Parallelising Compiler for Green Multicore Computing

Hironori Kasahara, Ph.D., IEEE Fellow, IPSJ Fellow
Professor, Dept. CS, 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

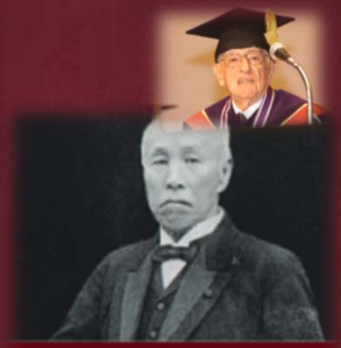
Reviewed Papers: 216, Invited Talks: 180, Granted Patents: 50 (Japan, US, GB, China), Articles in News Papers, Web News, Medias incl. TV etc.:

Committees in Societies and Government 260
IEEE Computer Society: President 2018, Executive Committee(2017-2019), BoG(2009-14), Strategic Planning Committee Chair 2018, Multicore STC Chair (2012-), Japan Chair(2005-07),
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.

早稲田大学

WASEDA University

Tokyo, Japan



1882 》

Okuma Shigenobu founded Tokyo Senmon Gakko (College)

The founding and opening ceremony of Tokyo Senmon Gakko (College) was held on October 21. At the ceremony, the Principal, Hikomaro Okuma, recited a text on the founding of the school, Aruma Ono delivered an address, and a declaration was made on the spirit of "Independence of Learning." The departments of political science, law, physical science, and English were established, and 80 students were admitted as the first batch of students.

The "Group of Four" who contributed to the development of Waseda University

The "Group of Four" refers to the four individuals who participated in the founding and management of Waseda University, and contributed to its development. Sanae Iokoi served as the first Principal and first President of the institution, and poured his efforts into raising the institution to the rank of university as well as establishing the school of science and engineering. Tomoyuki Aruma was the Director of the School of Commerce when it was first opened, and soon became the second Principal of the University. Shoyo Tsubouchi built the foundations for the present-day Department of Literature. Kenichi Ichijima worked hard to realize the economic independence of the university, and also contributed to the expansion of the library.

1903 》

Start of the Waseda-Keio baseball match (Sokeisen)

Along with the Cambridge-Oxford boat races and the Harvard-Yale football match, the Waseda-Keio baseball match (Sokeisen) is current among the three major university sporting events in the world. A tradition that can be traced back to 1903, its appeal lies in the baseball teams of the two universities battling to preserve the honor of their alma maters, as well as the grand scale of support offered from the stands.

《 1922

Visit by physicist Albert Einstein to Waseda University

On November 29, 1922, Professor Einstein visited Waseda University during his visit to Japan, and held a meeting with President Masanosuke Shinozawa who had once studied at Berlin University. At the welcome ceremony held on the central courtyard, more than 10,000 students and faculty welcomed Professor and Mrs. Einstein with enthusiastic applause. When they left, they were sent off with a chorus of the university's anthem.

Archaeological excavation of the Maikata site

In 1965, an archaeological team from Waseda University became the first Japanese people to launch an archaeological excavation mission at an ancient Egyptian site. In 1974, the team became the first in the history of archaeological excavations in Egypt to discover the "colossal statues" in Maikata. The team earned credibility with Egypt's Ministry of Antiquities, and was rated highly in Japan.

1928 》

Japan's first gold medalist

At the Amsterdam Olympics, Mikio Oda from Waseda University's track and field club became the first Japanese to win a gold medal for the triple jump. The same track and field team attended a sports meet for international students held in Paris on their way back to Japan from the Olympics, opening the path to participation in the Olympics later on.

1993 》

Visit to Waseda University by then U.S. President Bill Clinton

In 1993, Bill Clinton, then President of the United States of America, visited Waseda University. Thereafter, the university continued to welcome visits by many distinguished guests from around the world, including Hu Jintao, former President of the People's Republic of China in 2008, and former UN Secretary-General Ban Ki-moon in 2010.

《 1940

"Visas for life" from diplomat Chiune Sugihara

In 1918, Chiune Sugihara gained admission to the Department of English at Waseda University's Higher Normal School (the School of Education today). In 1940, Sugihara, who was then working at the Japanese Consulate in Lithuania, issued visas against orders from the Ministry of Foreign Affairs, thus saving about 6,000 Jews. His humanitarian act is highly appraised by the international community.

1956 》

The beginnings of the Ishibashi Cabinet, first alumnus of Waseda to become Prime Minister

In December 1956, Tanzan Ishibashi, former student of Waseda, was elected as President of the Liberal Democratic Party. In the nomination for the head of the government in both the upper and lower houses of the Diet held during the same month, Ishibashi defeated Mosaburo Suzuki, Chairman of the Socialist Party and also alumnus of Waseda, to become the Prime Minister of Japan. These are the origins of our first Prime Minister from Waseda.

2007 》

125th founding anniversary— Toward the "second establishment" of the university

Waseda University has continued to move forward toward its three goals of tackling the challenge of innovative advanced research, realizing lifelong learning across the institution, and fostering global citizens. At the ceremony held on October 21, 2007 to commemorate its 125th anniversary, 15th President Katsuhiko Shiina delivered the "Second Century Declaration of Waseda."

《 1962

Robert Kennedy attends student debate

In the midst of the protest against the Japan-U. S. Security Treaty in 1960, then U.S. Attorney General Robert Kennedy and his wife attended a student debate at the Okuma Auditorium. The couple was touched by the joint singing of the university's anthem by groups that were both for and against the Treaty. When they visited Japan again, they

《 2012

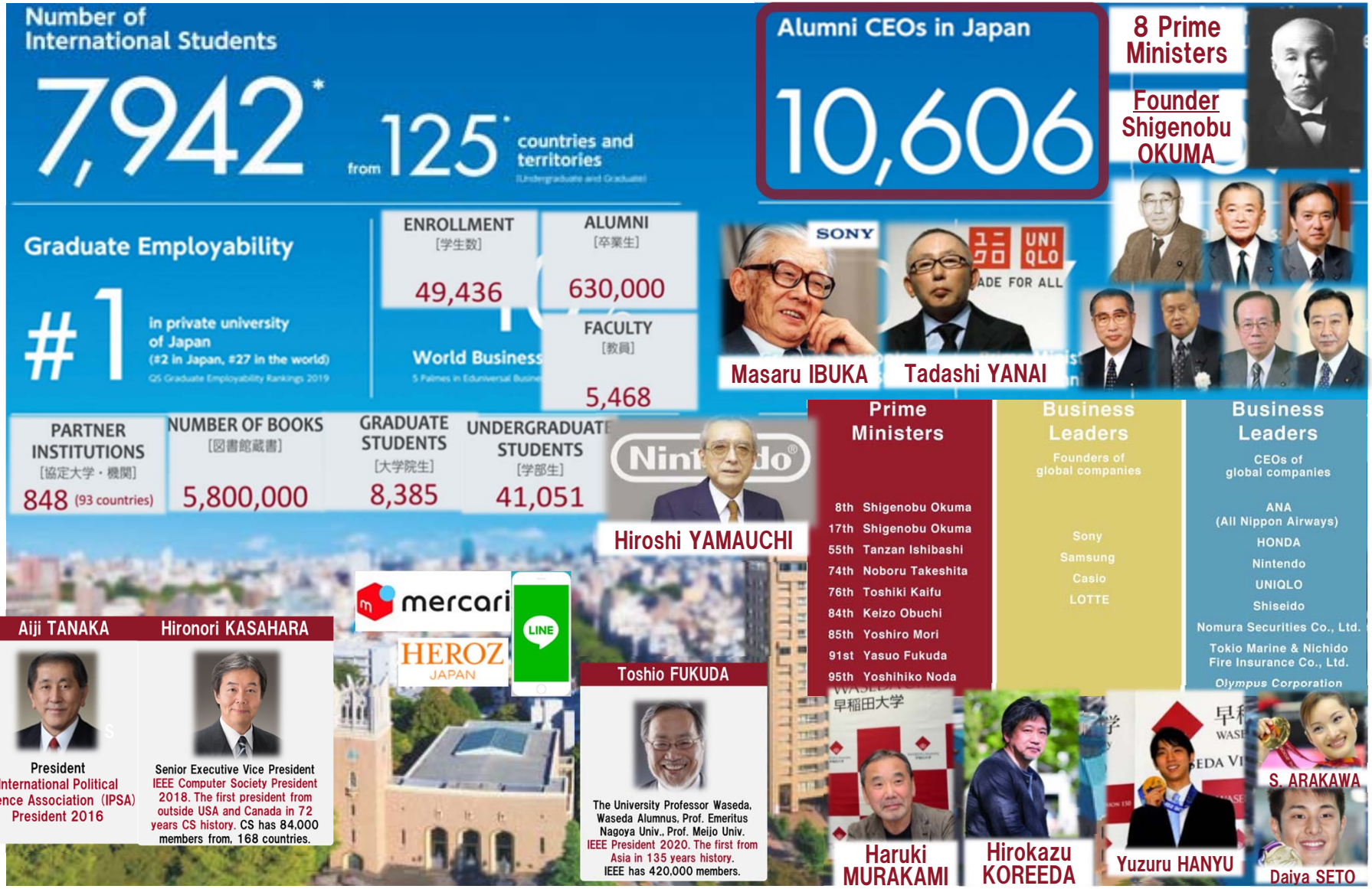
Formulation of Waseda Vision 150

Waseda Vision 150 was formulated in 2012 with a view to the 150th anniversary of the university's founding in 2032. Waseda University has dramatically improved the quality of education and research, and will continue to contribute to the world as a

confidential

WASEDA UNIVERSITY

About WASEDA - 早稲田大学



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

IEEE COMPUTER SOCIETY

IEEE

IEEE CS Awards Ceremonies with CS President 2018

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

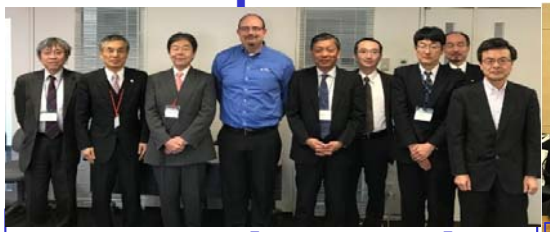
B. Ramakrishna Rau Award in MICRO, Fukuoka

Technical Achievement Award, in COMPSAC, Tokyo

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

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

Cooperation with International Organizations in 2018



IPSJ Leaders, March, IPSJ Convention, Tokyo

Japan (IPSJ), China(CCF), Korea(KIISE) in March, Waseda U., Tokyo

Okawa Foundation, CS Japan Chapter, Multicore STC & Japanese Government Symp.

MoU with UN ITU in AI for Good, May, Geneva

CCF China National Computer Congress, Oct. , Hangzhou

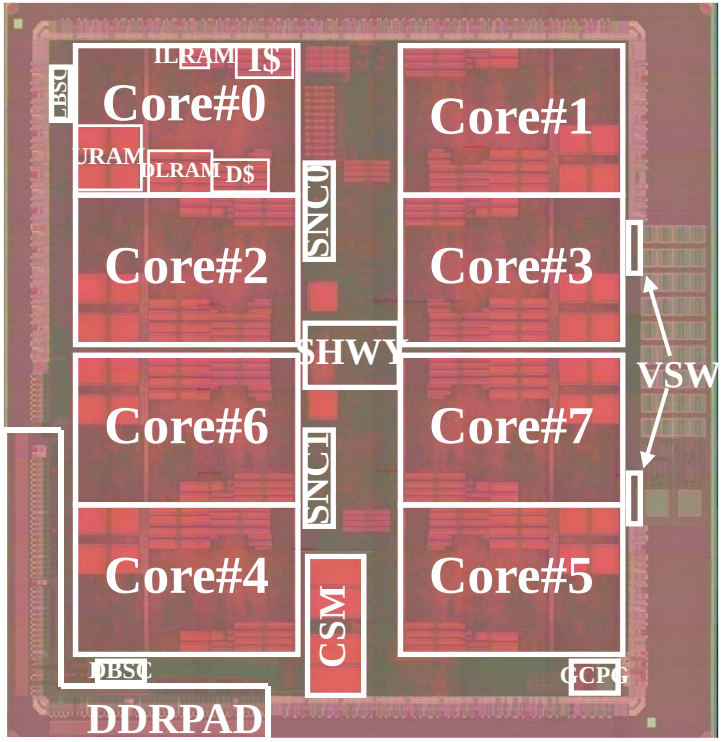
MoU with Baidu, July, Green Comp. C., Tokyo

IEEE CS China Office moderated Tencent-Waseda Univ. Joint Symposium, Nov., Waseda U., Tokyo

Russian Academy of Science: Russian Computer Science 70th Anniversary, Nov., Moscow

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 ∝ Frequency * Voltage^2

(Voltage ∝ Frequency)

⇒ Power ∝ Frequency^3

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

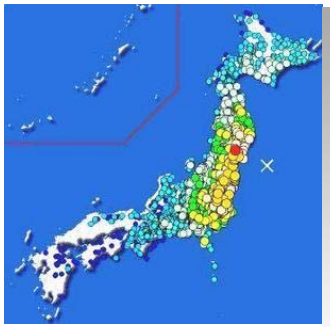
<Multicores>

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

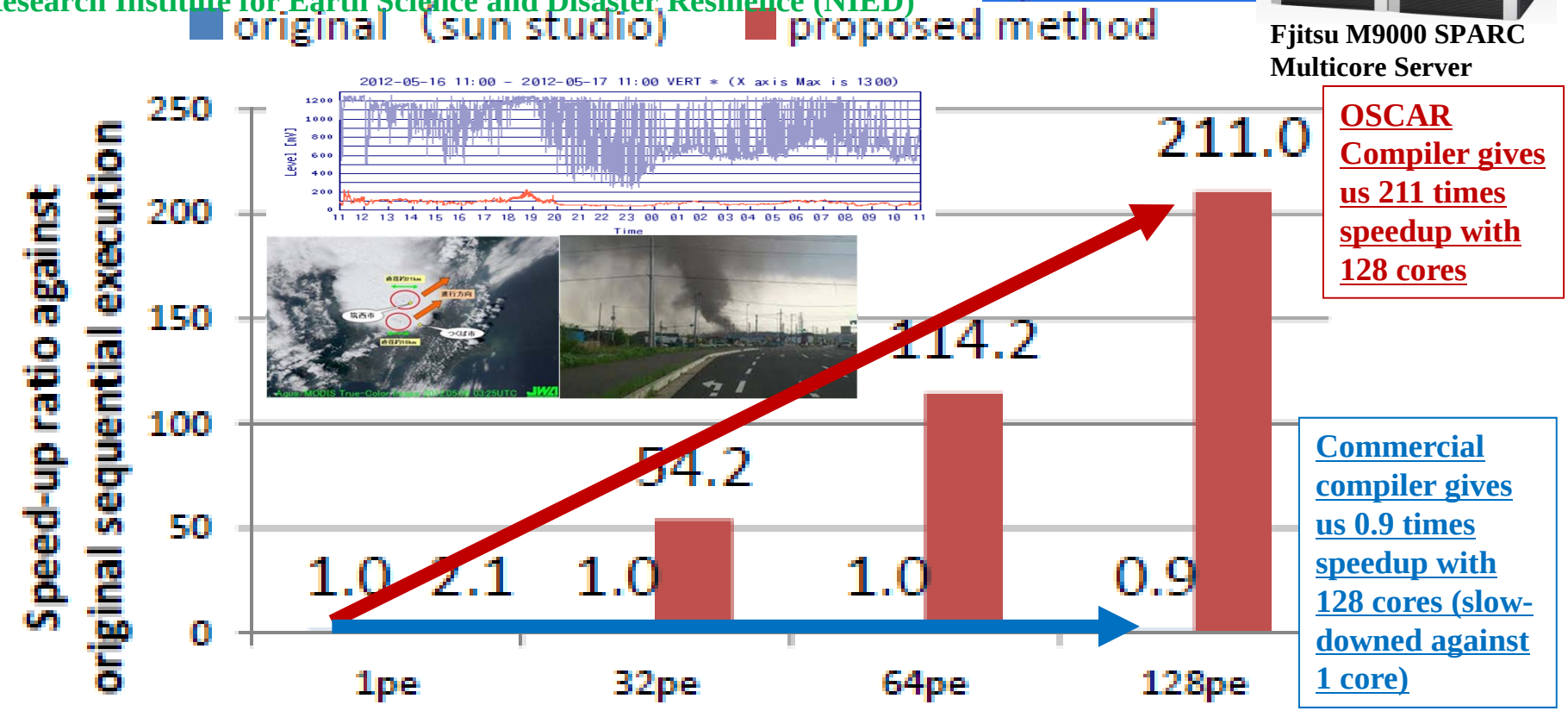
Parallel Soft is important for scalable performance of multicore (LCPC2015)

- 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)



Fjitsu M9000 SPARC Multicore Server



OSCAR Compiler gives us 211 times speedup with 128 cores

Commercial compiler gives us 0.9 times speedup with 128 cores (slow-downed against 1 core)

- 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

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)

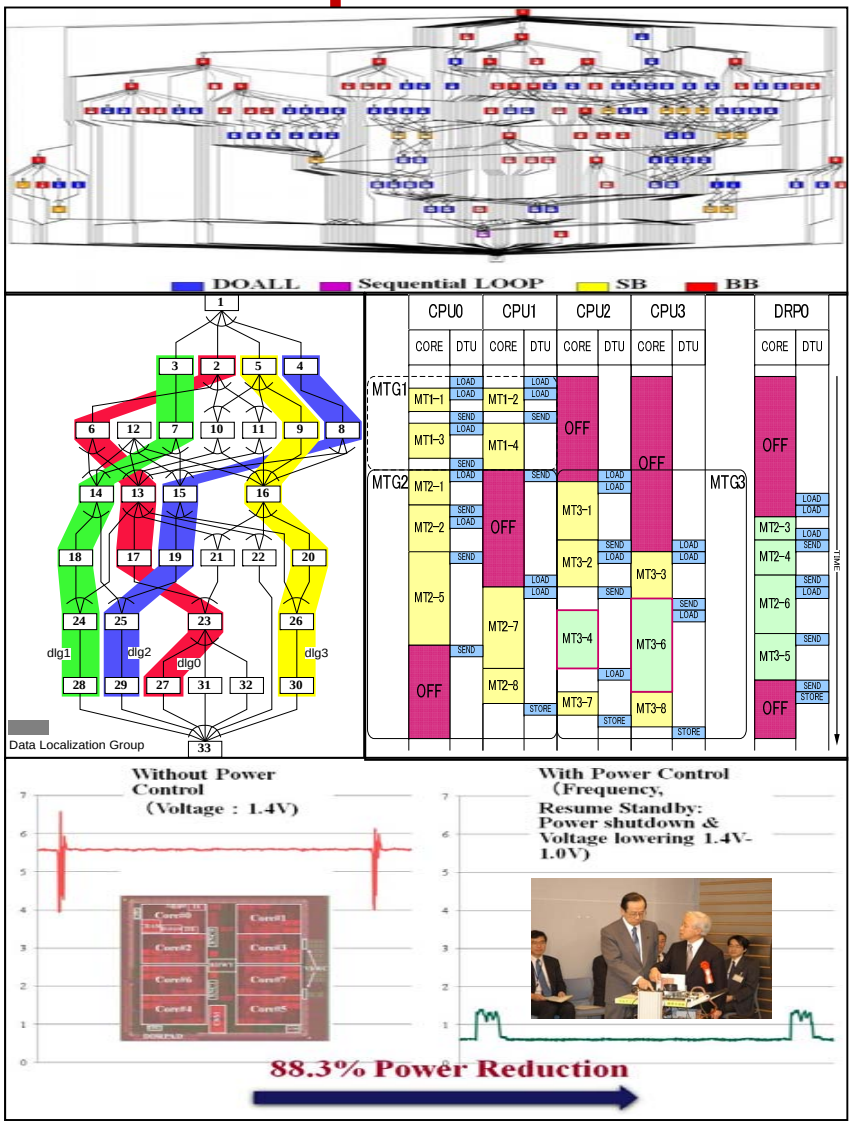
Data Transfer Overlapping(2016 partially)

Data transfer overlapping using Data Transfer Controllers (DMAs)

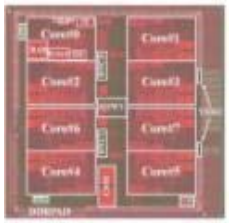
Power Reduction

(2005 for Multicore, 2011 Multi-processes, 2013 on ARM)

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

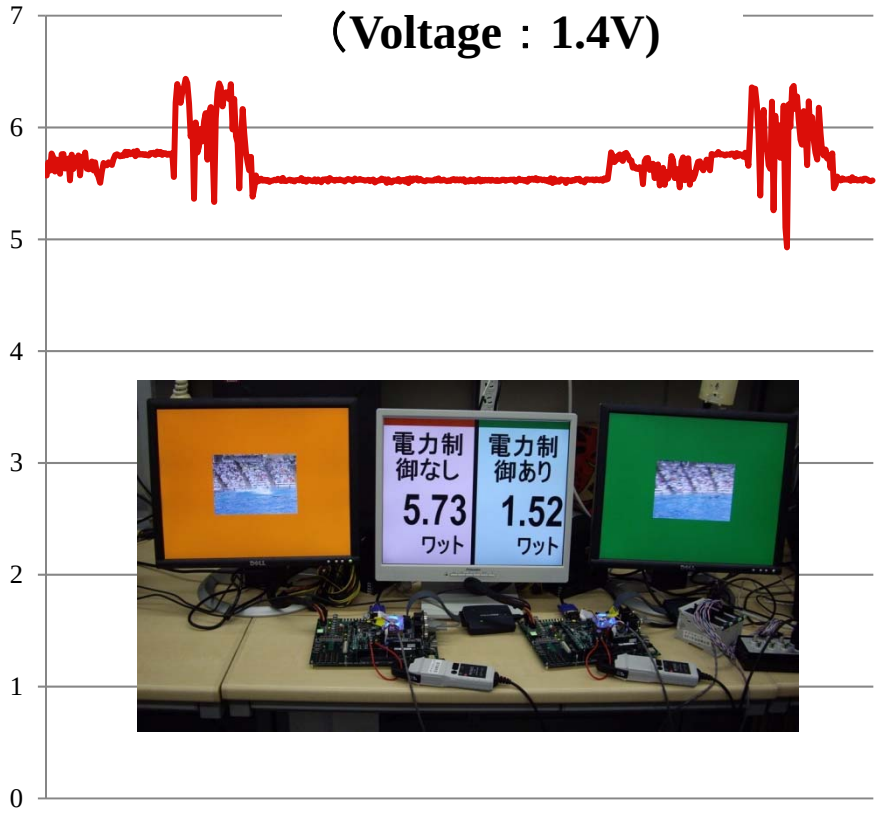


Power Reduction of MPEG2 Decoding to 1/4 on 8 Core Homogeneous Multicore RP-2 by OSCAR Parallelizing Compiler

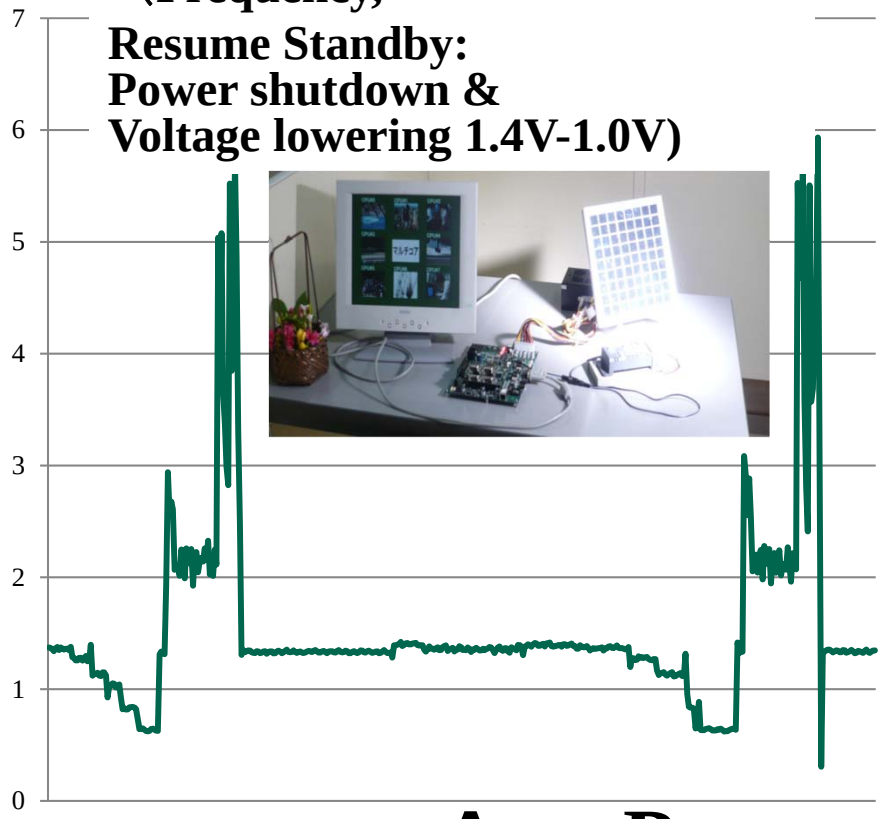


MPEG2 Decoding with 8 CPU cores

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 Green Multicore Processor for Real Time Consumer Electronics at Council of Science and Engineering Policy on April 10, 2008

http://www8.cao.go.jp/cstp/gaiyo/honkaigi/74index.html

第74回総合科学技術会議【平成20年4月10日】

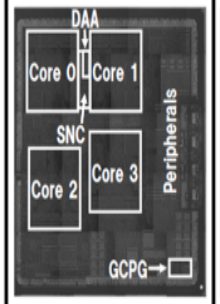
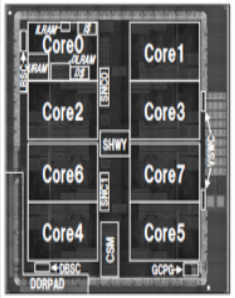
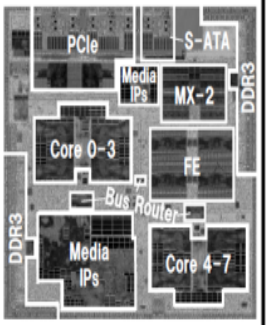
第74回総合科学技術会議の様子(1)

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)

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)

Prime Minister FUKUDA is touching our multicore chip during execution.

