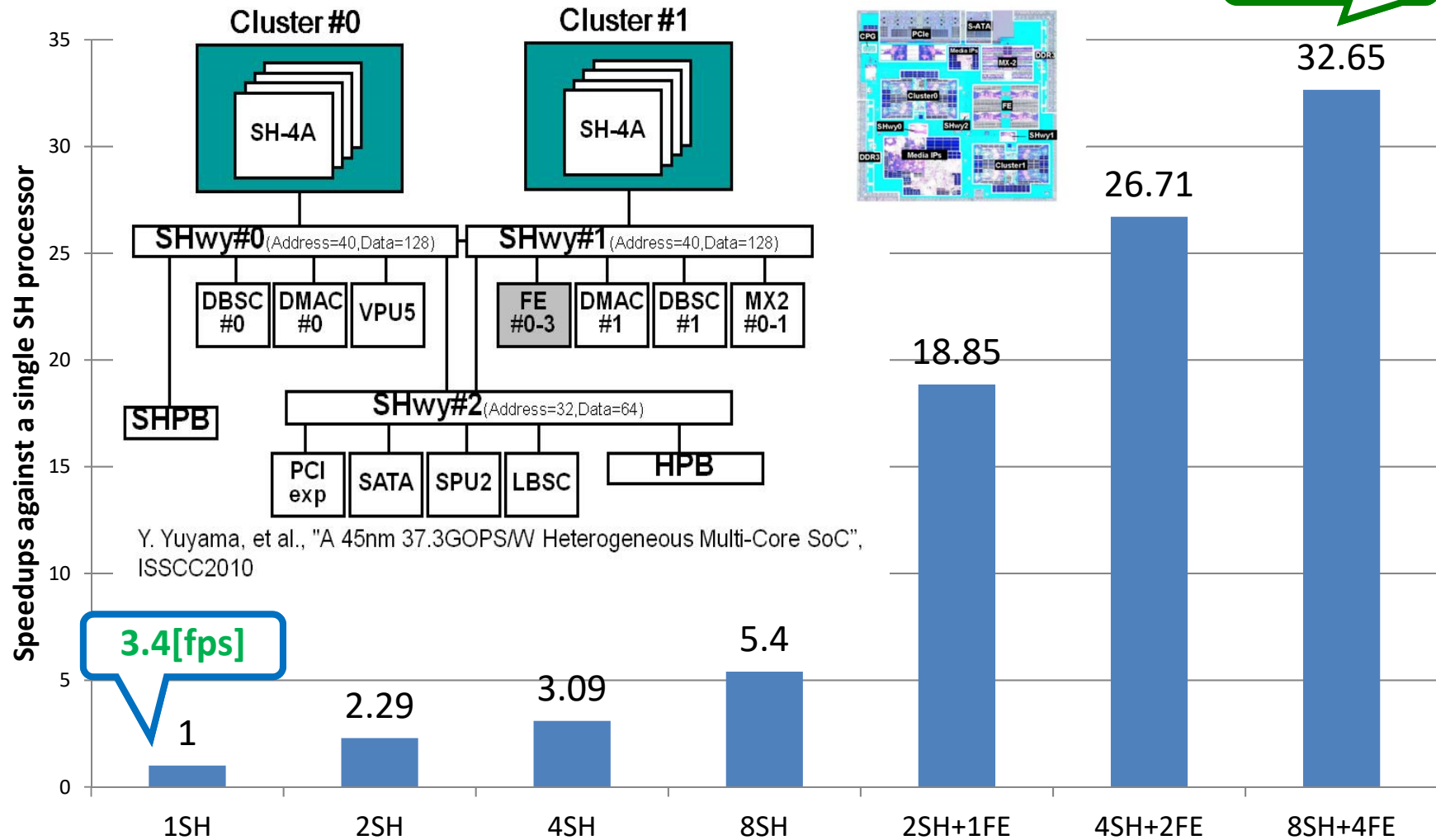


An Image of Static Schedule for Heterogeneous Multi-core with Data Transfer Overlapping and Power Control



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

111[fps]



3.4[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)

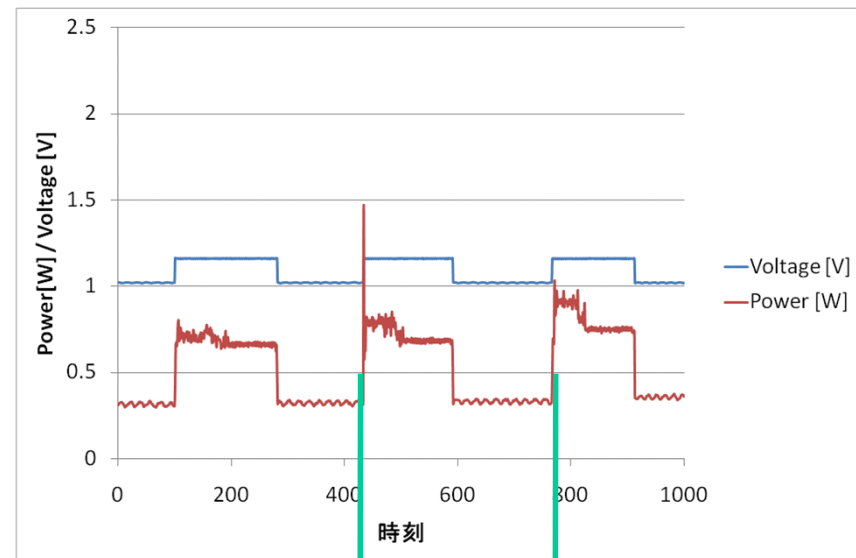
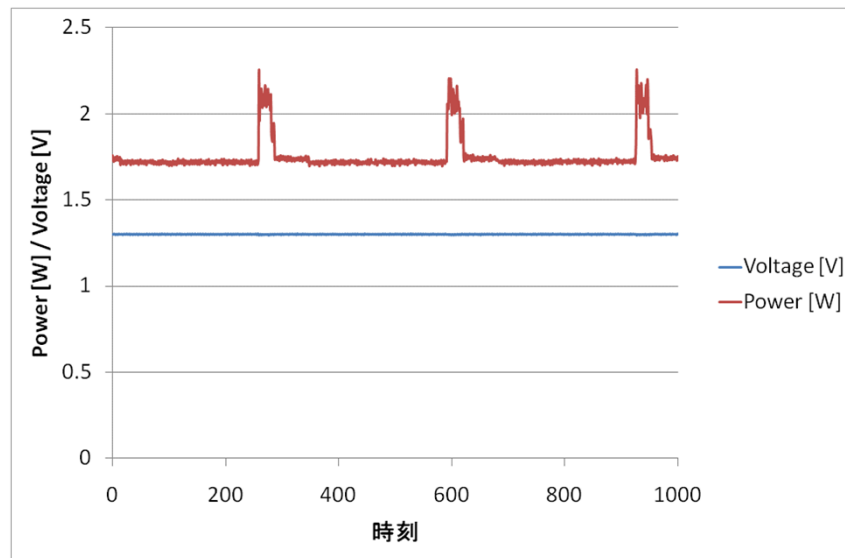
Without Power Reduction

With Power Reduction by OSCAR Compiler
70% of power reduction

Average: 1.76[W]



