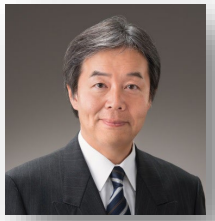
OSCAR Codesigned Compiler and Multicore Architecture



Prof. Hironori Kasahara, IEEE Life-Fellow, IPSJ Fellow

Senior Executive Vice President (2018-2022), Waseda University

IEEE Computer Society President 2018

Board Member: The Academy of Engineering of 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., 1989-90 Research Scholar: U. of Illinois, Urbana-Champaign, Center for Supercomputing R&D, 1997 Prof.,
2004 Director, Advanced Multicore Research Institute,
2017member: the Engineering Academy of Japan (2020-Board Mem) and the Science Council of Japan
2018 IEEE Computer Society President
Senior Vice President, Waseda Univ. (2018 Nov.-2022 Sept.)

AWARD: 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 Asia Academic 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
2019 Spirit of IEEE Computer Society Award,
2020 IPSJ Contribution Award,

Reviewed Papers: 232, Invited Talks: 230,
Granted Patents: 67 (Japan, US, GB, DE, China),
Articles in News Papers, Web News, TV etc.: 697

Committees in Societies and Government 287
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
JST Moonshot Project G3 Robot & AI Vice Chair,
【COCN】Board Member in Council of Competitiveness Nippon, etc.

Some of papers in and just after Ph.D. Course in Waseda U.

IEEE TRANSACTIONS ON COMPUTERS, VOL. C-33, NO. 11, NOVEMBER 1984 1023

Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing

HIRONORI KASAHARA, MEMBER, IEEE, AND SEINOSUKE NARITA, SENIOR MEMBER, IEEE

Courtesy of dexchao - Fotolia.com

104 IEEE JOURNAL OF ROBOTICS AND AUTOMATION, VOL. RA-1, NO. 2, JUNE 1985

Parallel Processing of Robot-Arm Control Computation on a Multimicroprocessor System

HIRONORI KASAHARA MEMBER, IEEE, AND SEINOSUKE NARITA, SENIOR MEMBER, IEEE

2nd International Conference on Superecomputing
Santa Clara, CA, USA May 3-8,1987

A PARALLEL PROCESSING SCHEME FOR THE SOLUTION OF SPARSE LINEAR EQUATIONS USING STATIC OPTIMAL-MULTIPROCESSOR-SCHEDULING ALGORITHMS

H.Kasahara*, T.Fujii*, H.Nakayama*, S.Narita*, and Leon O.Chua**

* Dept. of Electrical Eng., Waseda University, Tokyo, 160, Japan
** Dept. of Electrical Eng. and Computer Sciences, University of California, Berkeley, CA 94720, U.S.A.

PARALLEL PROCESSING OF ROBOT MOTION SIMULATION

H. Kasahara, H. Fujii and M. Iwata

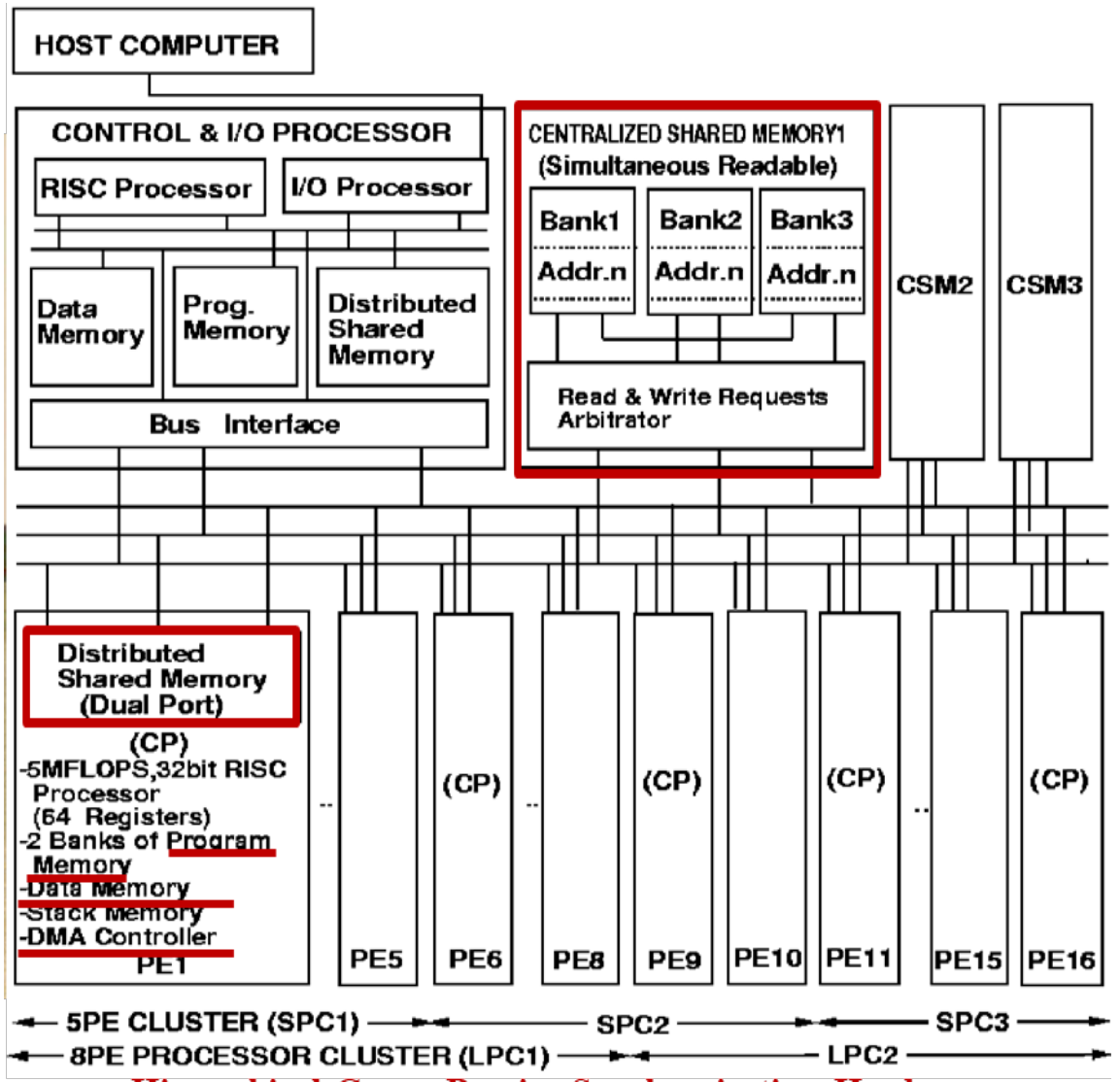
Department of Electrical Engineering, Waseda University, 3–4–1 Ohkubo Shinjuku-ku, Tokyo 160, Japan

The First Compiler Codesigned Multiprocessor

OSCAR (Optimally Scheduled Advanced Multiprocessor) in 1987

AMD29325 32-bit Floating-point unit

H. Kasahara, "OSCAR Fortran Multigrain Compiler", Stanford University, Hosted by Professor John L. Hennessy and Professor Monica Lam, May. 15. 1995.



AMD29325 32-bit Floating-point unit

Hierarchical Group Barrier Synchronization Hardware

SIGMA-1

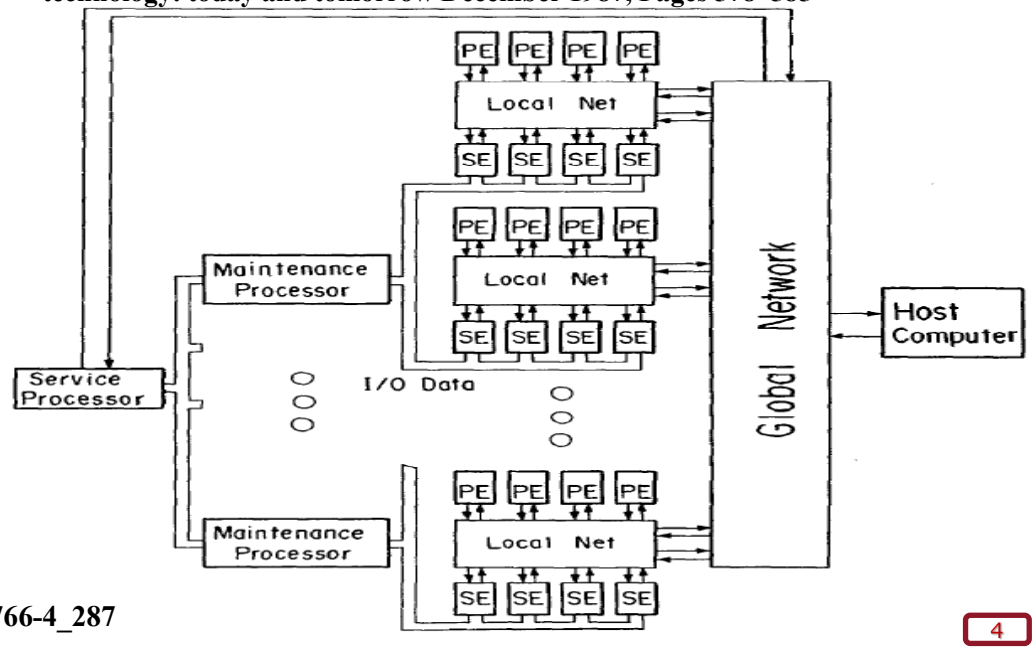
ETL(Electrotechnical Laboratory), Ministry of International Trade and Industry, **Japan**. **(AIST, METI)**

SIGMA-1 is a large-scale computer based on **fine-grain dataflow architecture** designed to show feasibility of fine-grain dataflow computer to highly parallel computation over conventional von Neumann computers. SIGMA-1 project was stated on 1984 and the **128 processing element (PE) started working on 1988**. SIGMA-1 was design and built at the Electrotechnical Laboratory (ETL for short), Ministry of International Trade and Industry, Japan. SIGMA-1 is still the largest scale dataflow computer so far built and it achieved more 100 Mflops as a maximum measured performance of the total system. As for the language for SIGMA-1**, ETL developed Dataflow-C language** as a subset of C programming language. Dataflow-C is **a single assignment language** that can compile C-like source programs to highly parallel executable dataflow machine codes (Fig. 1).

The SIGMA-1 Dataflow Computer

Toshitsugu Yuba, Kei Hiraki, Toshio Shimada, Satoshi Sekiguchi and Kenji Nishida

ACM '87: Proceedings of the 1987 Fall Joint Computer Conference on Exploring technology: today and tomorrow December 1987, Pages 578–585



https://link.springer.com/referenceworkentry/10.1007/978-0-387-09766-4_287

https://ieeexplore.ieee.org/document/130111/similar#similar

Parallel processing of near fine grain tasks using static scheduling on OSCAR (optimally scheduled advanced multiprocessor)

Publisher: IEEE | Cite This | PDF

H. Kasahara ; H. Honda ; S. Narita All Authors

Published in: Supercomputing '90:Proceedings of the 1990 ACM/IEEE Conference on Supercomputing

https://ieeexplore.ieee.org/document/1653942

Journals & Magazines > Computer > Volume: 15 Issue: 2

A Second Opinion on Data Flow Machines and Languages

Publisher: IEEE | Cite This | PDF

Gajski ; Padua ; Kuck ; Kuhn All Authors

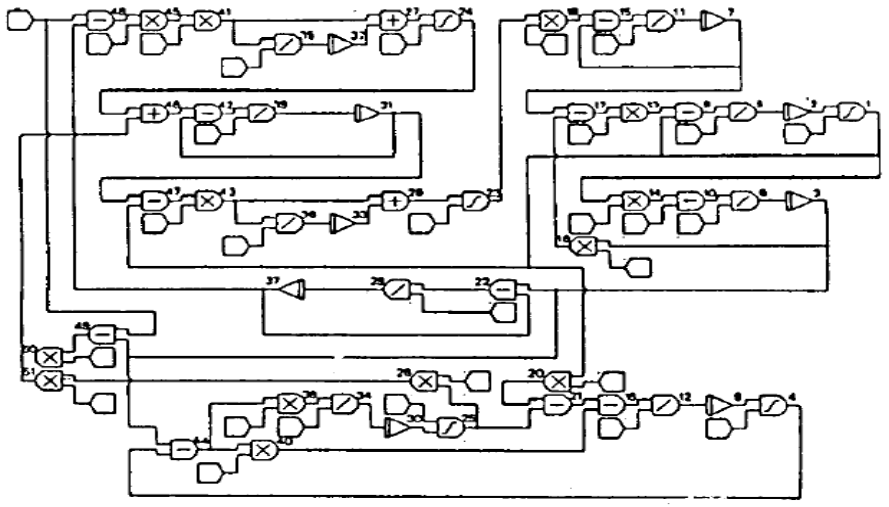
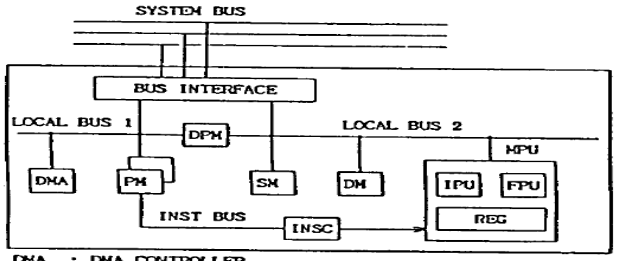


Fig.10 A kind of dataflow graph with near fine grain tasks



DMA : DMA CONTROLLER
PM : PROGRAM MEMORY (128KW×2BANK)
INSC : INSTRUCTION CONTROL UNIT
DPM : DUAL PORT MEMORY (32bit×2KW)
SM : STACK MEMORY (32bit×4KW)
DM : DATA MEMORY (32bit×256KW)
MPU : MAIN PROCESSING UNIT
IPU : INTEGER PROCESSING UNIT
FPU : FLOWTING PROCESSING UNIT
REG : REGISTER FILE (32bit×64lines)
INST BUS : INSTRUCTION BUS

Fig.2 OSCAR's processor element

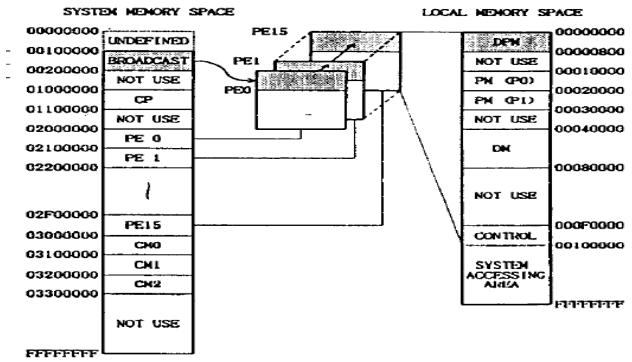


Fig.3 OSCAR's memory space

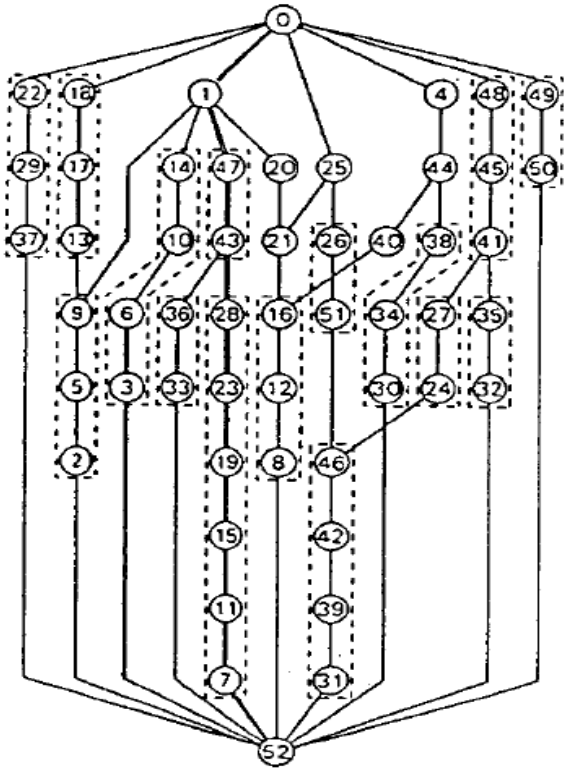
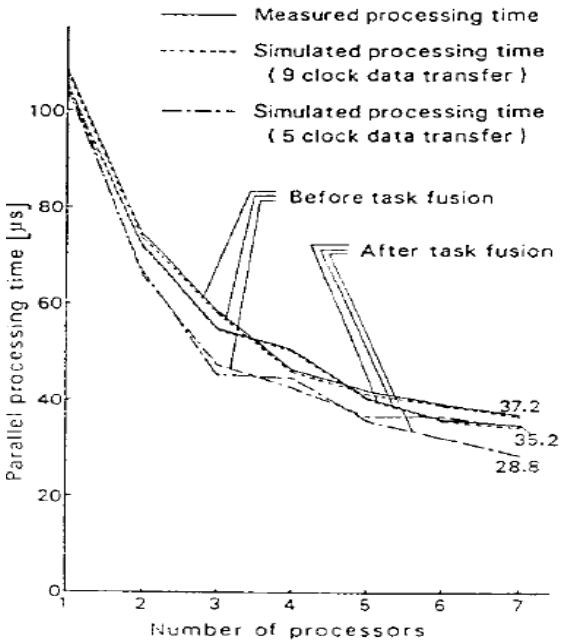


Fig.11 Task graph for Fig. 10



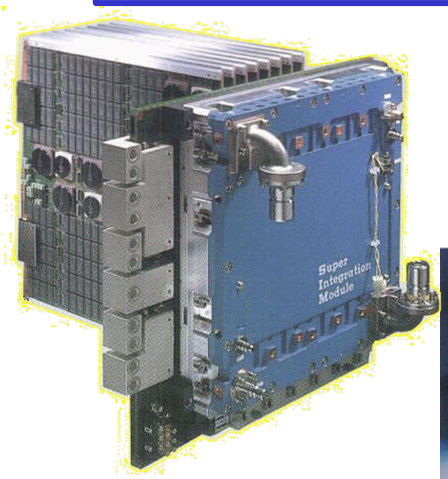
Parallel processing time of fig.10 on OSCAR

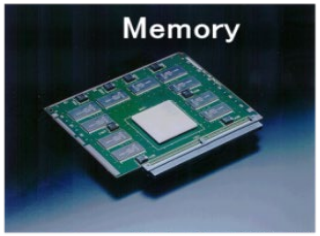
tioned in Section 3.1. The measured processing times were reduced from 108.7 [us] and 105.8 [us] for one PE to 37.2 [us] (1/2.92) and 35.2 [us](1/3.01) for seven PEs respectively.

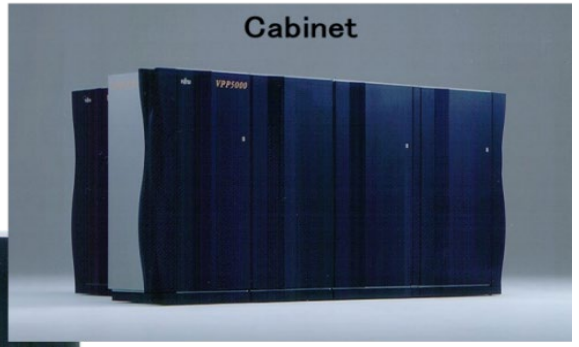
1993年 スーパーコンピュータVPP500、数値風洞(NWT)

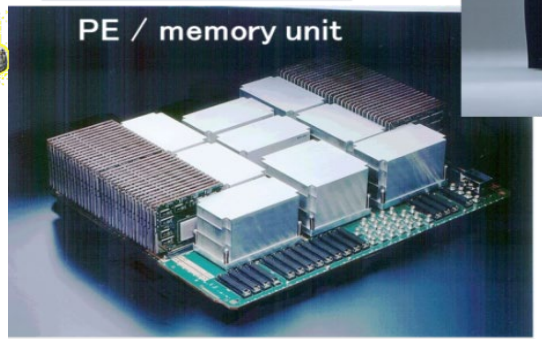
Mr. Hajime Miyoshi

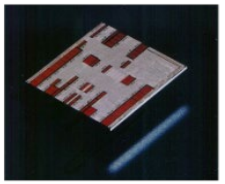
ACM/IEEE SC '94: Washington, D.C. November, 1994にて発表



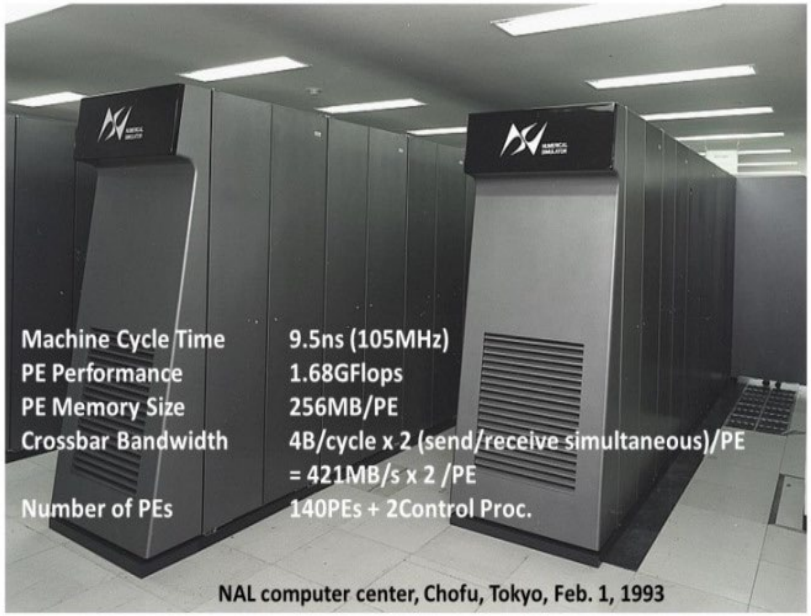




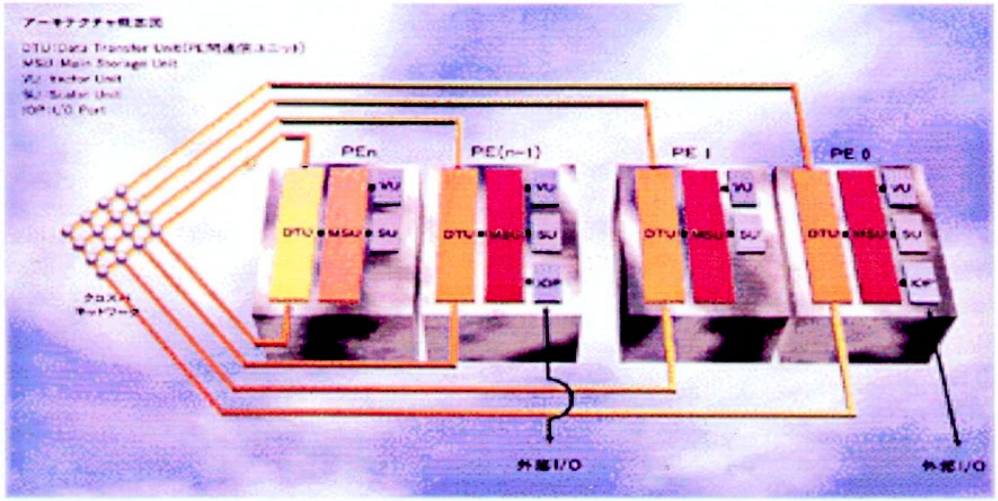




スパーコンピュータNWTの外観



商用VPP5000 (仏気象庁他)



Cedar Supercomputer

University of Illinois at Urbana-Champaign, CSRD (Center for Supercomputing R&D)

Prof. David Kuck

Univ. Illinois at Urbana-Champaign
Prof. Emeritus
Founder of Kuck & Associates

- **ILLIAC IV (Member)**
- **Parallelizing Compiler**
  - **Data dependence analysis**
  - **Automatic vectorization**
  - **Automatic loop parallelization**
  - **Loop restructuring**
  - **Cedar supercomputer (Alliant FX8)**
  - **OpenMP**

Intel Senior Fellow (Past CTO):

- **Intel Parallel Tuning Tools & Debuggers**

- **National Academy of Engineering Member**
- **IEEE ACM Eckert-Mauchly Award**
- **IEEE Charles Babbage Award**
- **ACM/IEEE Ken Kennedy Award**
- **Okawa Prize**
- **IEEE Computer Pioneer Award**

Hironori visited CSRD from 1989 to 1990.