Green Computing Systems R&D Center

Waseda University

Established by Prof. Kasahara supported by METI (Mar. 2011)

＜R & D Target＞

Hardware, Software, Application for Super Low-Power Manycore

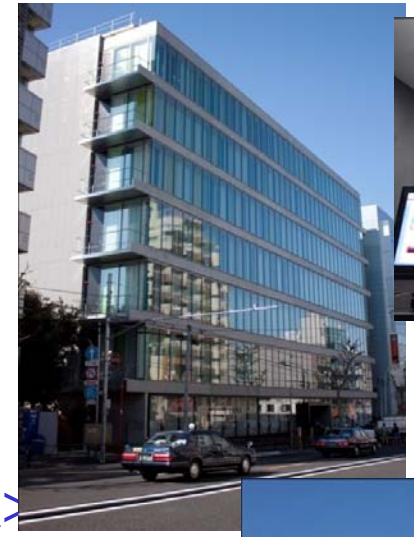
- **More than 64 cores**
- **Natural air cooling (No fan)**
  Cool, Compact, Clear, Quiet
- **Operational by Solar Panel**

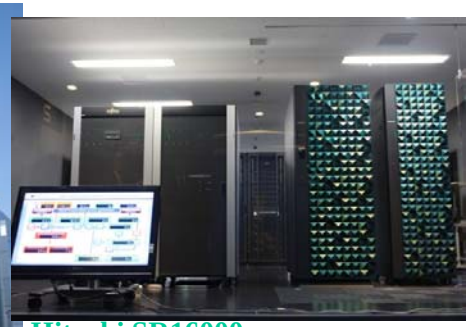
<Industry, Government, Academia>

Hitachi, Fujitsu, NEC, Renesas, Olympus, Toyota, Denso, Mitsubishi, Toshiba, OSCAR Technology, etc

＜Ripple Effect＞

- **Low CO_2 (Carbon Dioxide) Emissions**
- **Creation Value Added Products**
- **Automobiles, Medical, IoT, Servers**



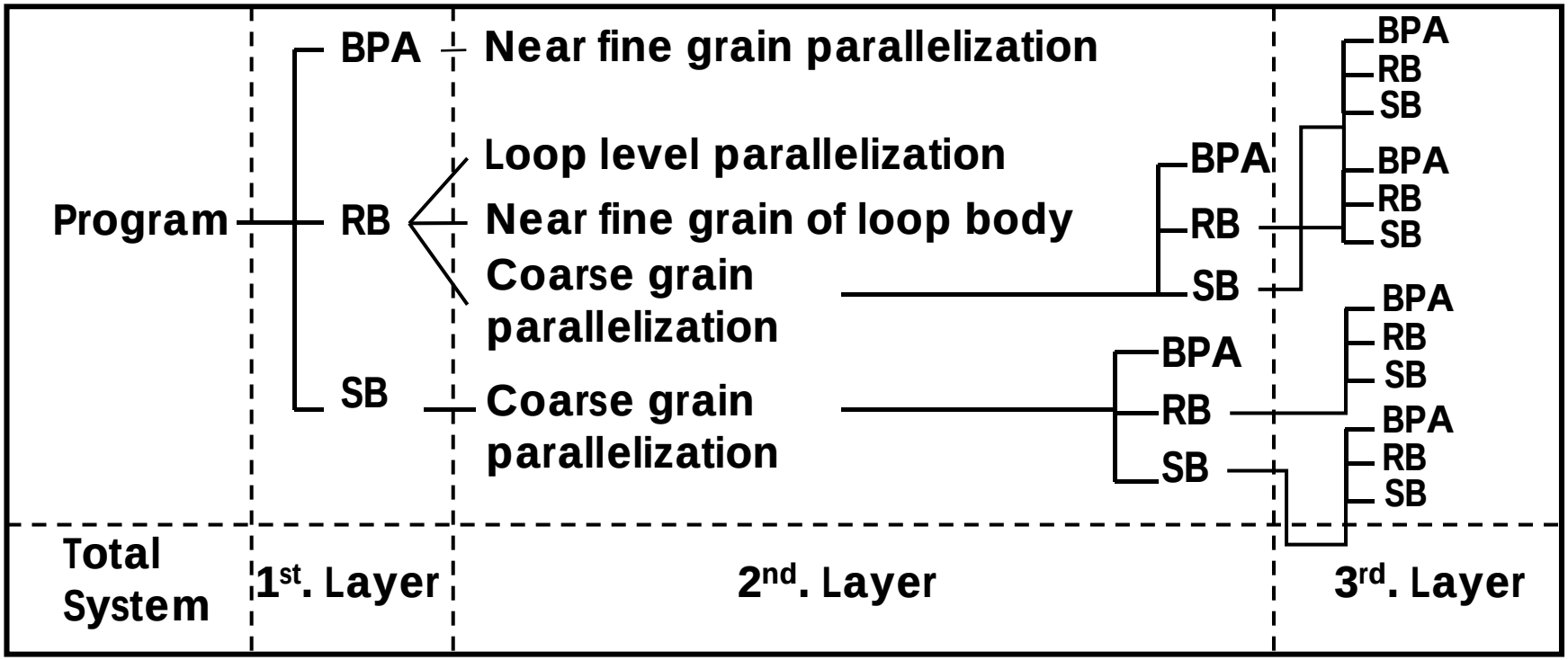


Hitachi SR16000:
Power7 128coreSMP
Fujitsu M9000
SPARC VII 256 core SMP

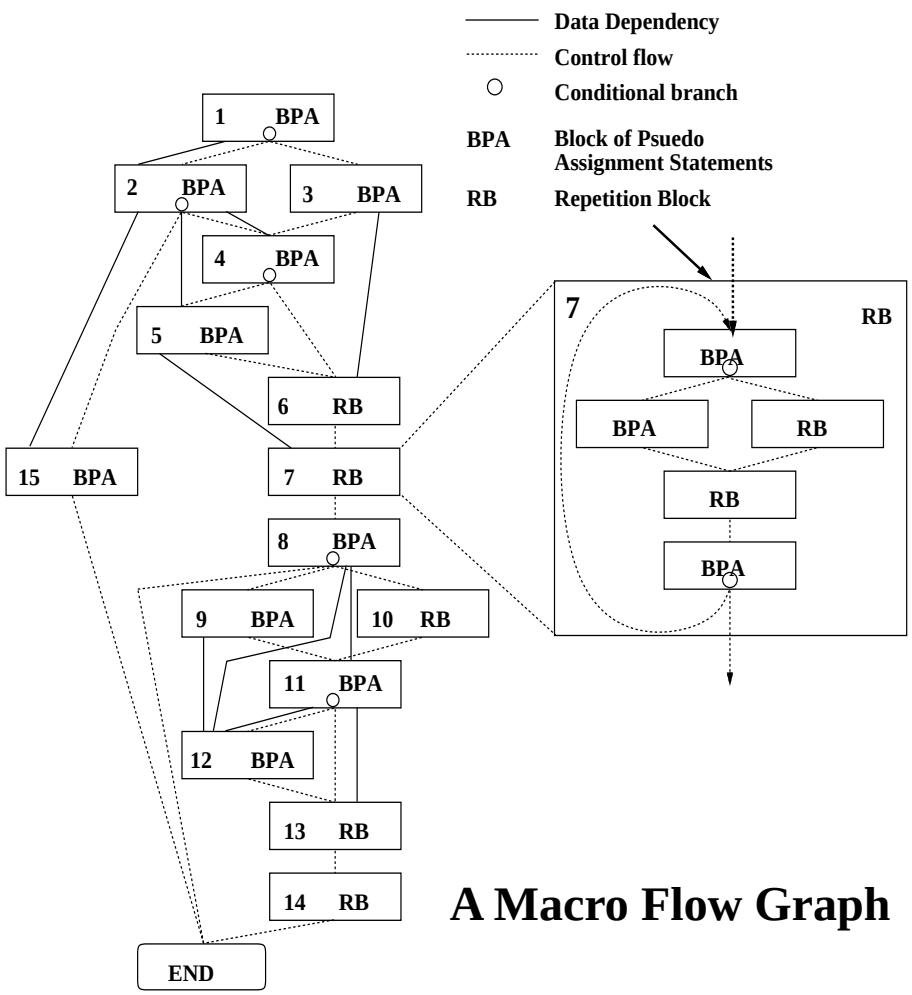
Beside Subway Waseda Station, Near Waseda Univ. Main Campus

Generation of Coarse Grain Tasks

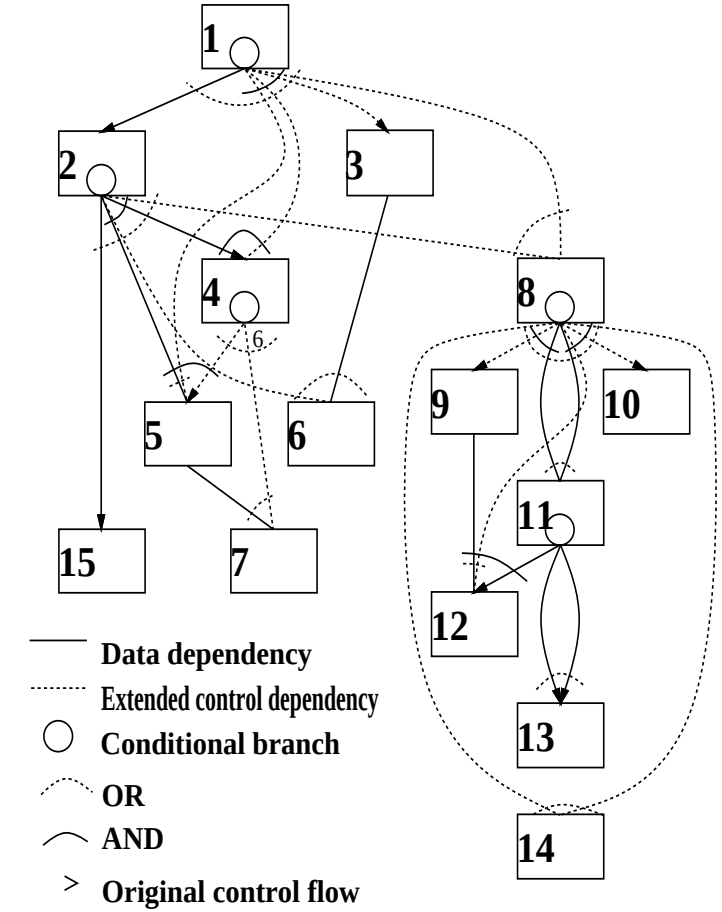
- **Macro-tasks (MTs)**
  - **Block of Pseudo Assignments (BPA): Basic Block (BB)**
  - **Repetition Block (RB) : natural loop**
  - **Subroutine Block (SB): subroutine**



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

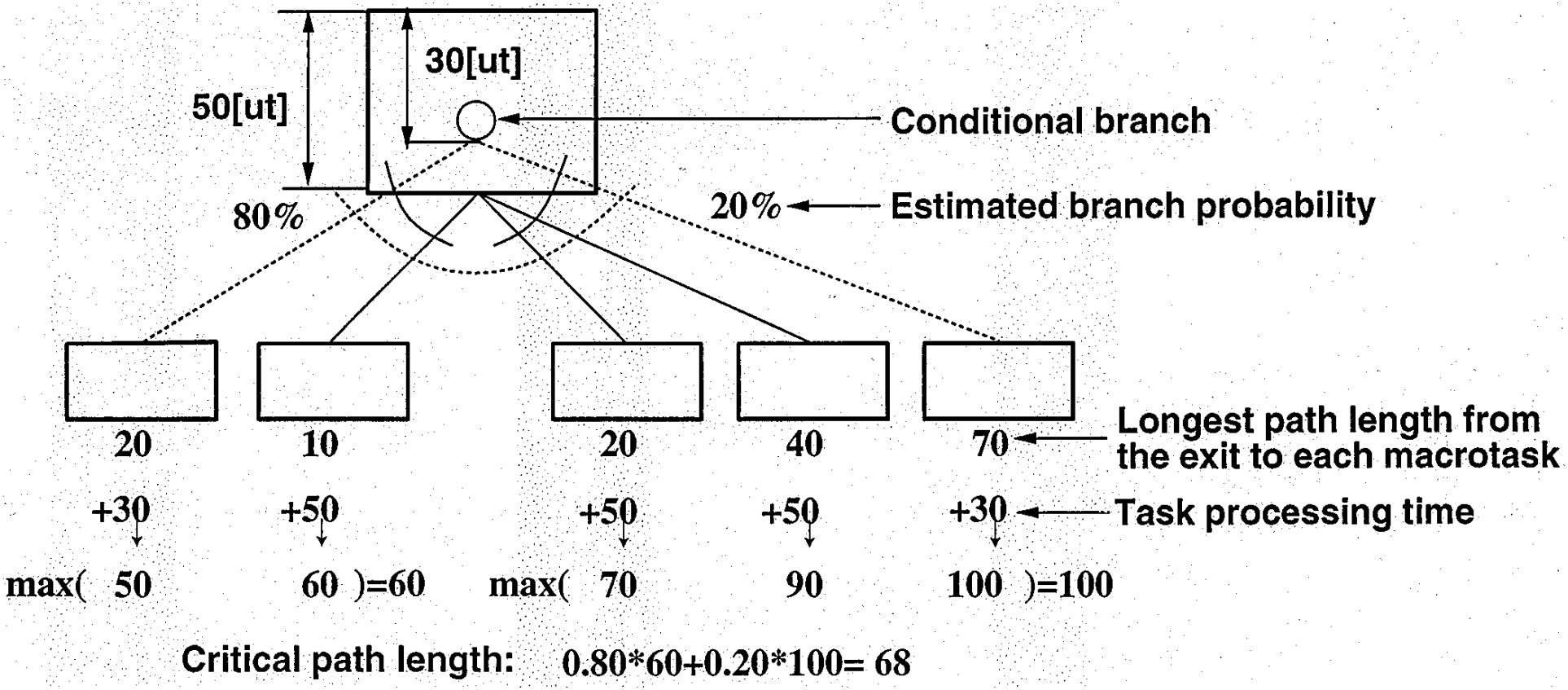


A Macro Flow Graph



A Macro Task Graph

PRIORITY DETERMINATION IN DYNAMIC CP METHOD



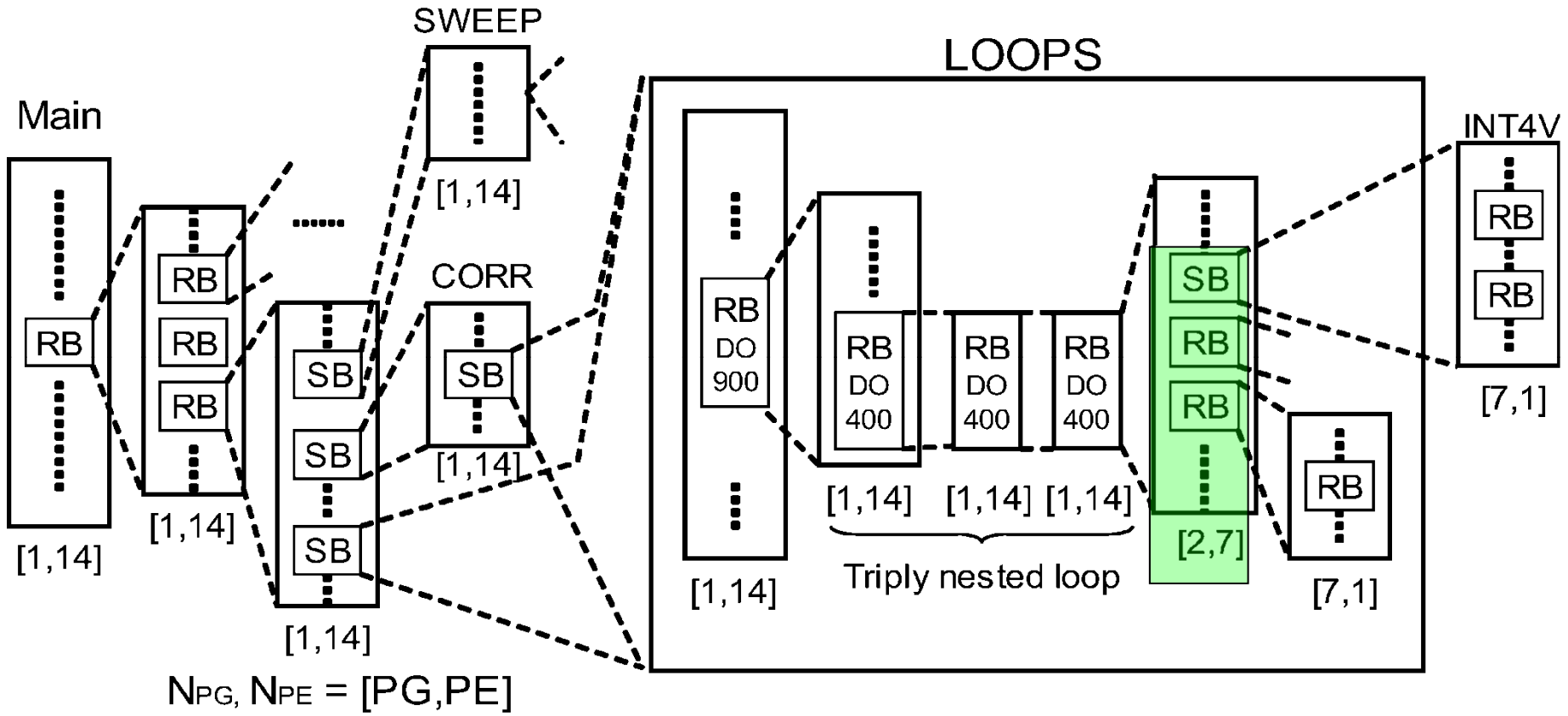
Earliest Executable Conditions

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

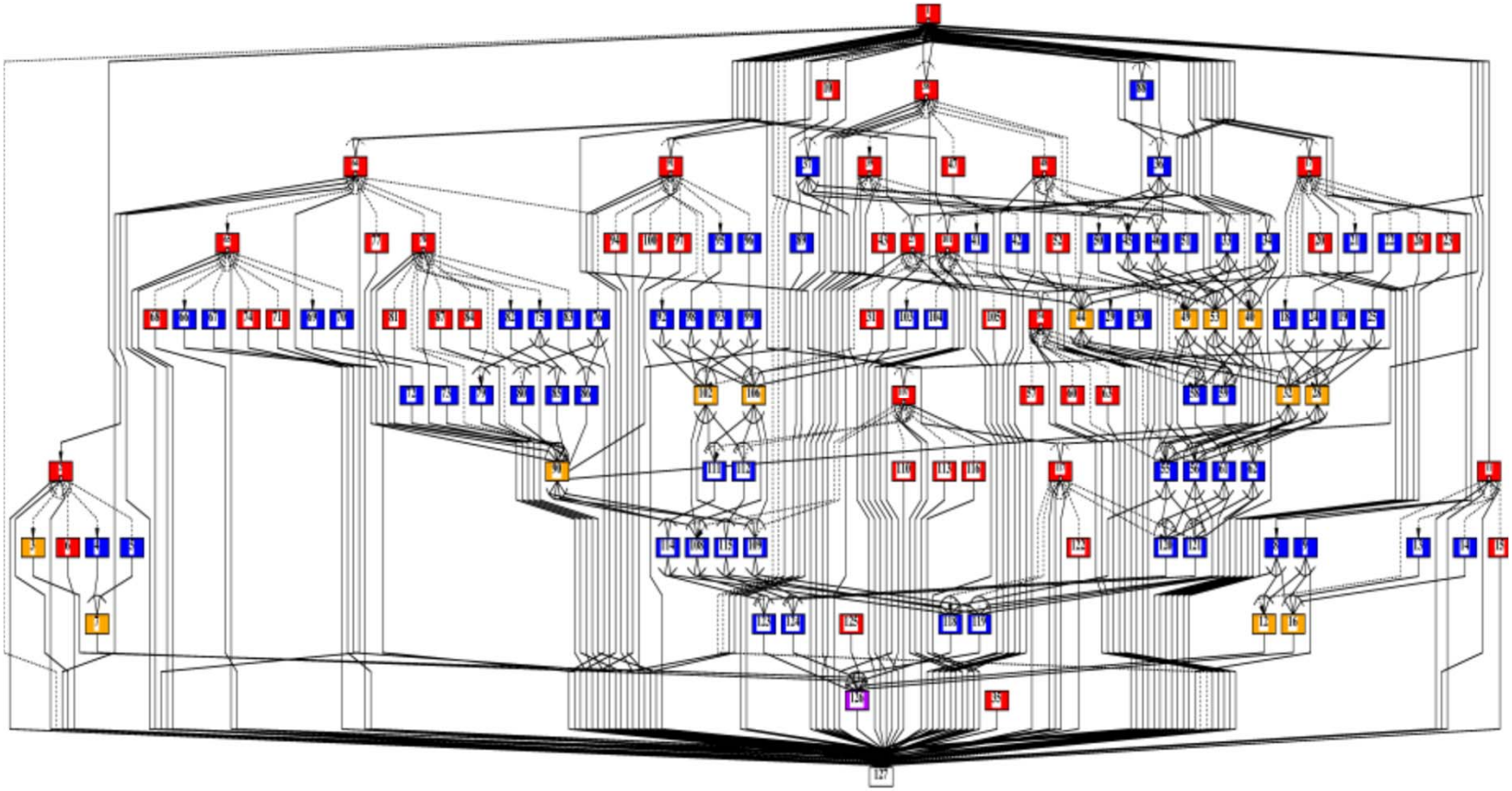
- **Using 14 processors**

Coarse grain parallelization within DO400



MTG of Su2cor-LOOPS-DO400

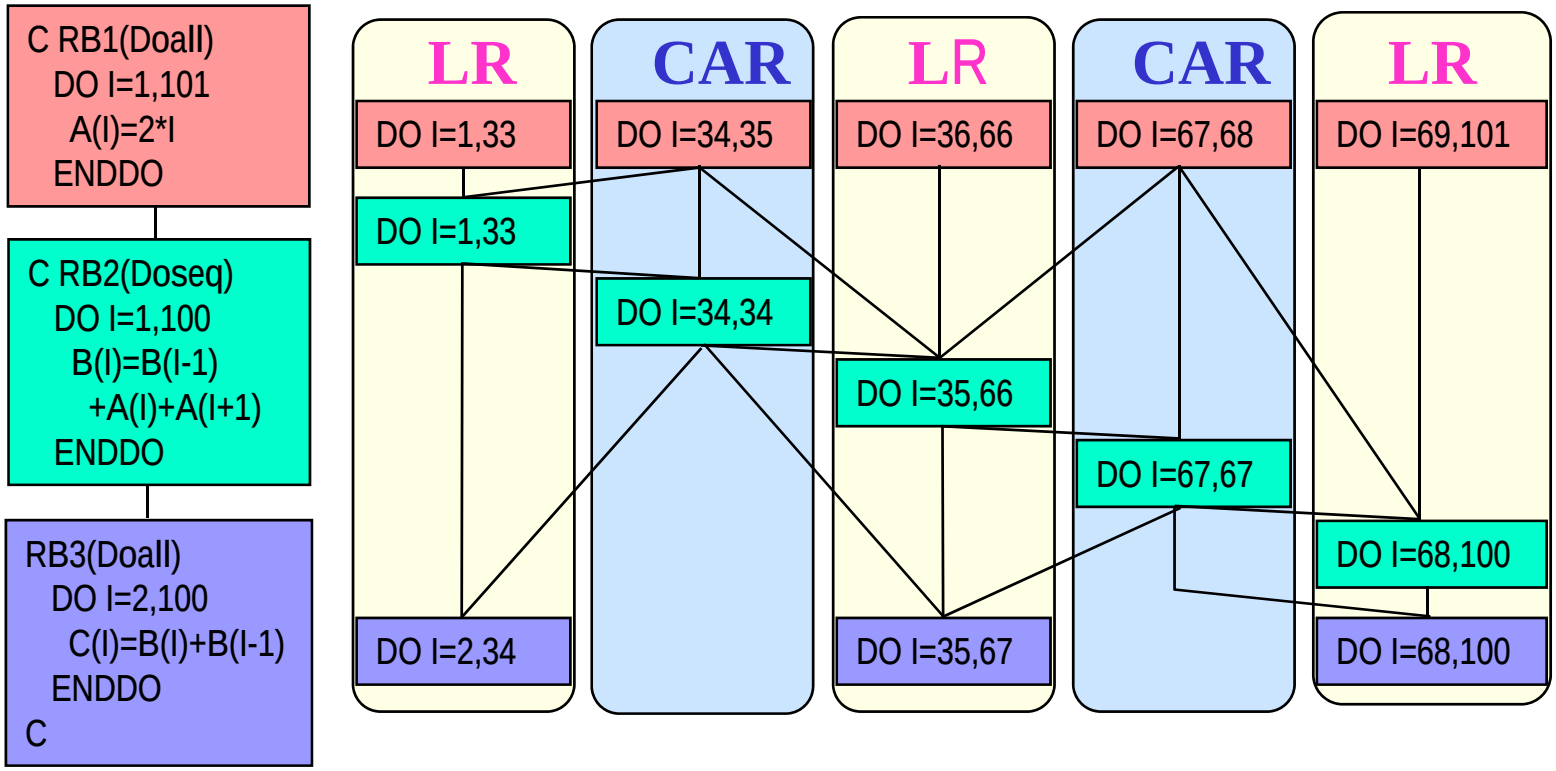
- **Coarse grain parallelism PARA_ALD = 4.3**