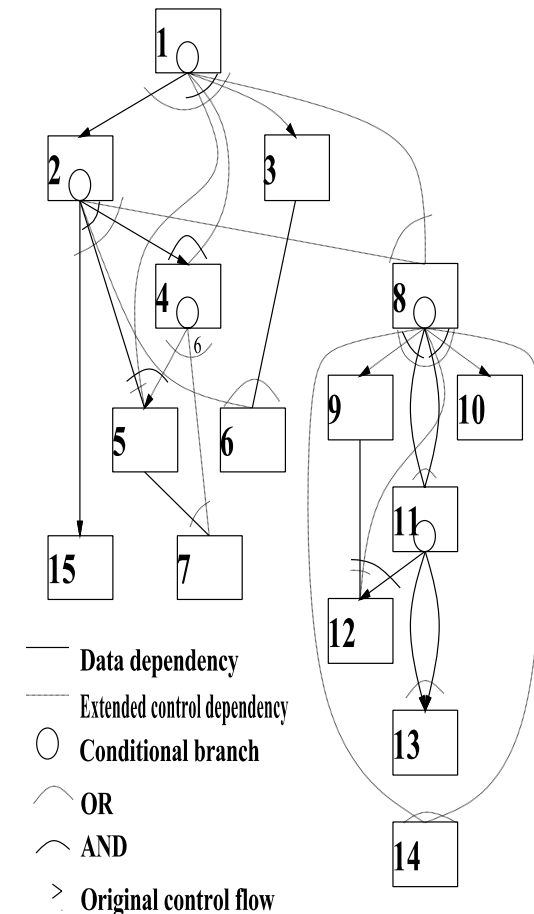
Average: 0.54[W]



**1cycle : 33[ms]
→30[fps]**

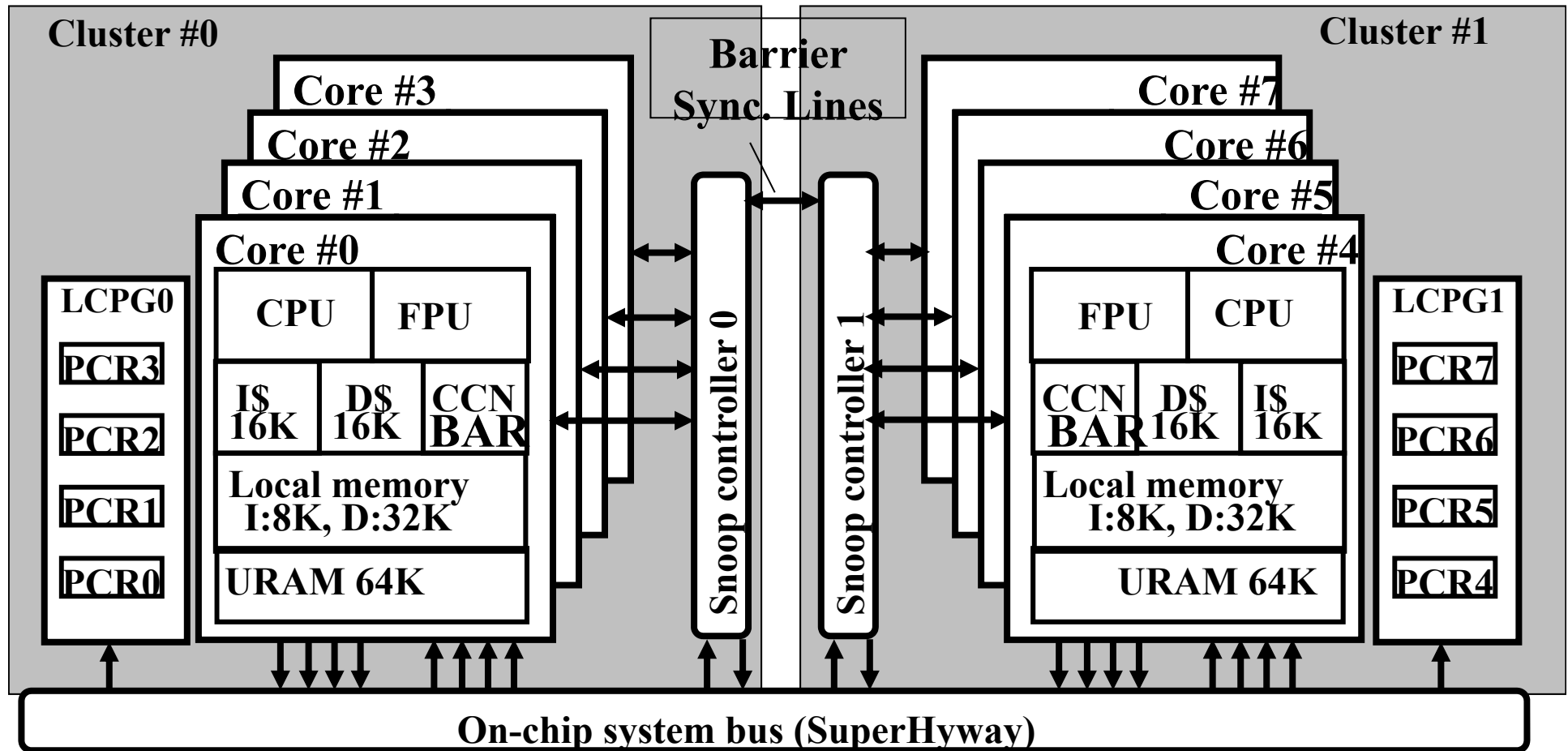
Software Coherence Control Method on OSCAR Parallelizing Compiler