Earth Simulator

2021年ノーベル物理学賞
プリンストン大 真鍋淑郎先生
大気・海洋大循環モデル

(http://www.es.jamstec.go.jp/)

Mr. Hajime Miyoshi

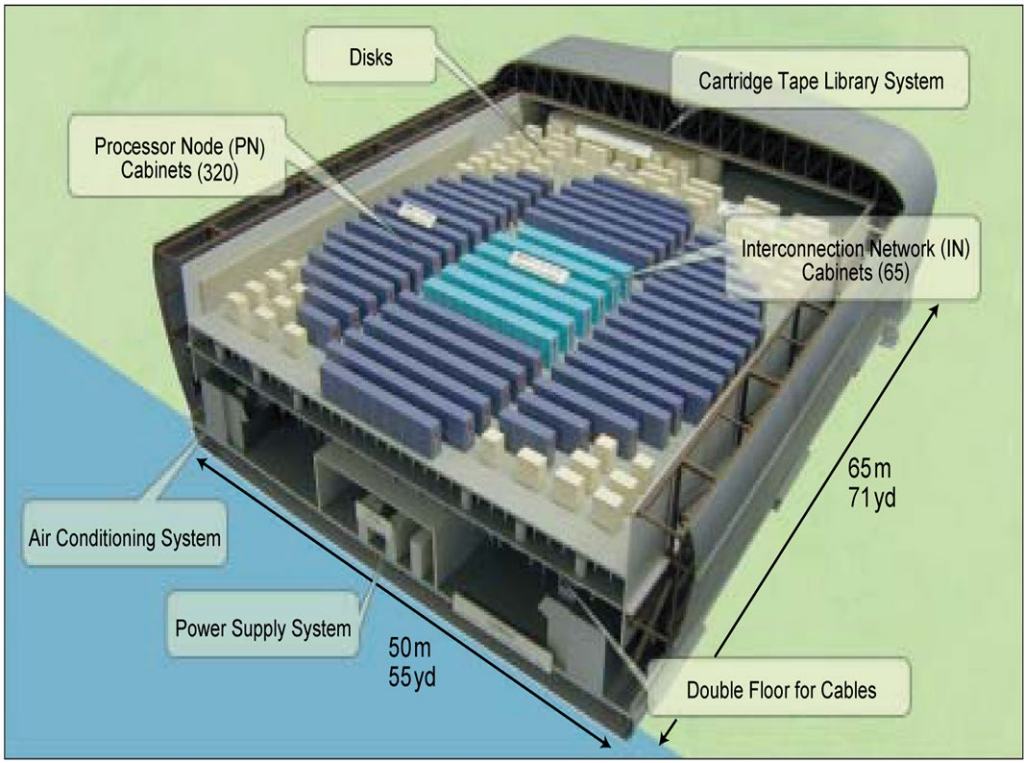
- Earth Environmental simulation like Global Warming, El Nino, PlateMovement for the all lives onr this planet.
- Developed in Mar. 2002 by STA (MEXT) and NEC with 400 M$ investment under Dr. Miyoshi's direction.
  (Dr.Miyoshi: Passed away in Nov.2001. NWT, VPP500, SX6)

40 TFLOPS Peak ($40*10^{12}$)

35.6 TFLOPS Linpack

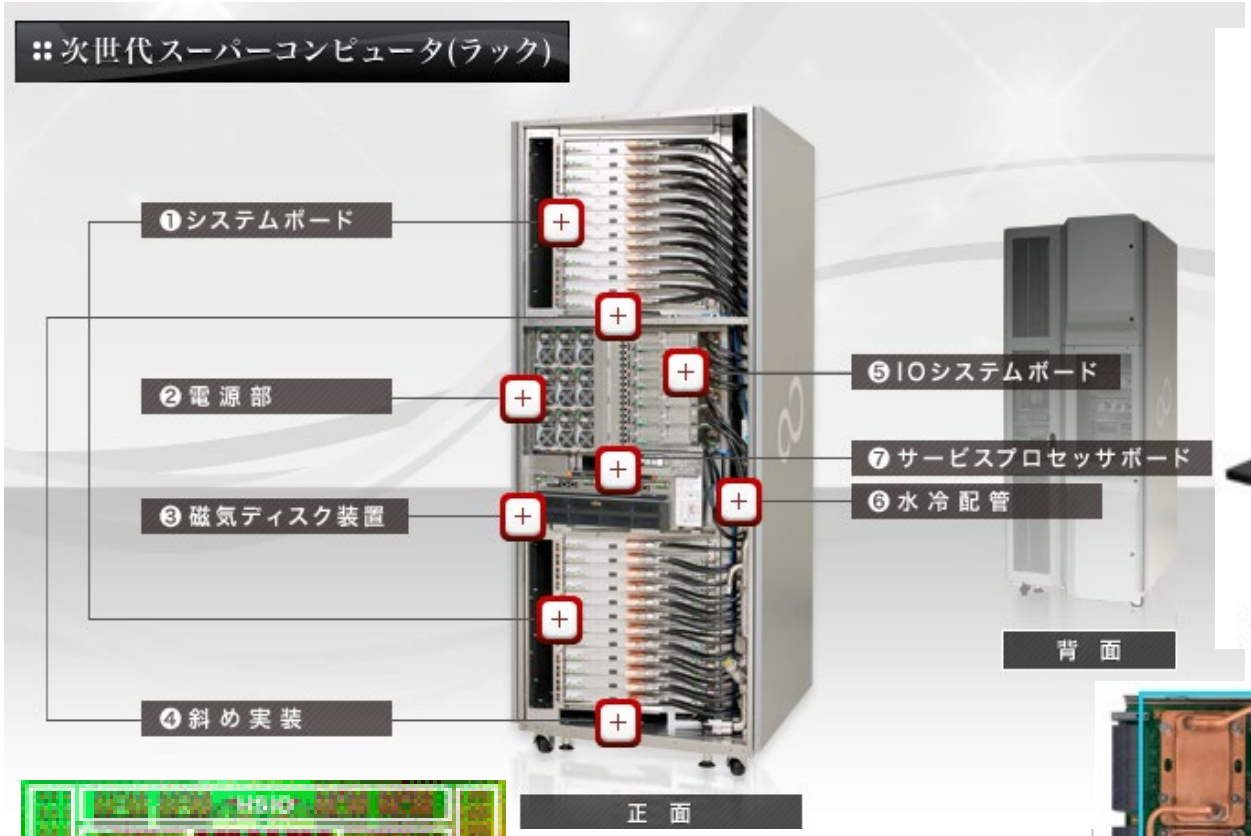
Image of Earth Simulator

4 Tennis Courts



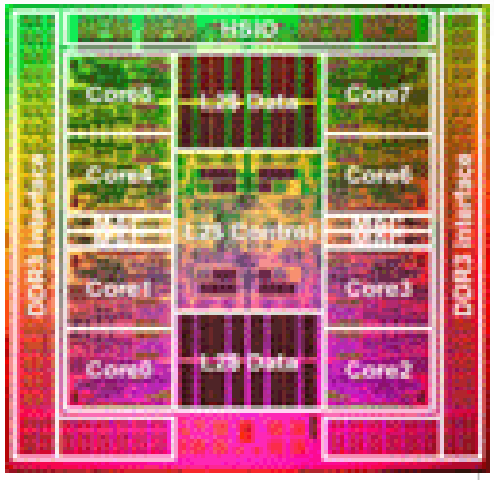
June 2002 Top1
Cores: 5,120, Rmax:35.86TFlop/s
Rpeak: 40.96TFlop/s, Power: 3.2MW

“K” Supercomputer by Riken, No.1 in TOP500, June 20 & Nov.2,2011

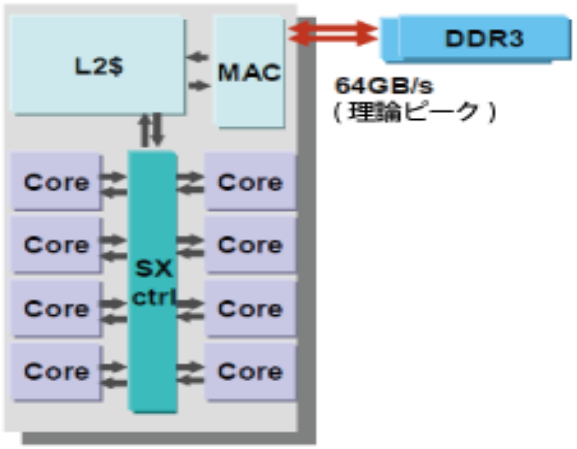


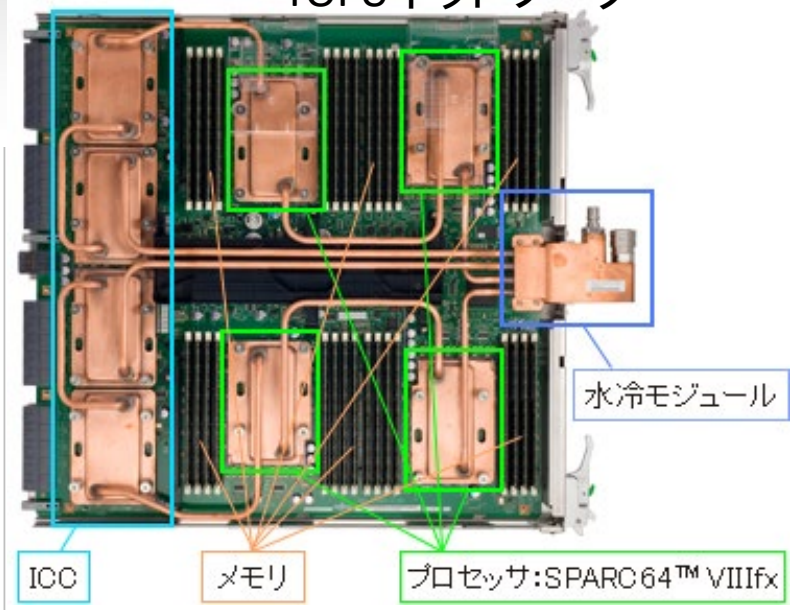
6次元メッシュ/トーラス(概念模型)

TOFUネットワーク



SPARC64™ VIIIfx
(提供:富士通(株))





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)

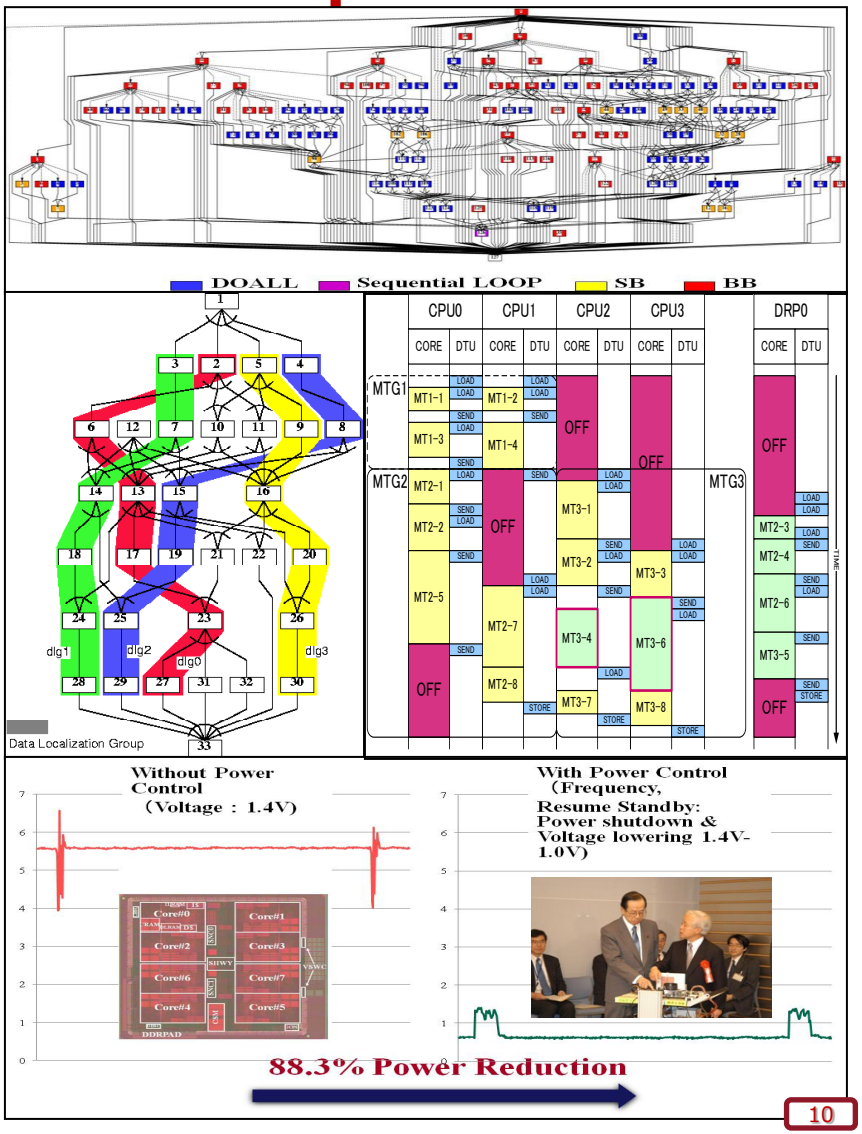
Data Transfer Overlapping(2016 partially)

Data transfer overlapping using Data Transfer Controllers (DMAs)

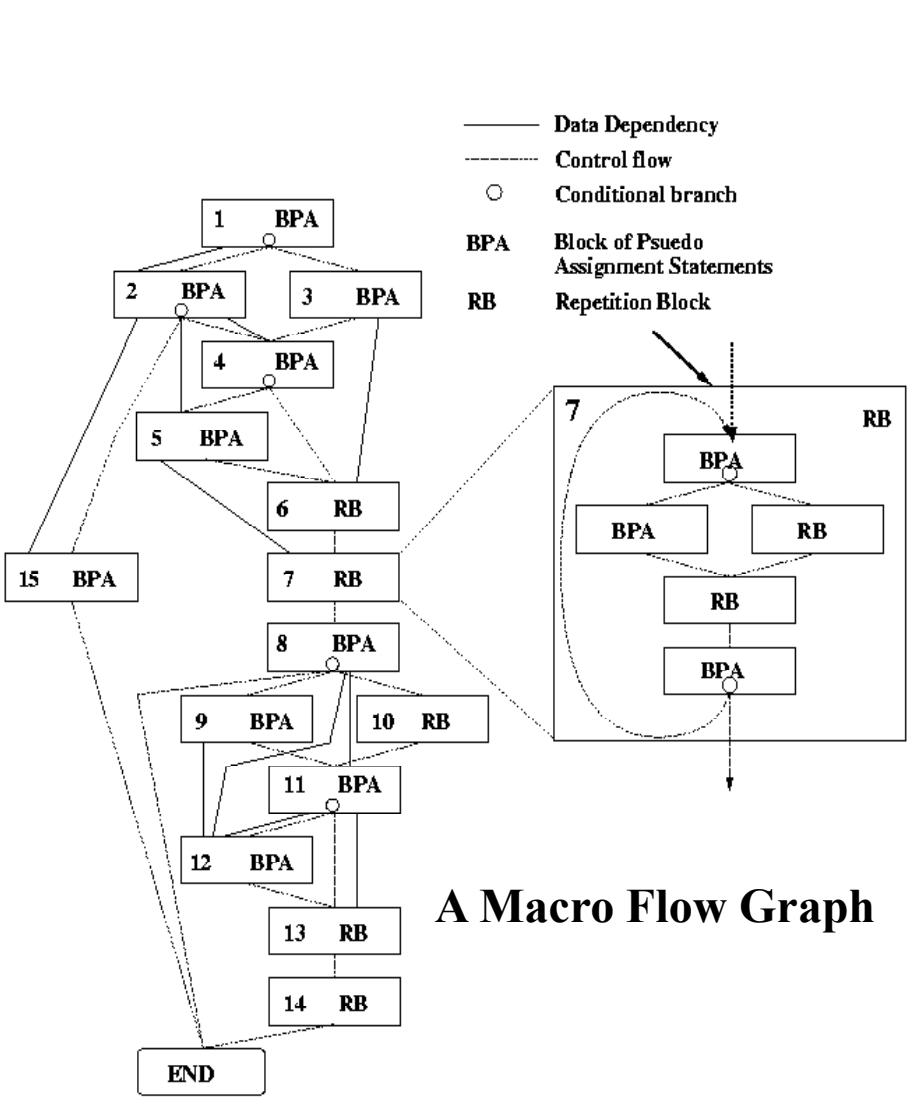
Power Reduction

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

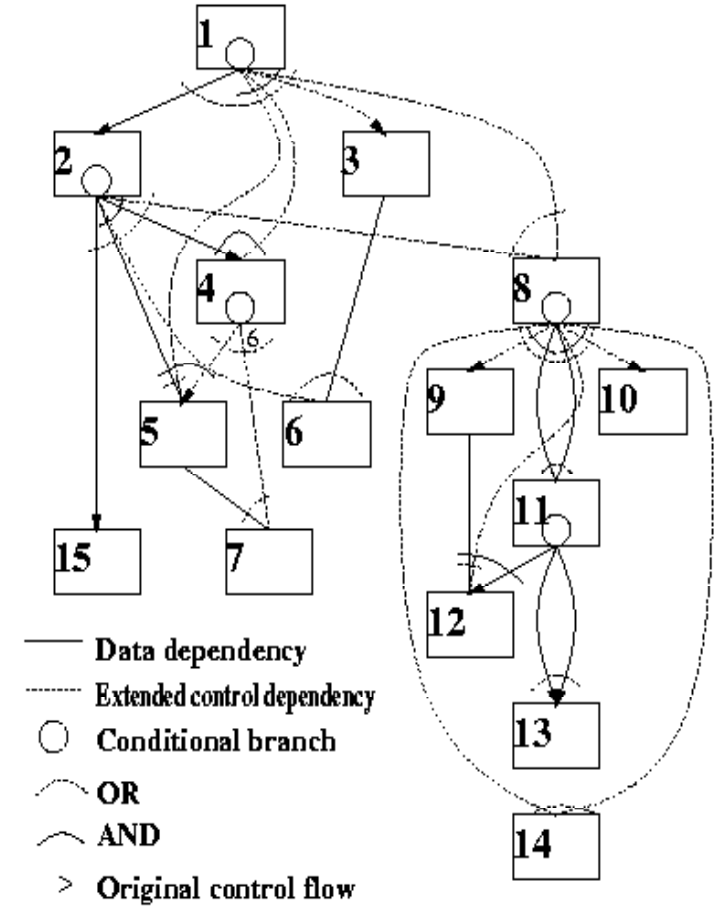
Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.