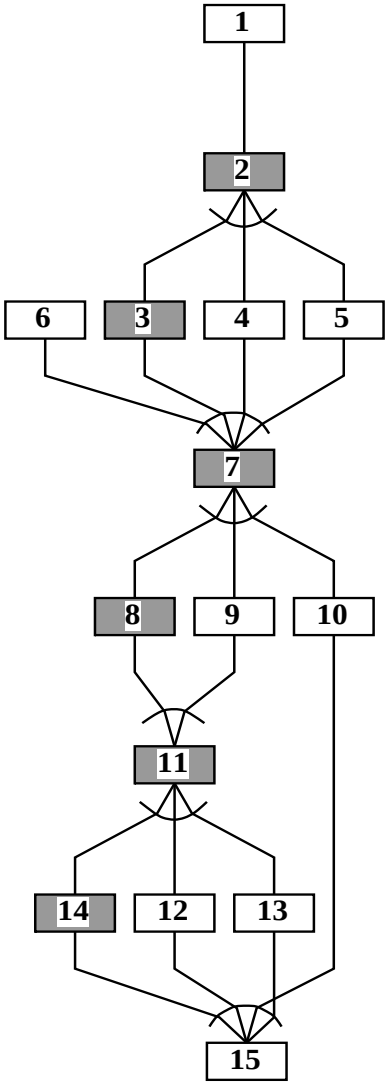
DOALL **Sequential LOOP** **SB** **BB**

Data-Localization: Loop Aligned Decomposition

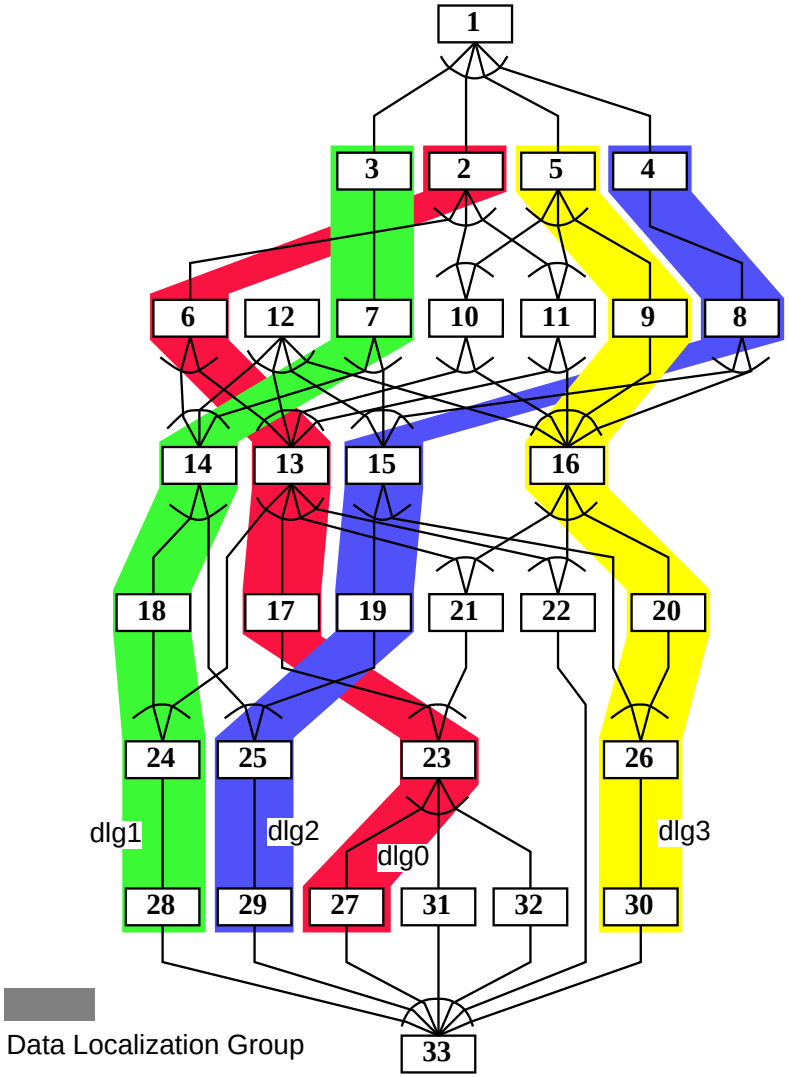
- **Decompose multiple loop (Doall and Seq) into CARs and LRs considering inter-loop data dependence.**
  - **Most data in LR can be passed through LM.**
  - **LR: Localizable Region, CAR: Commonly Accessed Region**



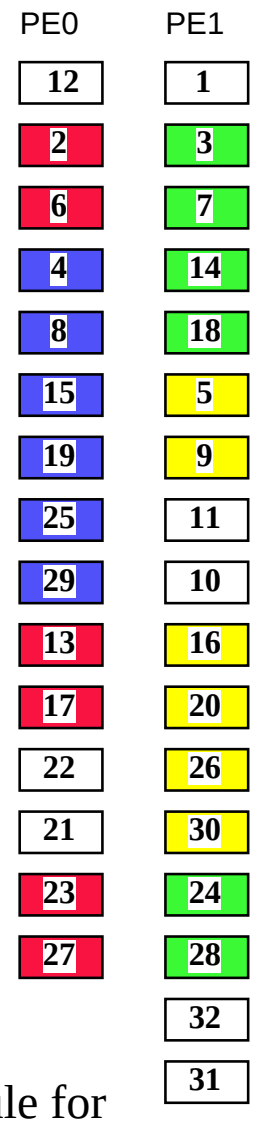
Data Localization



MTG



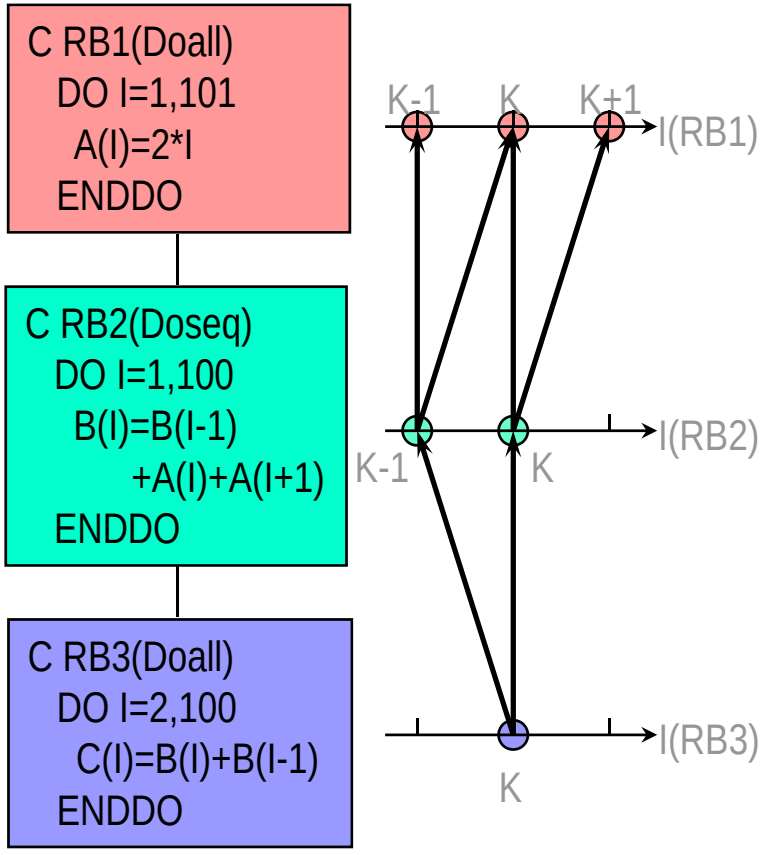
MTG after Division



A schedule for two processors

Inter-loop data dependence analysis in TLG

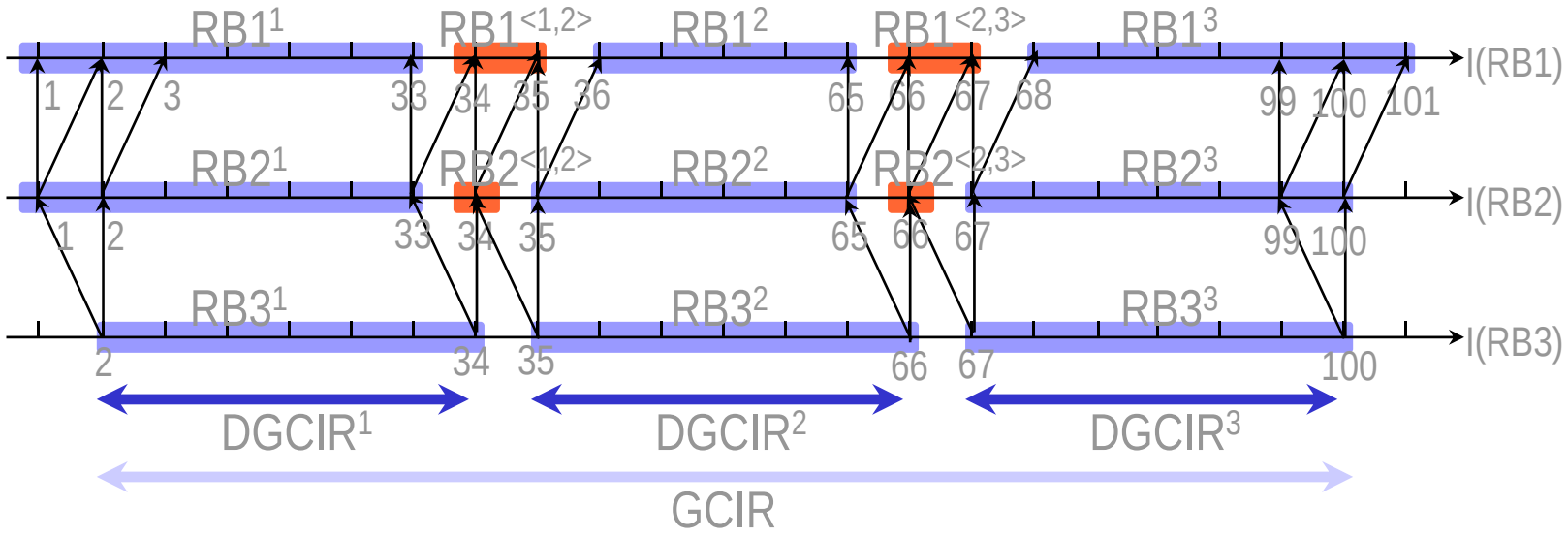
- Define exit-RB in TLG as Standard-Loop
- Find iterations on which a iteration of Standard-Loop is data dependent
  - e.g. K_{th} of RB3 is data-dep on $K\text{-}1_{th}$,K_{th} of RB2, on $K\text{-}1_{th}$,K_{th},$K\text{+}1_{th}$ of RB1



Example of TLG

Decomposition of RBs in TLG

- Decompose GCIR into $DGCIR^p(1 \leqq p \leqq n)$
  - 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.



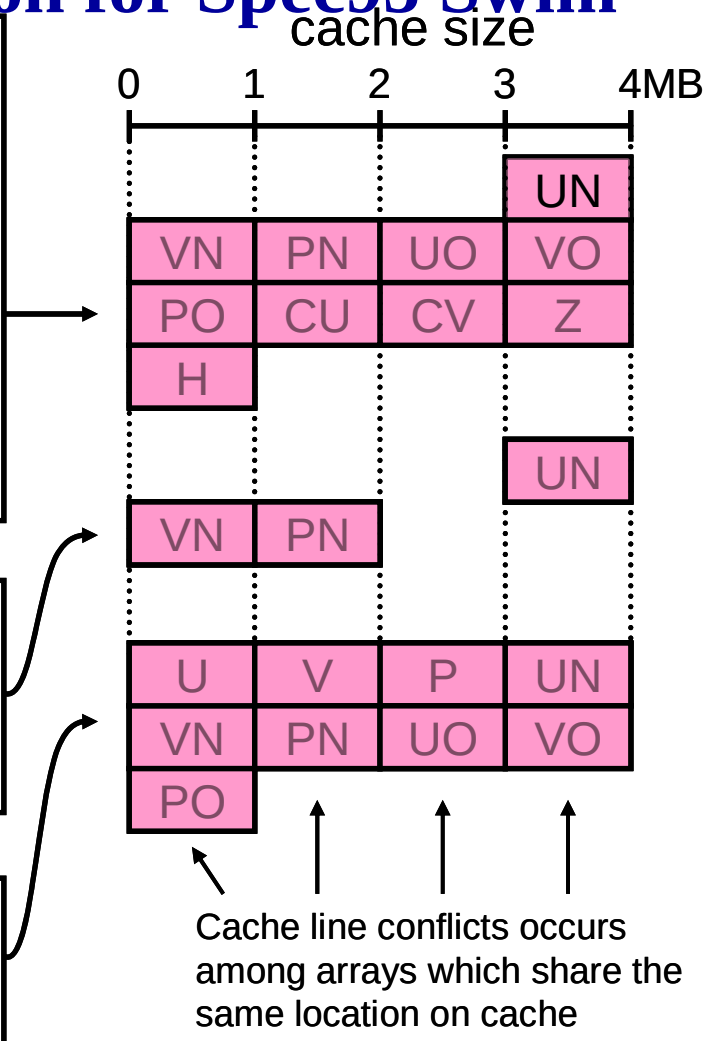
An Example of Data Localization for Spec95 Swim

```
      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)-TDTS8*(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

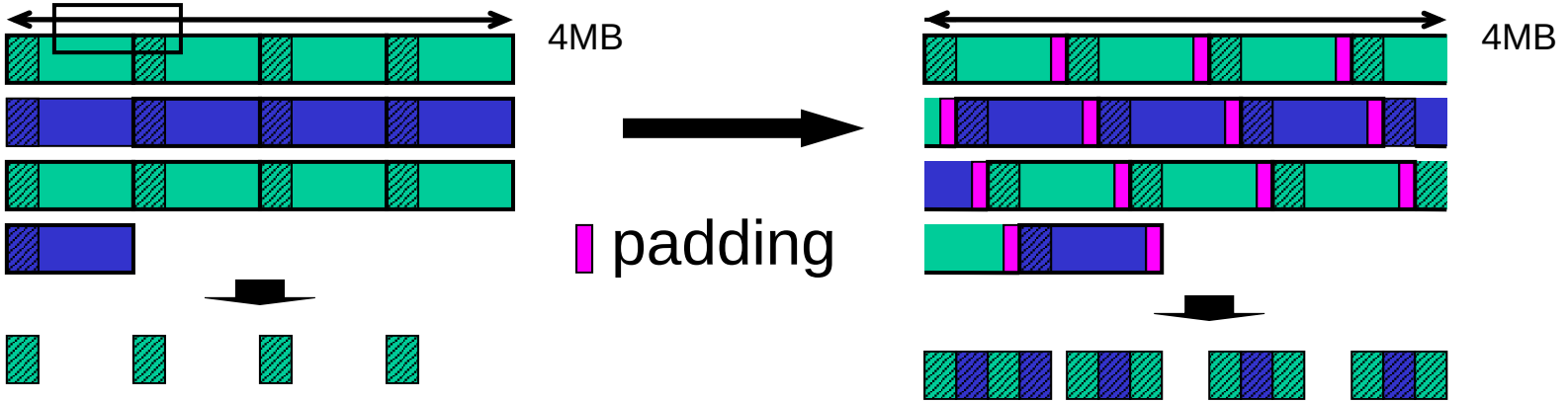
Declaration part of arrays in spec95 swim

before padding

```
PARAMETER (N1=513, N2=513)