- 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 46
condition analysis**

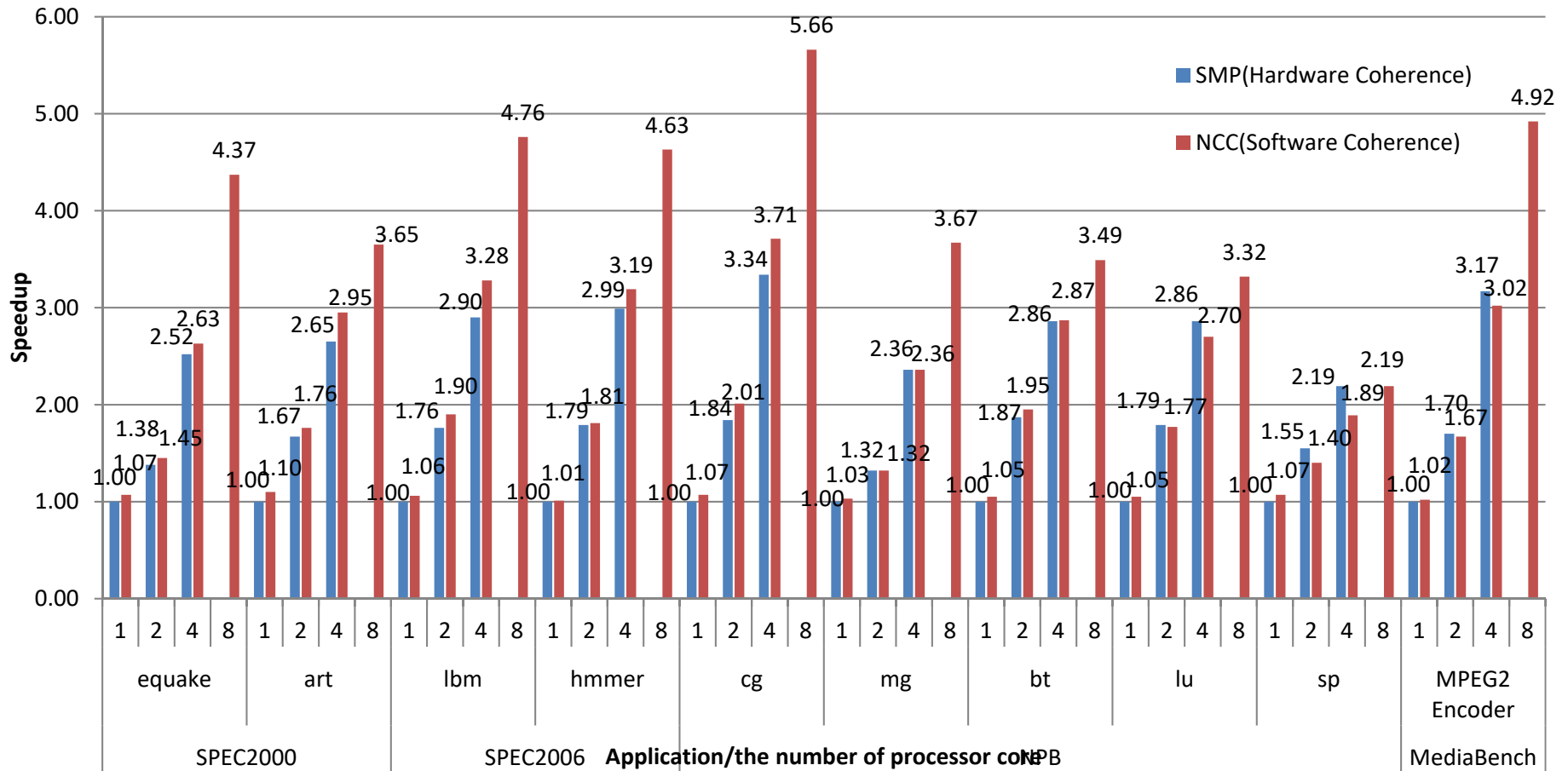
8 Core RP2 Chip Block Diagram



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

Automatic Software Coherent Control for Manycores

Performance of Software Coherence Control by OSCAR Compiler on 8-core RP2

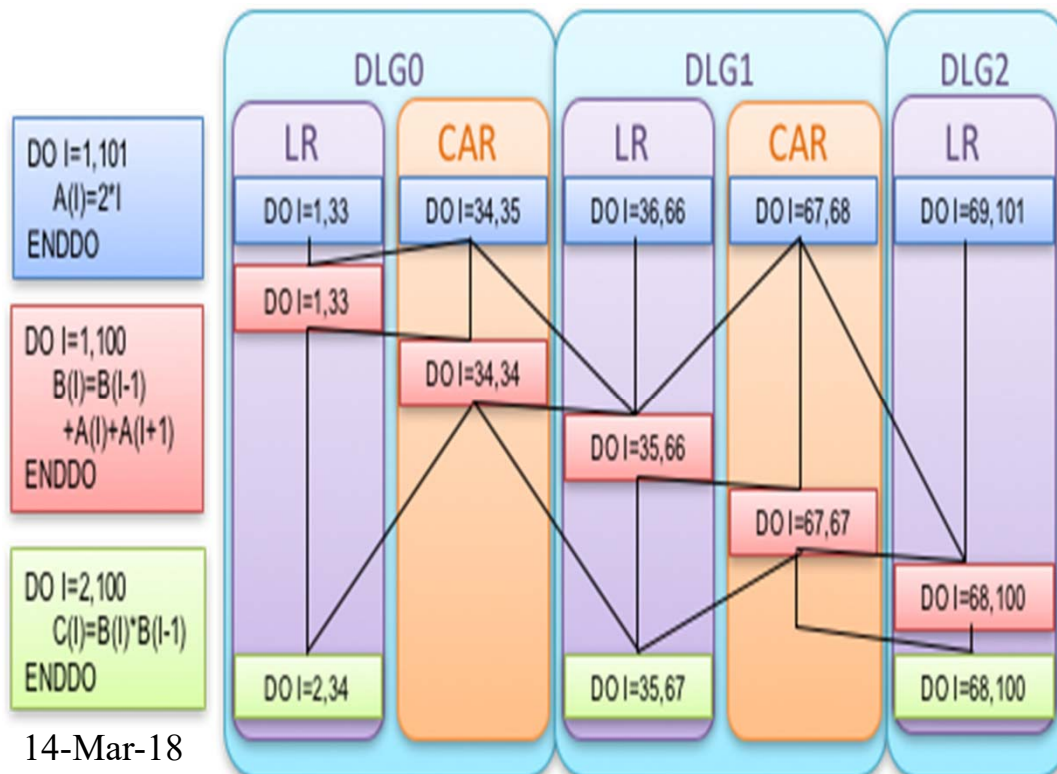


Automatic Local Memory Management

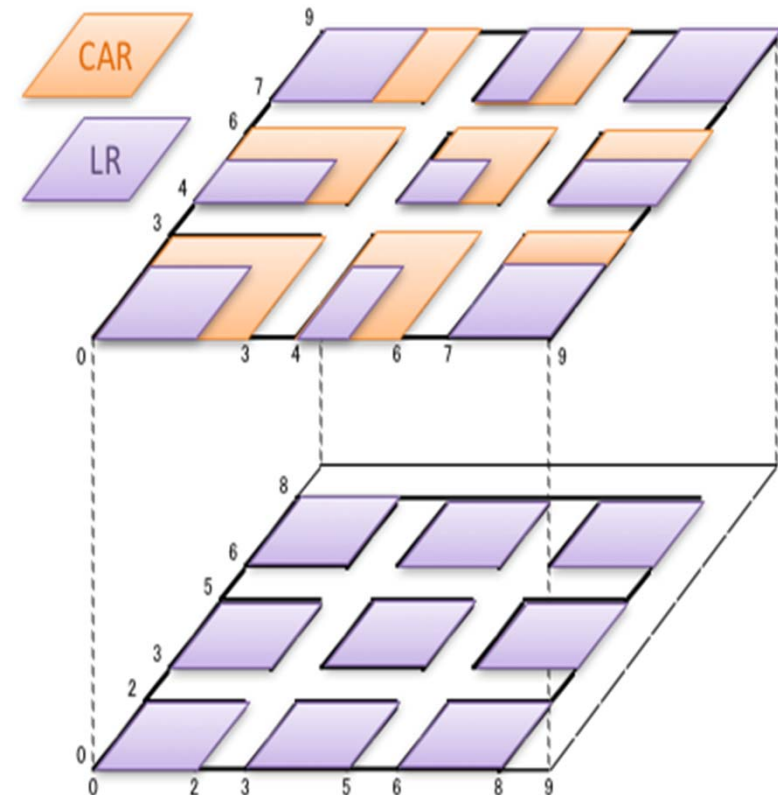
Data Localization: Loop Aligned Decomposition