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



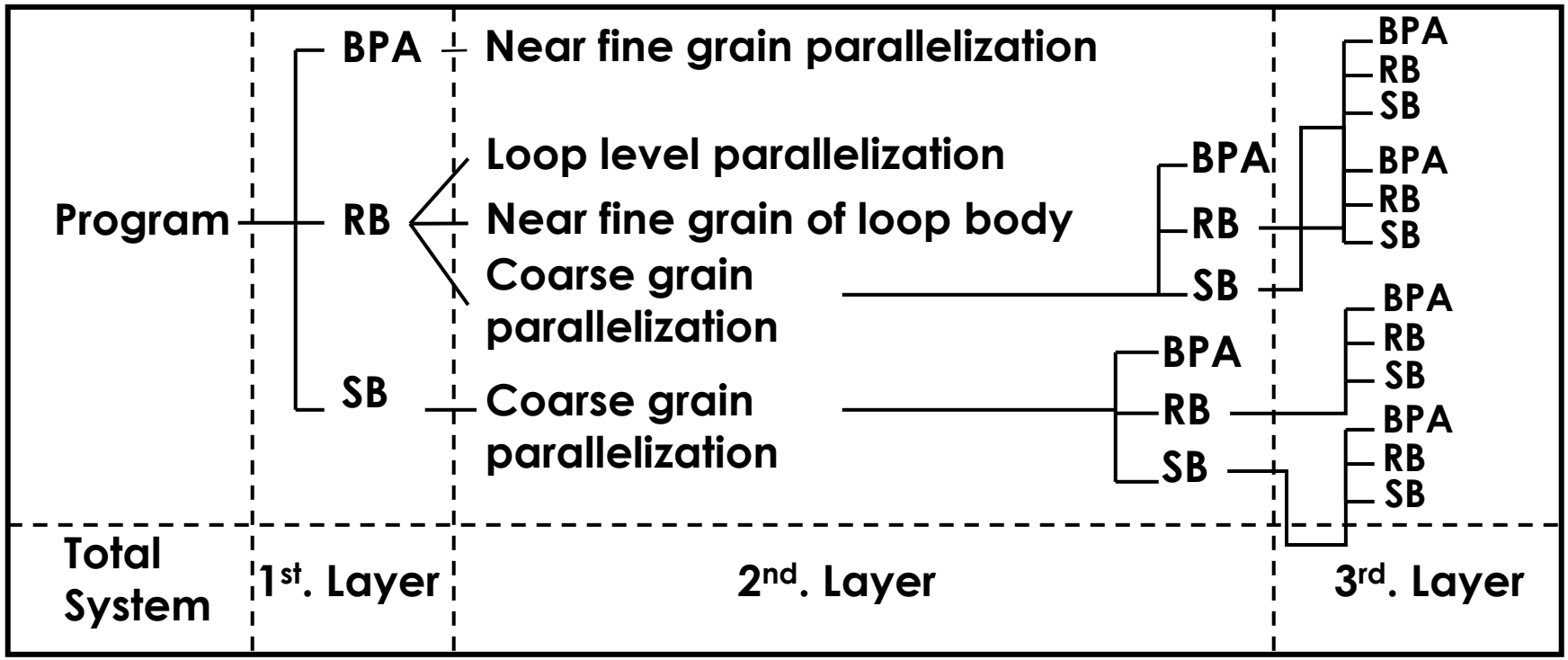
A Macro Flow Graph



A Macro Task Graph

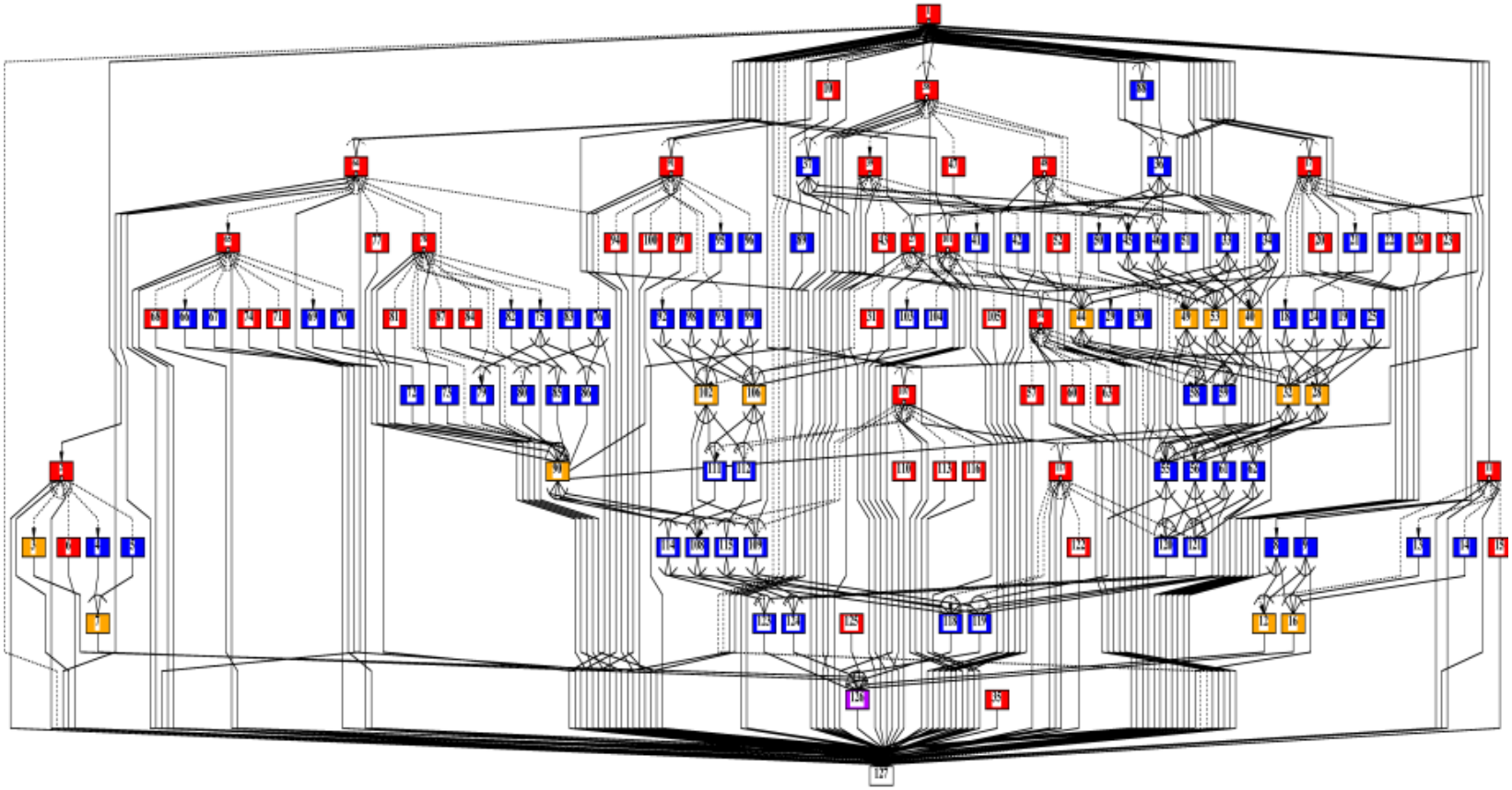
Generation of Coarse Grain Tasks

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



MTG of Su2cor-LOOPS-DO400

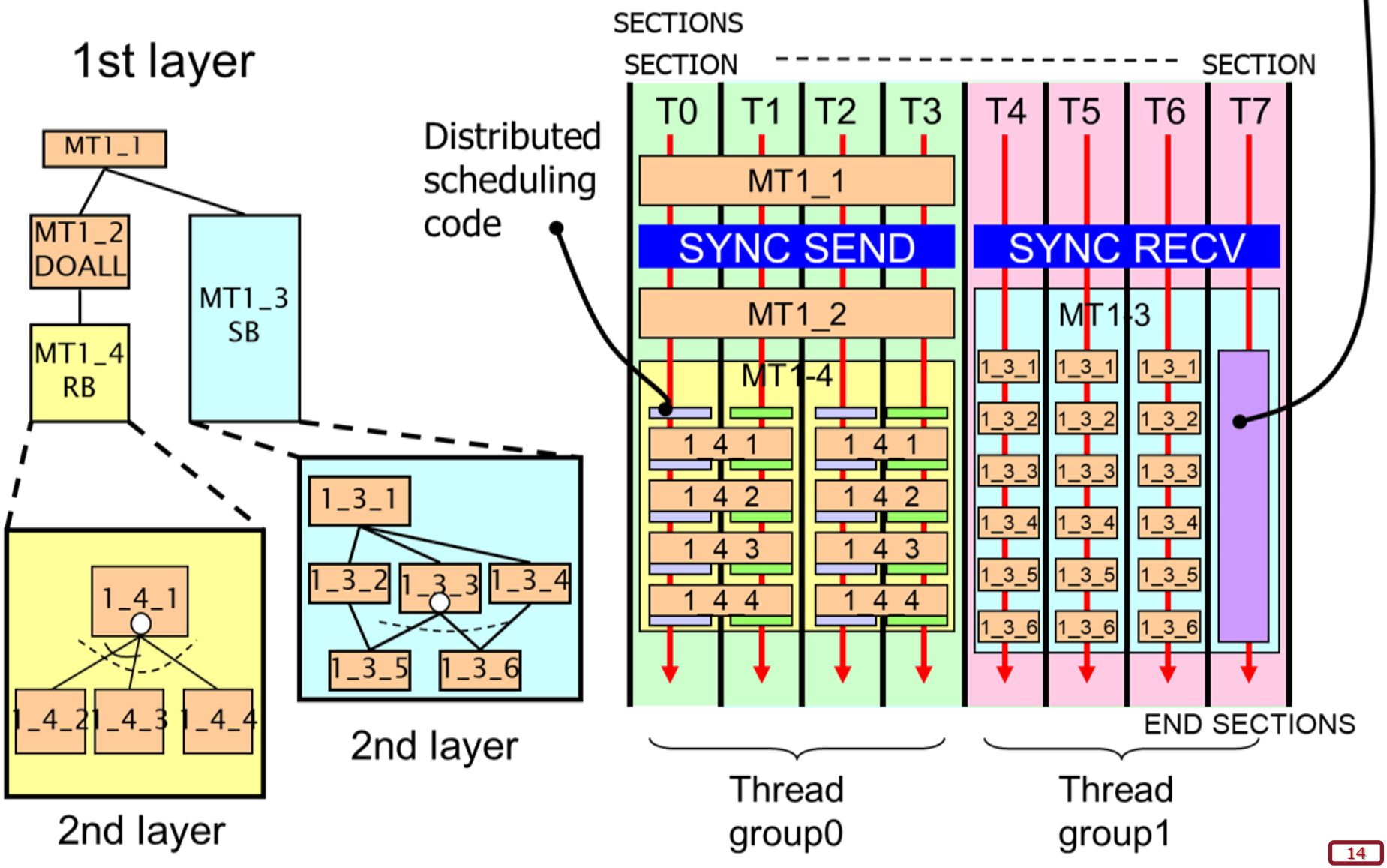
- **Coarse grain parallelism PARA_ALD = 4.3**



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

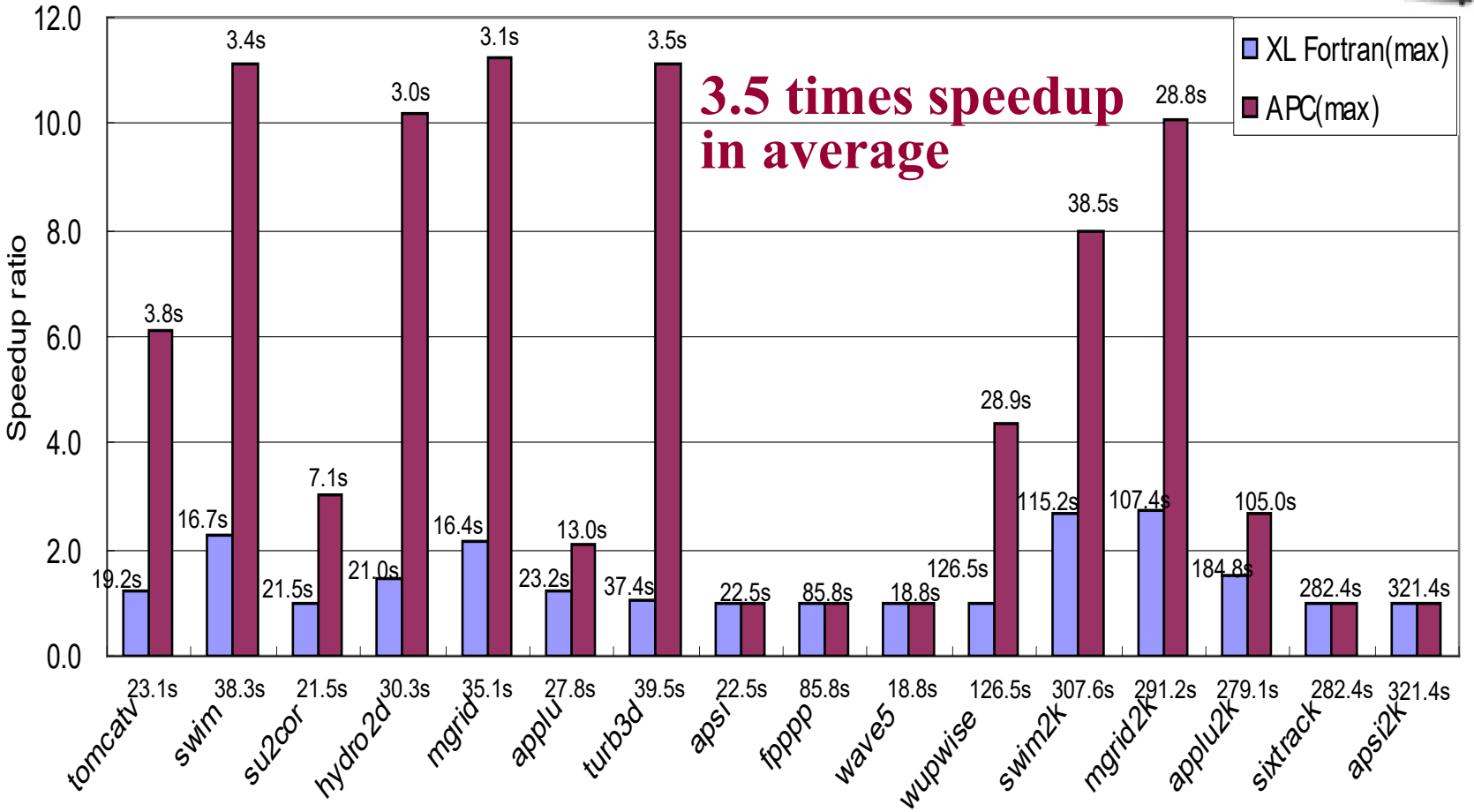
Generated Multigrain Parallelized Code

(The nested coarse grain task parallelization is realized by only OpenMP "section", "Flush" and "Critical" directives.)

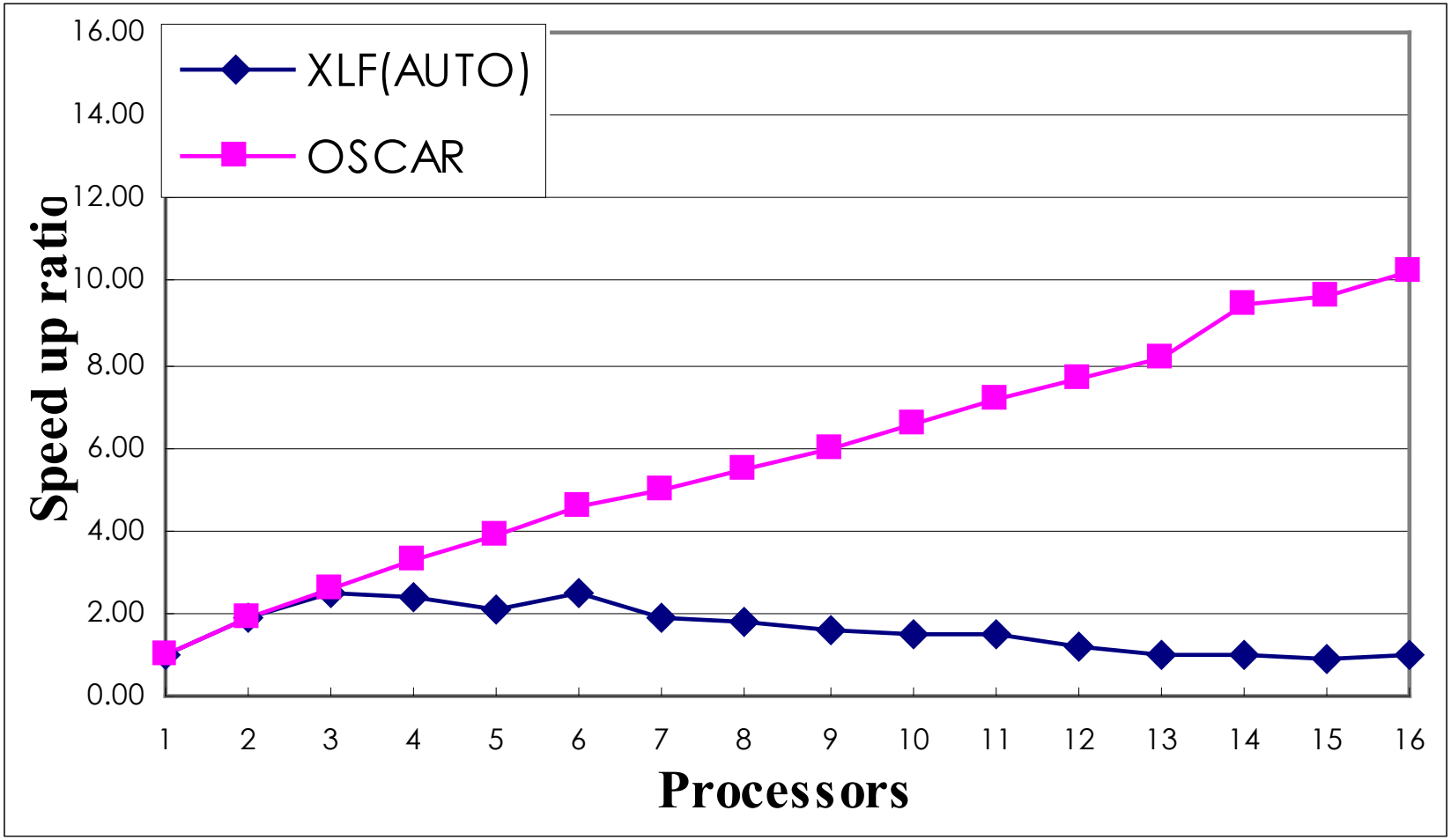


Performance of APC Compiler on IBM pSeries690 16 Processors High-end Server

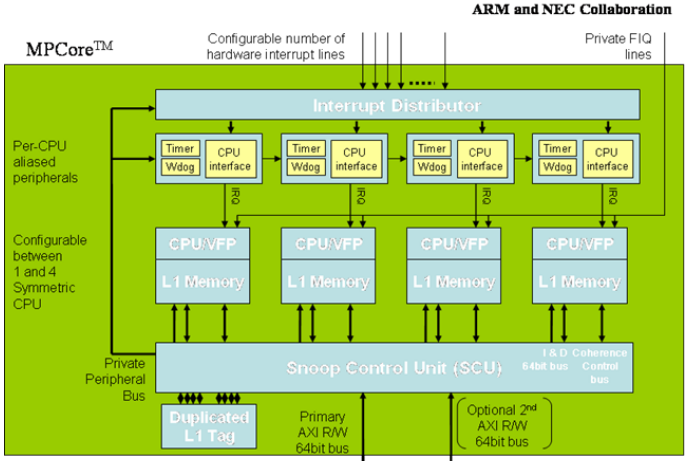
- IBM XL Fortran for AIX Version 8.1
  - Sequential execution : -O5 -qarch=pwr4
  - Automatic loop parallelization : -O5 -qsmp=auto -qarch=pwr4
  - OSCAR compiler : -O5 -qsmp=noauto -qarch=pwr4 (su2cor: -O4 -qstrict)

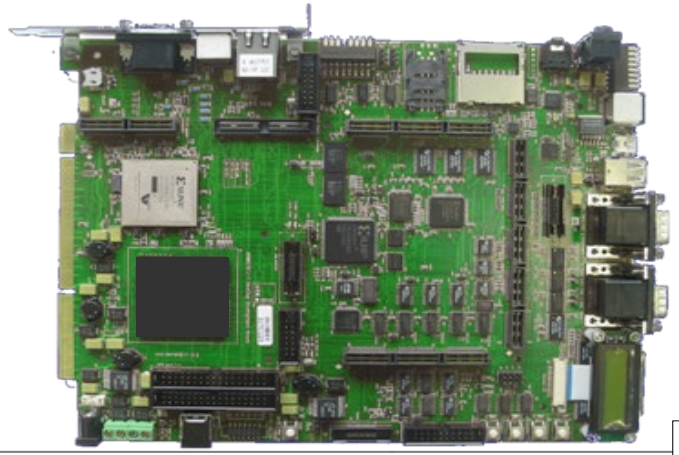


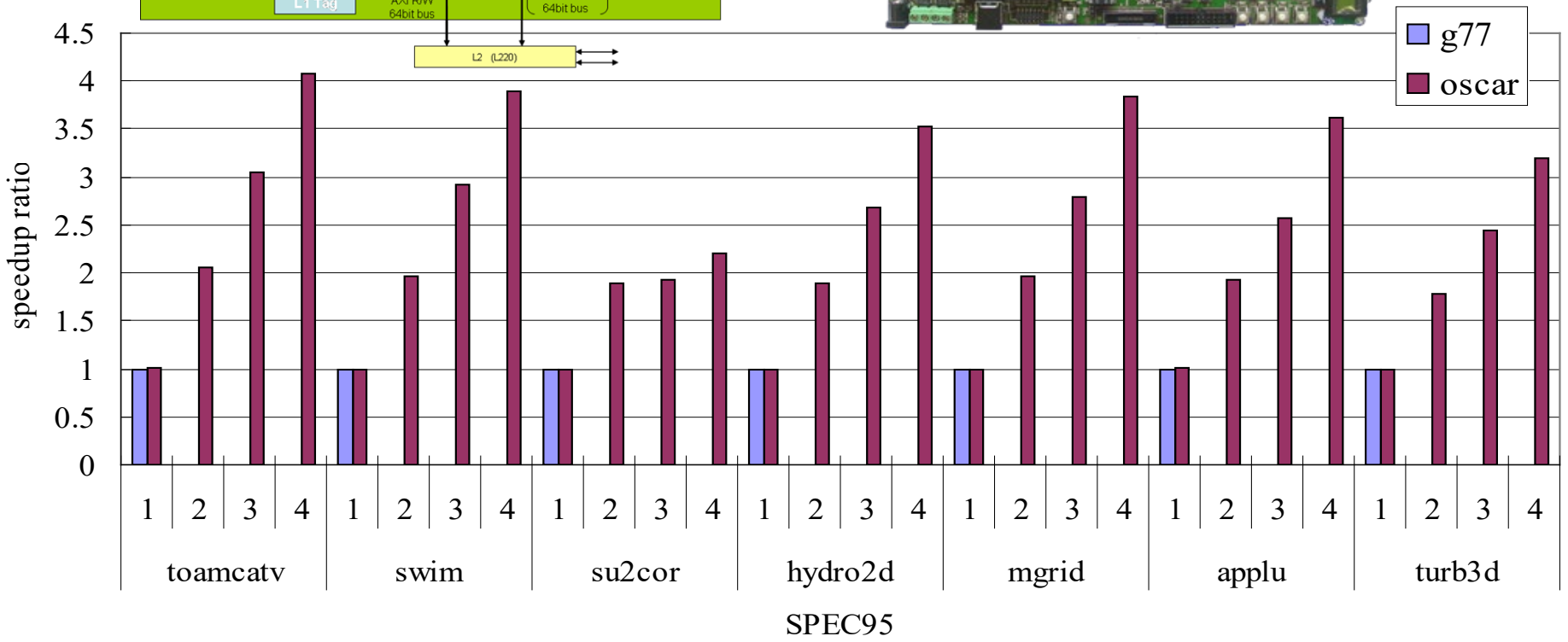
Performance of Multigrain Parallel Processing for 102.swim on IBM pSeries690