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

after padding

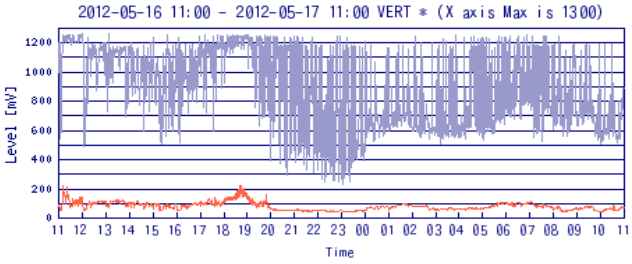
```
PARAMETER (N1=513, N2=544)

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

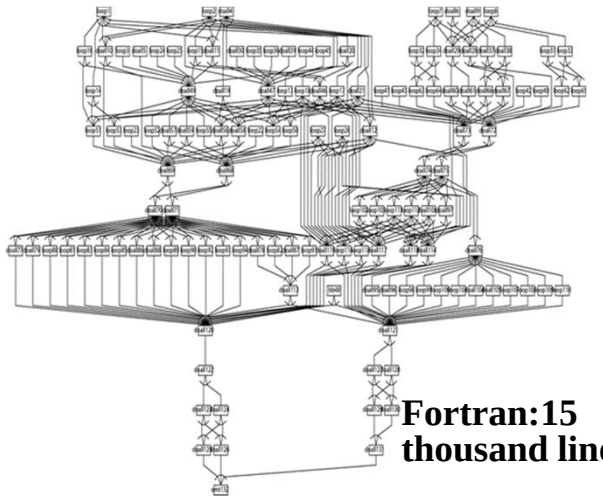


Box: Access range of DLG0

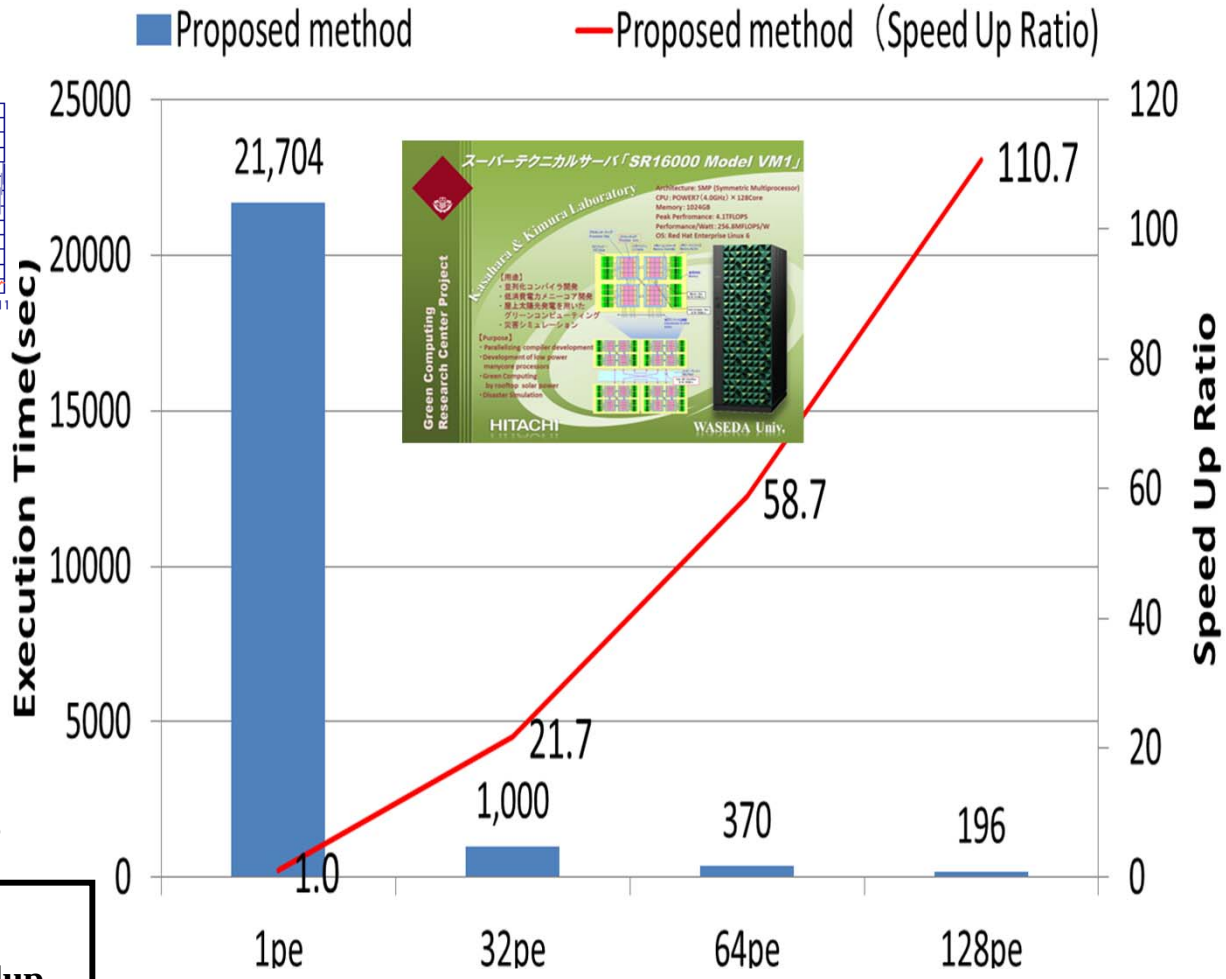
110 Times Speedup against the Sequential Processing for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000

（Power7 Based 128 Core Linux SMP） (LCPC2015)





Fortran:15 thousand lines

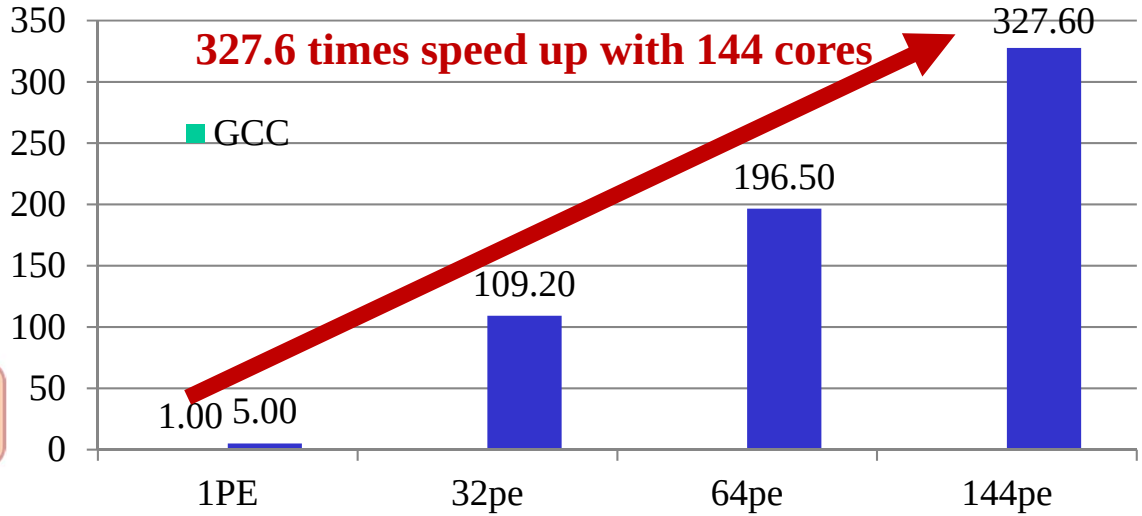


First touch for distributed shared memory and cache optimization over loops are important for scalable speedup

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

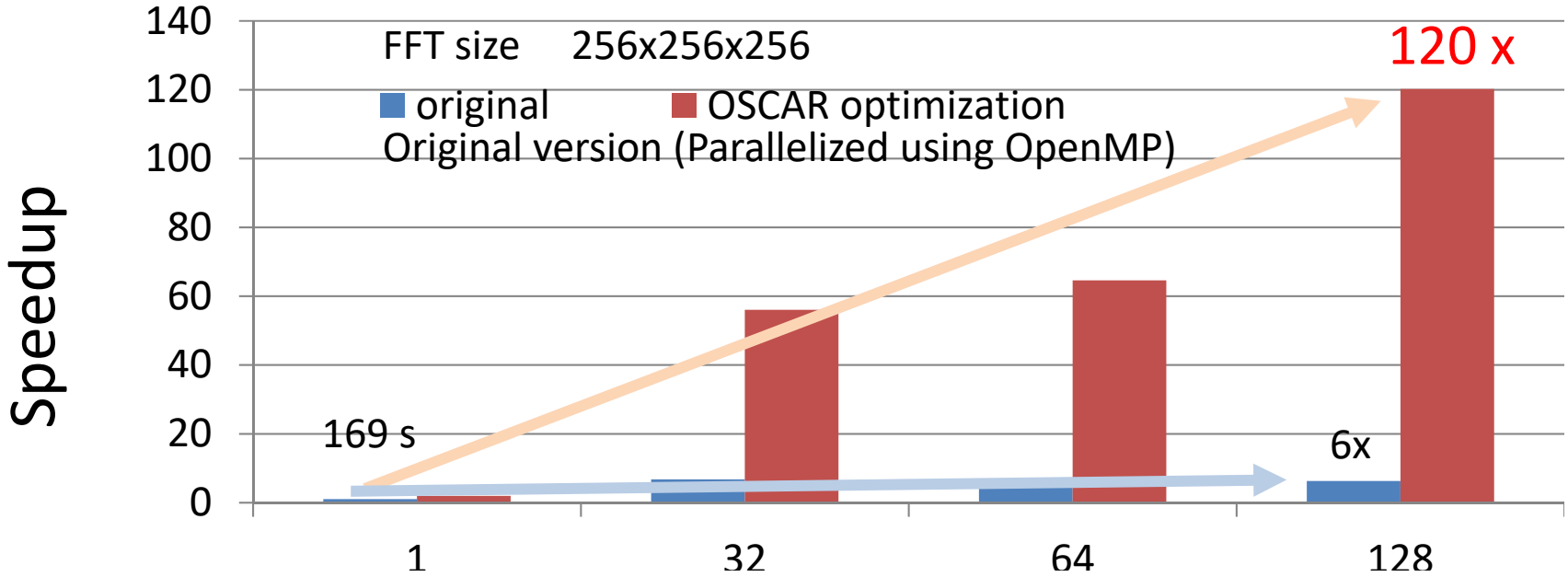
327 times speedup on 144 cores

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



- **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

Multicore Program Development Using OSCAR API V2.0

Sequential Application Program in Fortran or C
(Consumer Electronics, Automobiles, Medical, Scientific computation, etc.)

Homogeneous

Hetero

Manual parallelization / power reduction

Accelerator Compiler/ User
Add "hint" directives before a loop or a function to specify it is executable by the accelerator with how many clocks

Waseda OSCAR Parallelizing Compiler
- Coarse grain task parallelization
- Data Localization
- DMAC data transfer
- Power reduction using DVFS, Clock/ Power gating

OSCAR API for Homogeneous and/or Heterogeneous Multicores and manycores
Directives for thread generation, memory, data transfer using DMA, power managements

Parallelized API F or C program

Proc0
Code with directives
Thread 0

Proc1
Code with directives
Thread 1

Accelerator 1
Code

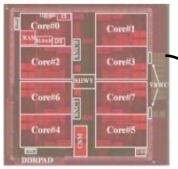
Accelerator 2
Code

Generation of parallel machine codes using sequential compilers

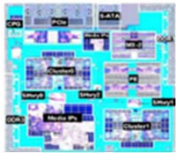
Low Power Homogeneous Multicore Code Generation
API Analyzer | Existing sequential compiler

Low Power Heterogeneous Multicore Code Generation
API Analyzer (Available from Waseda) | Existing sequential compiler

Server Code Generation
OpenMP Compiler



Homegeneous Multicore s from Vendor A (SMP servers)



Heterogeneous Multicores from Vendor B

Shred memory servers

Executable on various multicores

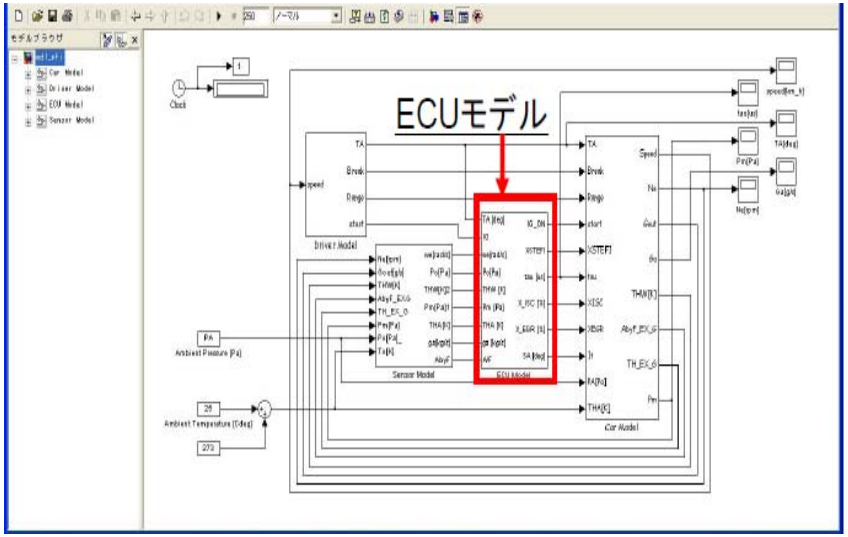
Hitachi, Renesas, NEC, Fujitsu, Toshiba, Denso, Olympus, Mitsubishi, Esol, Cats, Gaio, 3 univ.

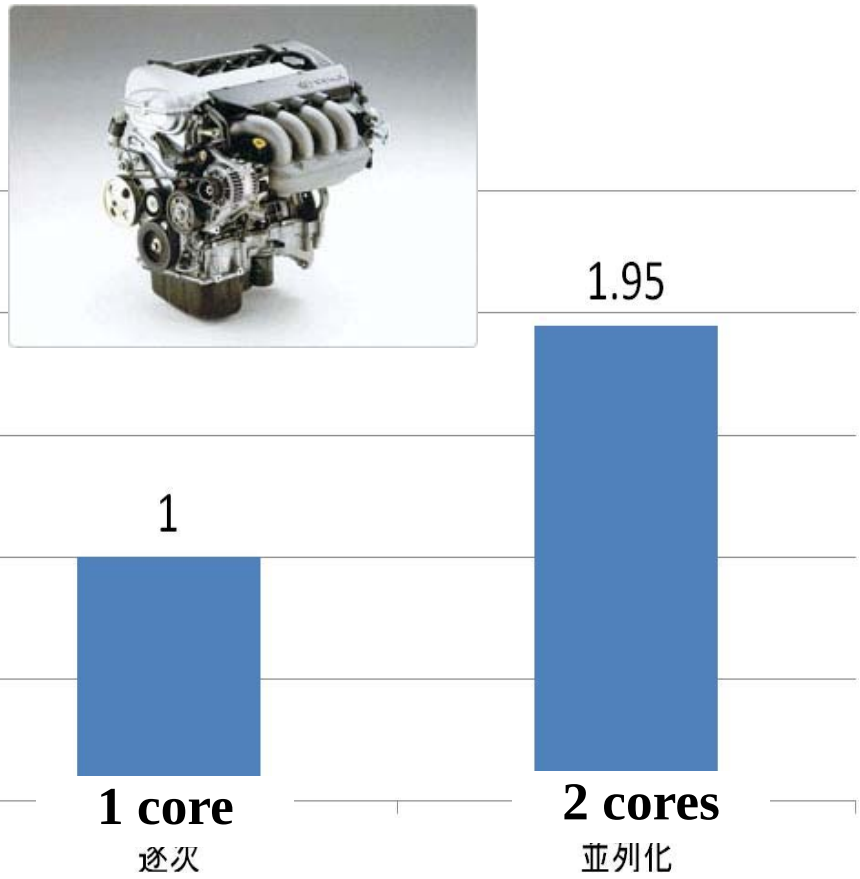
OSCAR: Optimally Scheduled Advanced Multiprocessor
API : Application Program Interface

Engine Control by multicore with Denso

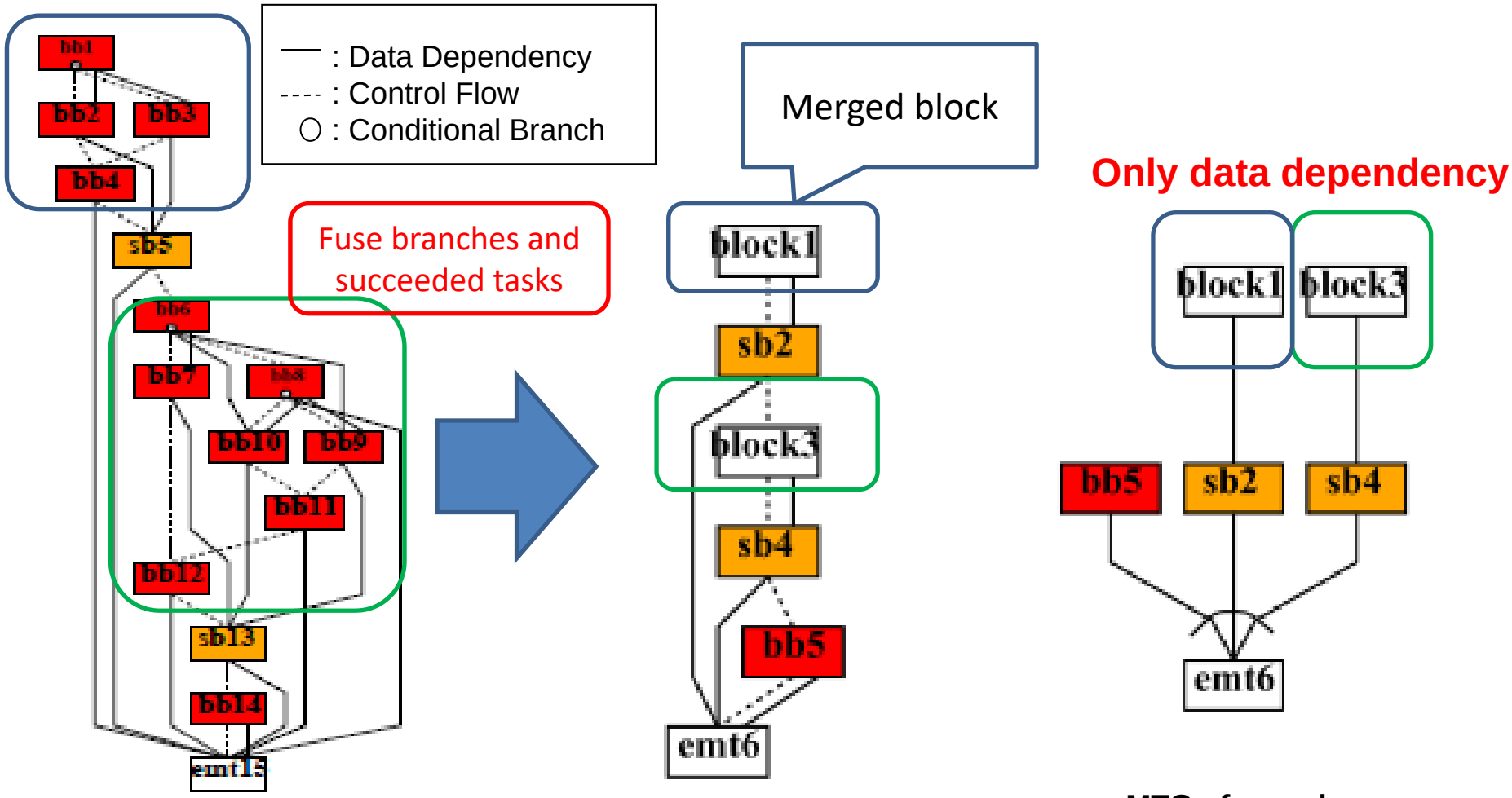
Though so far parallel processing of the engine control on multicore has been very difficult, Denso and Waseda succeeded 1.95 times speedup on 2core V850 multicore processor.

- 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



MFG of sample program before maro task fusion

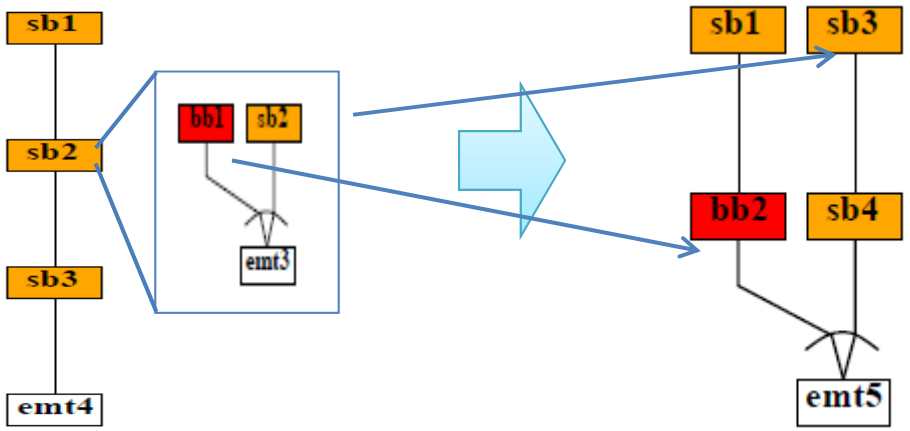
MFG of sample program after macro task fusion

MTG of sample program after macro task fusion

3.1 Restructuring : Inline Expansion

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

Improves coarse grain parallelism



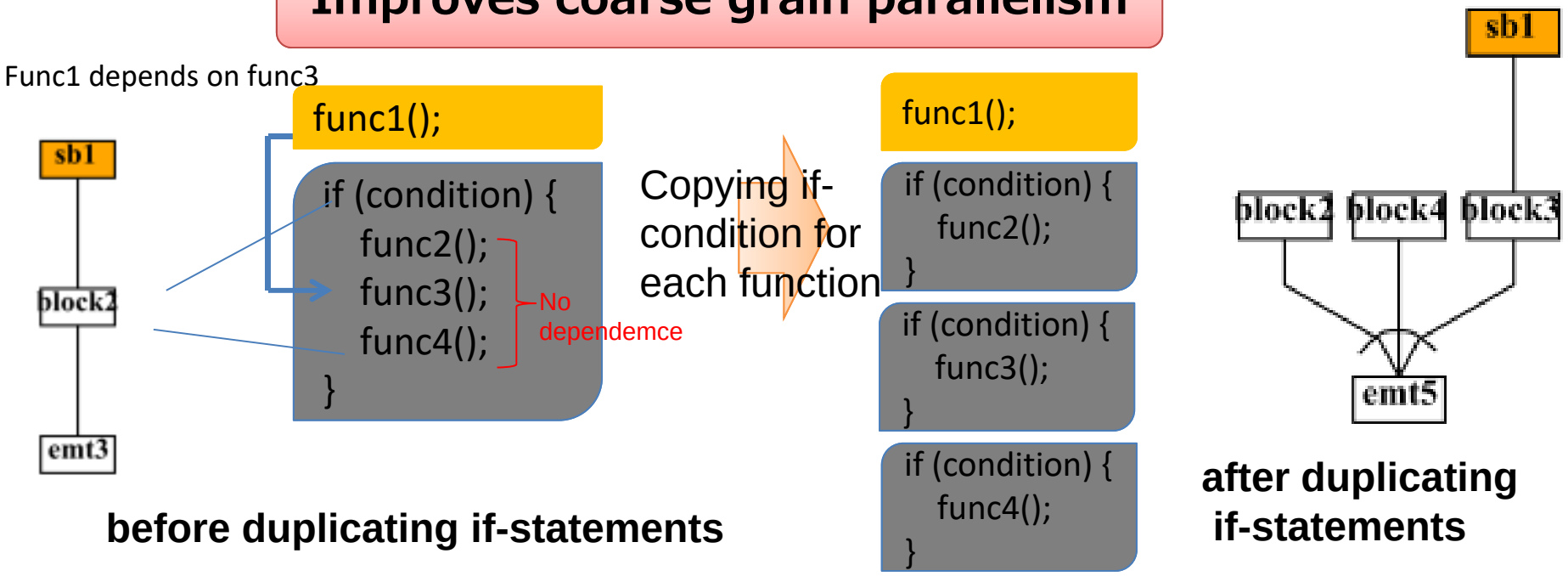
MTG before inline expansion

MTG after inline expansion

3.2 Restructuring: Duplicating If-statements

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

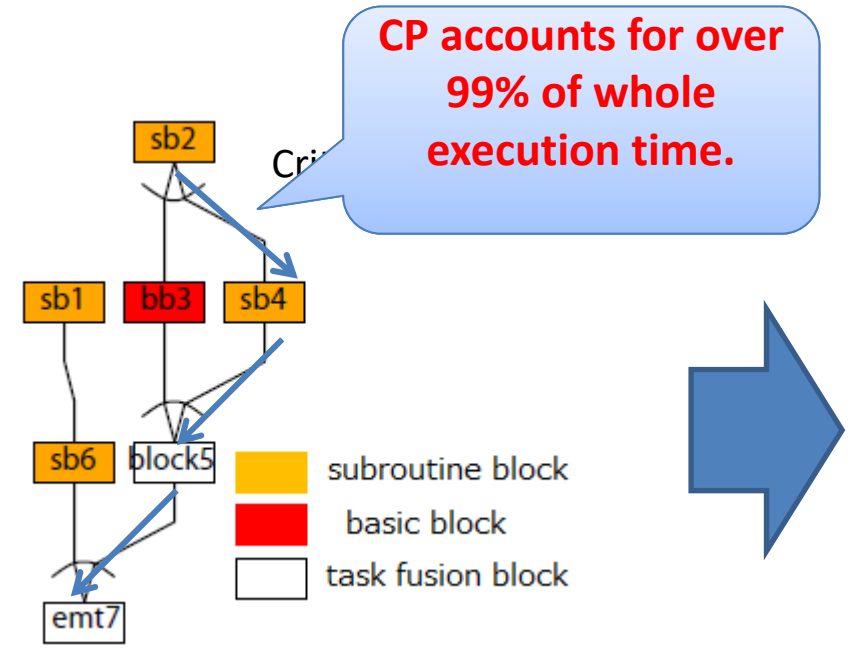
Improves coarse grain parallelism



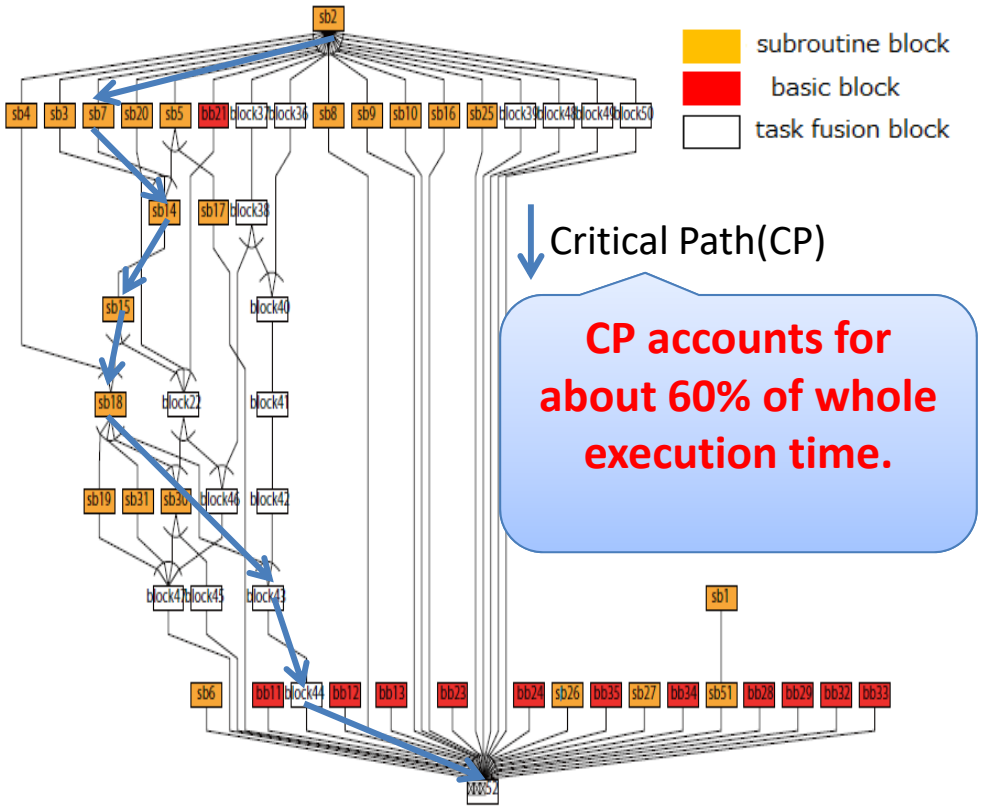
before duplicating if-statements

after duplicating if-statements

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



MTG of crankshaft program before restructuring

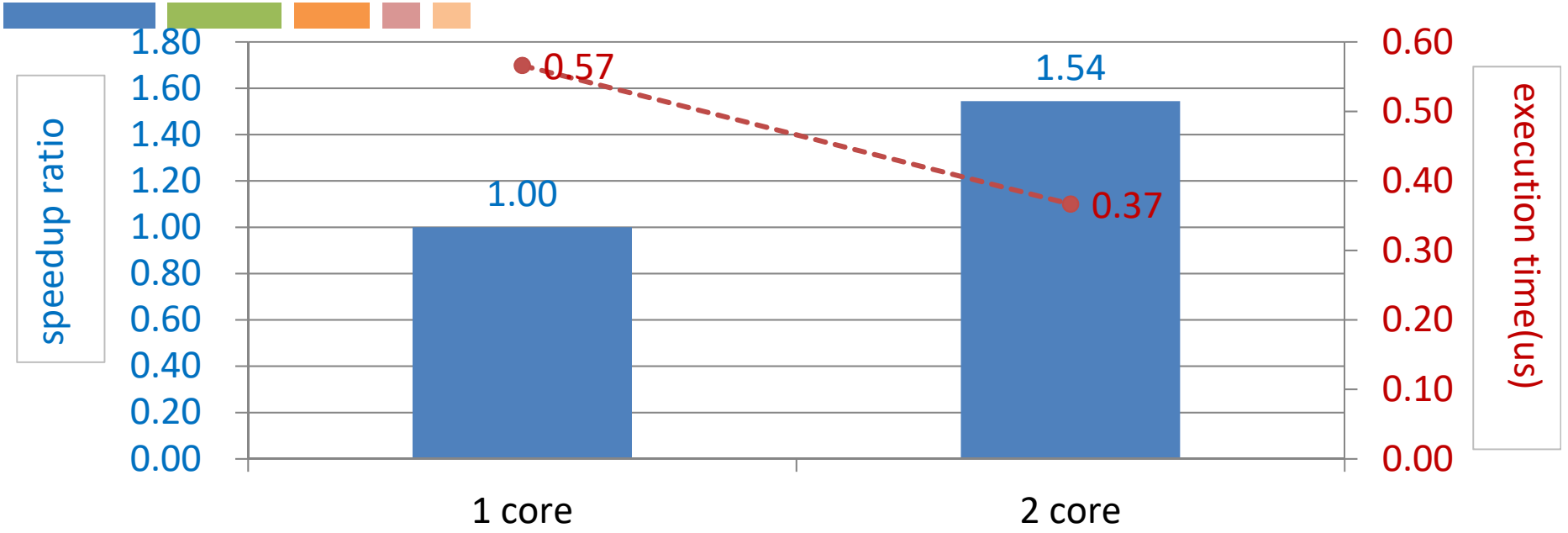


MTG of crankshaft program after restructuring

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

Successfully increased coarse grain parallelism

Evaluation of Crankshaft Program with Multi-core Processors

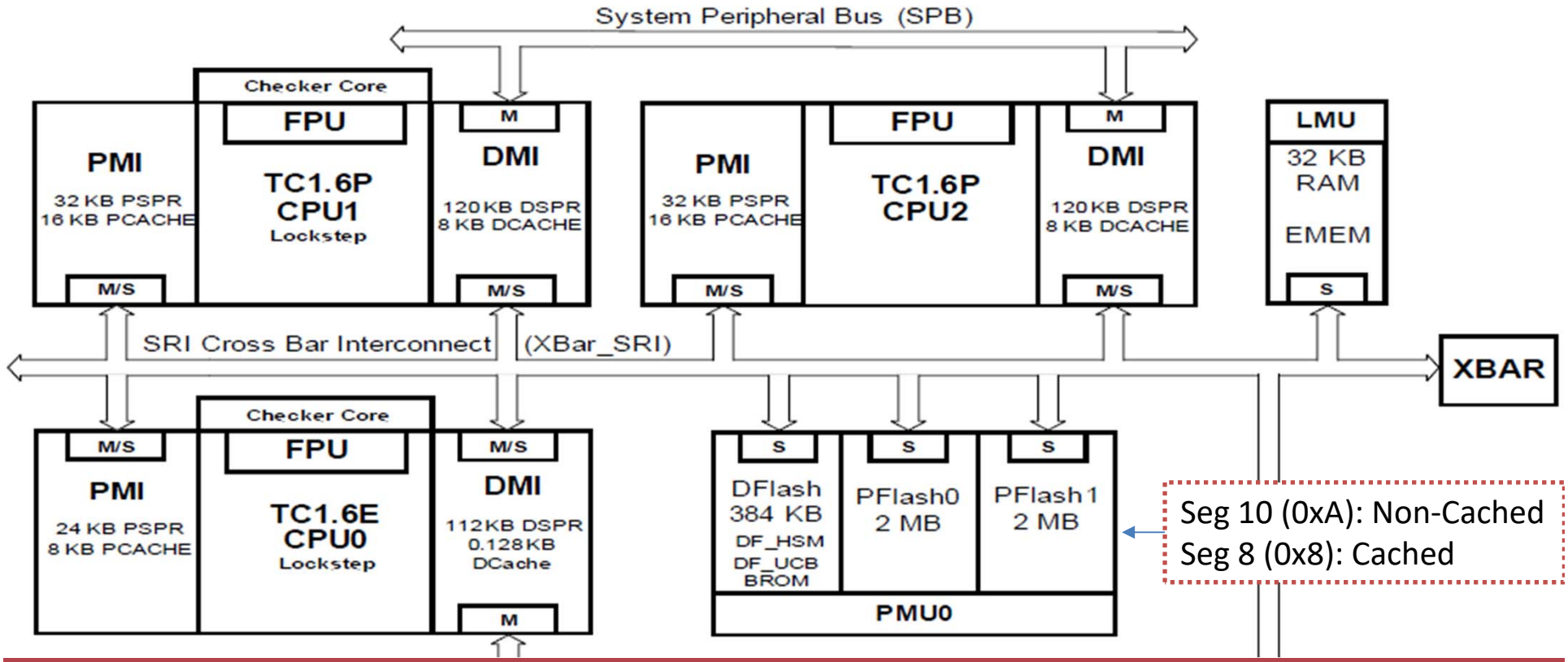


- Attain 1.54 times speedup on RPX
  - There are no loops, but only many conditional branches and small basic blocks and difficult to parallelize this program
- This result shows possibility of multi-core processor for engine control programs

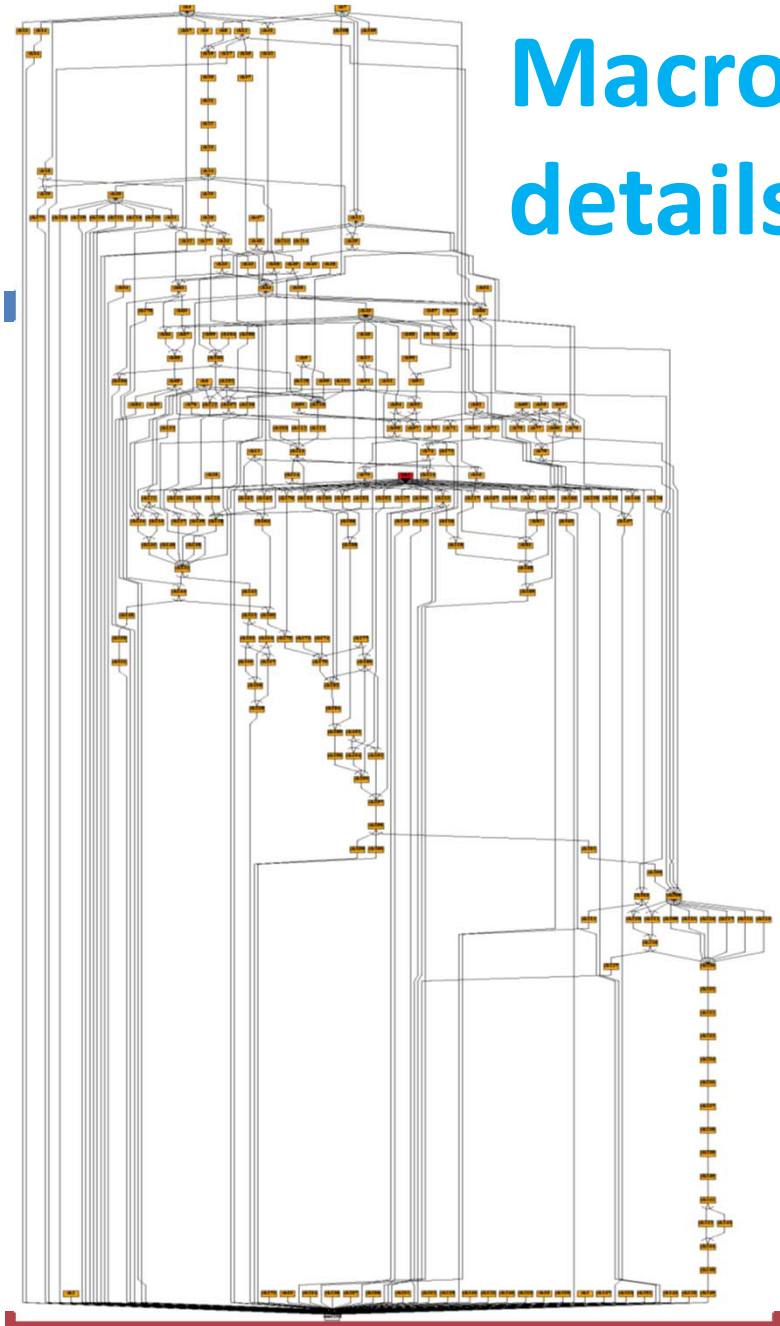
Infineon AURIX TC277

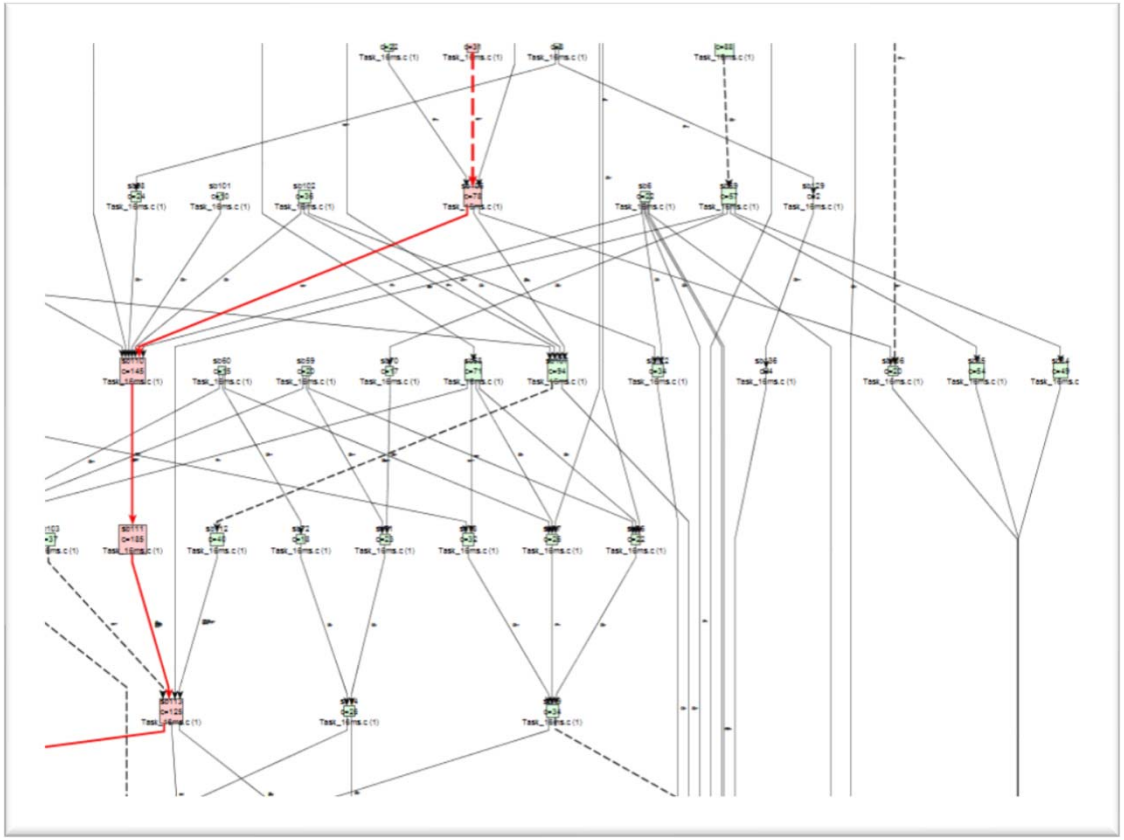
Abbreviations :

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 (EEEPROM)
S :	SRI Slave Interface
M :	SRI Master Interface



Macrotask Graph, Dependence details and schedules

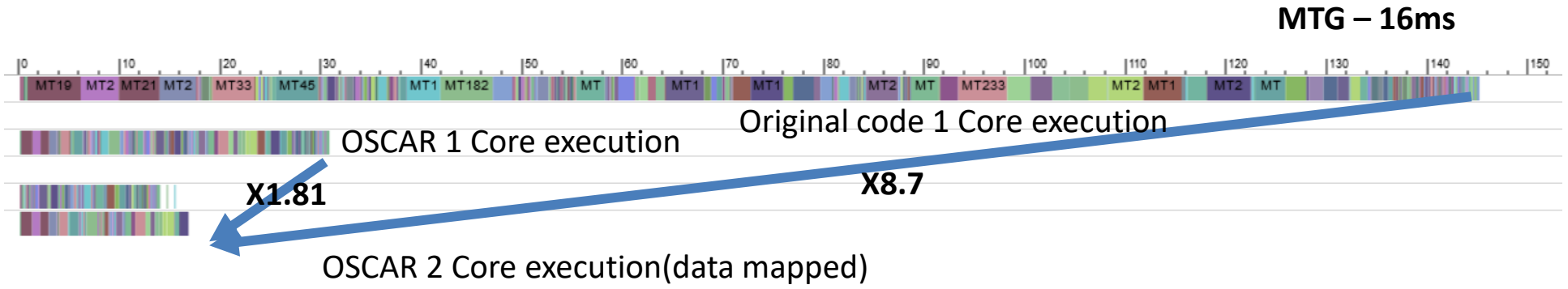




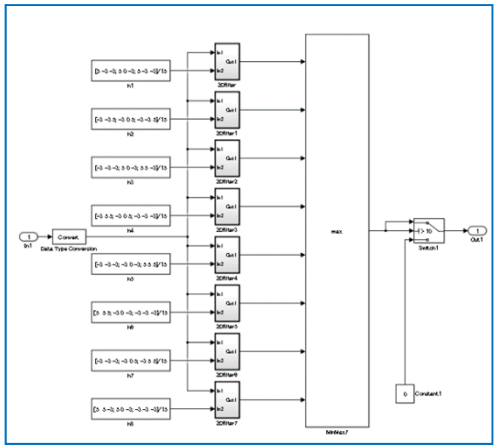
MTG – task_16ms

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.**



OSCAR Compile Flow for Simulink Applications



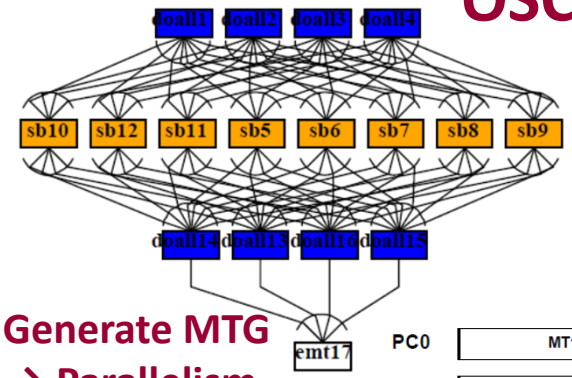
Simulink model

Generate C code
using Embedded Coder

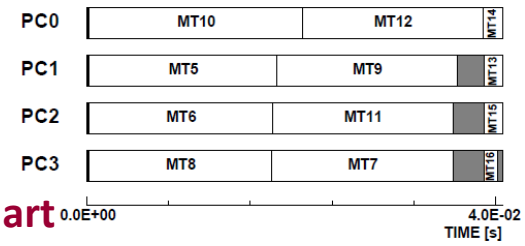
```
/* Model step function */
void VesselExtraction_step(void)
{
  int32_T i;
  real_T u0;