- 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

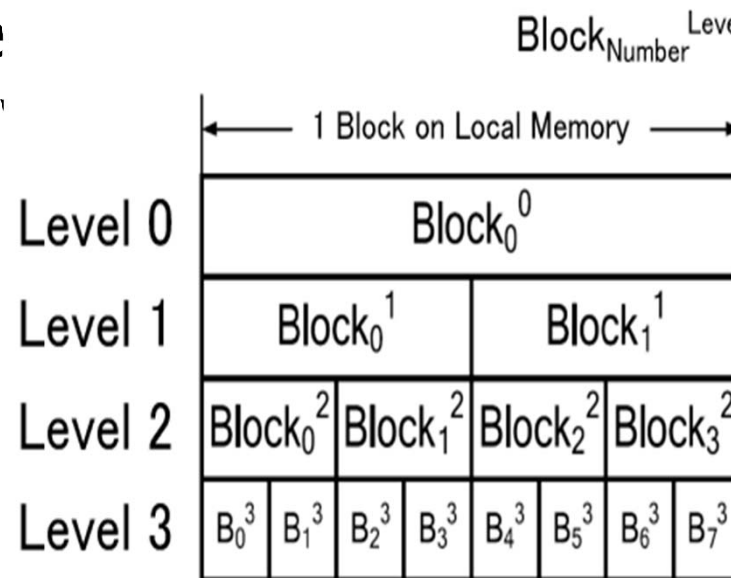


Multi-dimension Decomposition



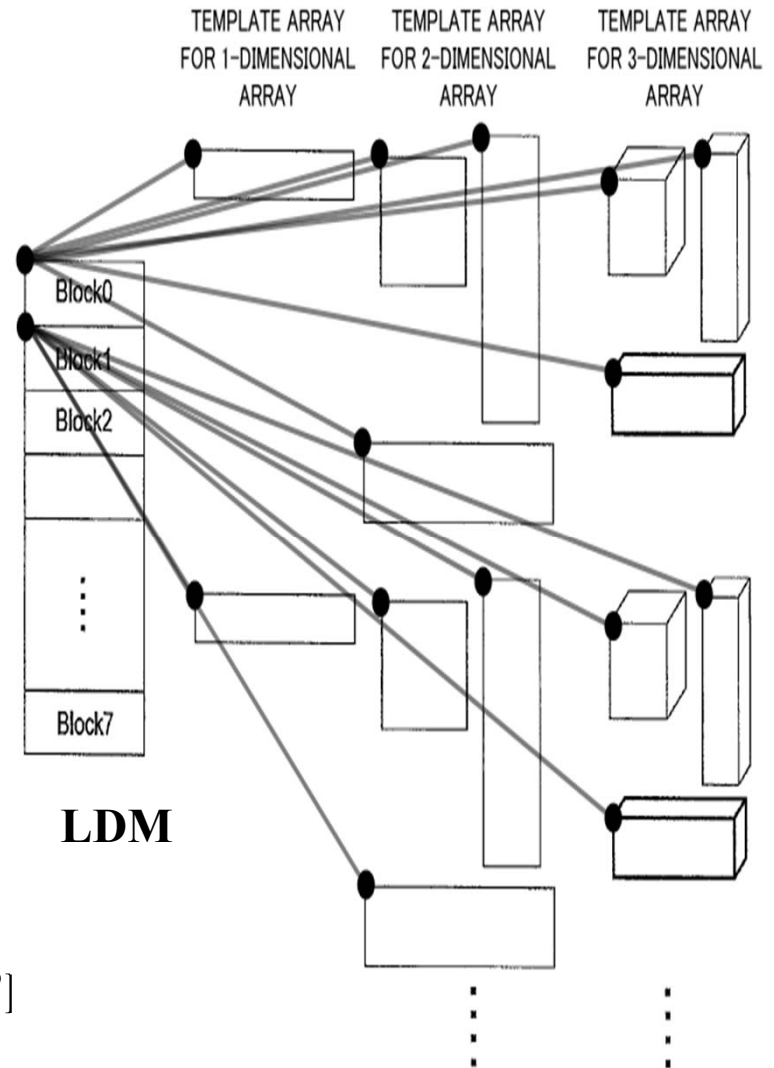
Adjustable Blocks

- 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

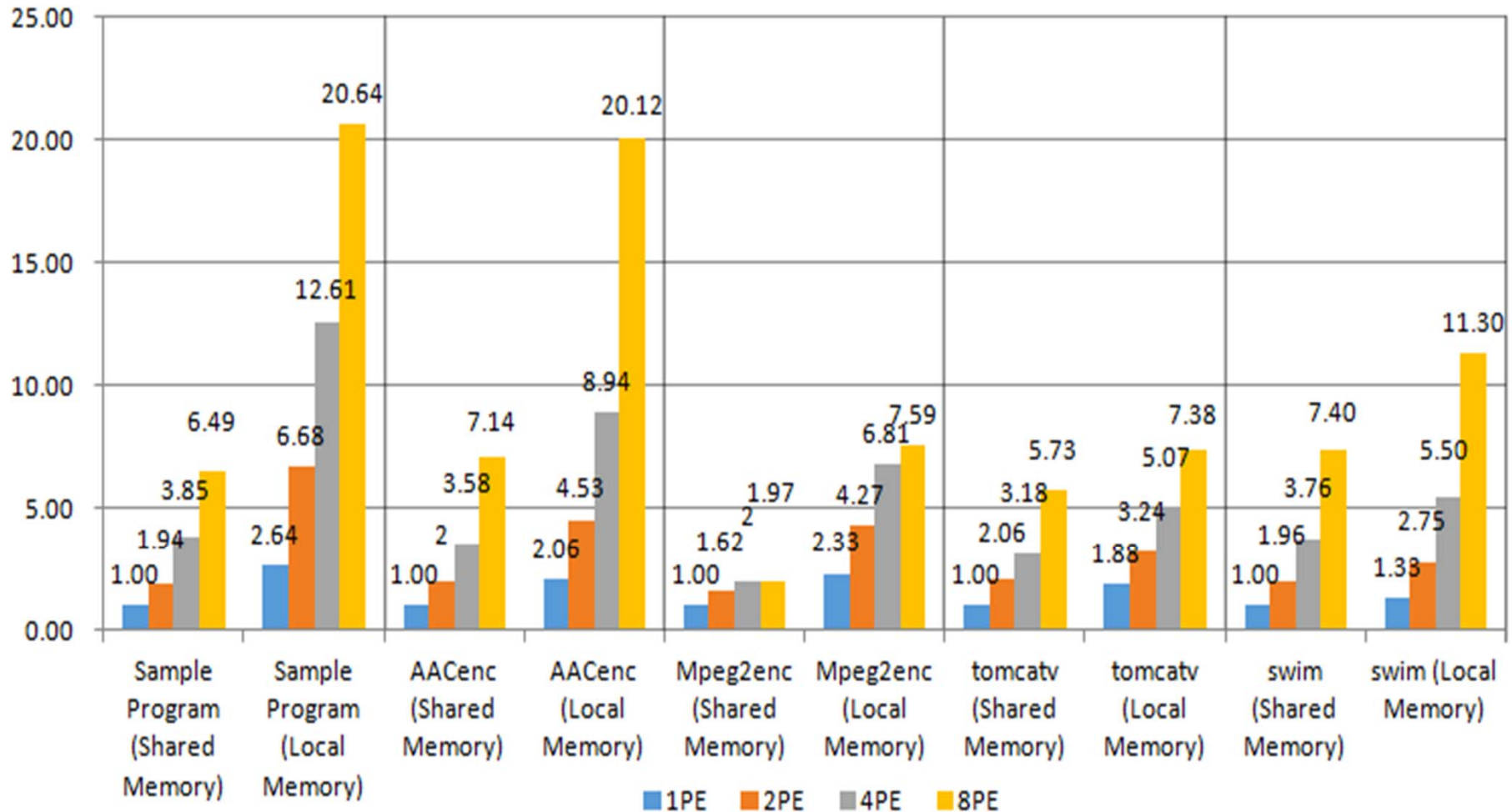
□ **Compiler Control Memory block Replacement**

- 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 the Proposed Local Memory Management Compared with Utilizing Shared Memory on Benchmarks Application using RP2