NEC/ARM MPCore Embedded 4 core SMP

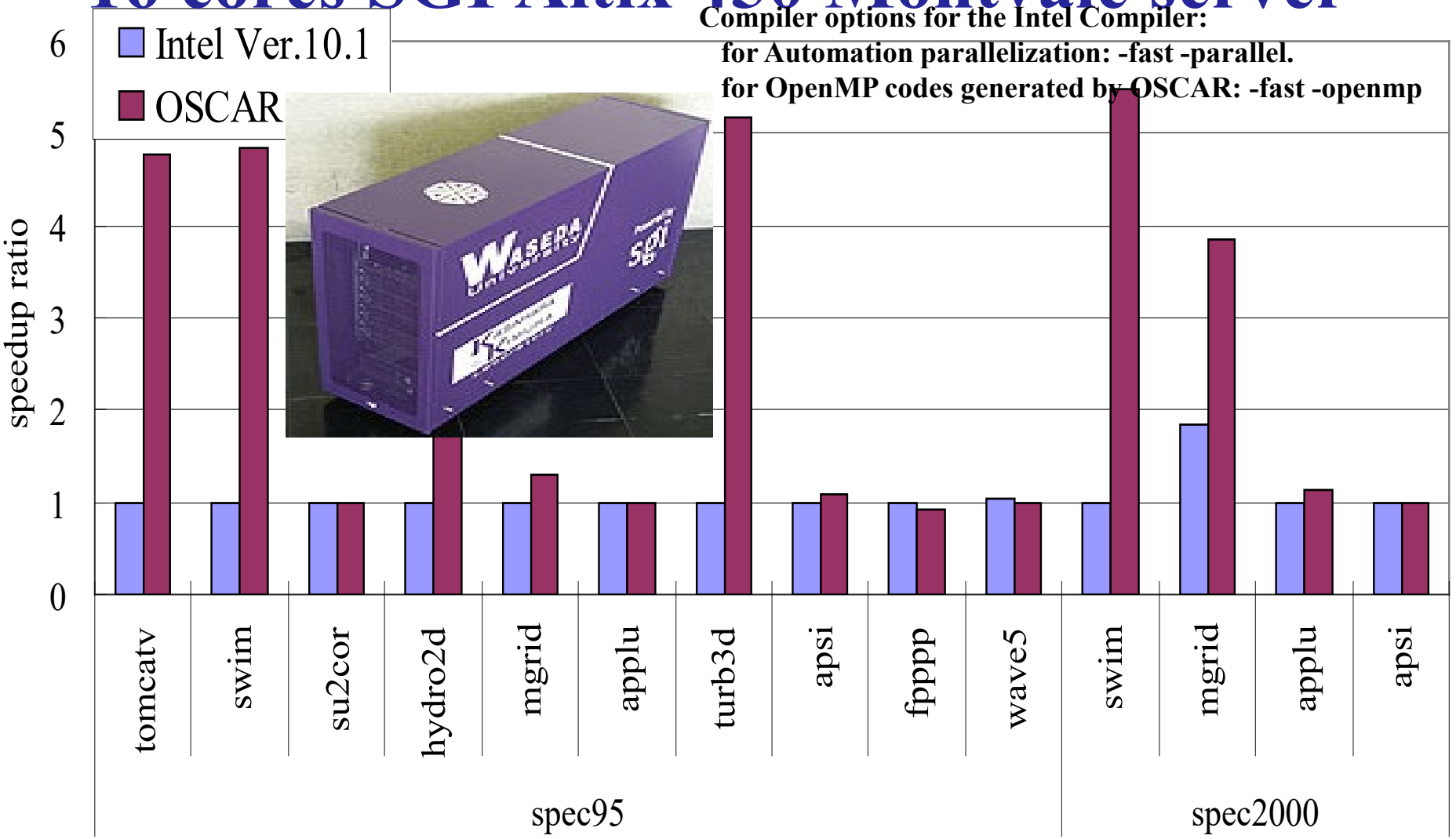






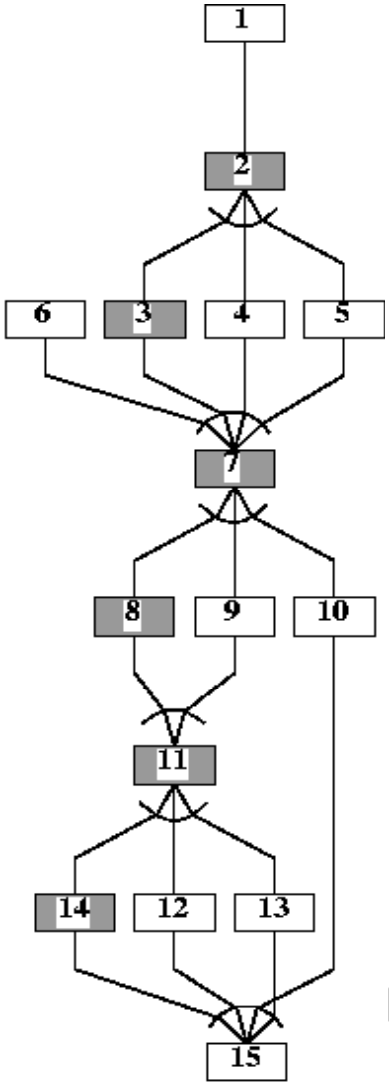
3.48 times speedup by OSCAR compiler against sequential processing

Performance of OSCAR compiler on 16 cores SGI Altix 450 Montvale server

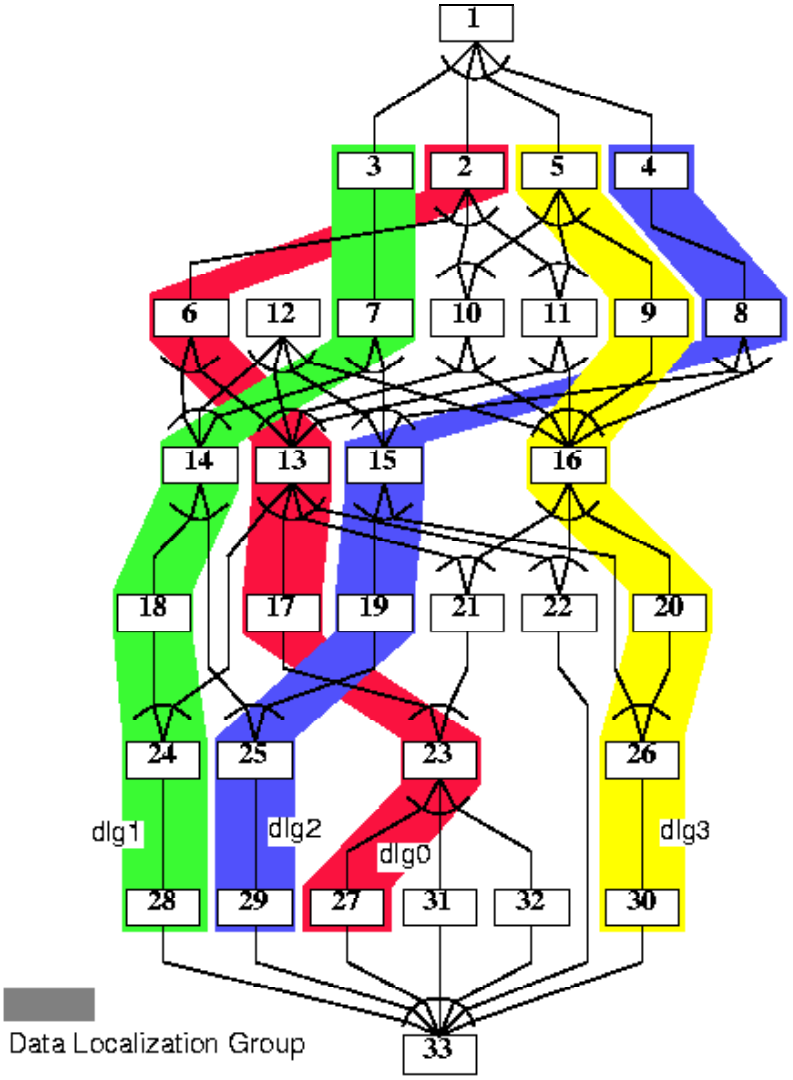


- **OSCAR compiler gave us 2.32 times speedup against Intel Fortran Itanium Compiler revision 10.1**

Data Localization



MTG

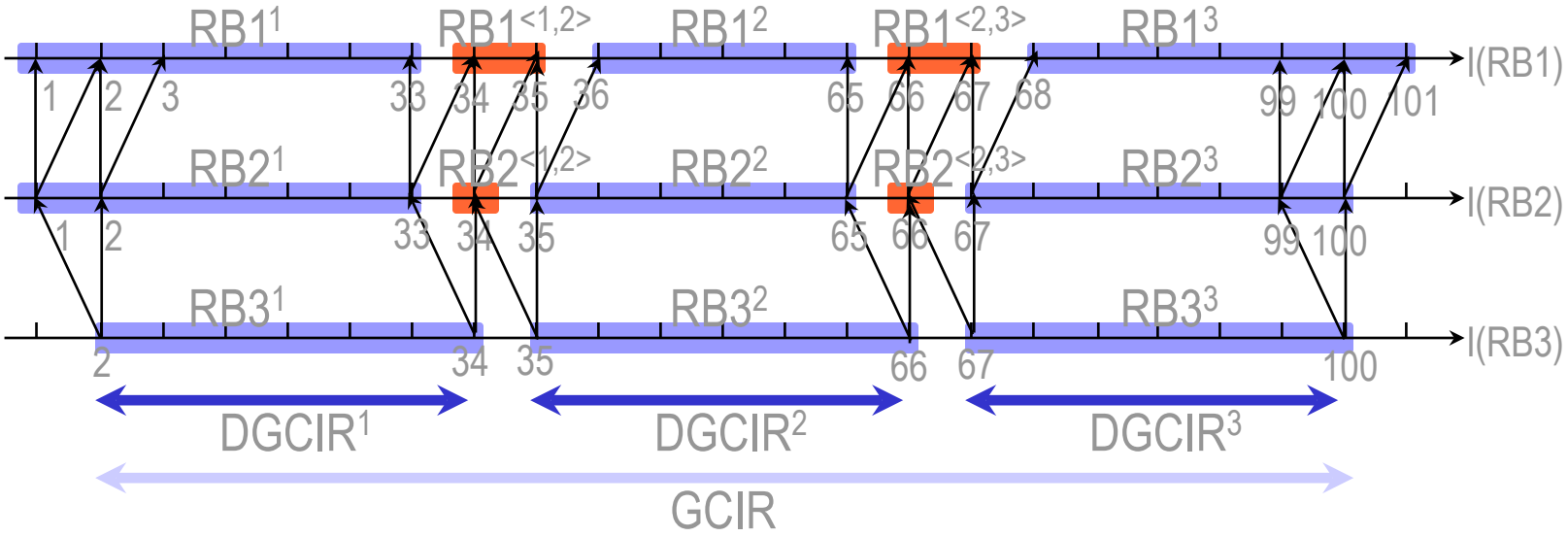


MTG after Division

A schedule for two processors

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.



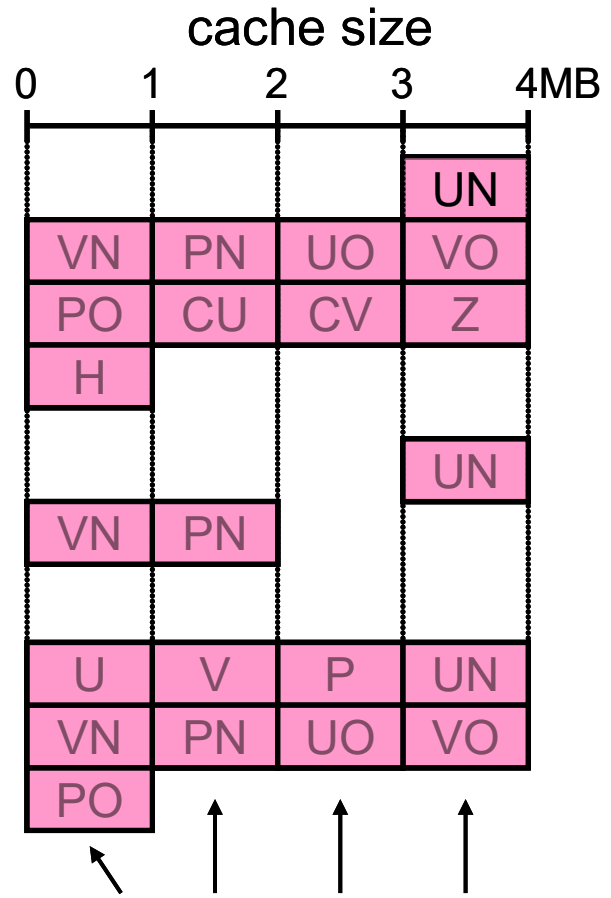
An Example of Data Localization for Spec95 Swim

```
      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

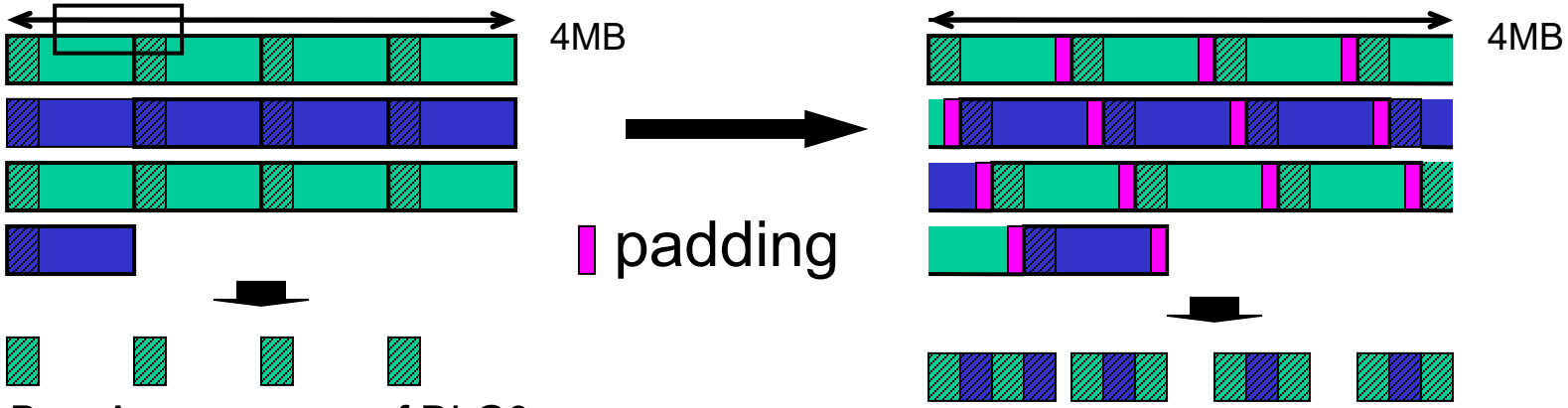
```
PARAMETER (N1=513, N2=513)