  /* 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>/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



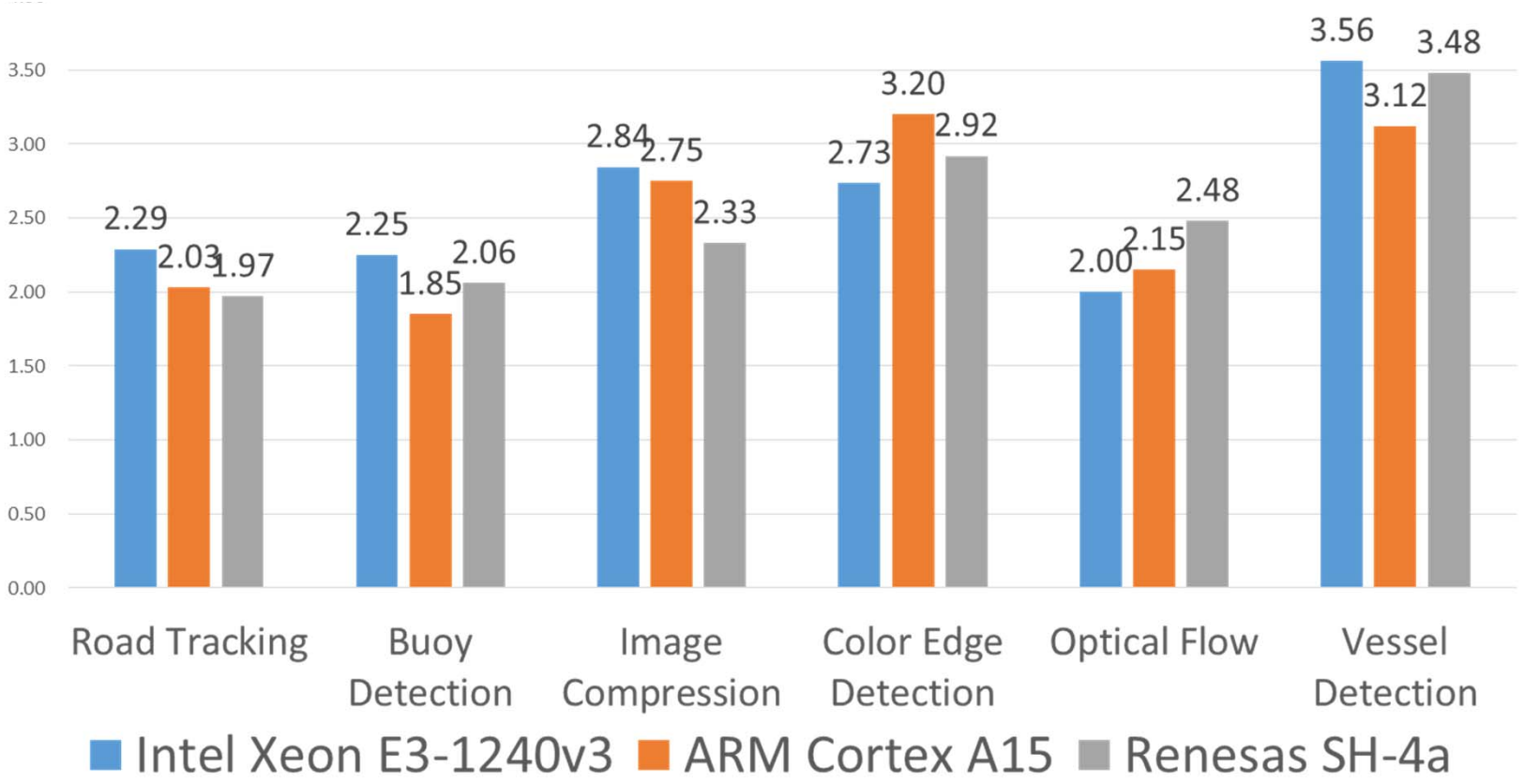
(2) Generate gantt chart
→ Scheduling in a multicore

```
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_PE0 ( ) ;

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

void thread_function_001 ( void )
{
    VesselExtraction_step_PE1 ( ) ;
}
```

(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/

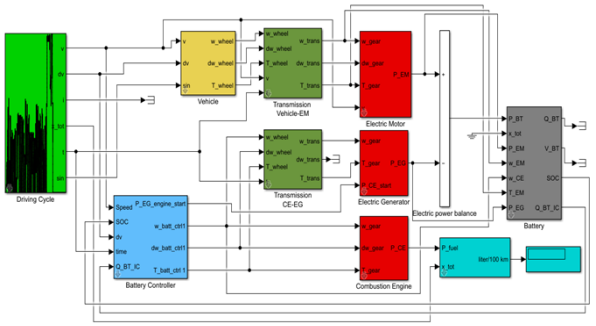
Automatic Pallalelization Tool of MATLAB/Simulink:

OSCAR Tech “OSCARator” https://www.oscartech.jp/en/

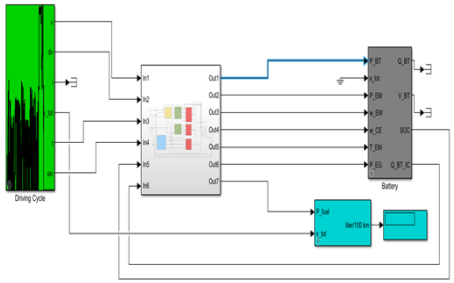
- **OSCARator is a simulation accelerator of MATLAB/Simulink on multicore processor**
  - **based on “OSCAR Compiler” Automatic Parallelization Technology developed by Kasahara and Kimura Lab. Waseda University**

Original Simulink Model

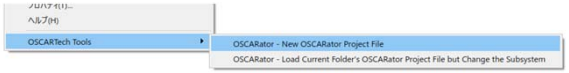
HybridVehicle

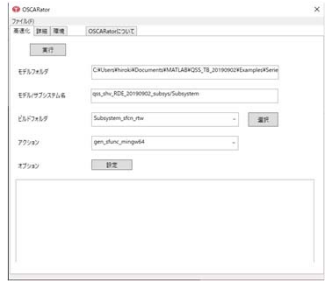


make a “Subsystem” with blocks which you want to accelerate

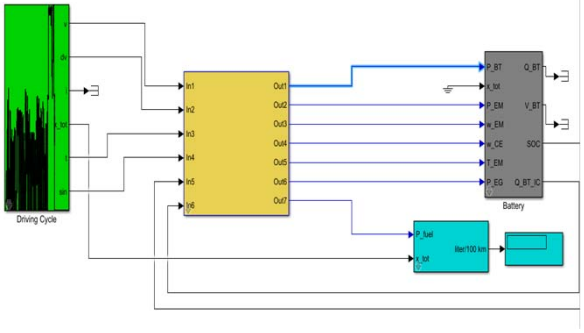


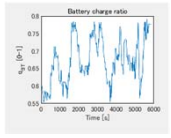
Start OSCARator from right click menu, OSCARator will automatically configure settings.



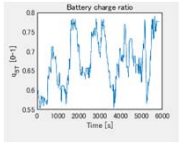


New Accelerated Simulink Model





Same Result and
Shorter Simulation Time



<FULLY AUTOMATIC>

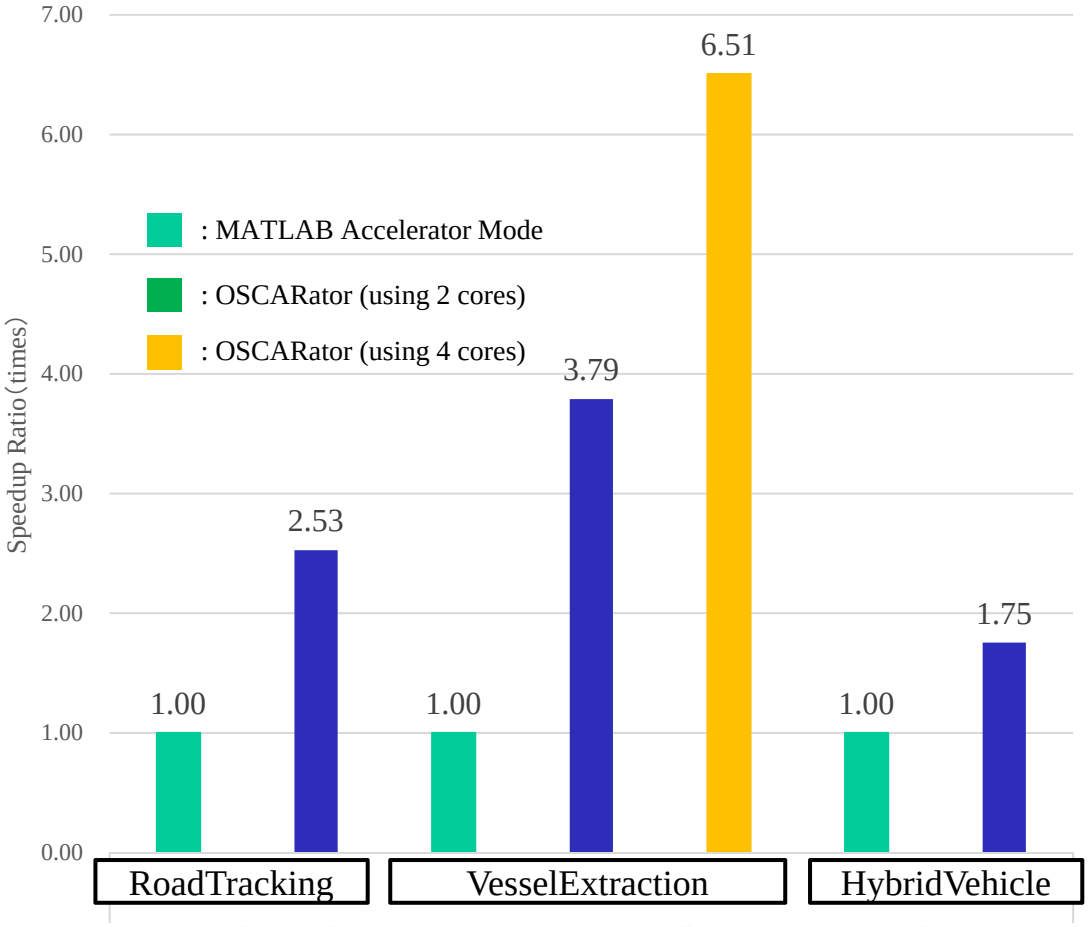
- **Simulink Coder C-Code Generation**
- **Automatic Parallelization**
- **S-Function MEX Build**
- **Replacing Subsystem with S-Function Block**

MEX: Dynamically linked subroutine executed in the MATLAB environment.