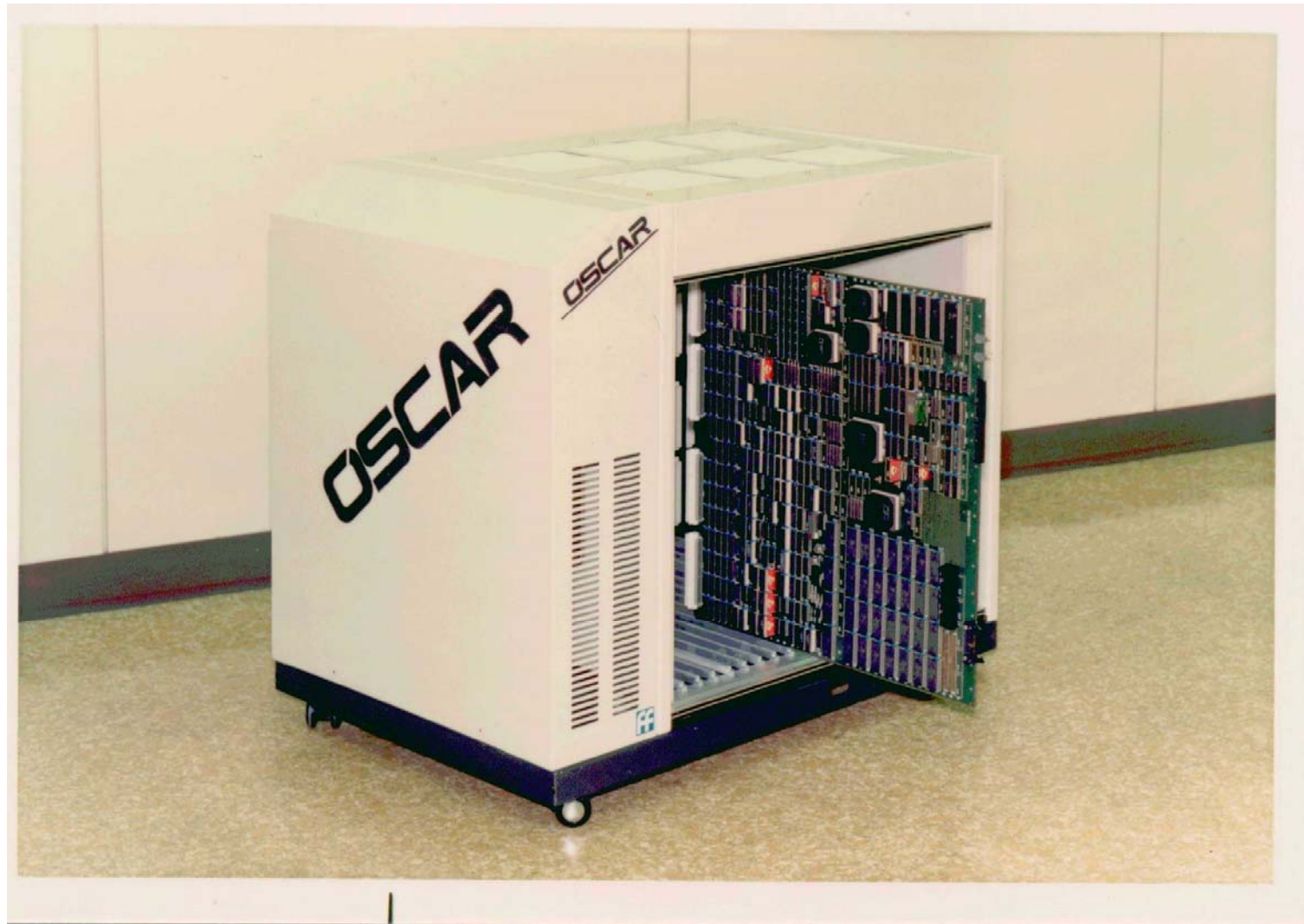


20.12 times speedup for 8cores execution using local memory against sequential execution using off-chip shared memory of RP2 for the AACenc

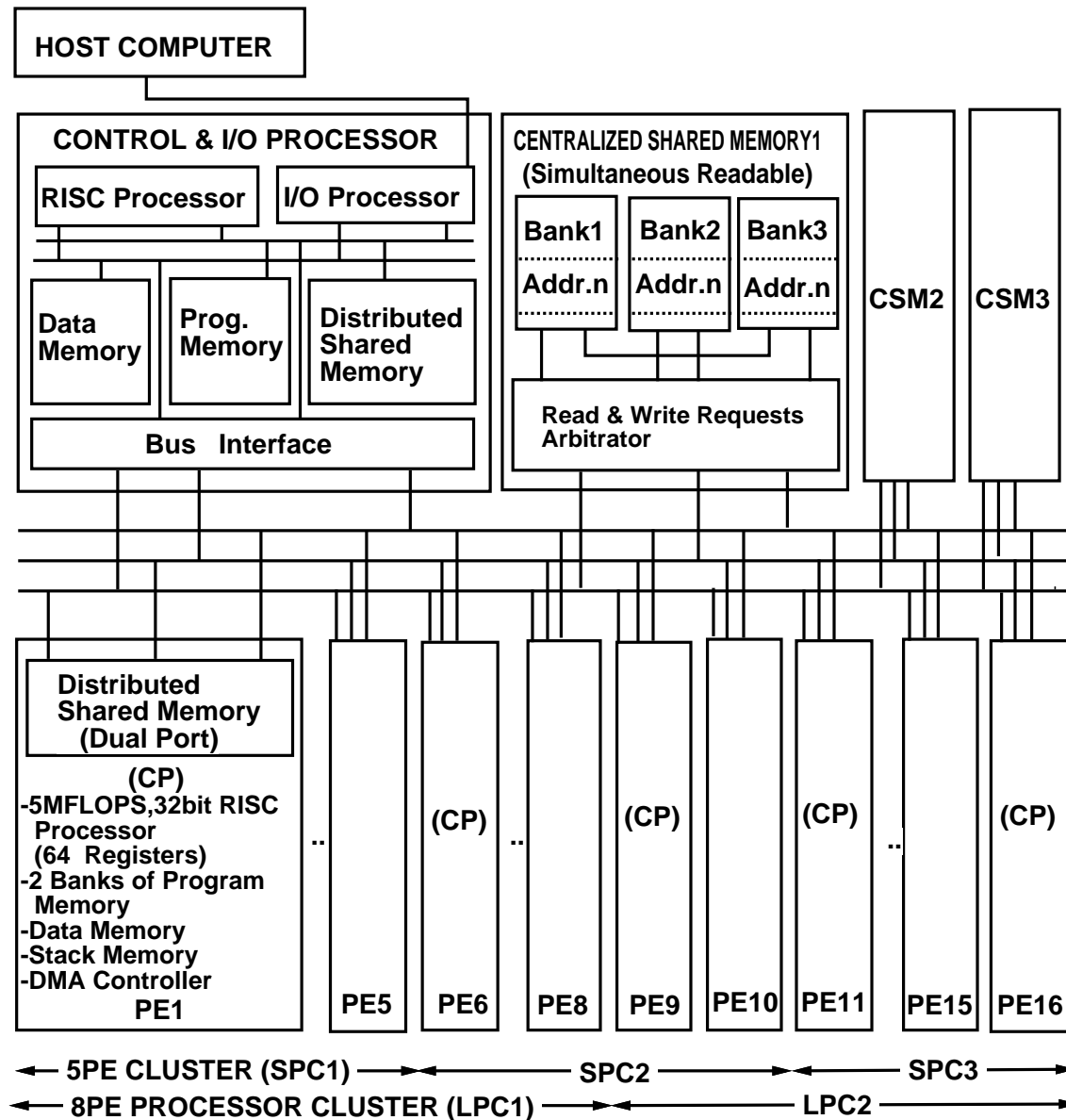
1987 OSCAR(Optimally Scheduled Advanced Multiprocessor)