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

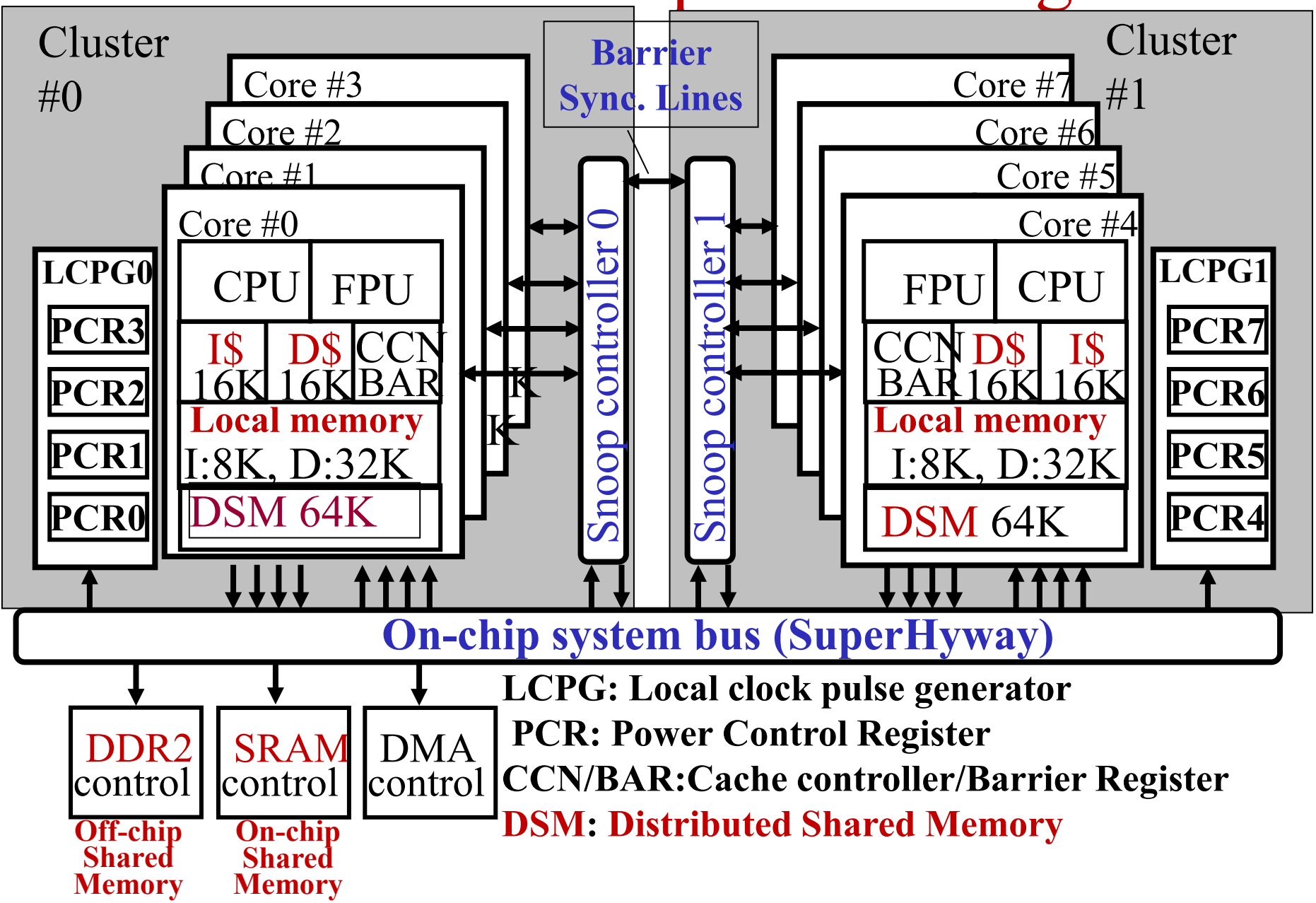
```
PARAMETER (N1=513, N2=544)

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

8 Core RP2 Chip Block Diagram



LCPG: Local clock pulse generator

PCR: Power Control Register

CCN/BAR:Cache controller/Barrier Register

DSM: Distributed Shared Memory

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)

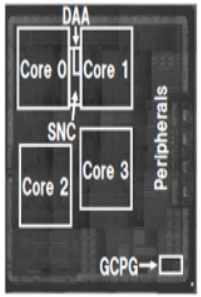
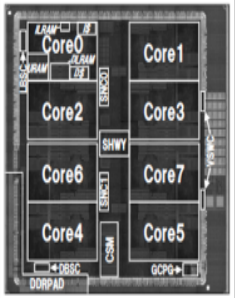
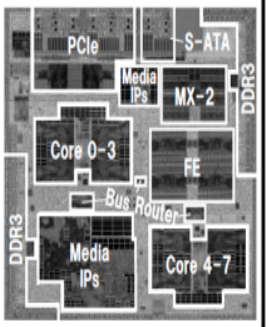
第74回総合科学技術会議の様子(2)


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)

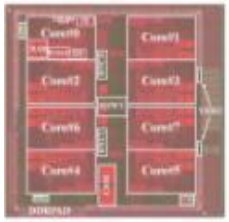
Codesign of Compiler and Multiprocessor Architecture since 1985

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)

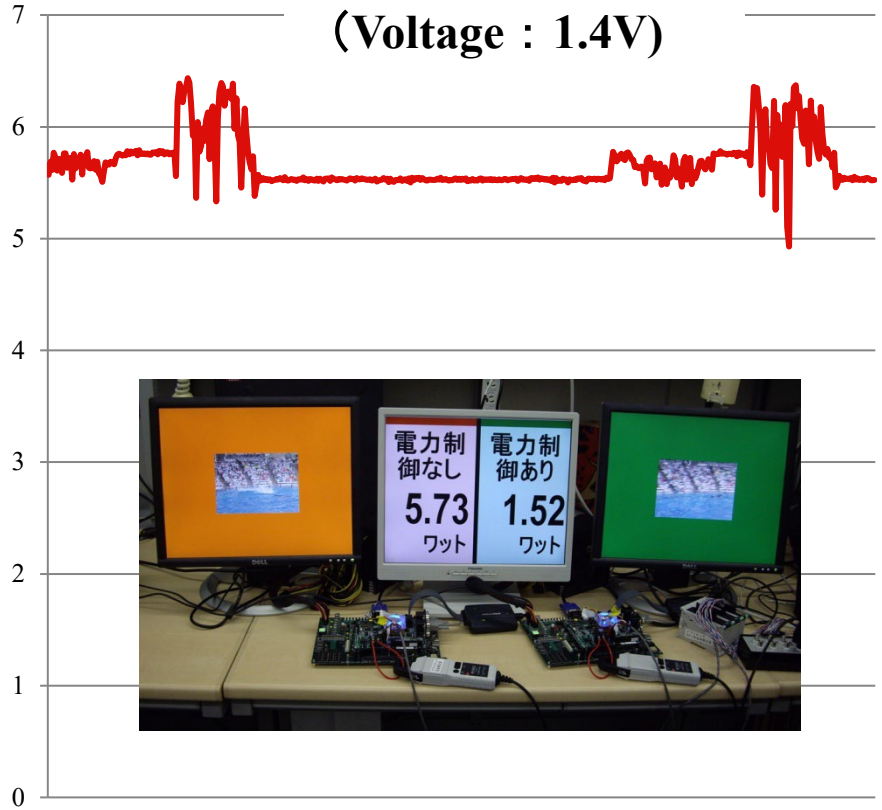
Prime Minister FUKUDA is touching our multicore chip during execution.

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

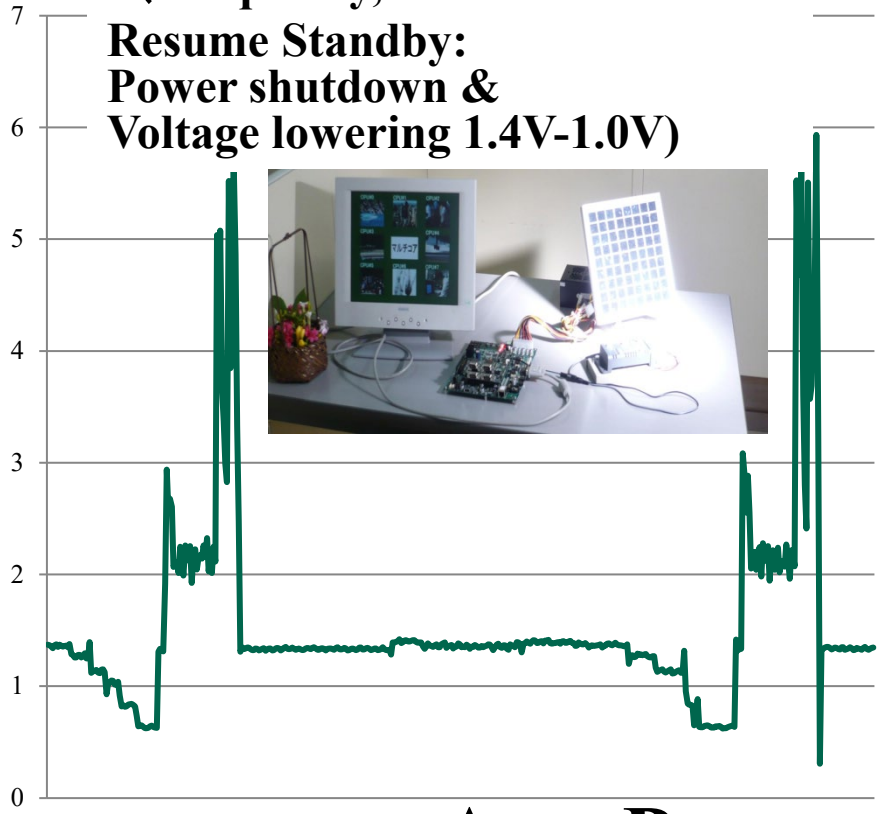


MPEG2 Decoding with 8 CPU cores

Without Power Control (Voltage：1.4V)



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



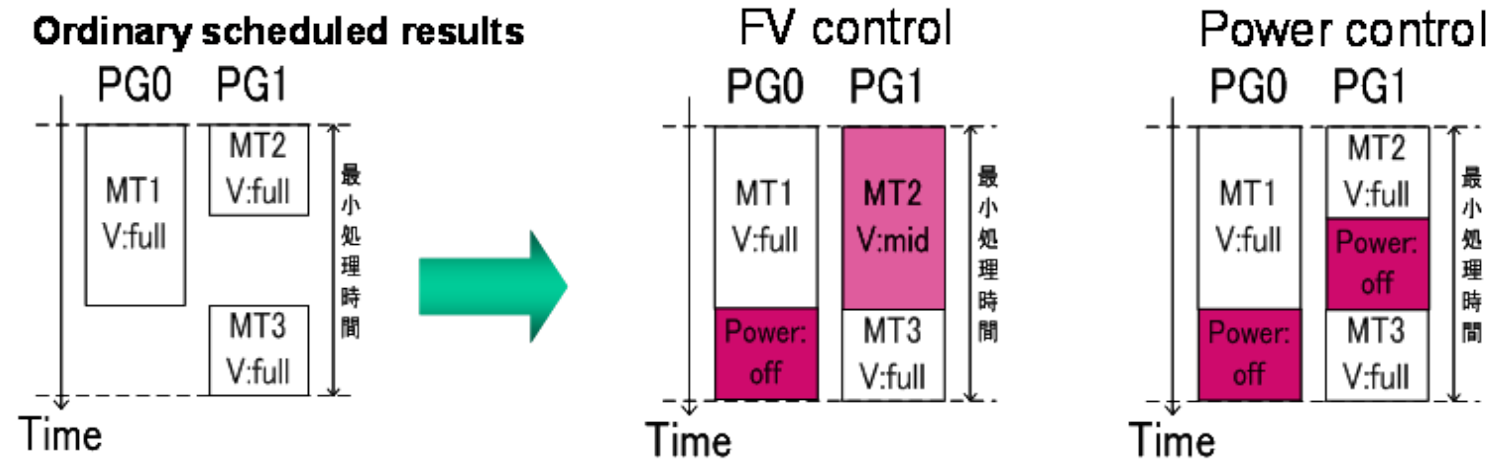
Avg. Power 5.73 [W]

73.5% Power Reduction

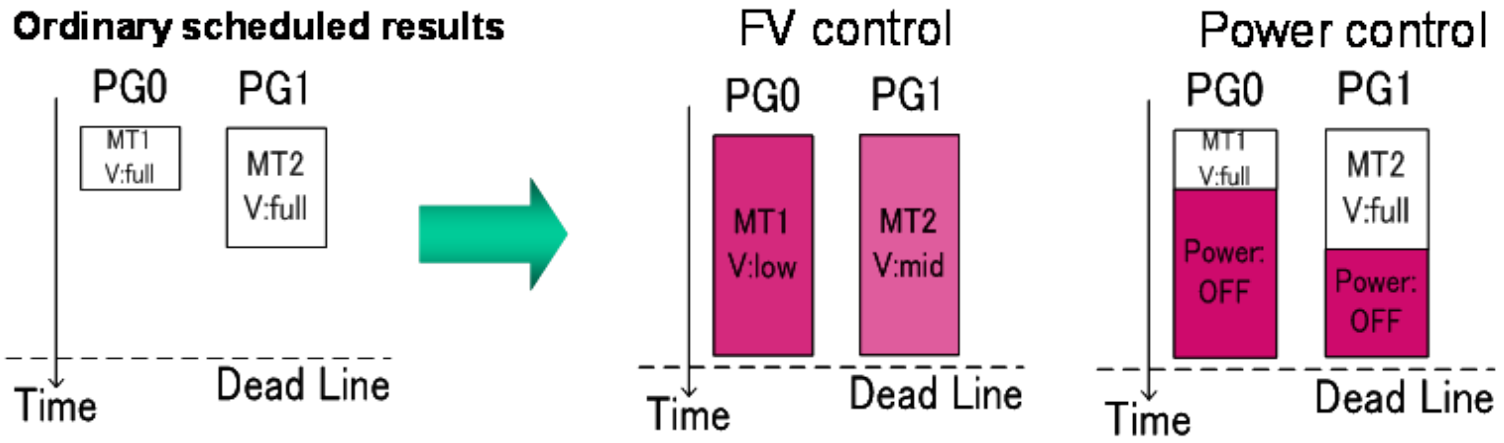
Avg. Power 1.52 [W]

Power Reduction by Power Supply, Clock Frequency and Voltage Control by OSCAR Compiler

- Shortest execution time mode



- Realtime processing mode with dead line constraints



An Example of Machine Parameters for the Power Saving Scheme

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

state	FULL	MID	LOW	OFF
frequency	1	1 / 2	1 / 4	0
voltage	1	0.87	0.71	0
dynamic energy	1	3 / 4	1 / 2	0
static power	1	1	1	0

- **State transition overhead**

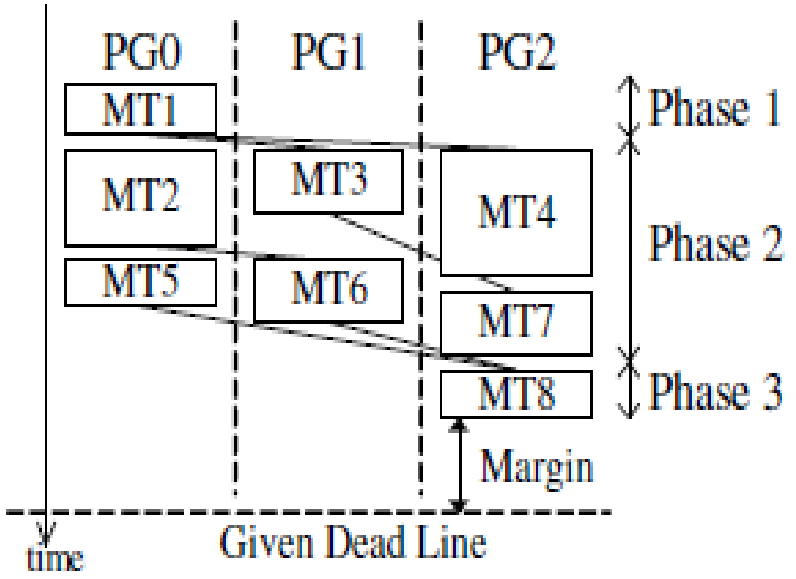
state	FULL	MID	LOW	OFF
FULL	0	40k	40k	80k
MID	40k	0	40k	80k
LOW	40k	40k	0	80k
OFF	80k	80k	80k	0

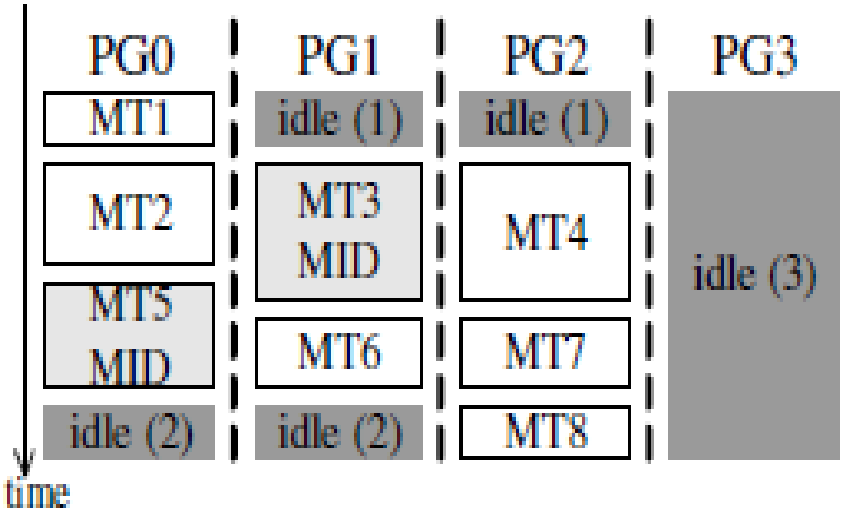
delay time [u.t.]

state	FULL	MID	LOW	OFF
FULL	0	20	20	40
MID	20	0	20	40
LOW	20	20	0	40
OFF	40	40	40	0

energy overhead [μJ]

Power Reduction Scheduling





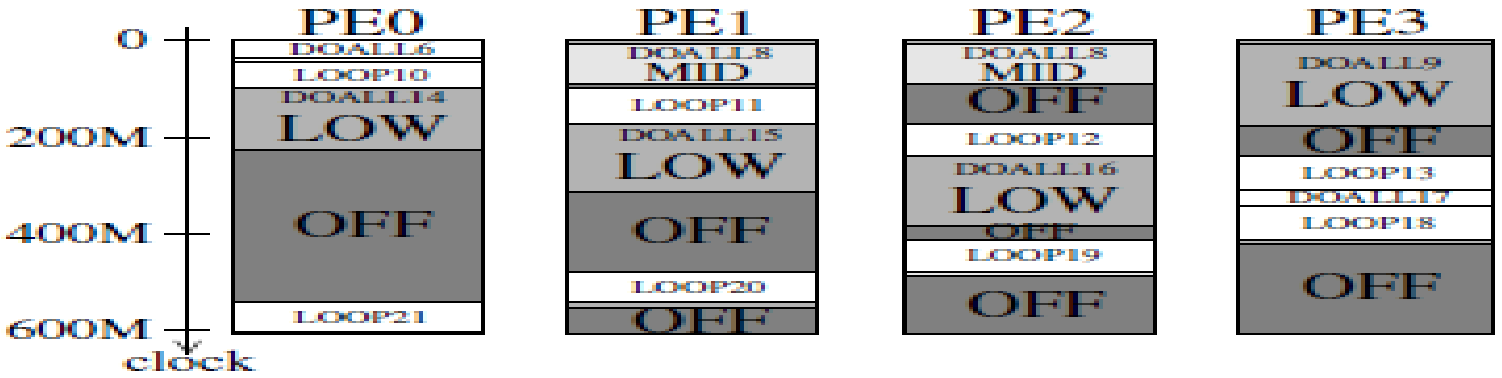
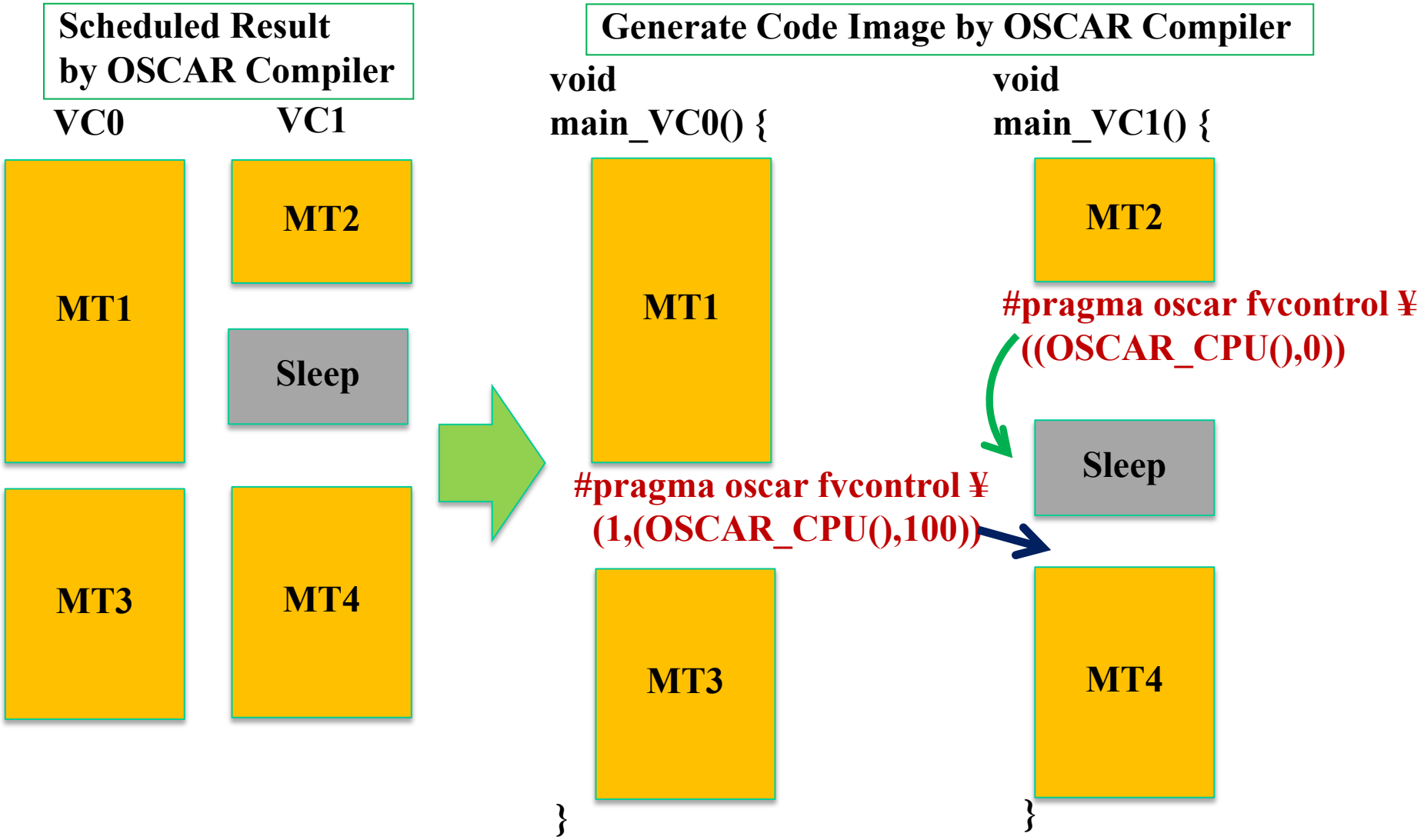


Fig. 6. V/F control of applu(4proc.)

Low-Power Optimization with OSCAR API



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

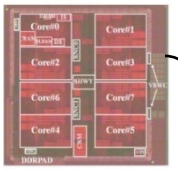
Accelerator 2
Code

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

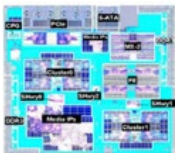
Low Power Heterogeneous Multicore Code Generation
API Analyzer (Available from Waseda) | Existing sequential compiler

Server Code Generation
OpenMP Compiler

Generation of parallel machine codes using sequential compilers



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

IEEE COMPUTER SOCIETY

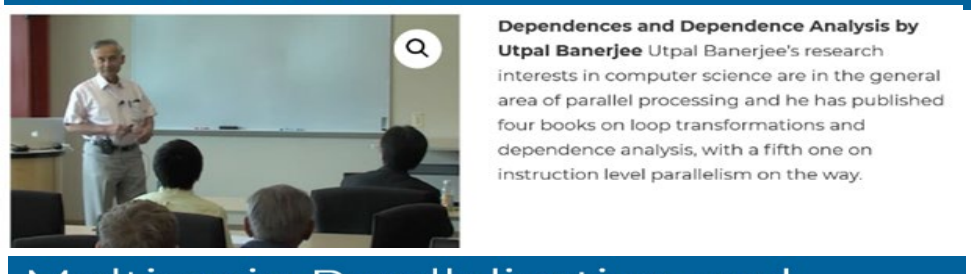
https://www.computer.org/product/education/multi-core-video-lectures-bundle

Created by Multicore STC Chair Hironori Kasahara

Multi-Core Lecture Series consists of 11 one-hour lectures by some of the world's leading researchers in the field. This series is not a course and it consists of the presentation for those who are in the research field. This is more intended for research information sharing than educational training. Topics that are covered during these lectures are listed below. This series also includes an hour discussion of the lecturers.

Video Presentations:

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