Speedup of Simulink Models by OSCARator on 4 cores Intel Core i5 Processor

https://www.oscartech.jp/en/

6.51 times speed up on 4 cores against 1 core MATLAB Accerelator Mode for VesselExtraction

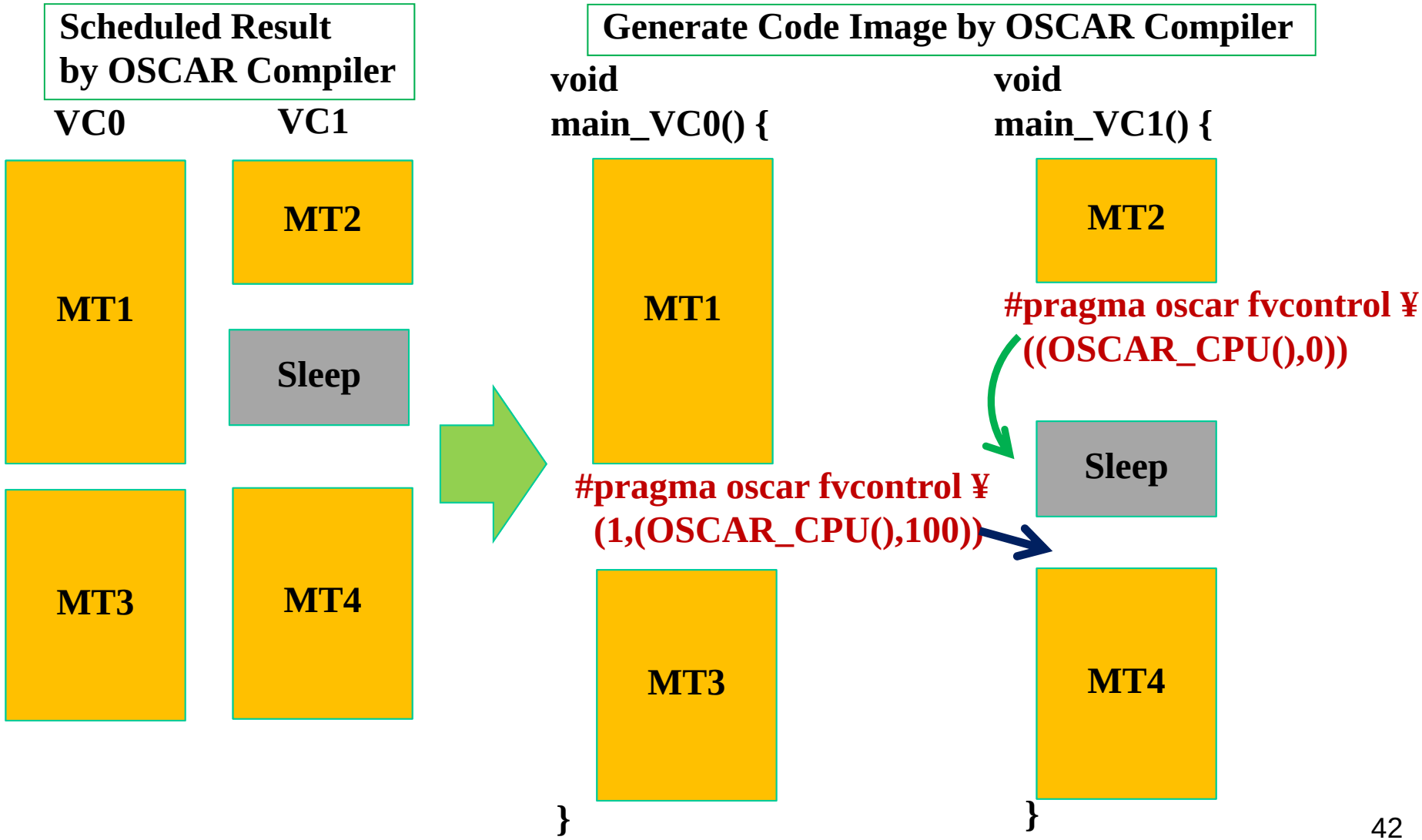


Intel Core i5 7400T 2.4GHz (4 cores)
16GB (SODIMM 2400MHz)
Windows 10 Pro (1903)
MATLAB R2019a Update 5
MinGW GCC 6.3

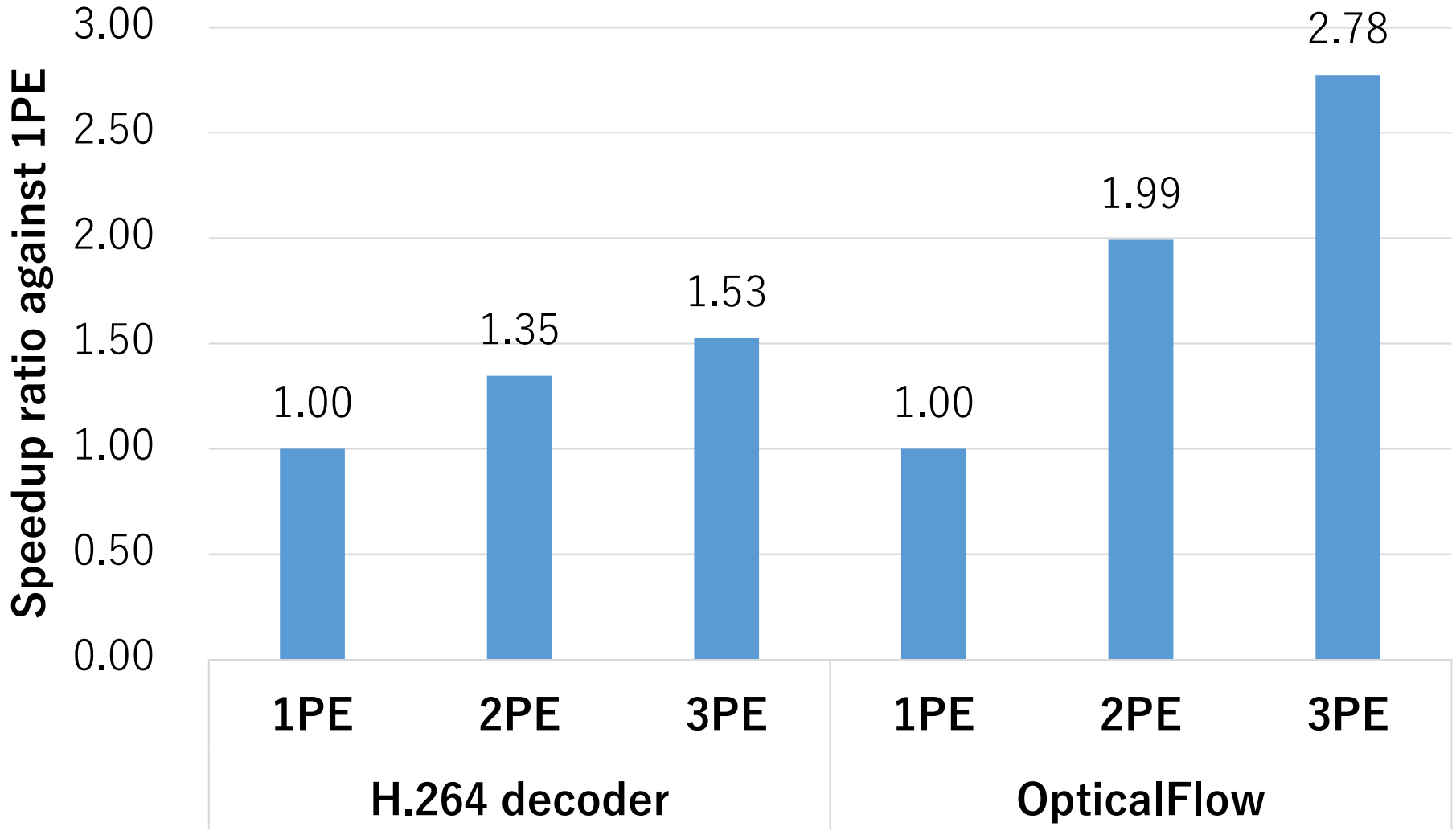
- RoadTracking
  - from Computer Vision Toolbox
  - https://jp.mathworks.com/help/vision/examples/color-based-road-tracking.html
- VesselExtraction
  - from MATLAB Central
  - modified for Simulink Model
  - https://www.mathworks.com/matlabcentral/fileexchange/24990-retinal-blood-vessel-extraction
- HybridVehicle
  - Hybrid Vehicle Powertrain
  - developed by Kusaka Lab. Waseda University
  - http://www.f.waseda.jp/jin.kusaka/

(Compared with MATLAB Accelerator Mode Simulation)

Low-Power Optimization with OSCAR API

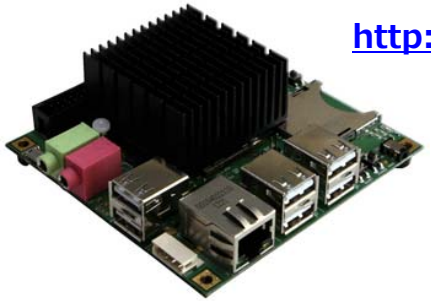


Speedup for H.264 and Optical Flow on ARM Cortex-A9 Android 3 cores by OSCAR Automatic Parallelization



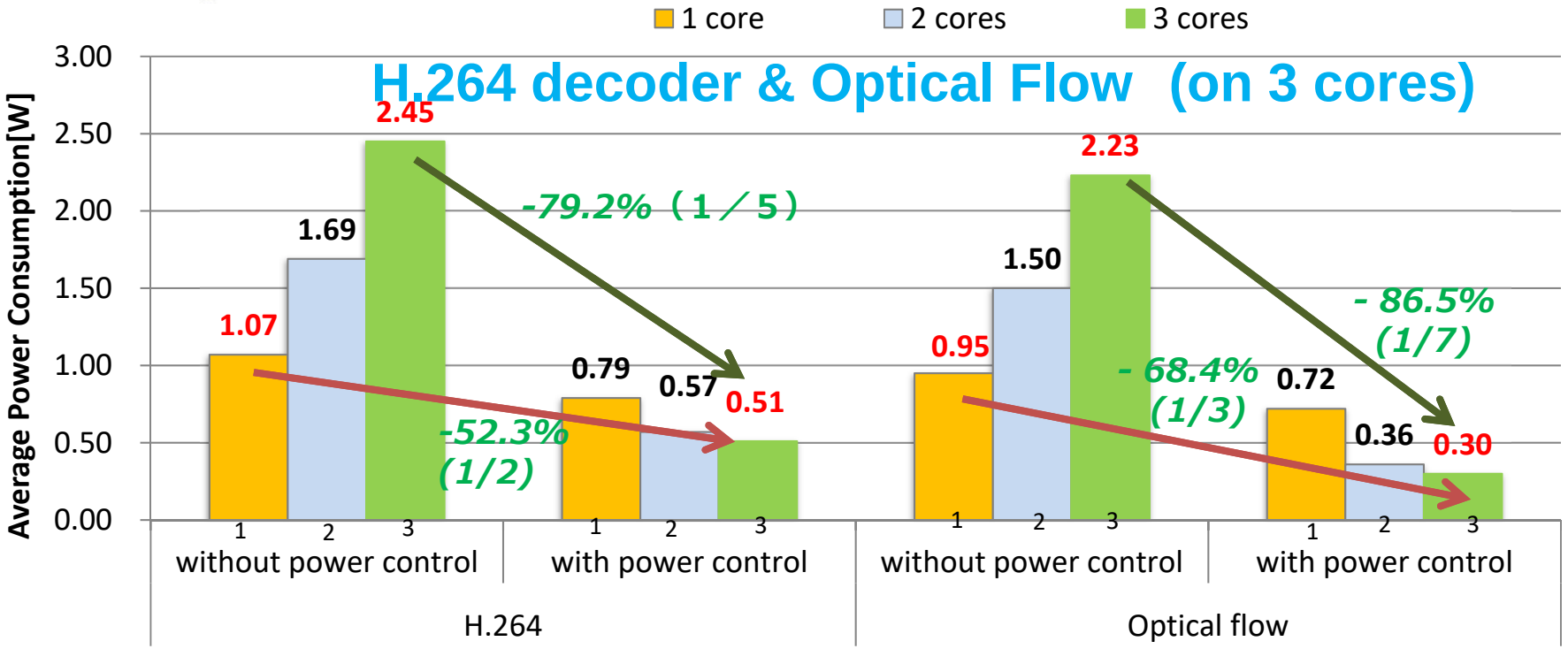
Automatic Power Reduction on ARM CortexA9 with Android

http://www.youtube.com/channel/UCS43lNYEIkC8i_KIgFZYQBQ



ODROID X2

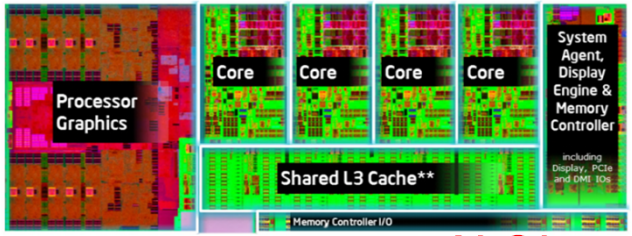
Samsung Exynos4412 Prime, ARM Cortex-A9 Quad core
1.7GHz～0.2GHz, used by Samsung's Galaxy S3



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

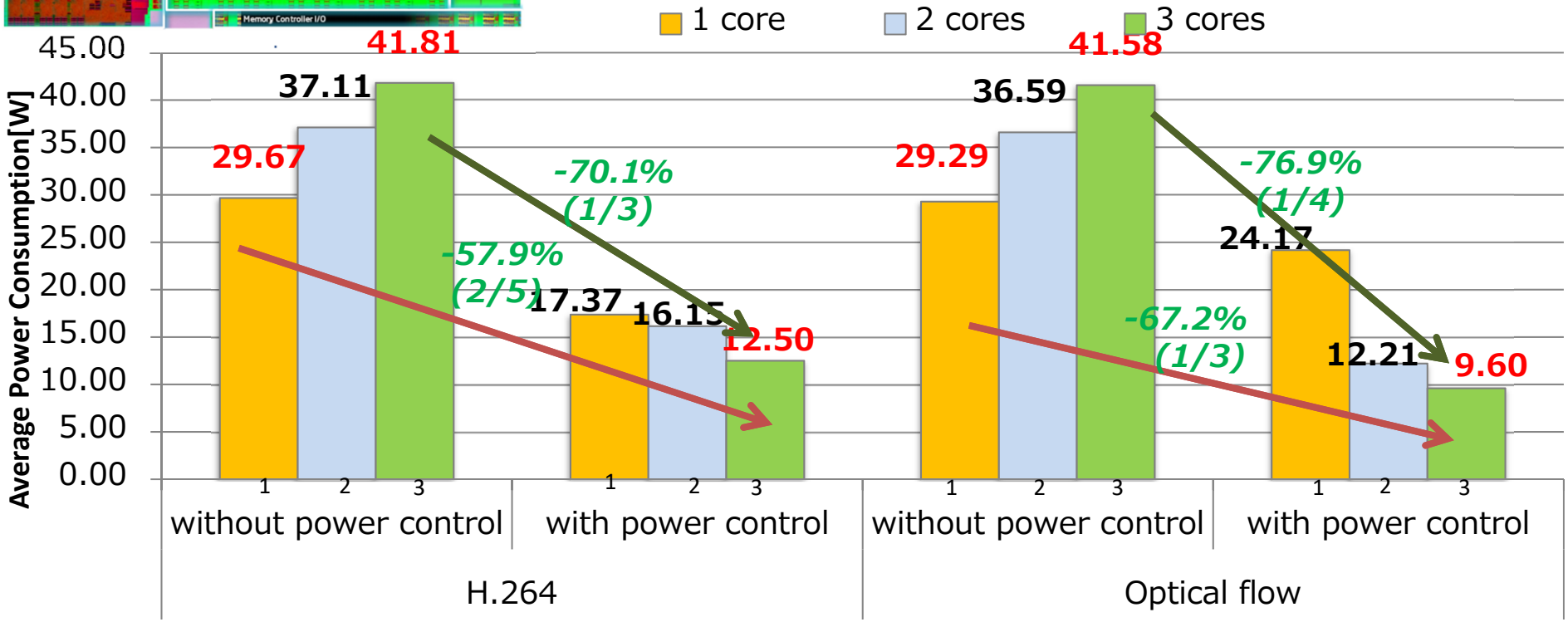
Automatic Power Reuction on Intel Haswell

H.264 decoder & Optical Flow (3cores)