Co-design of Compiler and Architecture

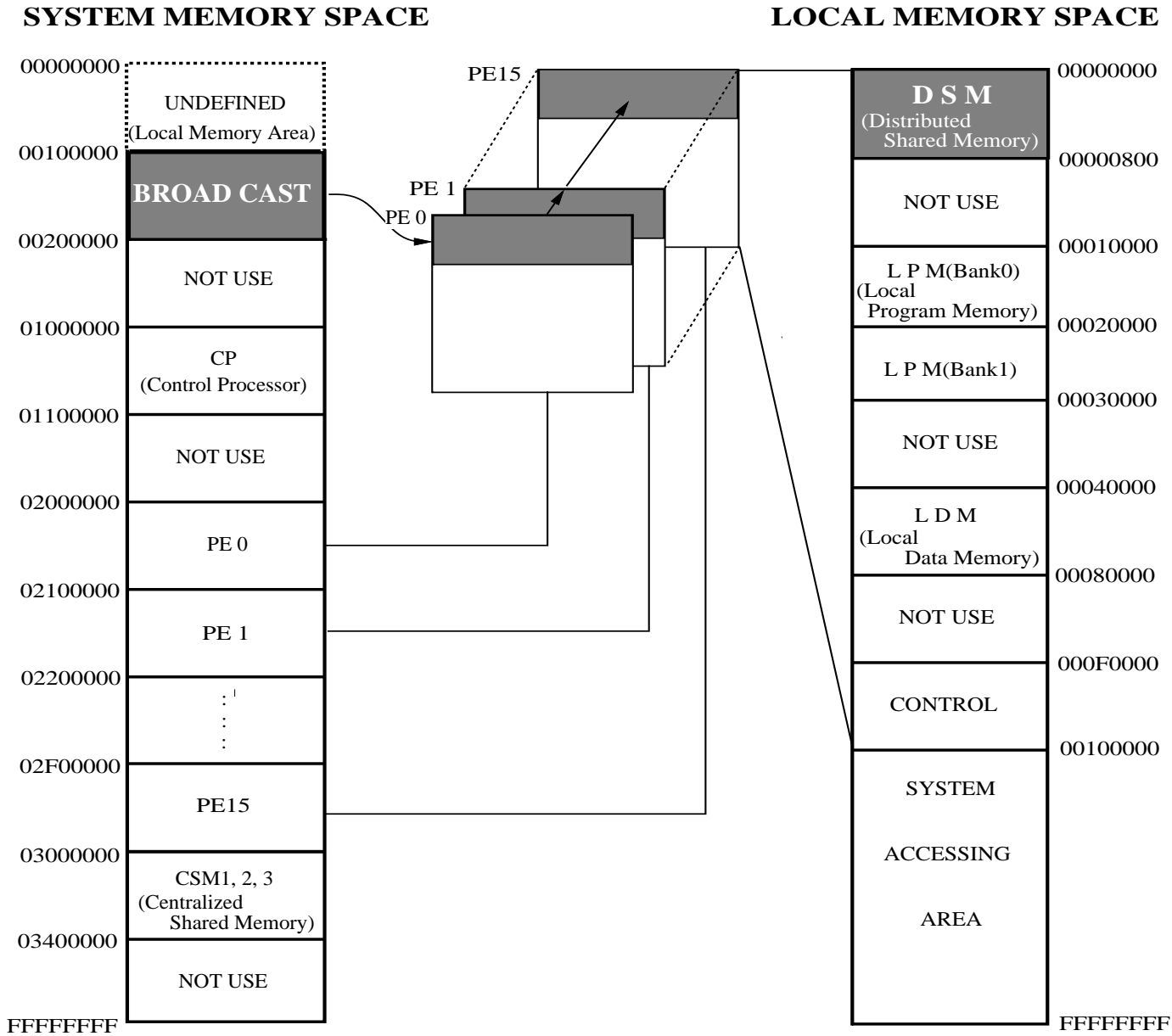
Looking at various applications, design a parallelizing compiler and design a multiprocessor/multicore-processor to support compiler optimization



OSCAR(Optimally Scheduled Advanced Multiprocessor)

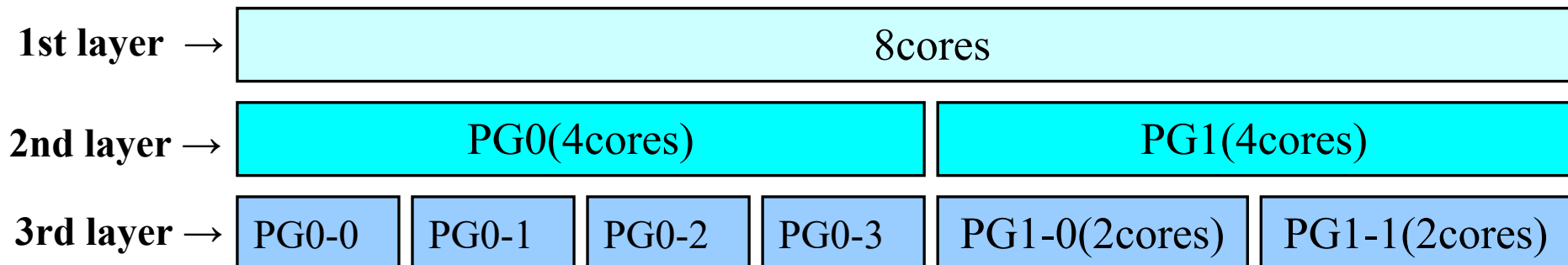


OSCAR Memory Space (Global Address Space)