Dependences and Dependence Analysis by Utpal Banerjee Utpal Banerjee's research interests in computer science are in the general area of parallel processing and he has published four books on loop transformations and dependence analysis, with a fifth one on instruction level parallelism on the way.

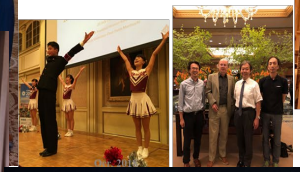
Multigrain Parallelization and Power Reduction by Hironori Kasahara. Professor Kasahara has been researching on OSCAR Automatic Parallelizing and Power Reducing Compiler and OSCAR Multicore architecture for more than 30 years, and led four Japanese national projects on parallelizing compilers, multicores, and green computing.

IEEE Computer Society

The first President from outside North America 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

215
International Conferences

12 Magazines

35 Journals

47 Total Publications

847,000+
Articles in CSDL

6
New Standards

230
Active Standards

IEEE754, 802

12,000+
Volunteers

615
Committees/ Boards

2,352+
Meetings/ Teleconferences

373,100+
Community Members

168
Countries with CS Members

634
Chapters

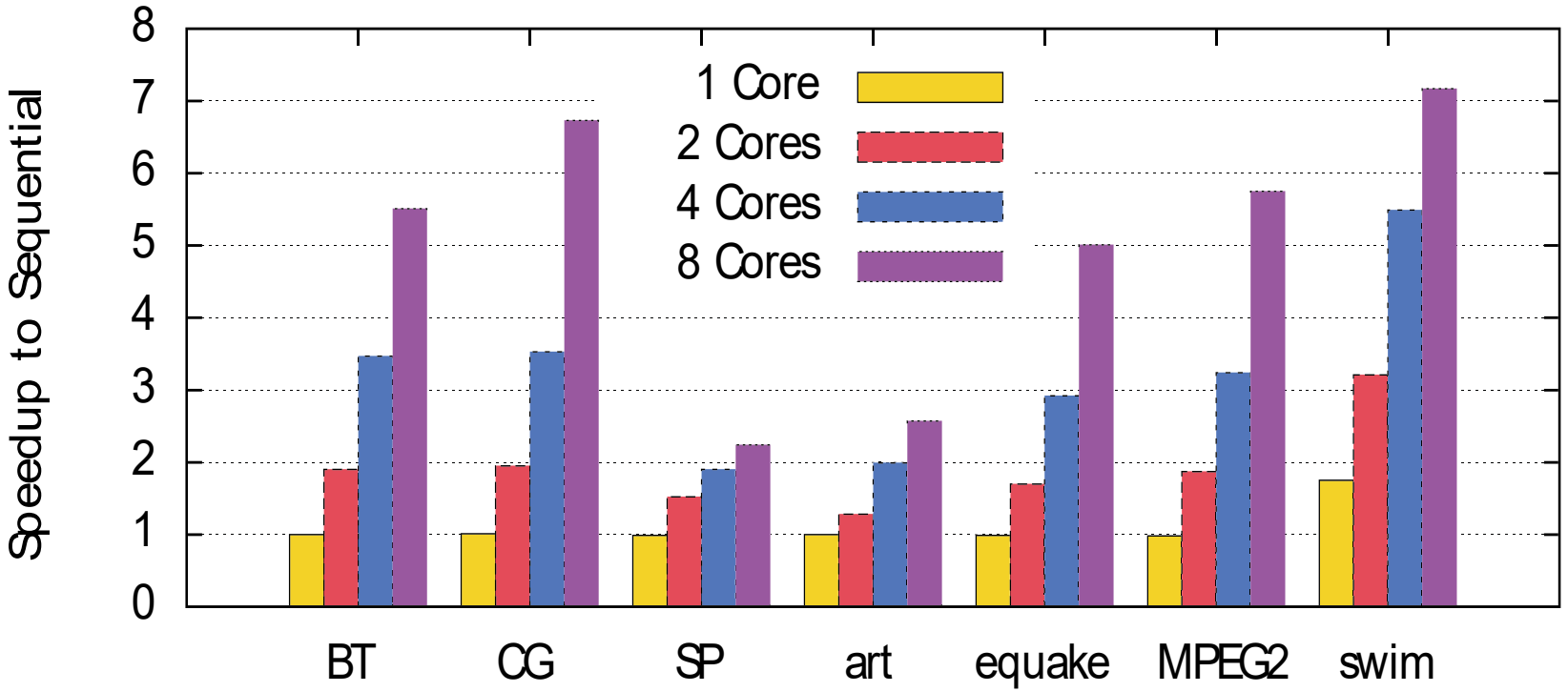
ACM/IEEE SC (SuperComputing) 19, Denver, Nov.17-22, 2019

Cornel Univ. Prof. Steven Squyres: Mars Exploration、CalTech. Dr. Katie Bouman: Visualization of Black Hole

Intel Xeon E5-2650v4 – Benchmark results on upto 8 cores

- x86-64 based Architecture, 12 Cores, 2.2 GHz – 2.9 GHz
- 30 MiB shared L3 cache,
- L3 Cache: Shared by all cores
- speedup to sequential version
- gcc as backend

- NAS parallel benchmark suite
  - BT: Block Tri-diagonal solver
  - CG: Conjugate Gradient computation
  - SP: Scalar Penta-diagonal solver
- SPEC2000
  - art: Image recognition / Neural networks
  - equake: Seismic wave propagation simulation
  - swim: Shallow water modelling - Fortran 77
- MediaBench II : MPEG2 encoding



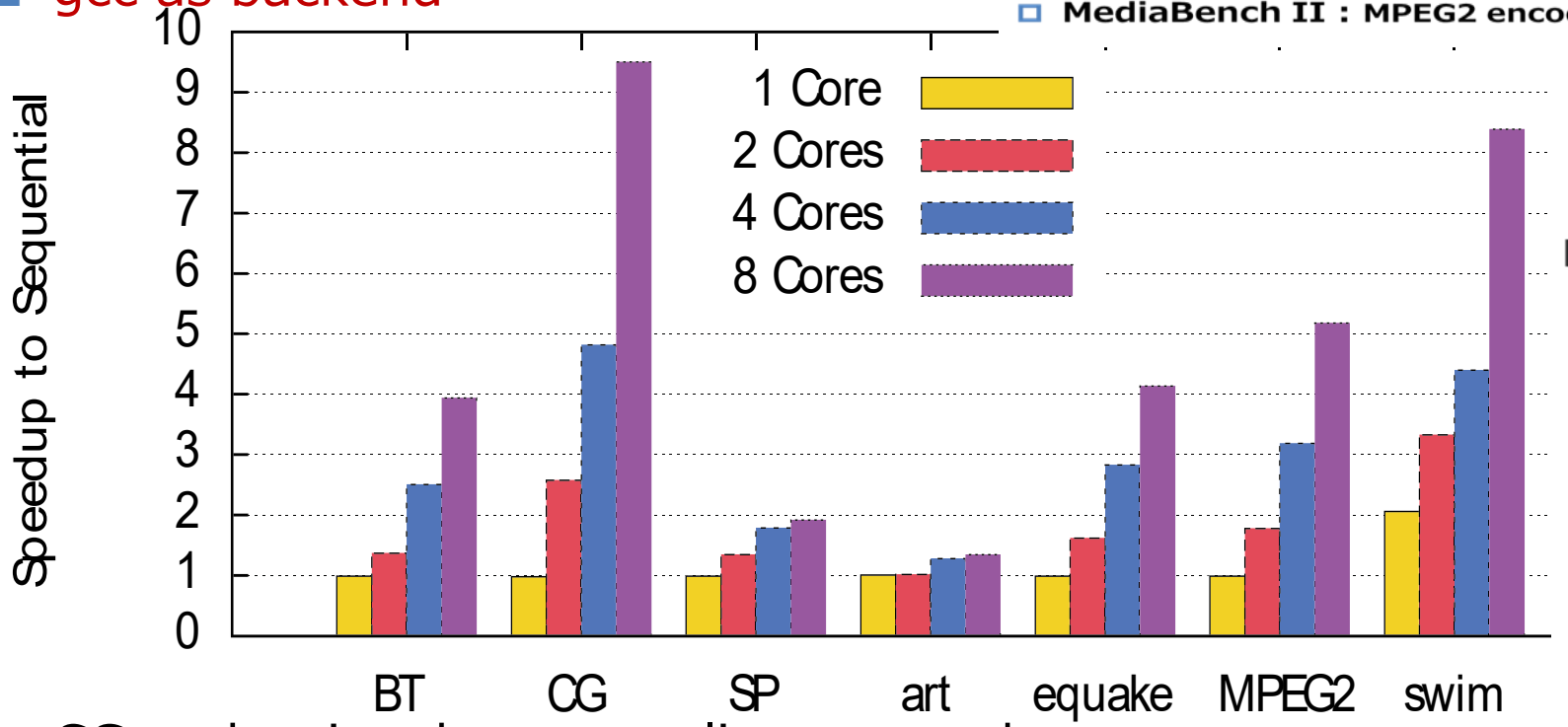
- swim shows superlinear speedup and 1 core speedup
  - seq.: 58.1 s, 1 core OSCAR: 33.2 s, 4 core OSCAR: 10.5 s

AMD EPYC 7702P – Benchmark results on upto 8 cores

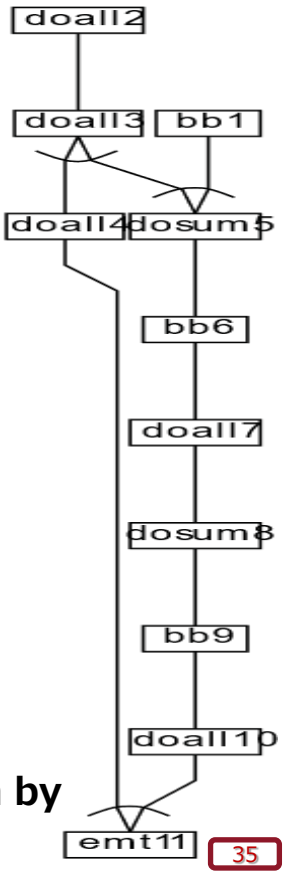
- x86-64 based Architecture, 64 Cores, 2.0 GHz – 3.35 GHz
- 16 MiB L3 cache per 4 core cluster,
- shared within the cluster

- NAS parallel benchmark suite
  - BT: Block Tri-diagonal solver
  - CG: Conjugate Gradient computation
  - SP: Scalar Penta-diagonal solver
- SPEC2000
  - art: Image recognition / Neural networks
  - equake: Seismic wave propagation simulation
  - swim: Shallow water modelling - Fortran 77
- MediaBench II : MPEG2 encoding

- speedup to sequential version
- gcc as backend



- CG and swim show superlinear speedup
  - CG: seq.: 0.86 s, 8 core OSCAR: 0.09 s

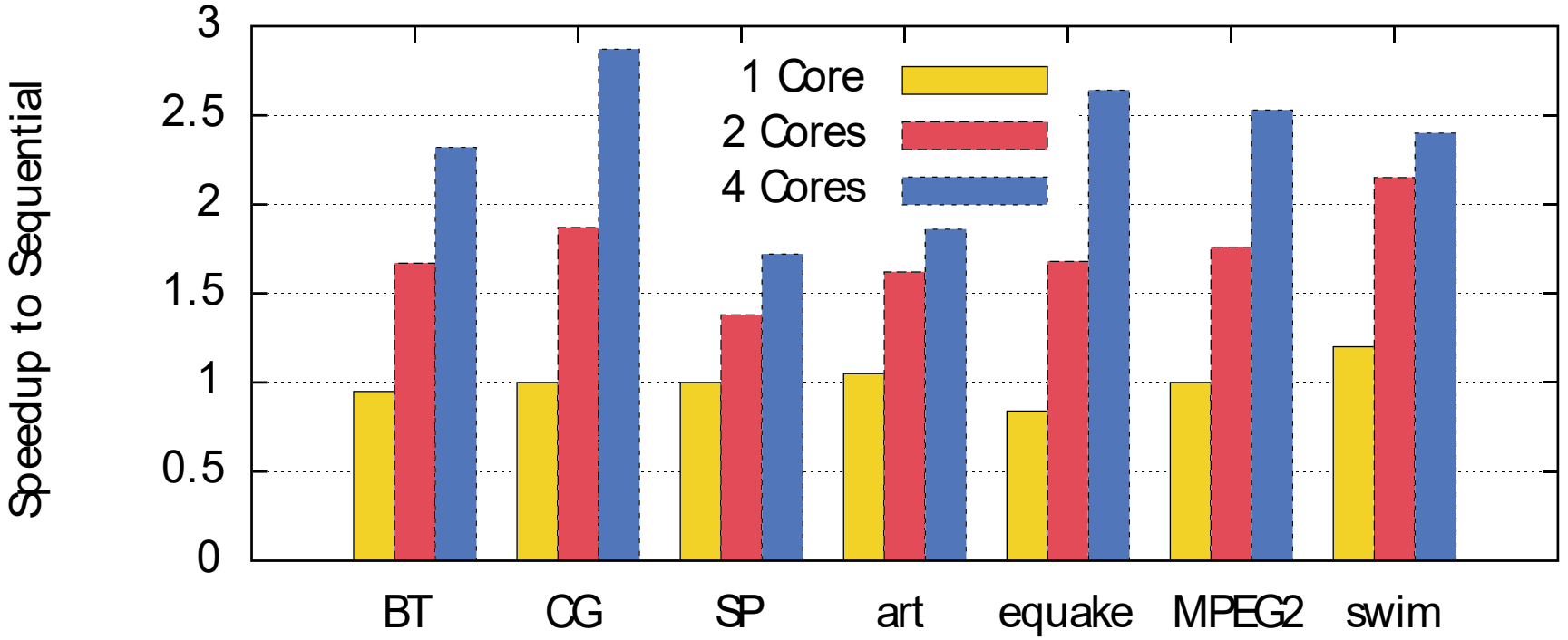


CG macrotask graph
L3 cache Optimization by Data Localization

NVIDIA Carmel ARMv8.2 – Benchmark results on upto 4 cores

- Arm v8.2 based Architecture, 6 Cores, 1.4 GHz
- 4 MiB shared L3 cache,
- L3 Cache: Shared across all cores
- speedup to sequential versior
- gcc as backend

- NAS parallel benchmark suite
  - BT: Block Tri-diagonal solver
  - CG: Conjugate Gradient computation
  - SP: Scalar Penta-diagonal solver
- SPEC2000
  - art: Image recognition / Neural networks
  - equake: Seismic wave propagation simulation
  - swim: Shallow water modelling - Fortran 77
- MediaBench II : MPEG2 encoding



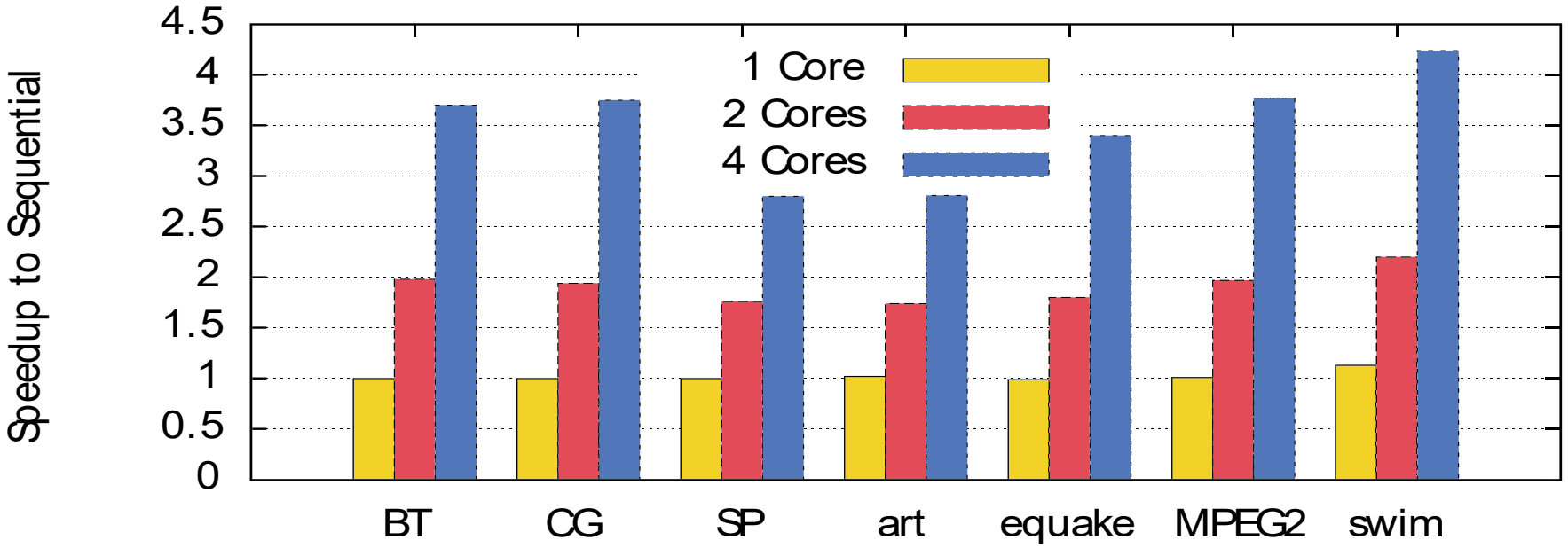
- overall good speedup is observed
  - equake: seq.: 19.0 s, 4 core OSCAR: 7.18 s

SiFive Freedom U740 – Benchmark results on upto 4 cores

- RISC-V based Architecture, 4 Cores, 1.2 GHz
- 2 MiB shared L2 cache
- L2 Cache: Shared across all cores