H81M-A, Intel Core i7 4770k

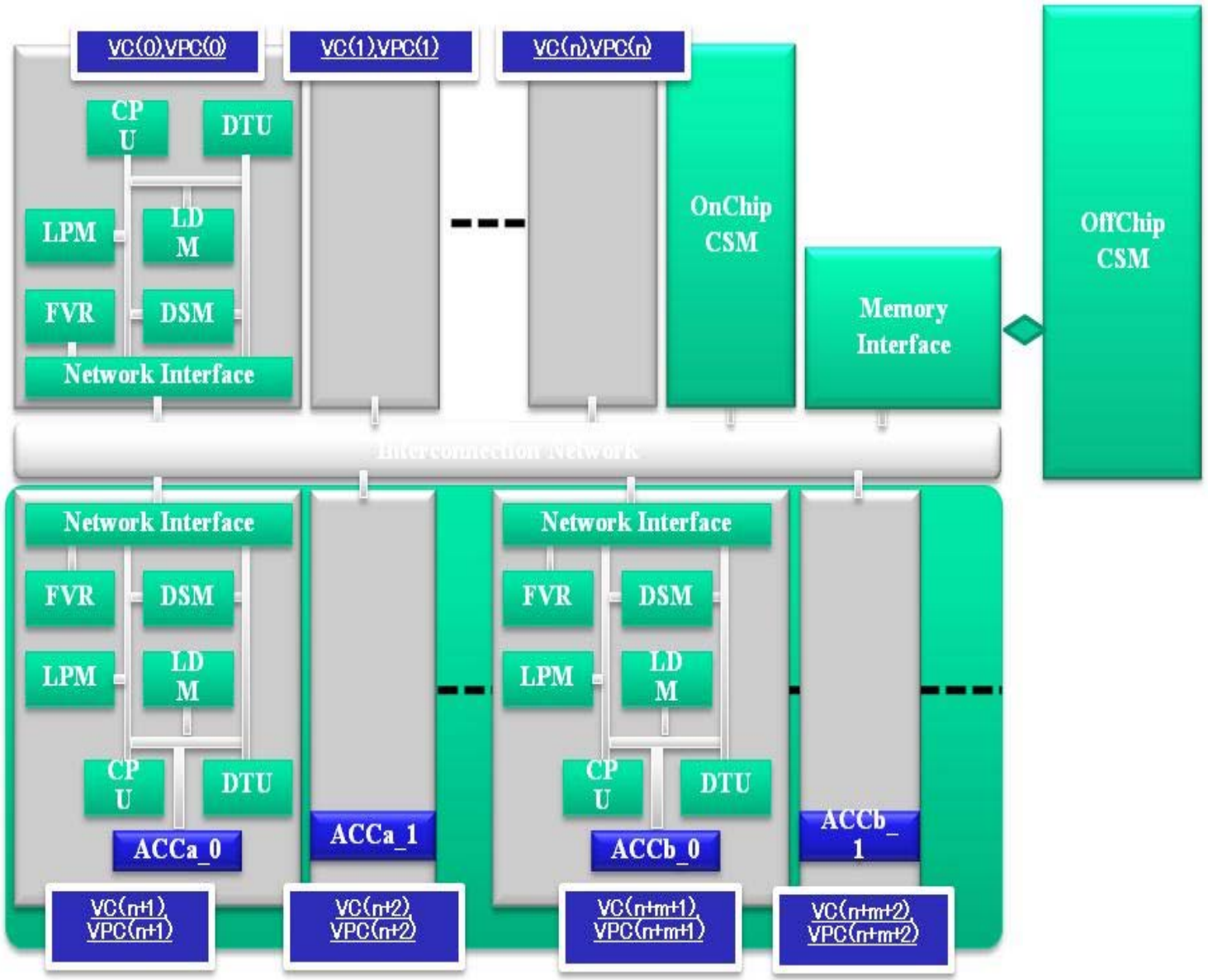
Quad core, 3.5GHz～0.8GHz



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

OSCAR Heterogeneous Multicore



DTU
- Data Transfer Unit

LPM
- Local Program Memory

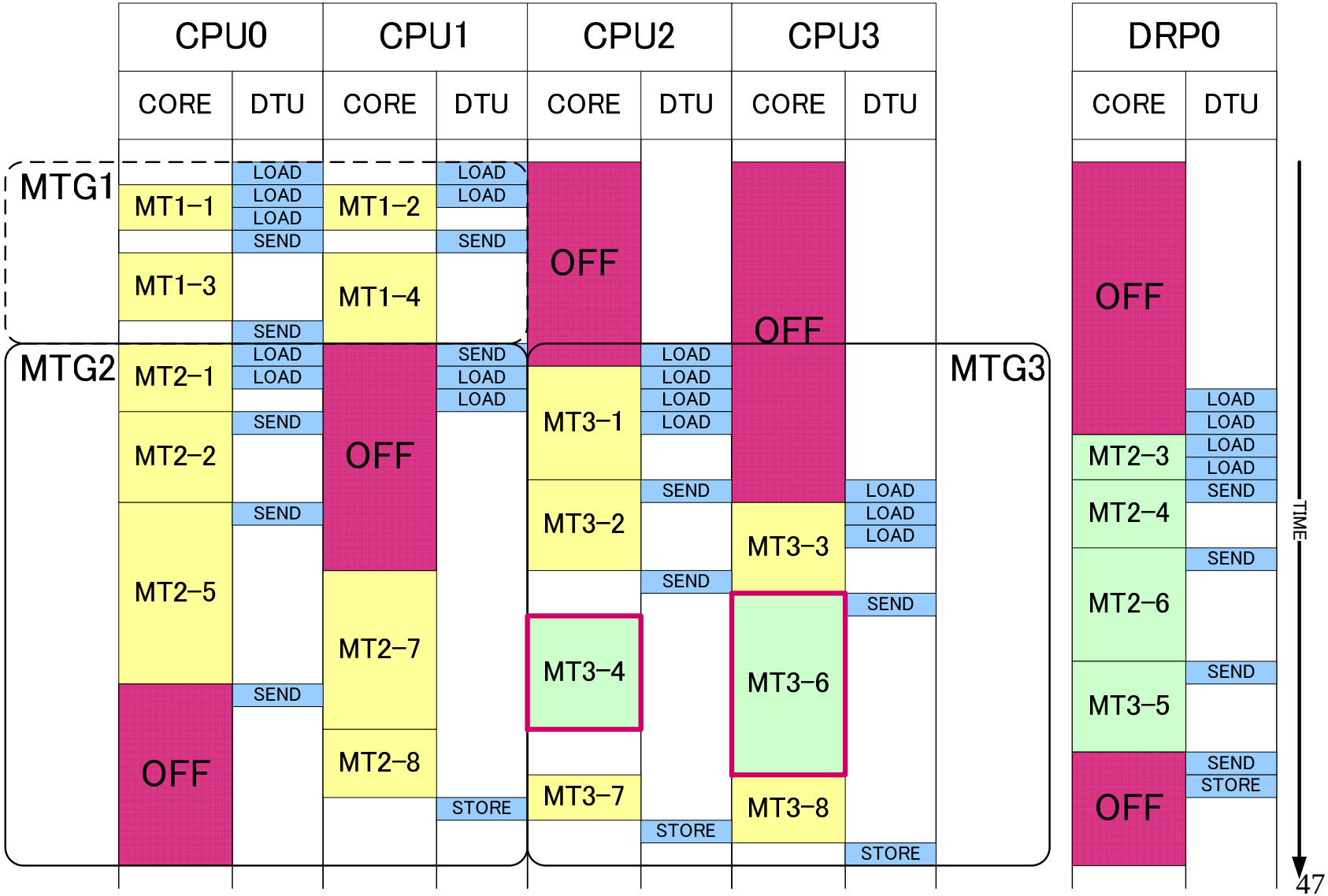
LDM
- Local Data Memory

DSM
- Distributed Shared Memory

CSM
- Centralized Shared Memory

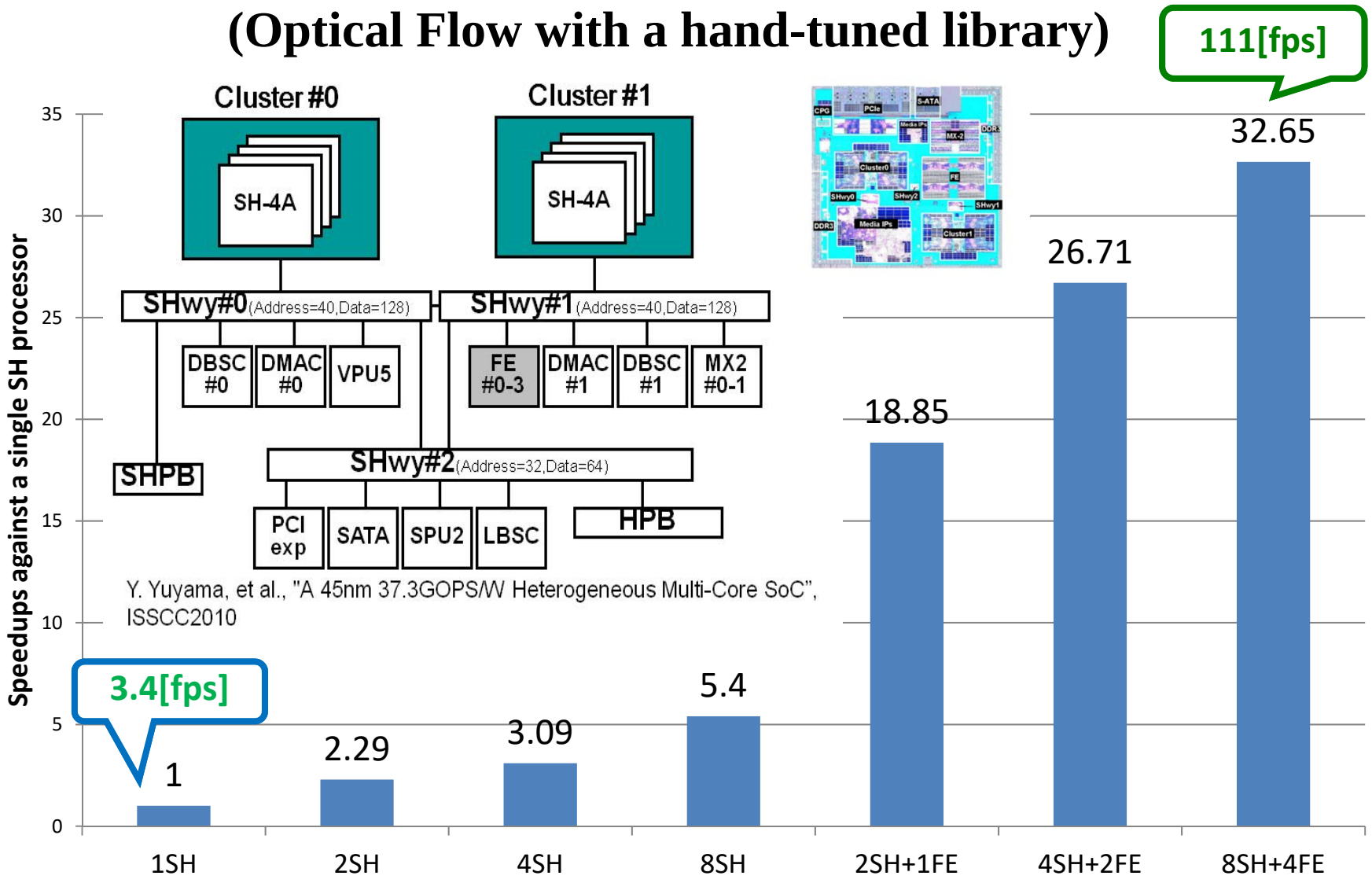
FVR
- Frequency/Voltage Control Register

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)



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

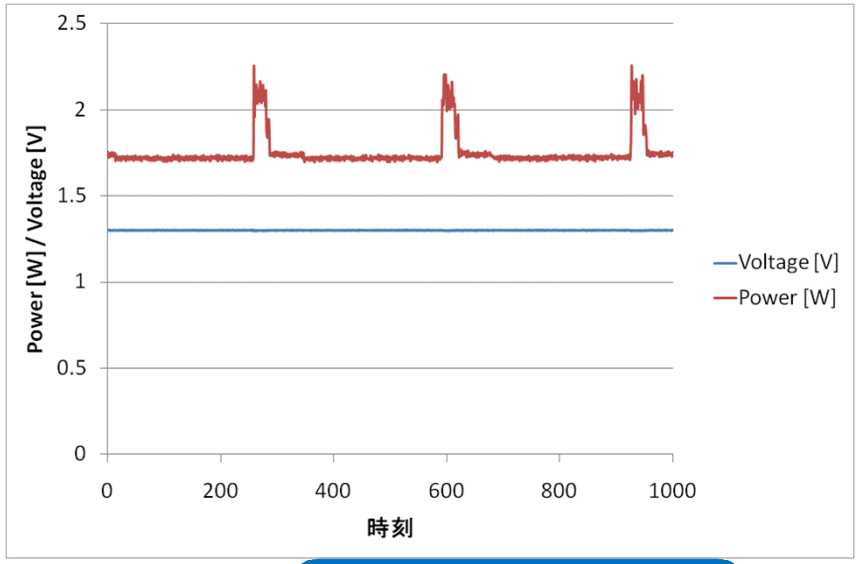
Without Power Reduction

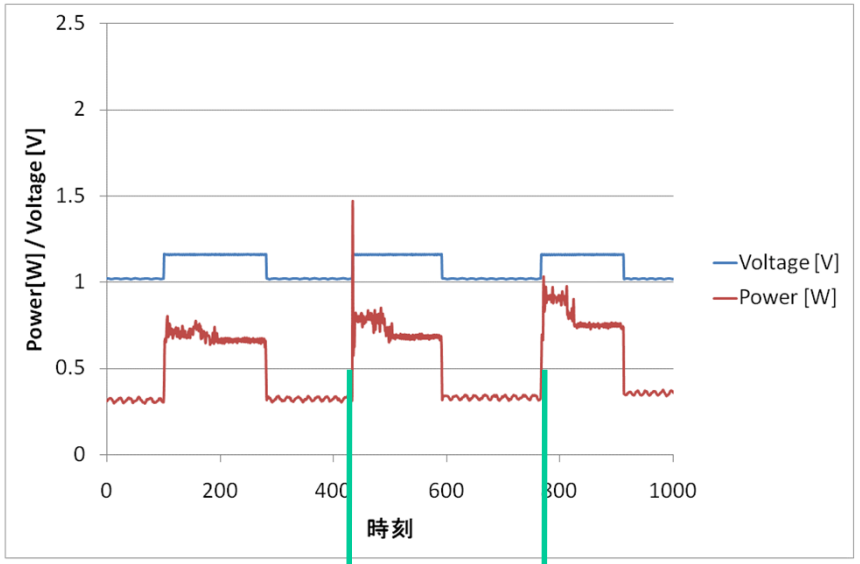
With Power Reduction by OSCAR Compiler

70% of power reduction

Average:1.76[W]

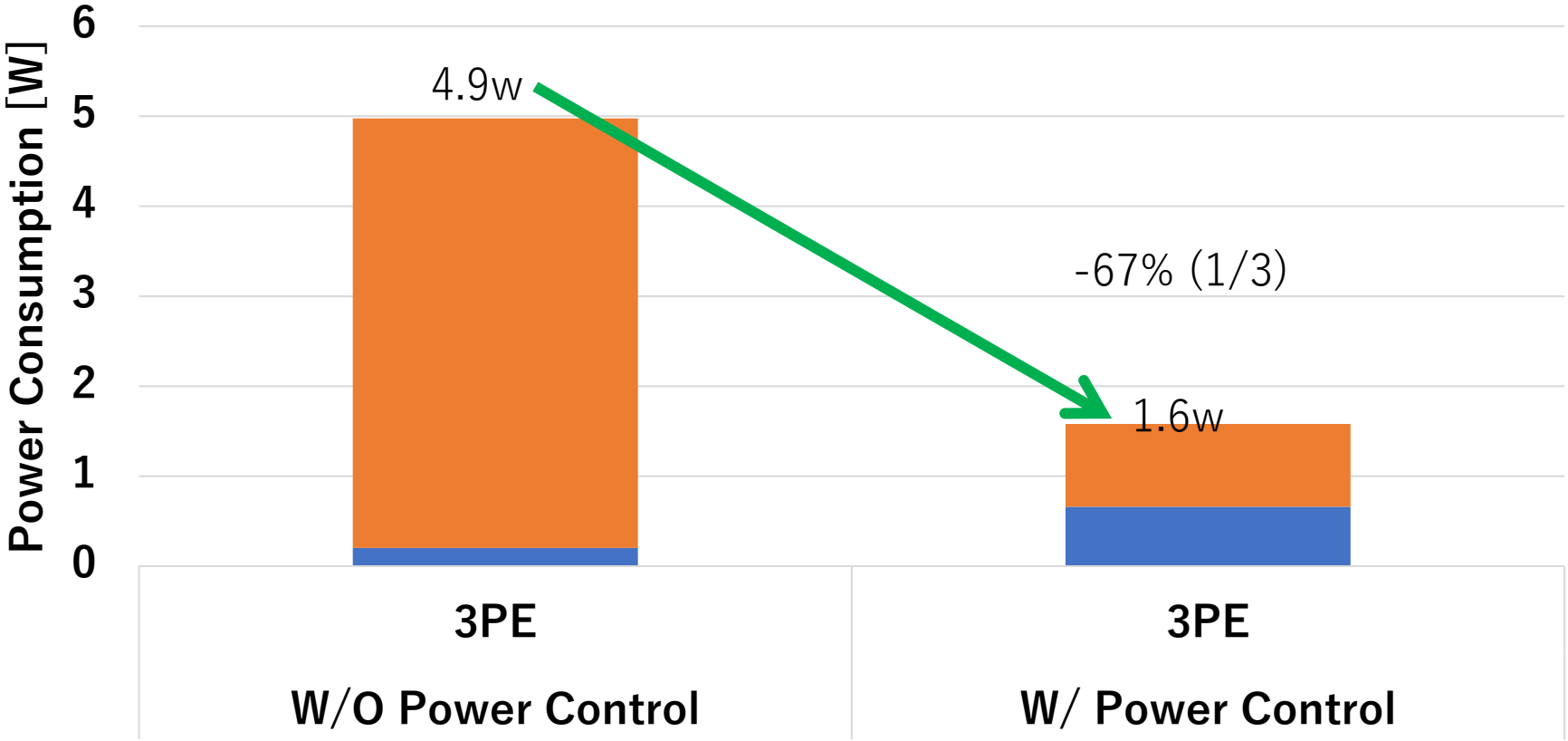
Average:0.54[W]





1cycle : 33[ms]
→30[fps]

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



- ODROID-XU3
  - 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/Heterogeneous Multicores and Manycores

(LCPC2009 Homogeneous, 2010 Heterogeneous)

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)

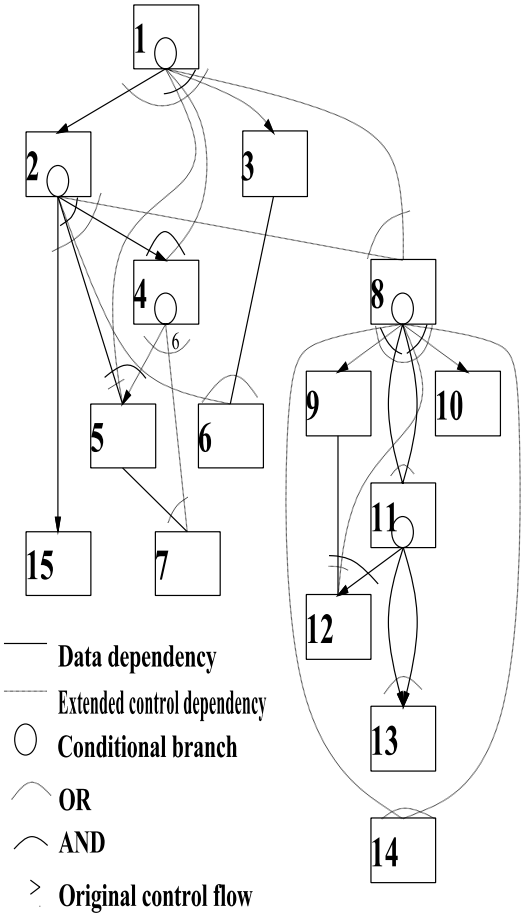
- Power Control API
  - fvcontrol
  - get_fvstatus
- Timer API
  - get_current_time
- Accelerator
  - accelerator_task_entry
- Cache Control
  - cache_writeback
  - cache_selfinvalidate
  - complete_memop
  - noncacheable
  - aligncache

2 hint directives for OSCAR compiler
- accelerator_task
- oscar_comment

from V2.0

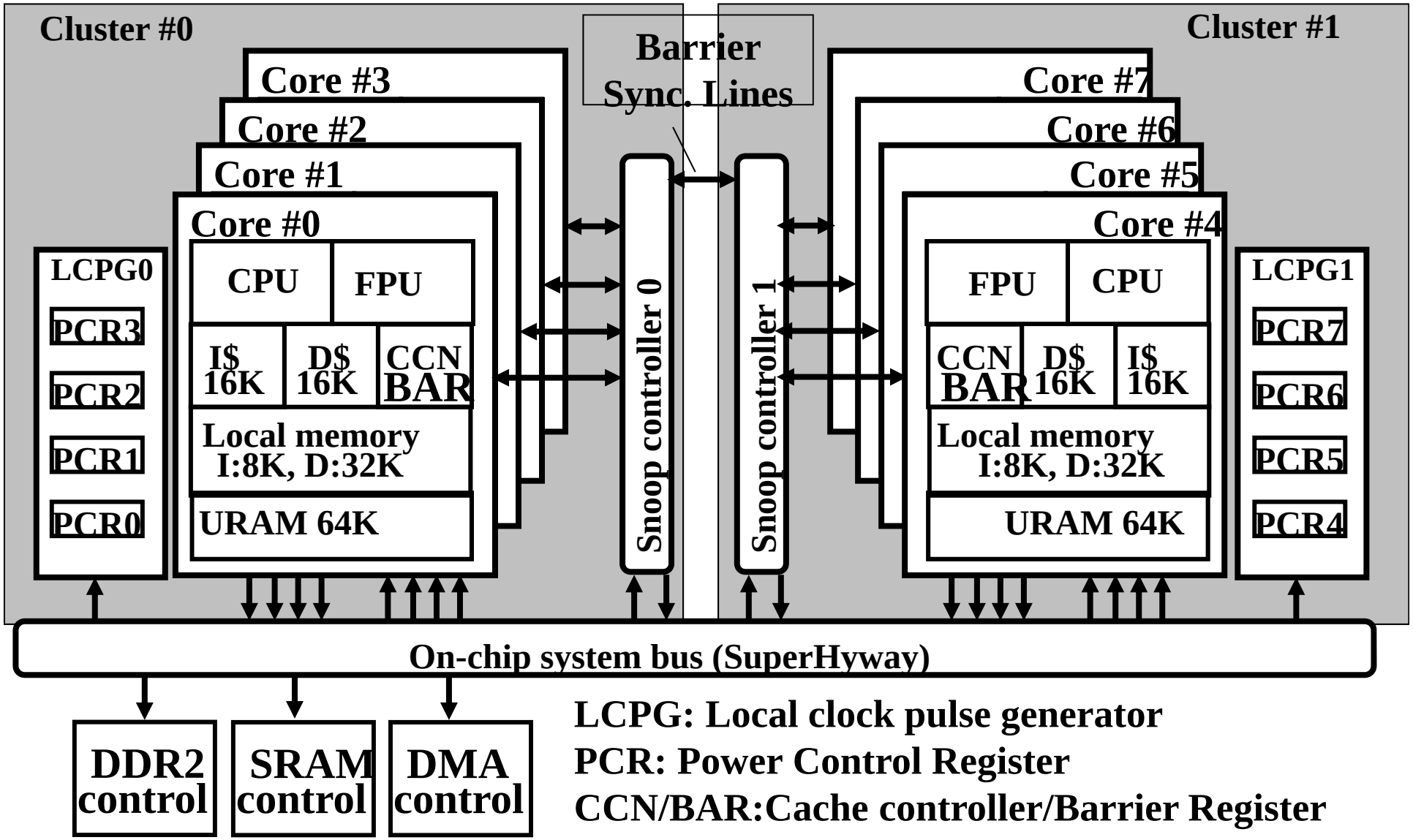
Software Coherence Control Method on OSCAR Parallelizing Compiler

- 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

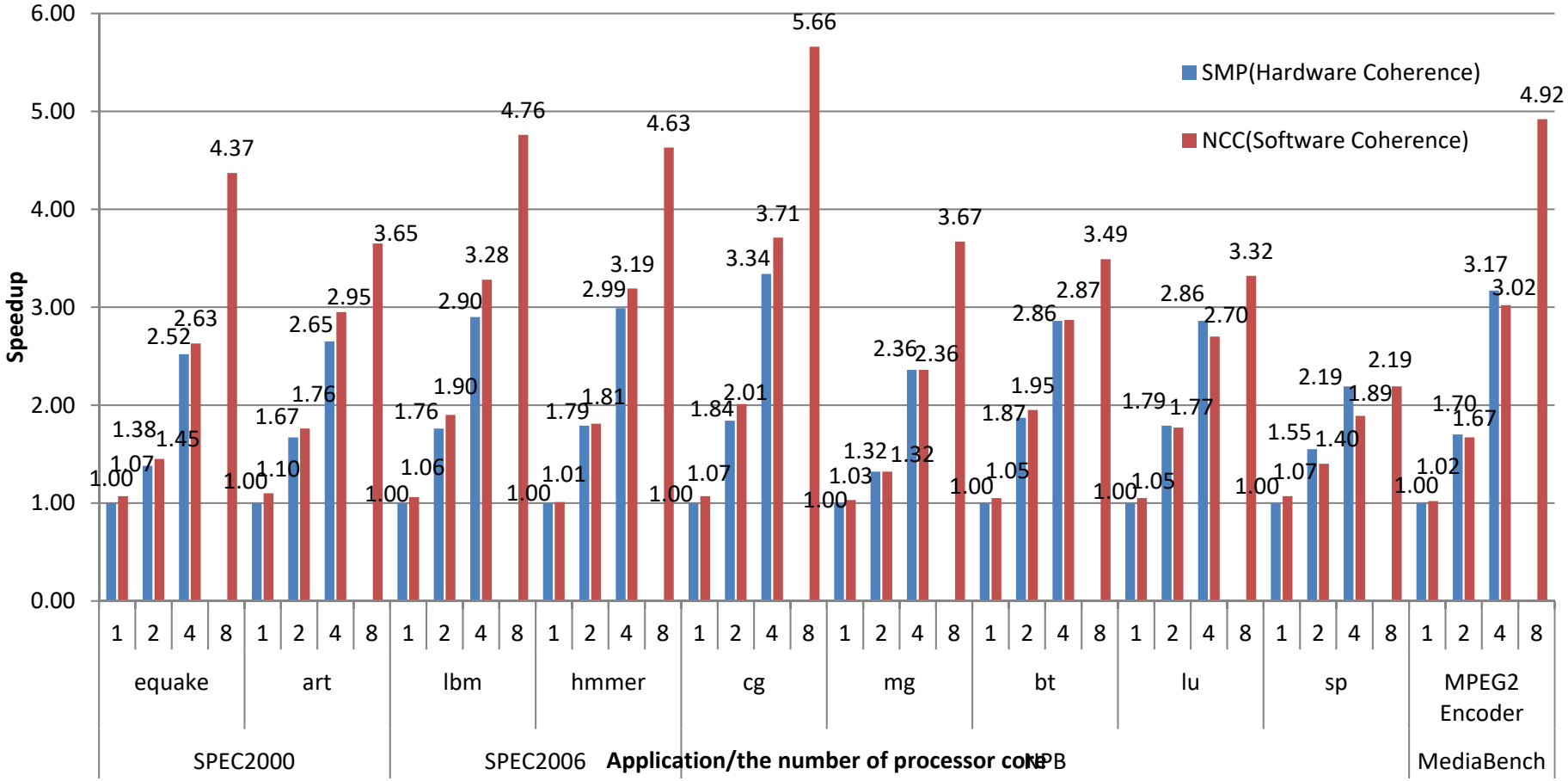
PCR: Power Control Register

CCN/BAR:Cache controller/Barrier Register

URAM: User RAM (Distributed Shared Memory)

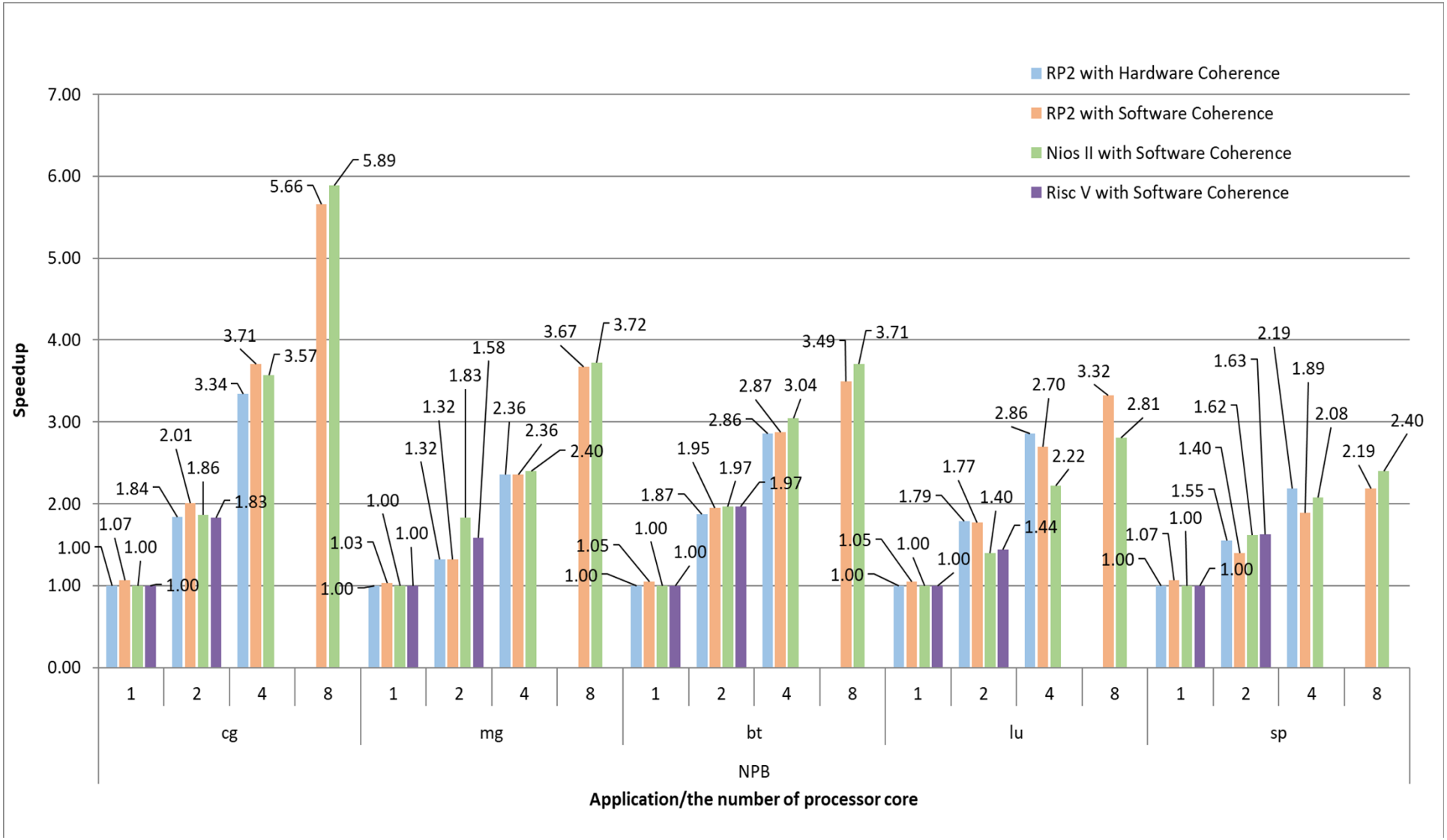
Automatic Software Coherent Control for Manycores