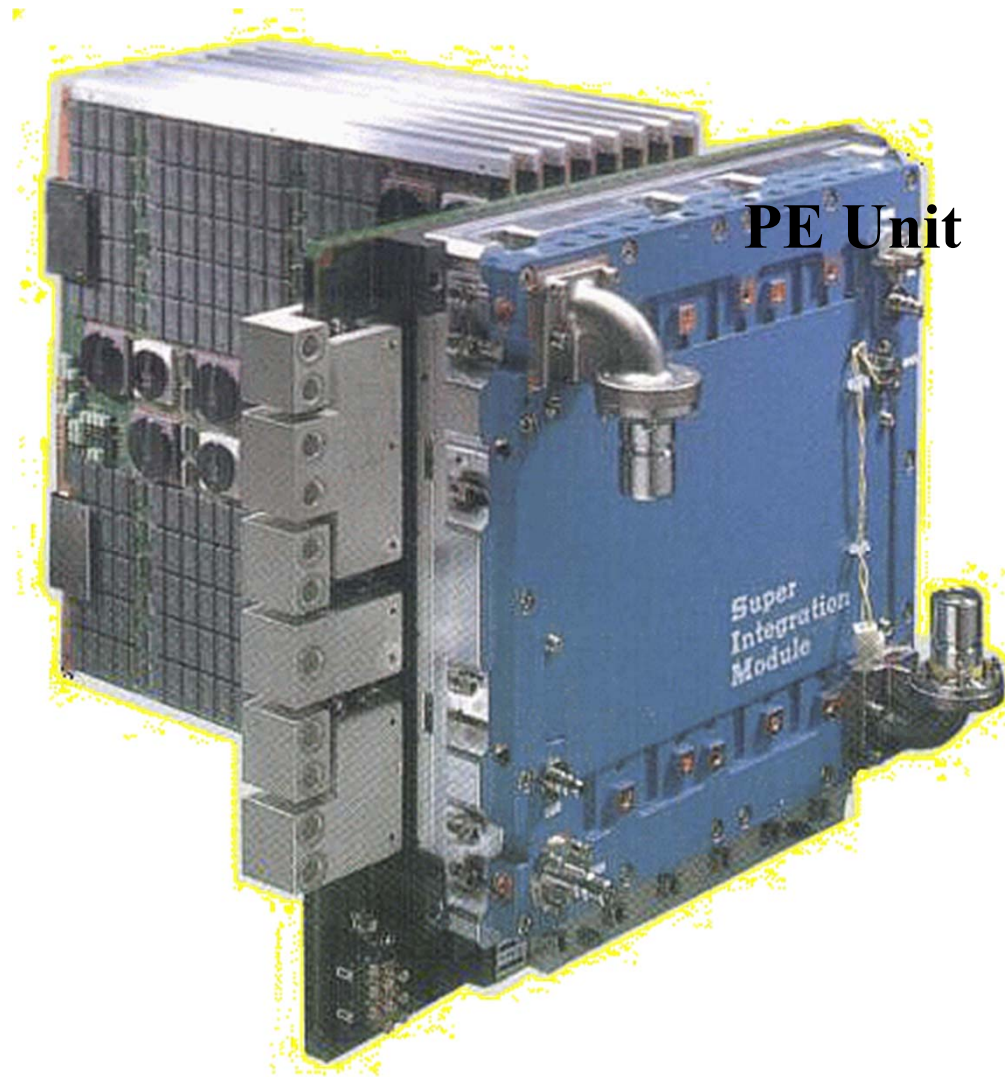


Hierarchical Barrier Synchronization

- **Specifying a hierarchical group barrier**
 - **#pragma oscar group_barrier (C)**
 - **!\$oscar group_barrier (Fortran)**



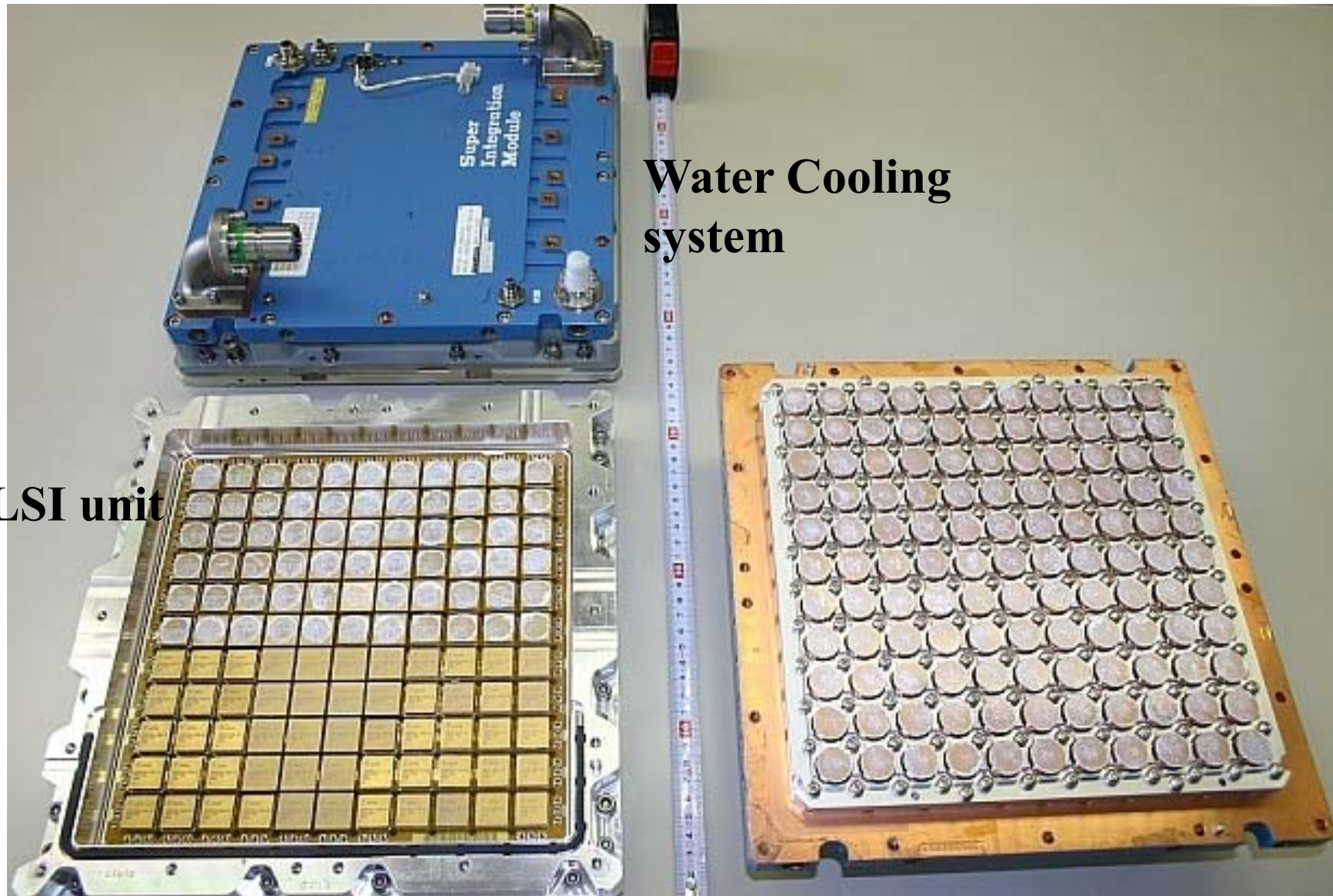
VPP500/NWT



PE Unit

LIMITED

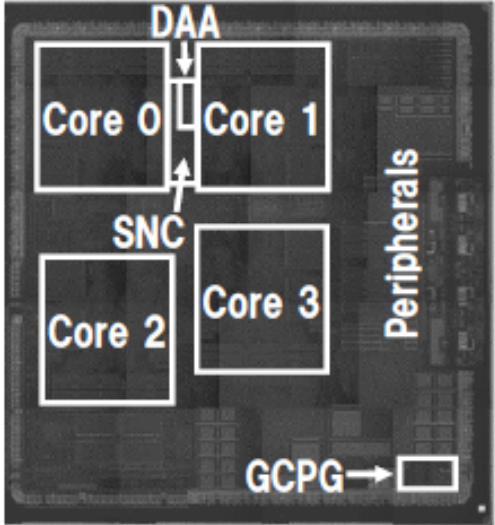
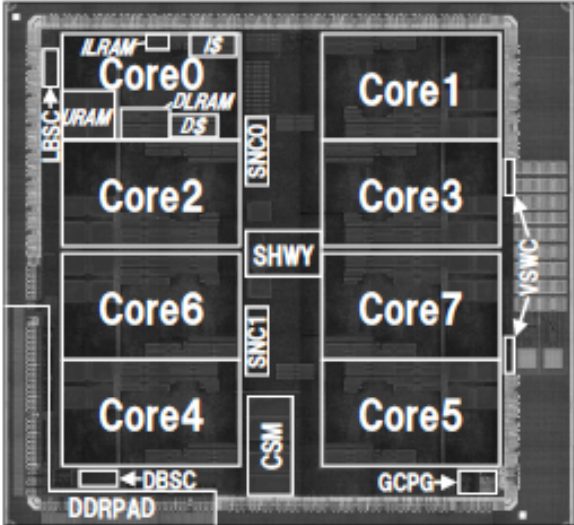
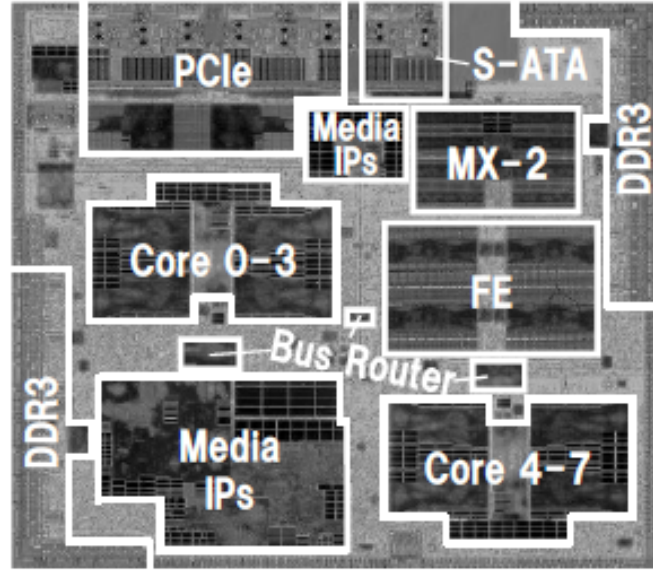
VPP500/NWT