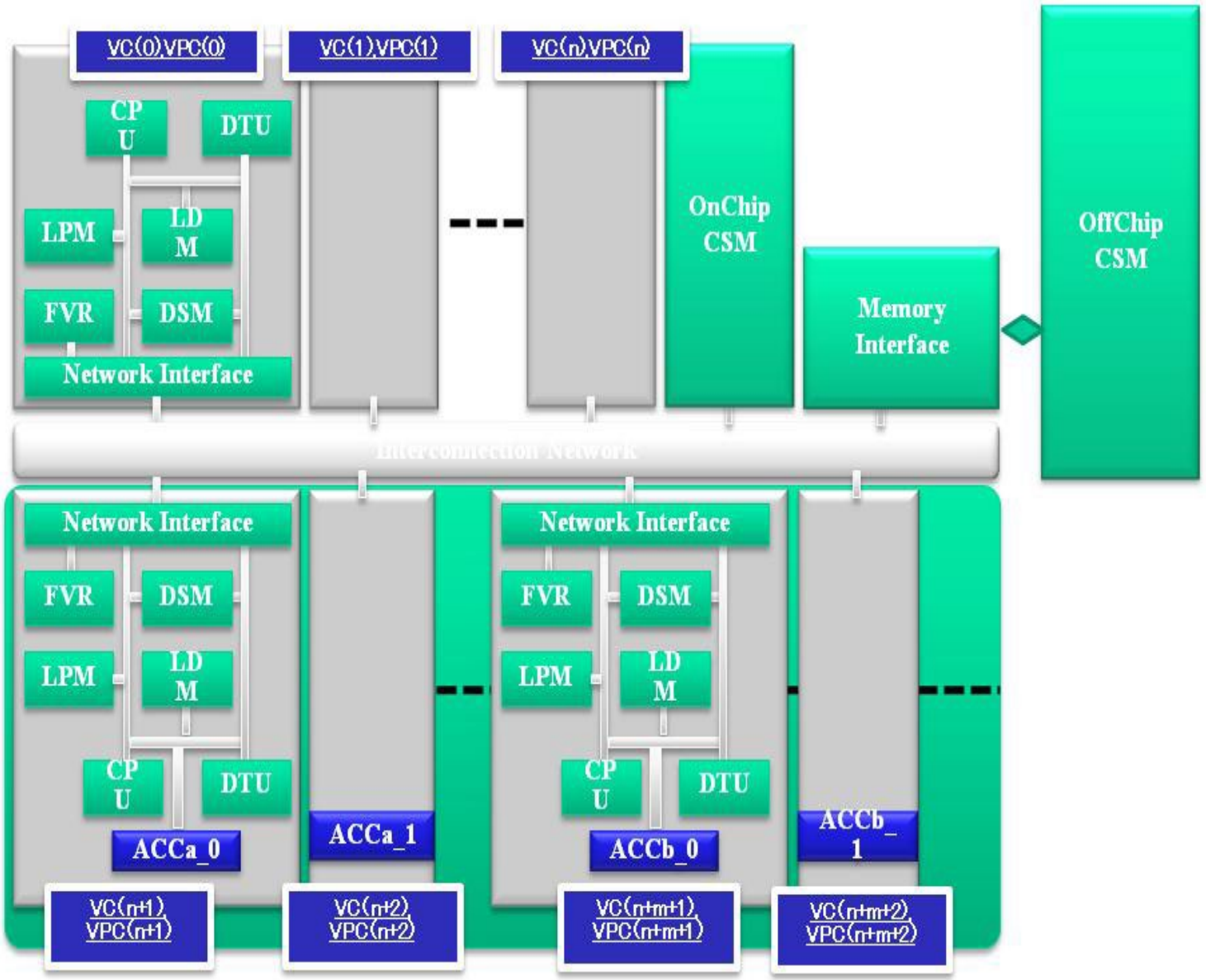
- NAS parallel benchmark suite
  - BT: Block Tri-diagonal solver
  - CG: Conjugate Gradient computation
  - SP: Scalar Penta-diagonal solver
- SPEC2000
  - art: Image recognition / Neural networks
  - equake: Seismic wave propagation simulation
  - swim: Shallow water modelling - Fortran 77
- MediaBench II : MPEG2 encoding

- speedup to sequential version
- gcc as backend



- overall good speedup is observed, swim superlinear
  - BT: seq.: 2041 s, 4 core OSCAR: 551 s

Heterogeneous Multicore Architecture targeted by OSCAR API



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

OSCAR API Ver. 2.0 for Homogeneous/Heterogeneous Multicores and Manycores

(LCPC2009 Homogeneous, 2010 Heterogeneous)

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

List of Directives (22 directives)

- Parallel Execution API
  - parallel sections (*)
  - flush (*)
  - critical (*)
  - execution
- Memoay Mapping API
  - threadprivate (*)
  - distributedshared
  - onchipshared
- Synchronization API
  - groupbarrier
- Data Transfer API
  - dma_transfer
  - dma_contiguous_parameter
  - dma_stride_parameter
  - dma_flag_check
  - dma_flag_send

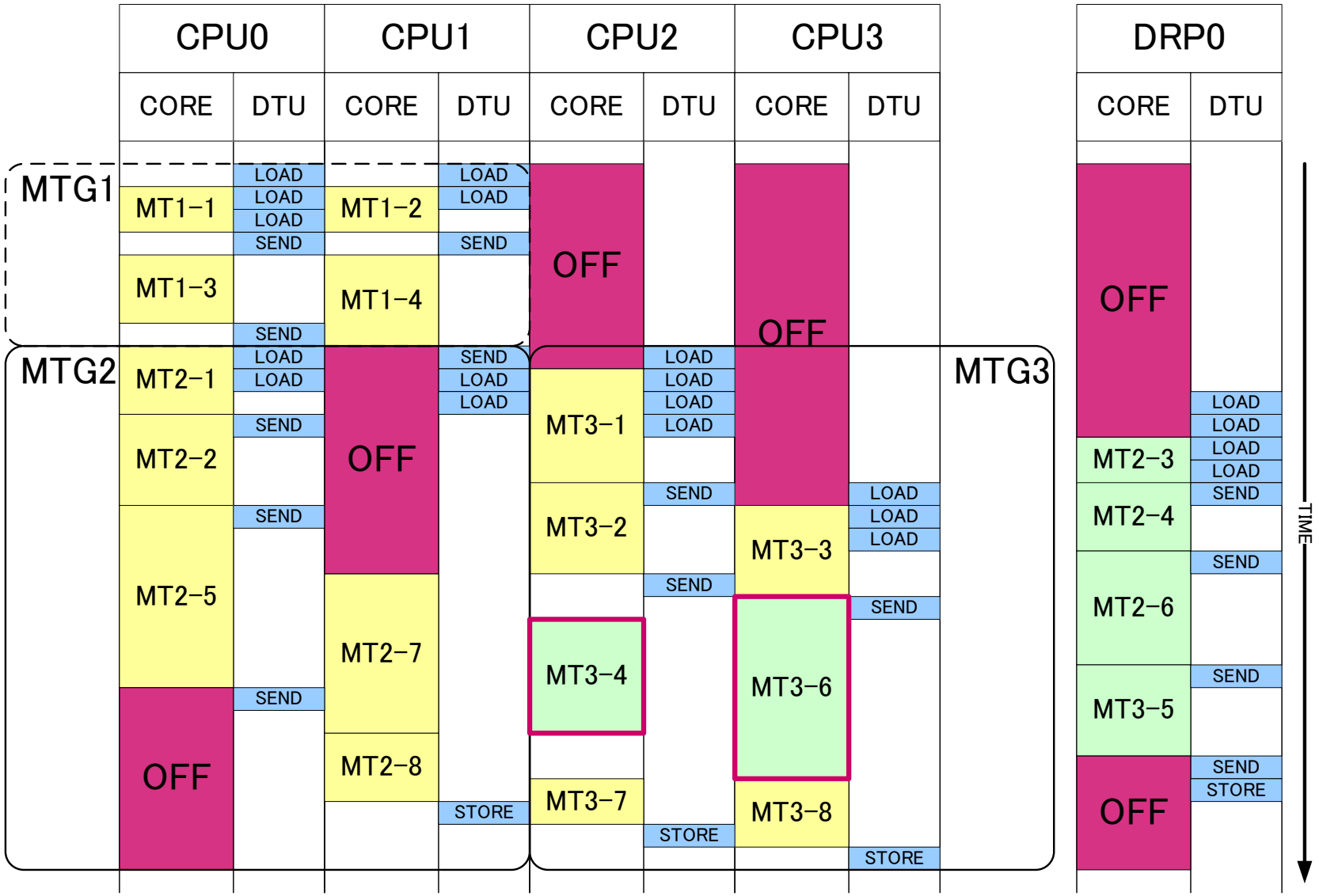
(* from OpenMP)

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

2 hint directives for OSCAR compiler
- accelerator_task
- oscar_comment

from V2.0

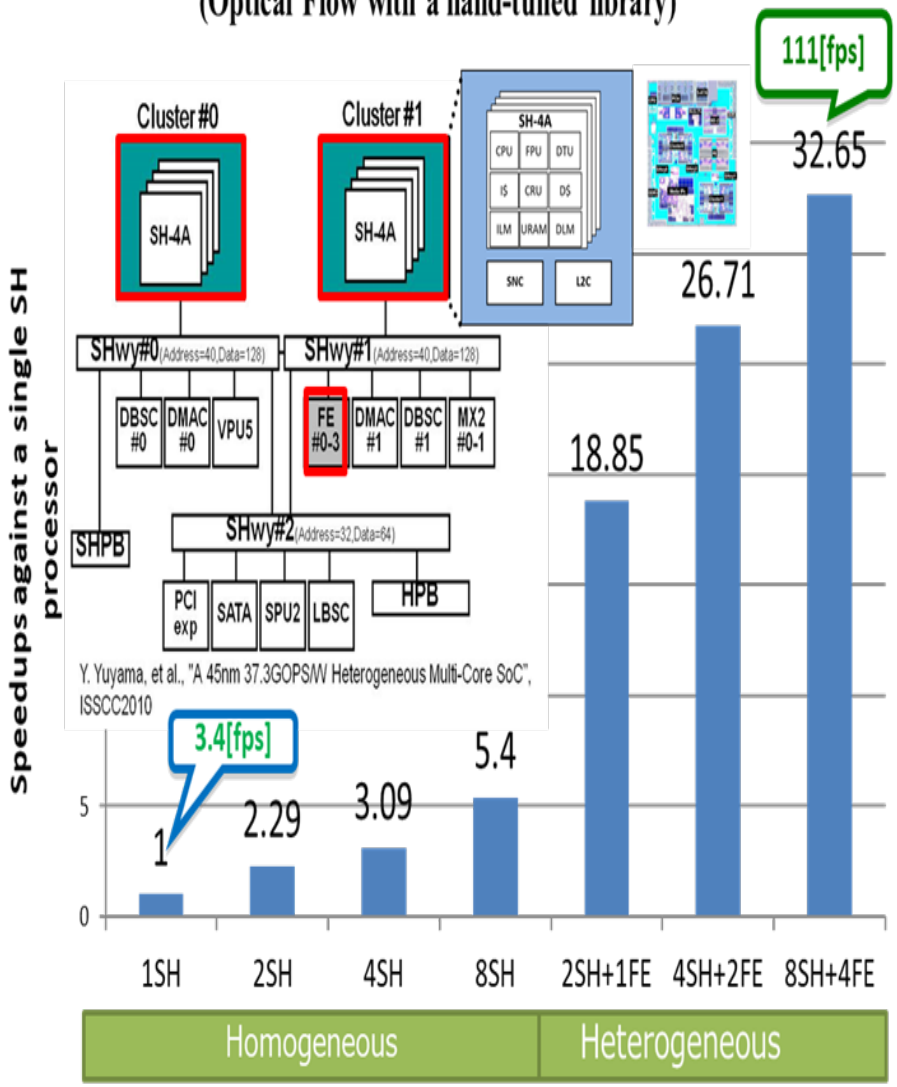
An Image of Static Schedule for Heterogeneous Multi-core with Data Transfer Overlapping and Power Control



Speedups & Power Reduction on RP-X Heterogeneous Multicore with 8 CPUs and 4 DRPs

33 Times Speedup Using OSCAR Compiler and API on Renesas RP-X with 8 CPUs & 4 DRP Accelerators

(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)

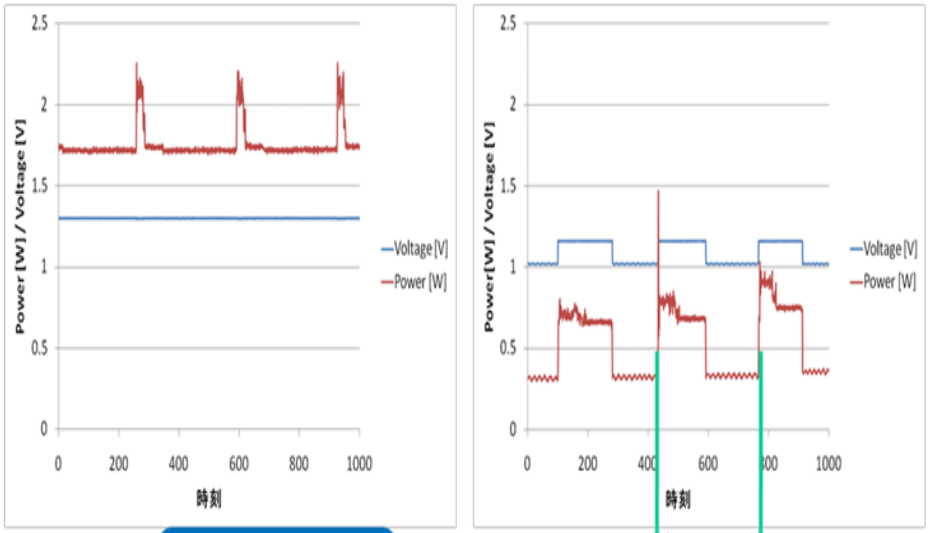
Without Power Reduction

With Power Reduction by OSCAR Compiler

70% of power reduction

Average:1.76[W]

Average:0.54[W]



1cycle : 33[ms]
→30[fps]

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

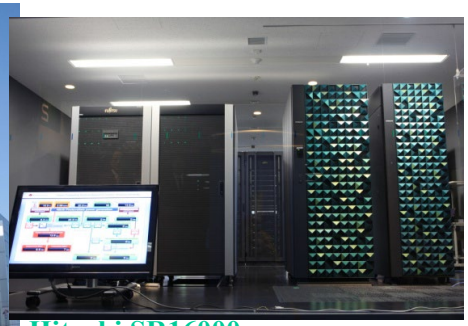
- **More than 64 cores**
- **Natural air cooling (No fan)**
  Cool, Compact, Clear, Quiet
- **Operational by Solar Panel**

<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

The 25th International Workshop on Languages and Compilers for Parallel Computing (LCPC2012), September 11-13, 2012 at Waseda U. Green Computing Center

A Strategic Initiative of Computing: Systems and Applications (SISA)- Integrating HPC, Big Data, AI and Beyond, Jan.18-19, 2017

A Strategic Initiative of Computing:Systems and Applications (SISA) --Integrating HPC, Big Data, AI and Beyond-- Jan. 18-19, 2017

Opening: Prof. Gao, Prof. Kasahara

Waseda VP Shuji Hashimoto

I. Architecture and Applications

Keynote: William J. Dally,
NVIDIA and Stanford University, USA

- Kimihiko Hirao, RIKEN, Japan
- G. W. Yang, Tsinghua Univ. China
- J. Sexton, IBM, USA

II . System Software and Applications

Keynote： Rick. Stevens ANL, USA

- S. Mikhail Smelyanskiy Intel USA
- Fred. Streitz, LLNL USA
- R. Govind, IIS, India
- H. Hironori Kasahara, Waseda Univ,

III. Extreme Scale and Beyond

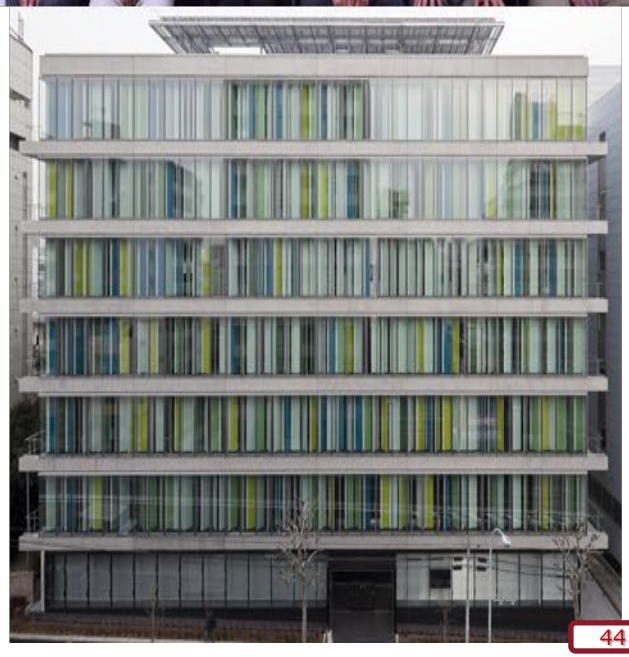
Keynote: Paul Messina ANL, USA

- Motoaki Saito, PEZY, Japan
- Eiji Ishida, MEXT, Japan
- Depei Qian, BUAA, China
- Toshiyuki Shimizu, Fujitsu, Japan

IV. Integration of HPC, Big Data, and AI

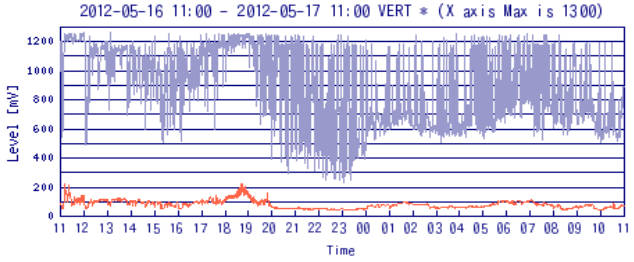
Keynote: Thomas Sterling, Indiana Univ., USA

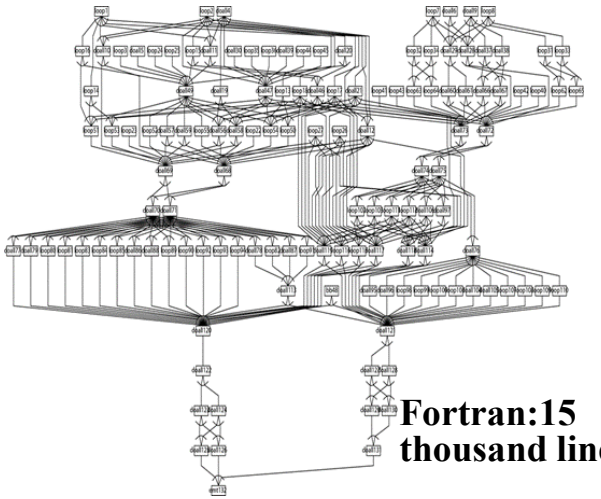
- Masaru Kitsuregawa, NII and Univ. of Tokyo, Japan