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



OSCAR Software Cache Coherent Control for NIOS and RISCV cores on FPGA

1.86x Speedups for NIOS and 1.83x for RISCV using 2 cores for NPB CG

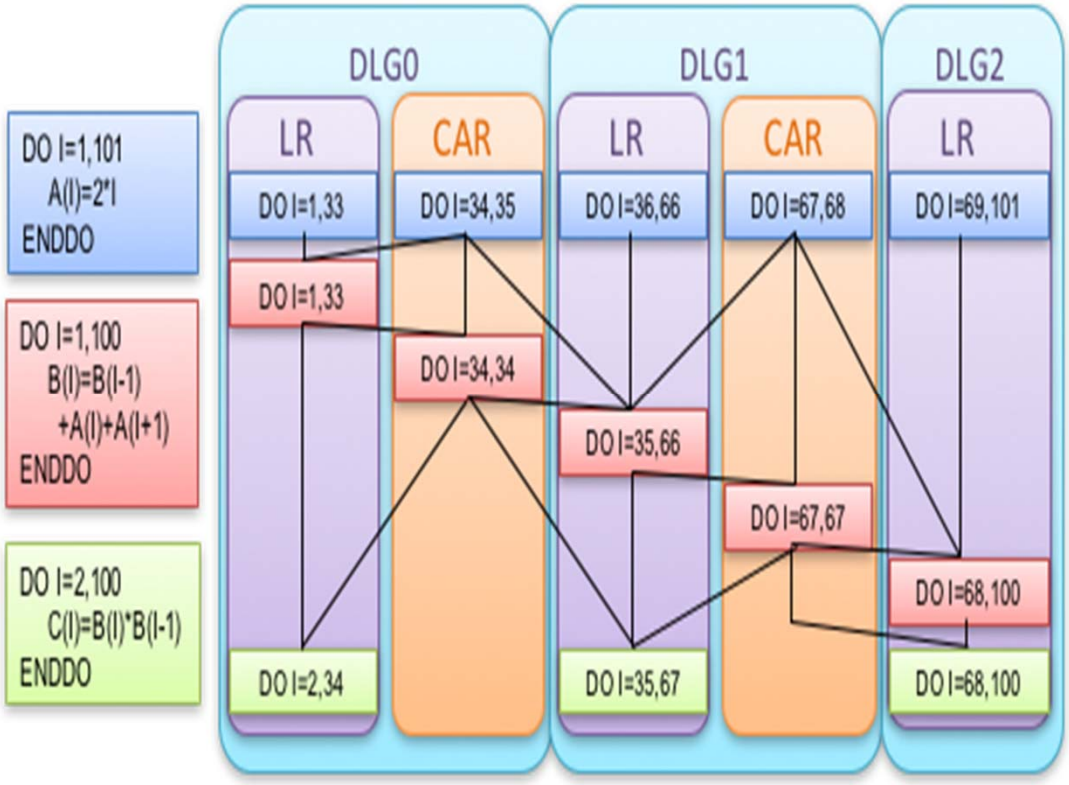


Automatic Local Memory Management

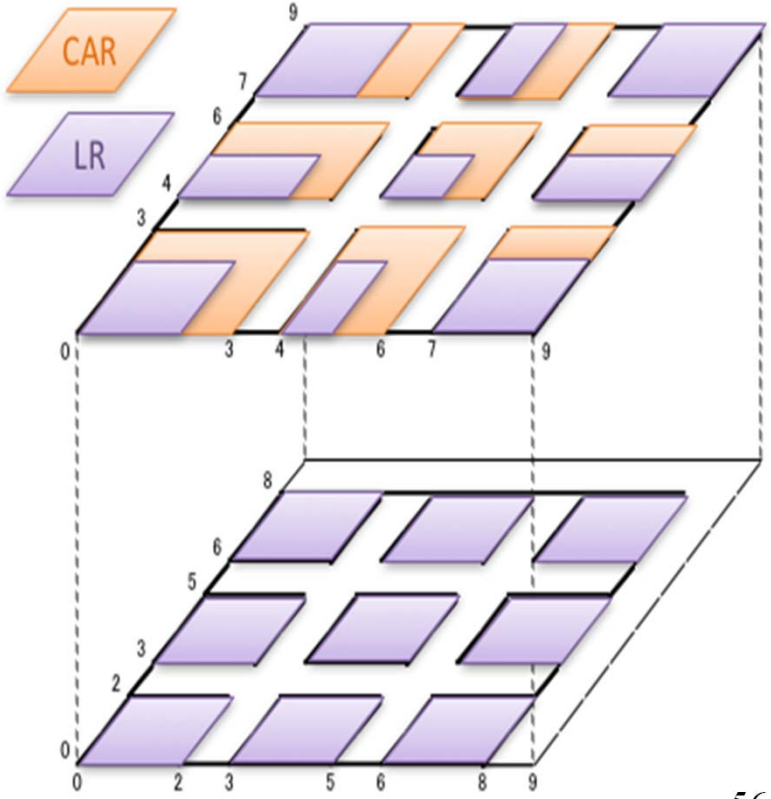
Data Localization: Loop Aligned Decomposition

- **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

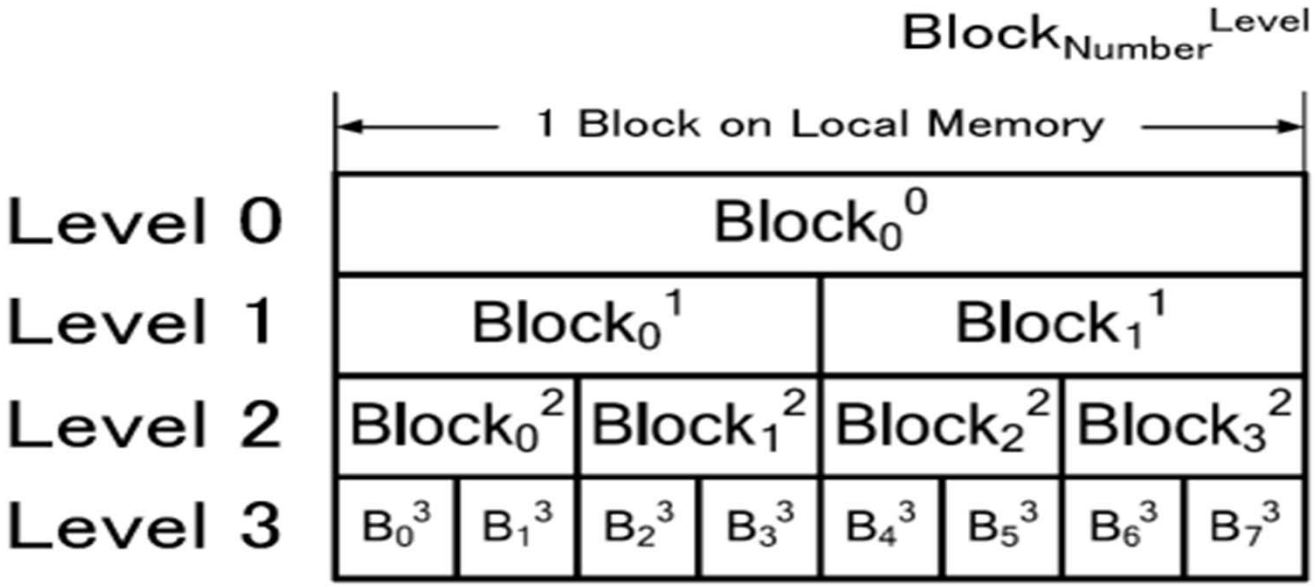


Multi-dimension Decomposition



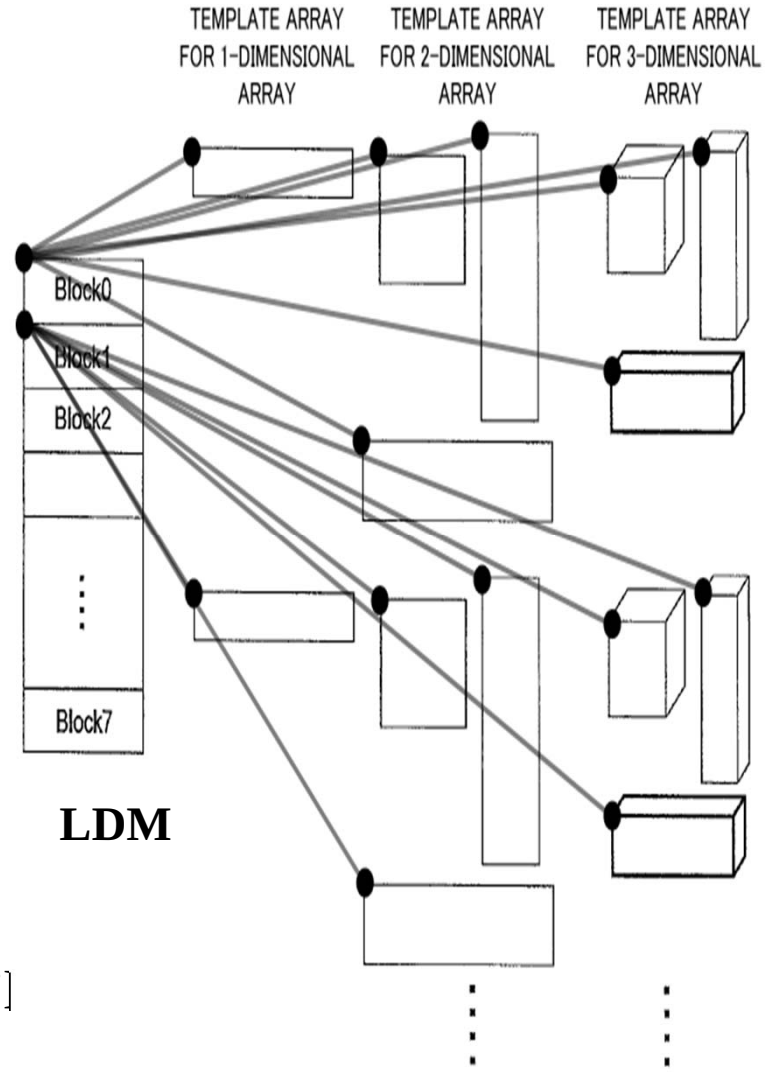
Adjustable Blocks

- 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 divisible size to handle small arrays and scalar variables



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] -> TA[offset + i' * L + j' * M + k']
  - Template Arrays provide readability
    - A[i][j][k] -> TA[Block#][i'][j'][k']

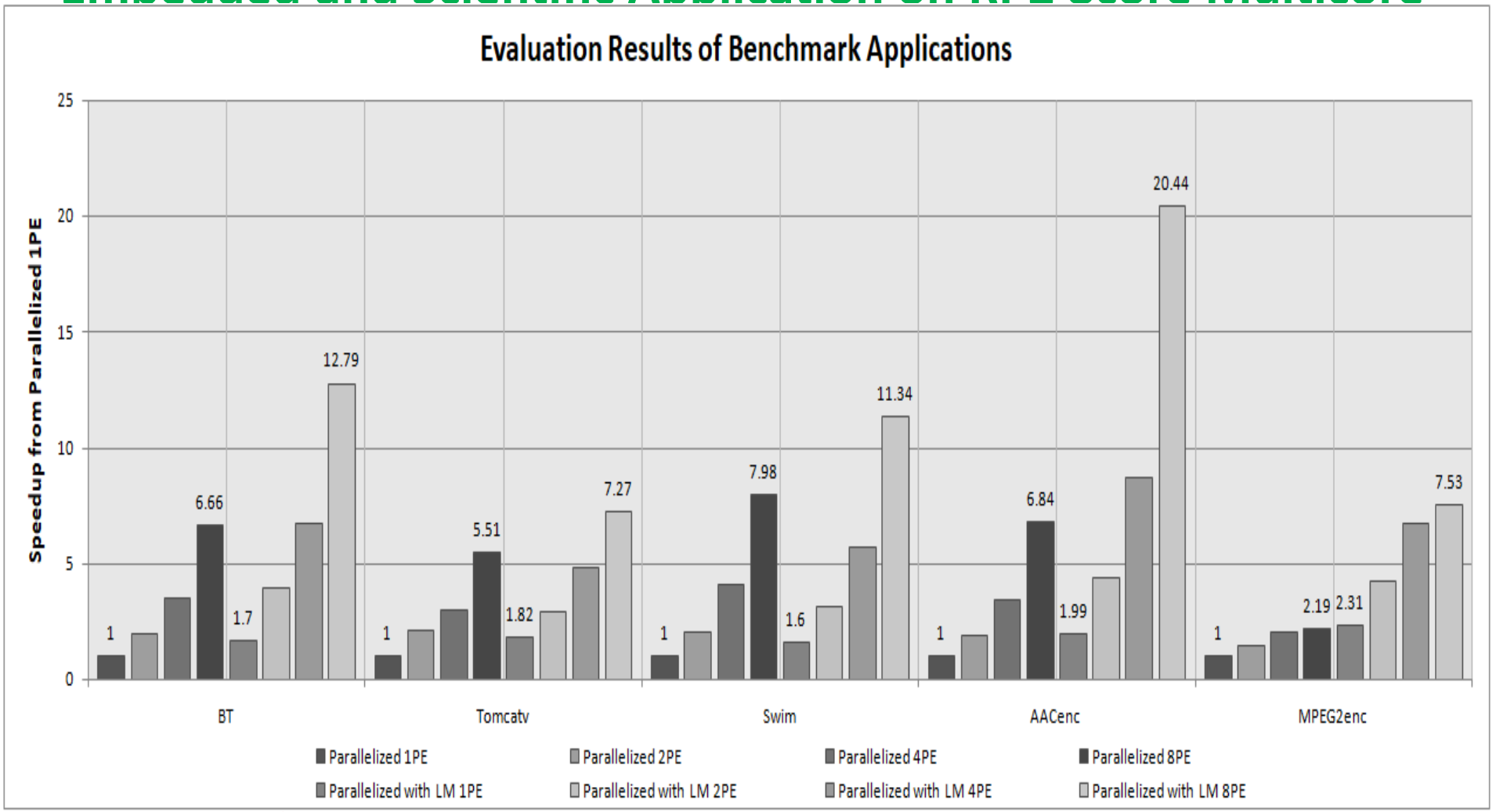


LDM

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

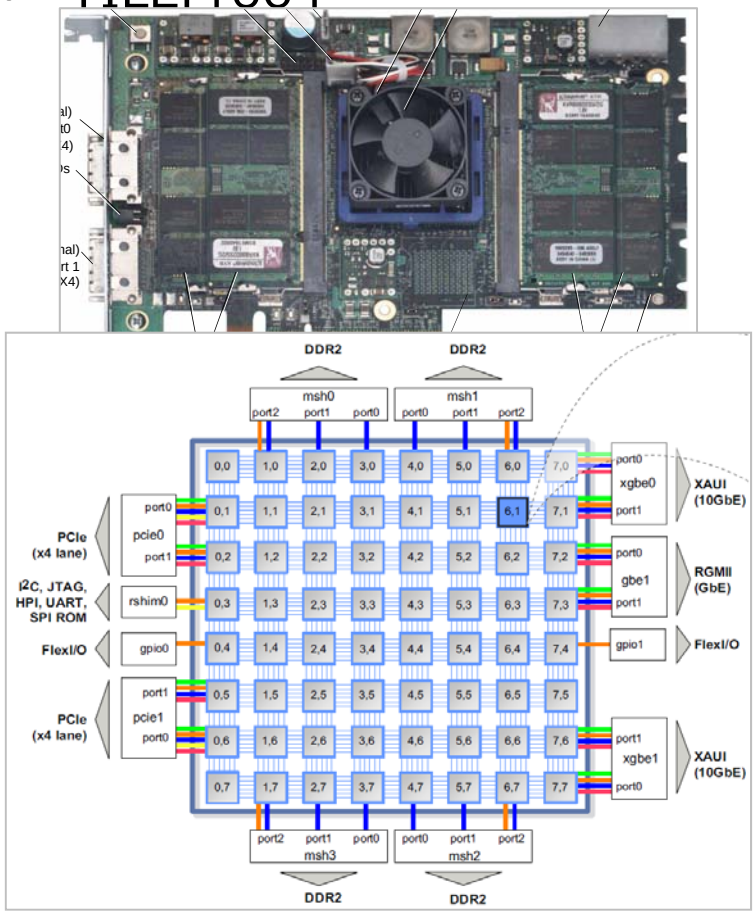


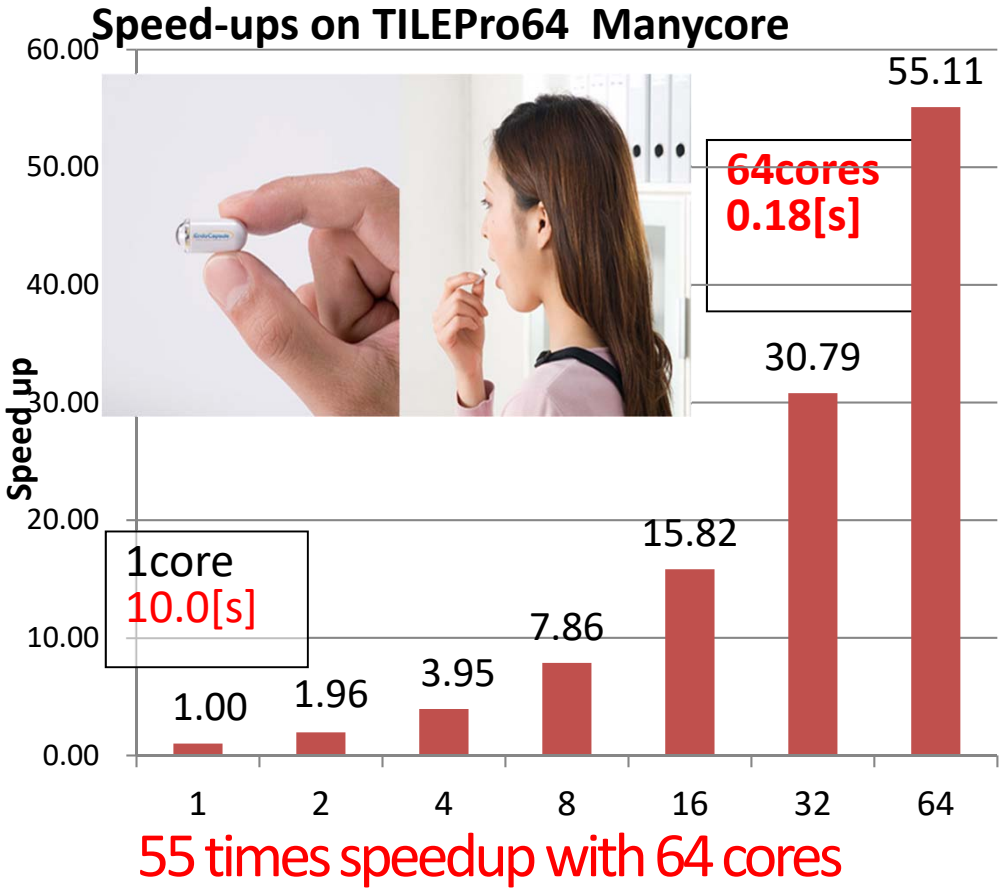
Maximum of 20.44 times speedup on 8 cores using local memory against sequential execution using off-chip shared memory

Automatic Parallelization of JPEG-XR for Drinkable Inner Camera (Endo Capsule)

10 times more speedup needed after parallelization for 128 cores of Power 7. Less than 35mW power consumption is required.

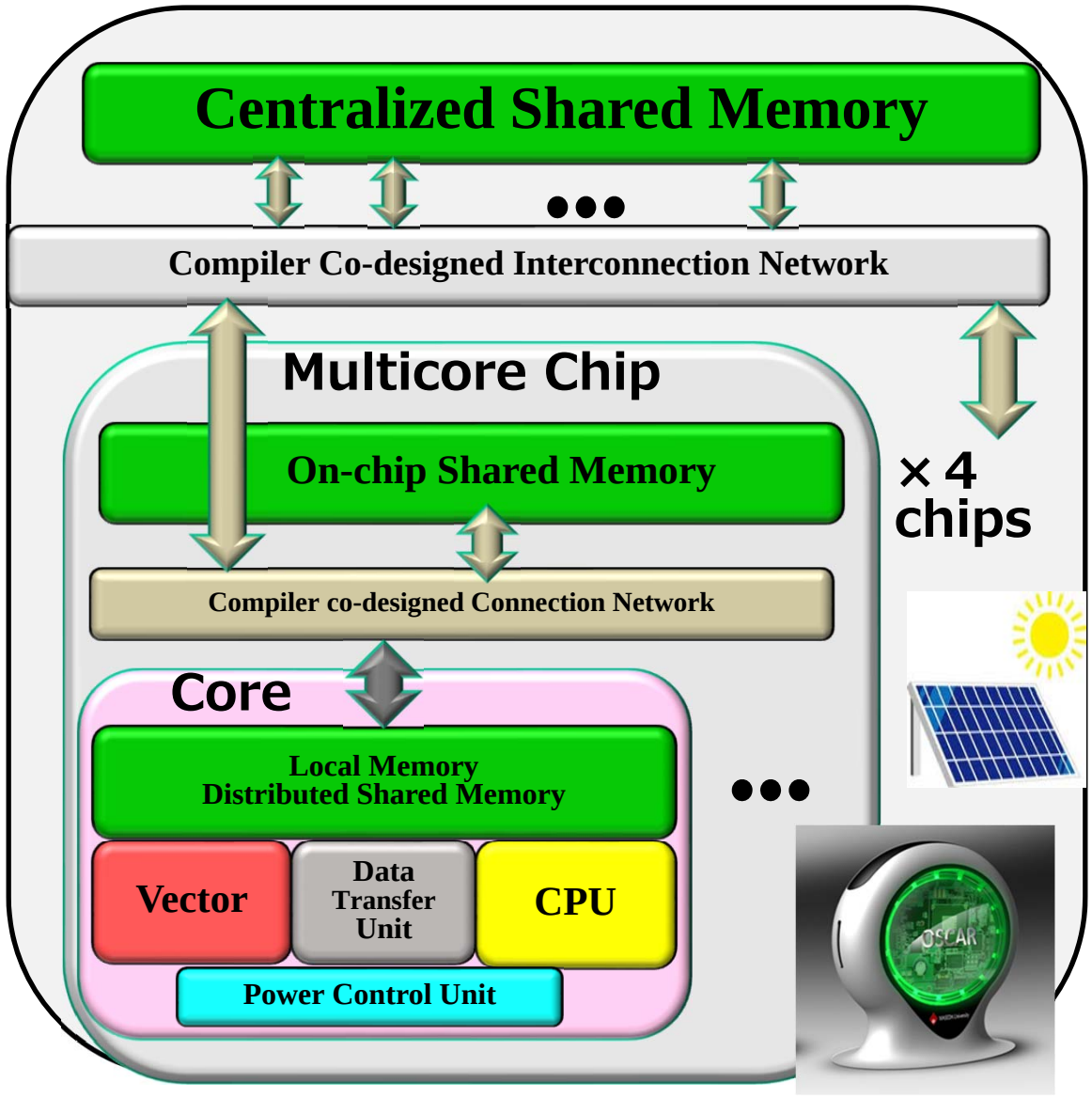
- TILEPro64





55 times speedup with 64 cores

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



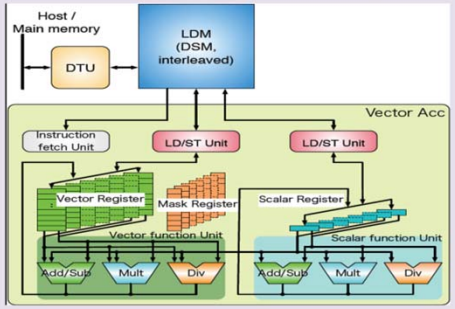
Target:

- **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 with Automatic Parallelizing Compiler

Next Generation Automobiles

- Safer, more comfortable, energy efficient, environment friendly
- Cameras, radar, car2car communication, internet information integrated brake, steering, engine, moter 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 fun, 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 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.
- Local memory management for automobiles and software coherent control have been patented and already realized by OSCAR compiler.