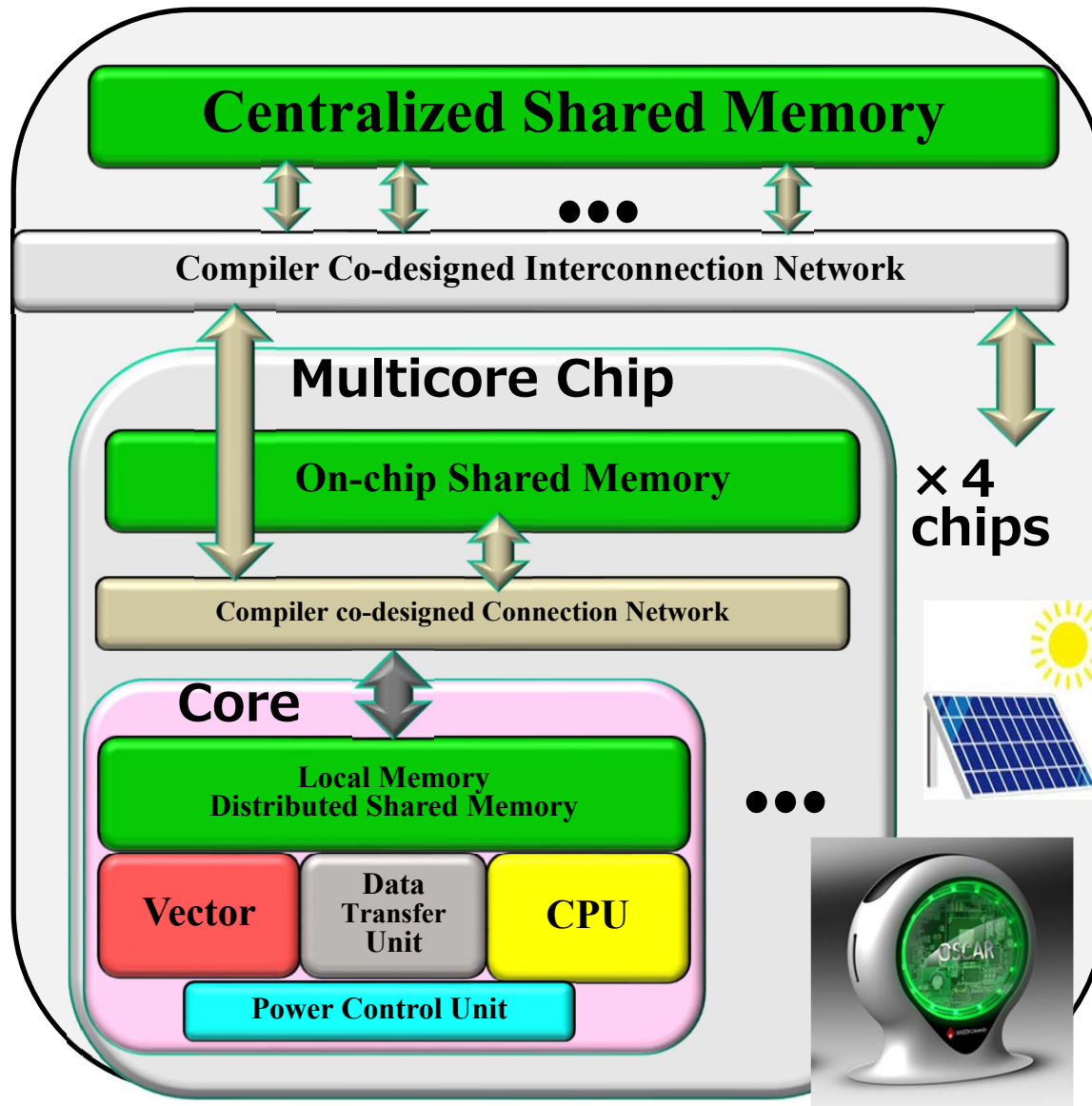
PE LSI unit

**Water Cooling
system**

4 core multicore RP1 (2007) , 8 core multicore RP2 (2008) and 15 core Heterogeneous multicore RPX (2010) developed in NEDO Projects with Hitachi and Renesas

RP-1 (ISSCC2007 #5.3)	RP-2 (ISSCC2008 #4.5)	RP-X (ISSCC2010 #5.3)
		
90nm, 8-layer, triple-Vth, CMOS	90nm, 8-layer, triple-Vth, CMOS	45nm, 8-layer, triple-Vth, CMOS
97.6 mm ² (9.88 x 9.88 mm)	104.8 mm ² (10.61 x 9.88 mm)	153.8 mm ² (12.4 x 12.4 mm)
1.0V (internal), 1.8/3.3V (I/O)	1.0-1.4V (internal), 1.8/3.3V (I/O)	1.0-1.2V (internal), 1.2-3.3V (I/O)
600MHz ,4.32 GIPS,16.8 GFLOPS	600MHz , 8.64 GIPS, 33.6 GFLOPS	648MHz, 13.7GIPS, 115GOPS, 36.2GFLOPS
11.4 GOPS/W (32b換算)	18.3 GOPS/W (32b換算)	37.3 GOPS/W (32b換算)

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



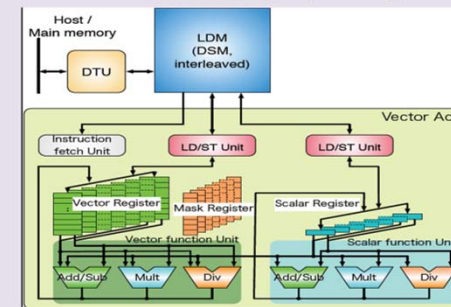
Target:

- 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

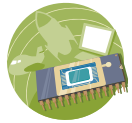


Function Units [tentative]

- **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



Future Multicore Products



Next Generation Automobiles

- Safer, more comfortable, energy efficient, environment friendly
- Cameras, radar, car2car communication, internet information integrated brake, steering, engine, motor control

Smart phones



- From everyday recharging to less than once a week
- Solar powered operation in emergency condition
- Keep health

Advanced medical systems



Cancer treatment, Drinkable inner camera

- Emergency solar powered
- No cooling fan, No dust, clean usable inside OP room



Personal / Regional Supercomputers



Solar powered with more than 100 times power efficient : FLOPS/W

- Regional Disaster Simulators saving lives from tornadoes, localized heavy rain, fires with earth quakes

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, 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 Tiler 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.**
- **Local memory management for automobiles and software coherent control have been patented and already realized by OSCAR compiler.**