- Thomas Schulthess, ETH, Swiss
- Moriyuki Takamura/Toshiaki Kitamura, Oscar Tech, Japan



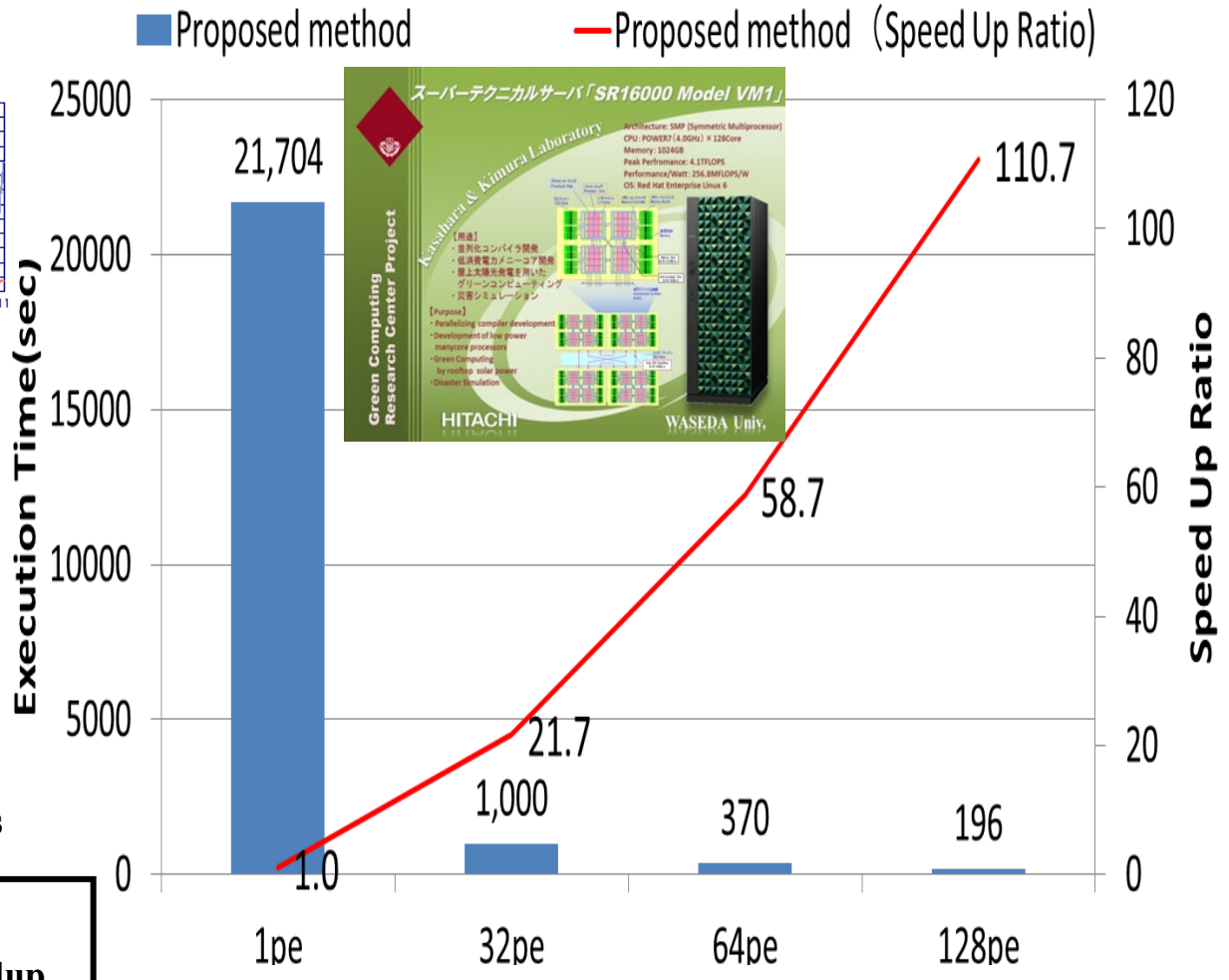
110 Times Speedup against the Sequential Processing for GMS Earthquake Wave Propagation Simulation on Hitachi SR16000

（Power7 Based 128 Core Linux SMP） (LCPC2015)





Fortran:15 thousand lines

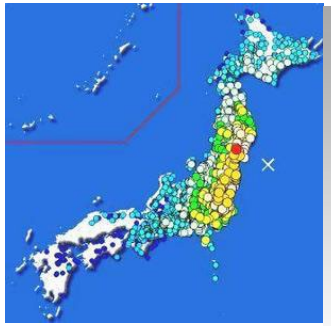


First touch for distributed shared memory and cache optimization over loops are important for scalable speedup

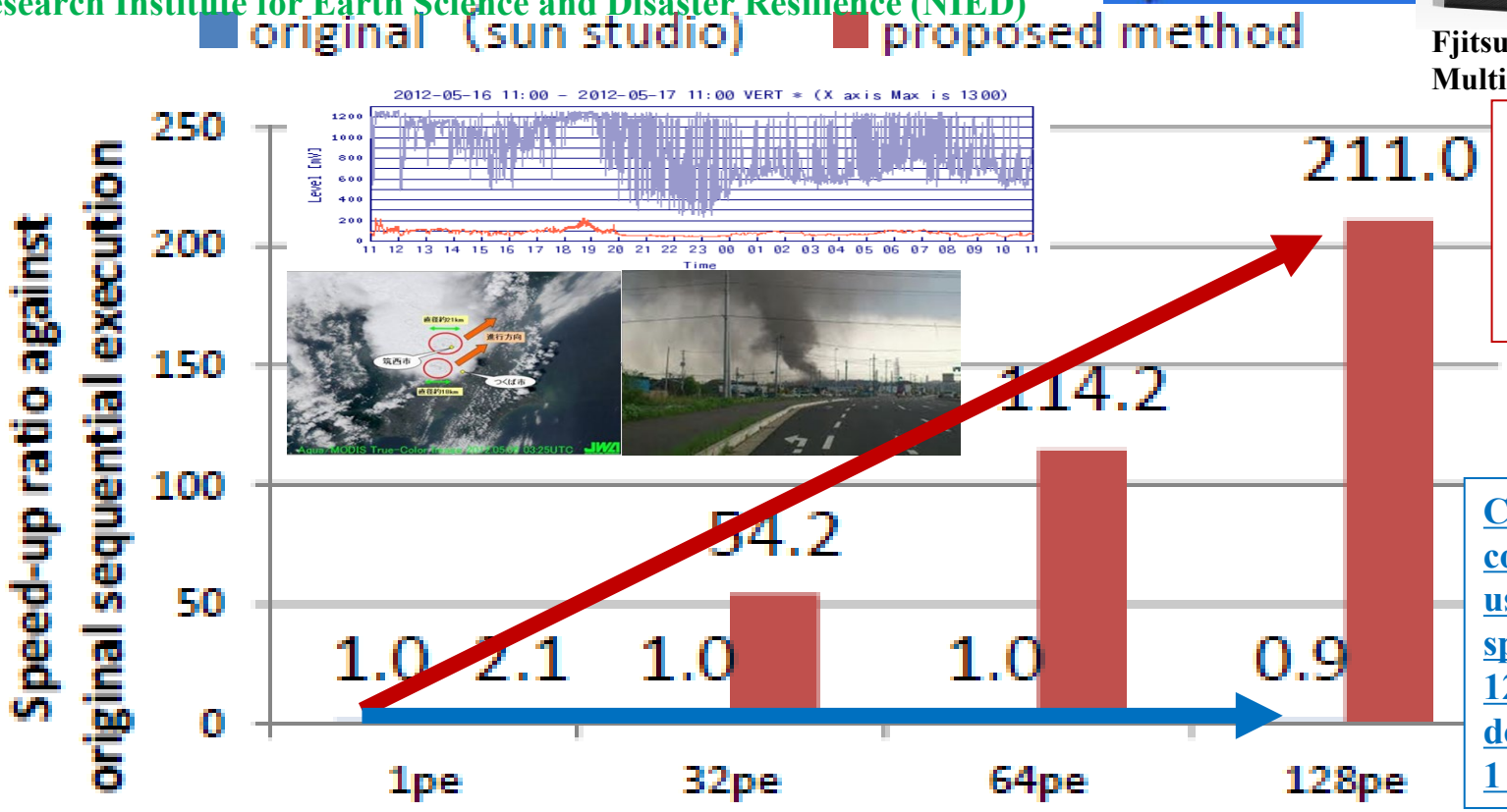
Parallel Soft is important for scalable performance of multicore (LCPC2015)

- 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



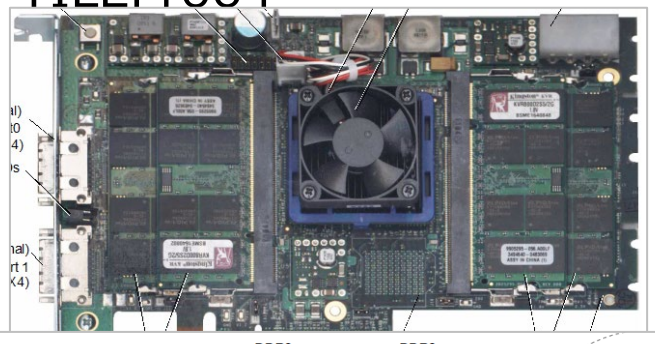
OSCAR Compiler gives us 211 times speedup with 128 cores

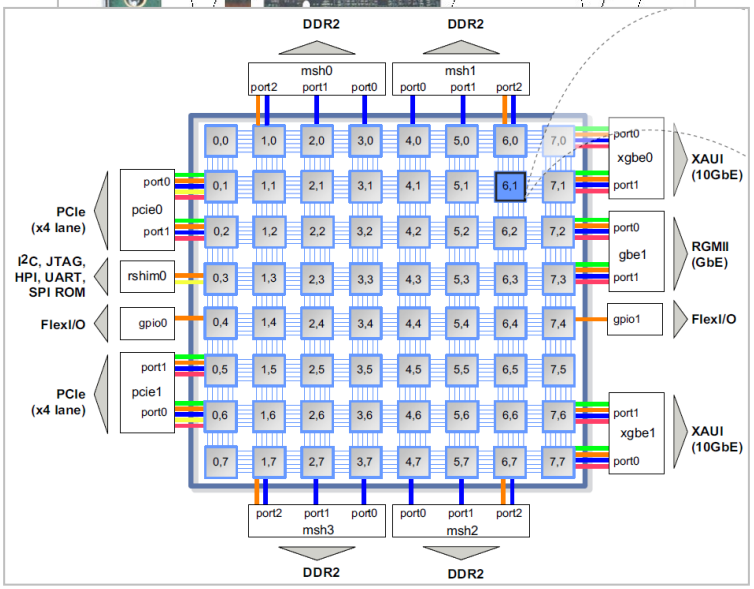
Commercial compiler gives us 0.9 times speedup with 128 cores (slow-downed against 1 core)

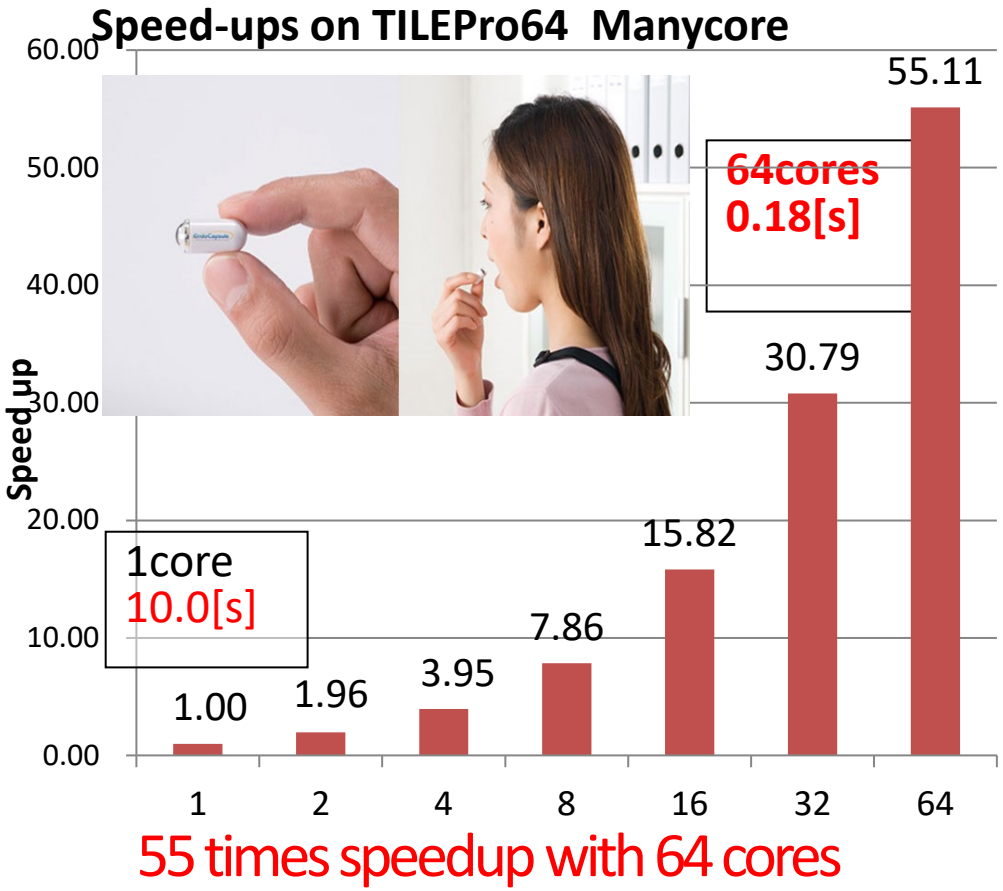
- Automatic parallelizing compiler available on the market gave us no speedup against execution time on 1 core on 64 cores
  - Execution time with 128 cores was slower than 1 core (0.9 times speedup)
- Advanced OSCAR parallelizing compiler gave us 211 times speedup with 128cores against execution time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

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

- TILEPro64



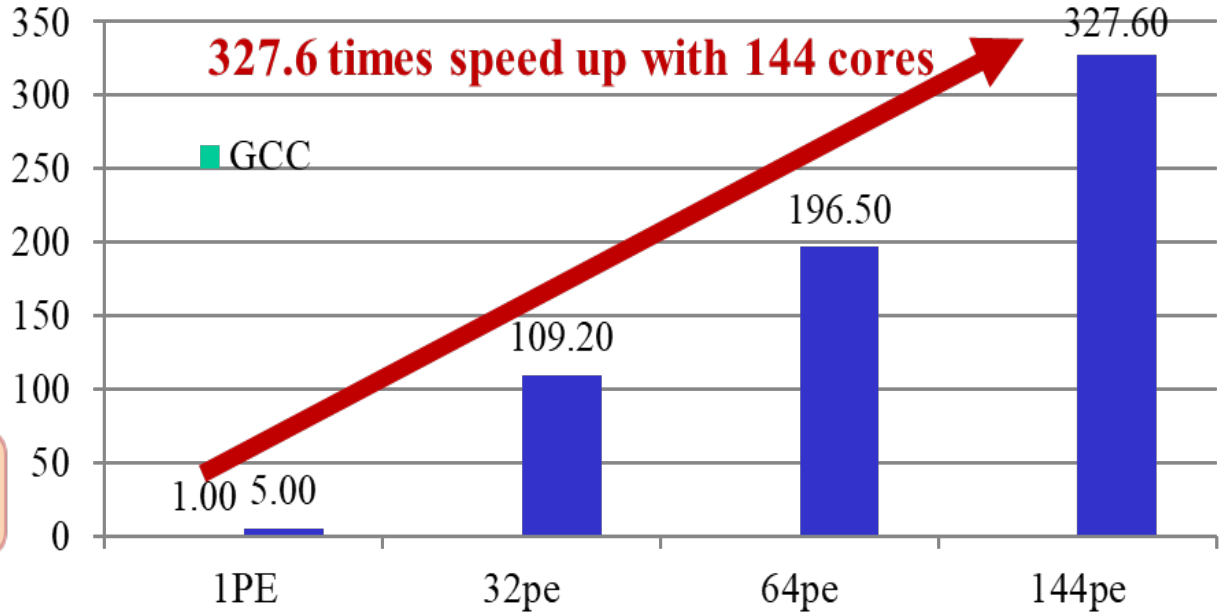




Performance on Multicore Server for Latest Cancer Treatment Using Heavy Particle (Proton, Carbon Ion)

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

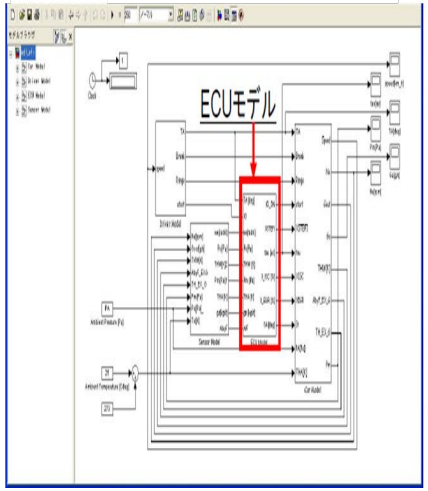
日本乗用車のエンジン制御計算をデンソー2コアECU上で、1．95倍の速度向上に成功。(見神、梅田)

欧州農耕作業車エンジン制御計算をインフィニオン2コアプロセッサ上で8.7倍の高速化に成功。

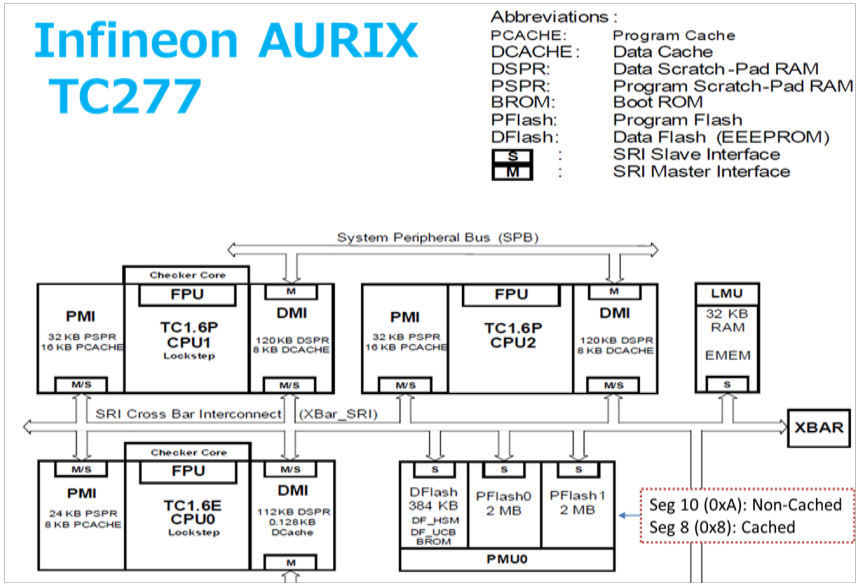
Engine Control by multicore with Denso

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.

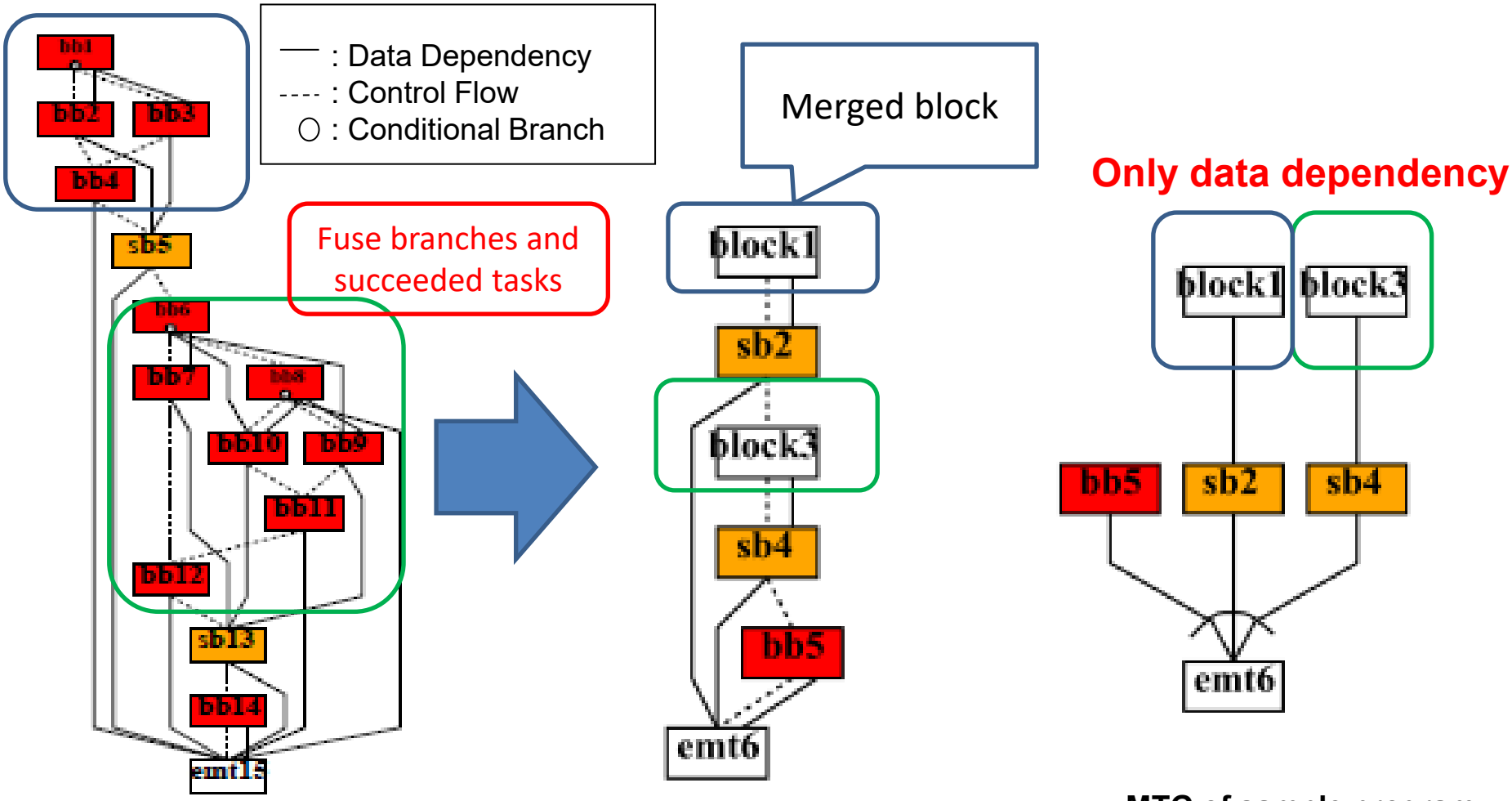
- Hard real-time automobile engine control by multicore using local memories
- Millions of lines C codes consisting conditional branches and basic blocks



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



Macro Task Fusion for Static Task Scheduling



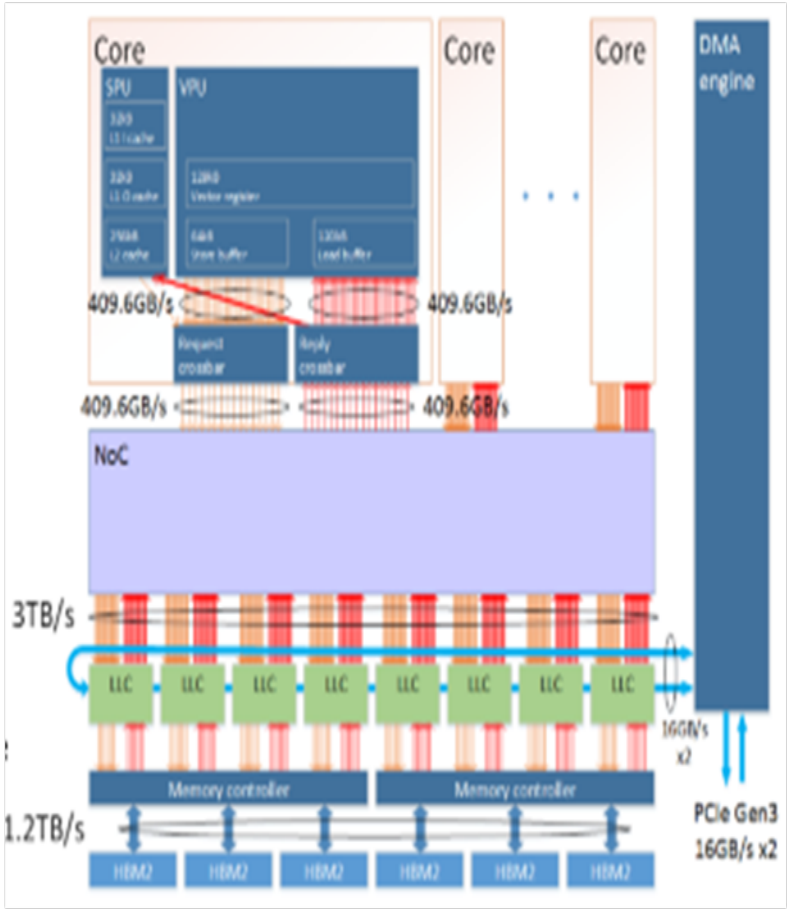
MFG of sample program before maro task fusion

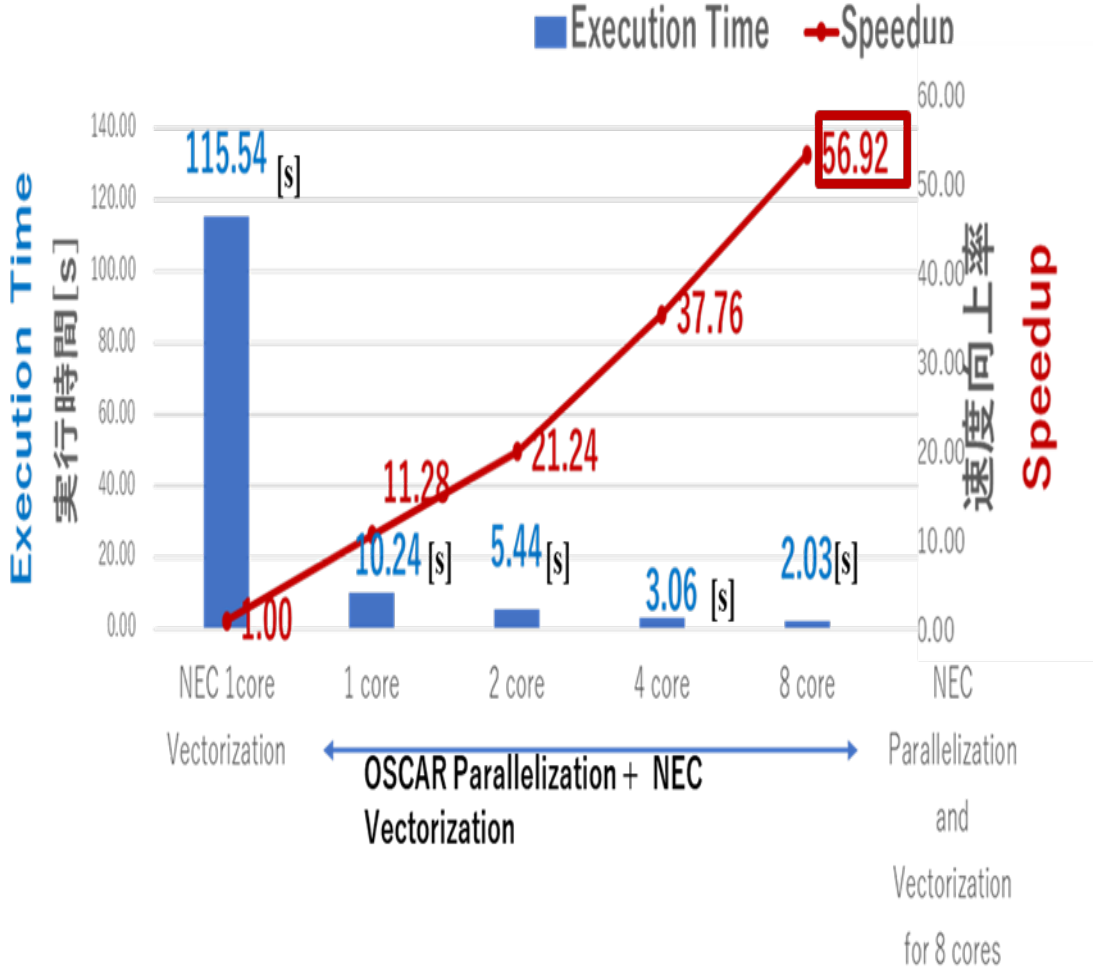
MFG of sample program after macro task fusion

MTG of sample program after macro task fusion

Speedups of NPB/CG Scientific Code by OSCAR Compiler on NEC SX-Aurora TSUBASA A100-1 8 cores 10C VE

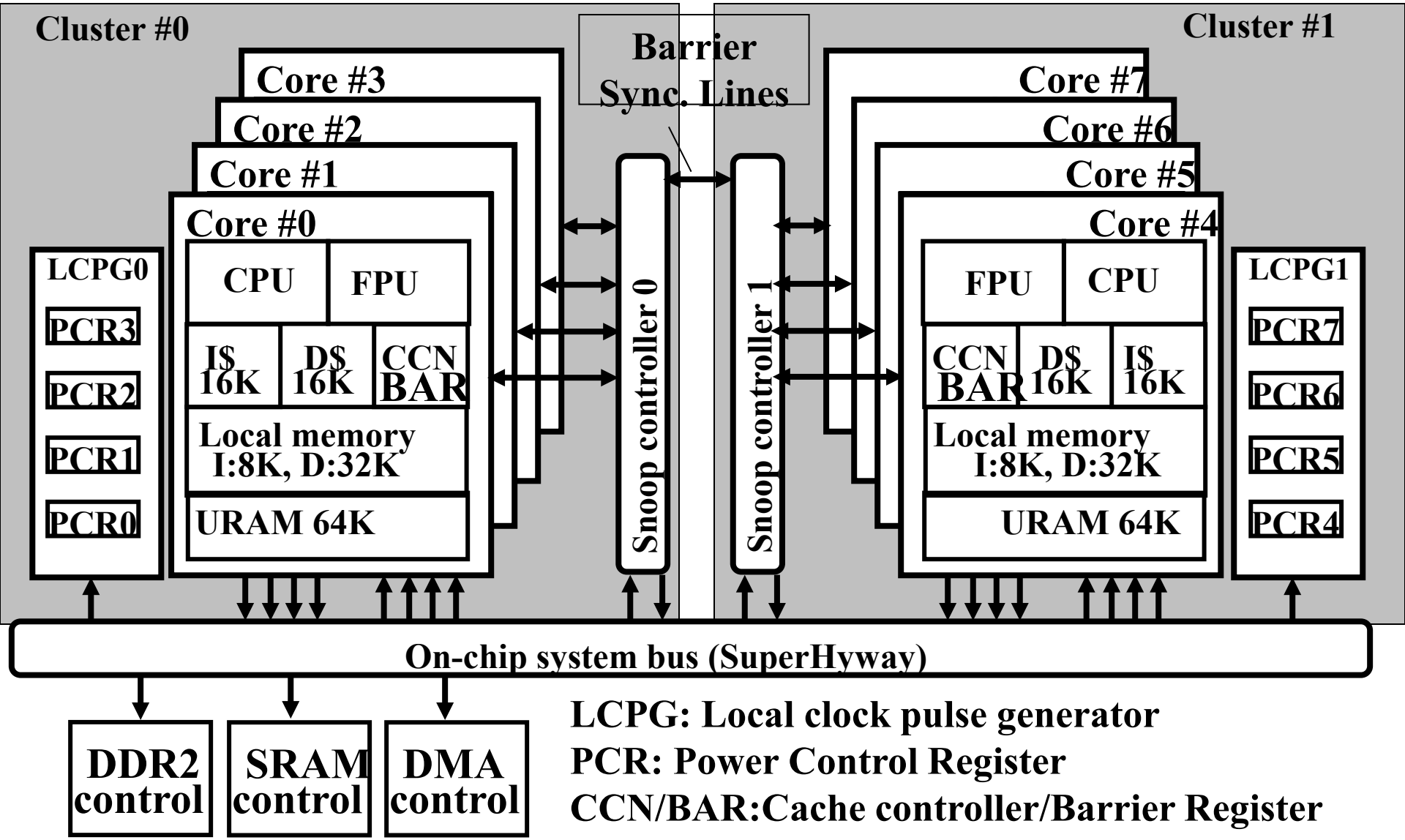
57 times speedup for 8 vector cores by OSCAR Parallelization & NEC Vectorization against NEC 1 core Vectorization





NEC Corporation. (2021) SX-Aurora TSUBASA Performance Tuning Guide. [Online]. Available: https://www.hpc.nec/documents/guide/pdfs/AuroraVE TuningGuide.pdf

8 Core RP2 Chip Block Diagram



LCPG: Local clock pulse generator

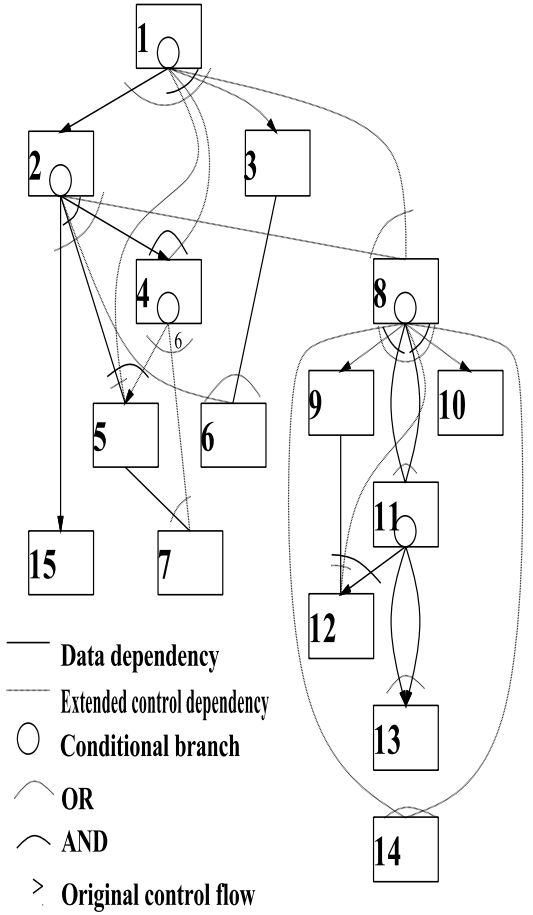
PCR: Power Control Register

CCN/BAR:Cache controller/Barrier Register

URAM: User RAM (Distributed Shared Memory)

Software Coherence Control Method on OSCAR Parallelizing Compiler

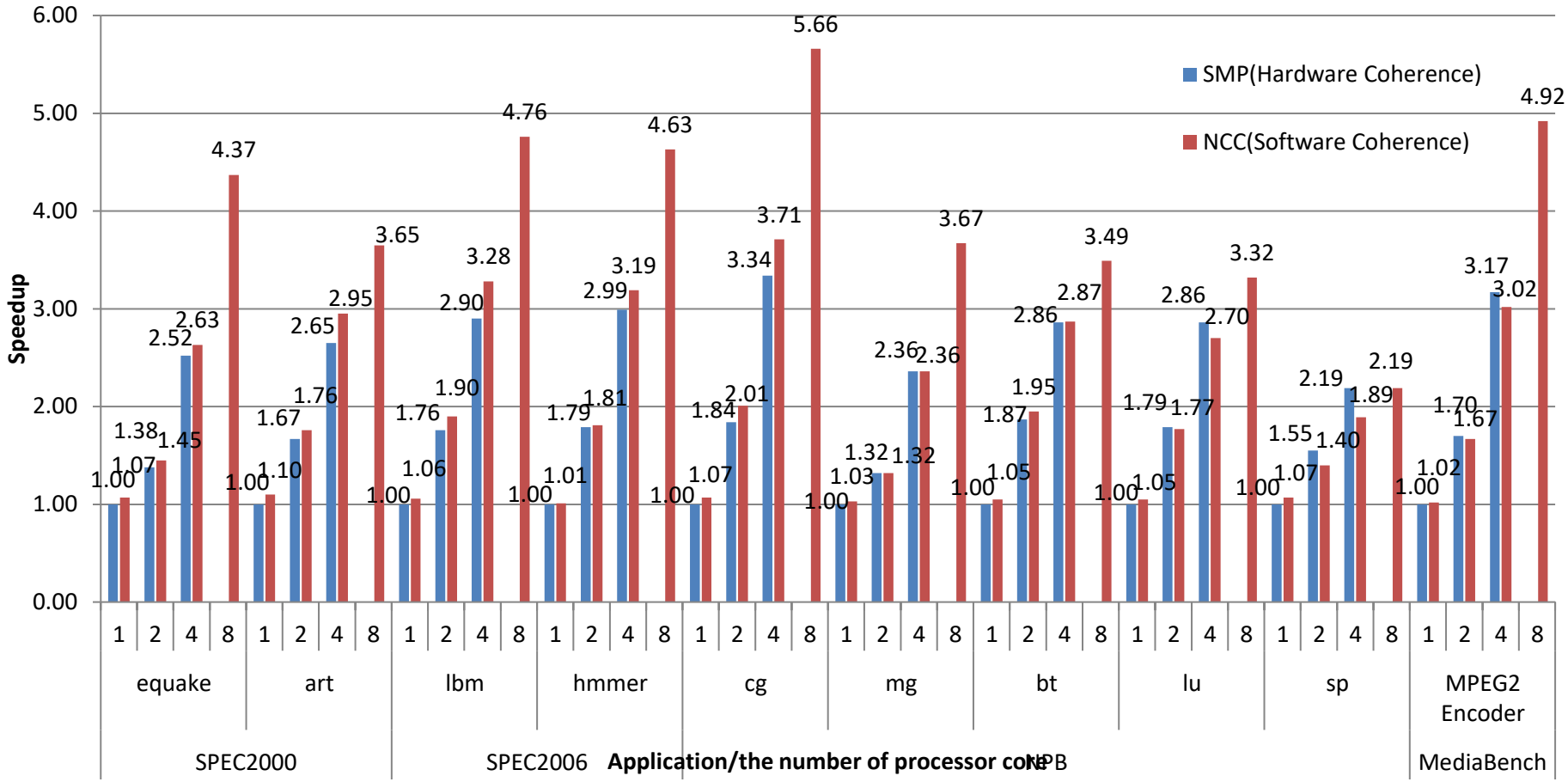
- Coarse grain task parallelization with earliest condition analysis (control and data dependency analysis to detect parallelism among coarse grain tasks).
- OSCAR compiler automatically controls coherence using following simple program restructuring methods:
  - To cope with stale data problems:
  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



MTG generated by earliest executable condition analysis

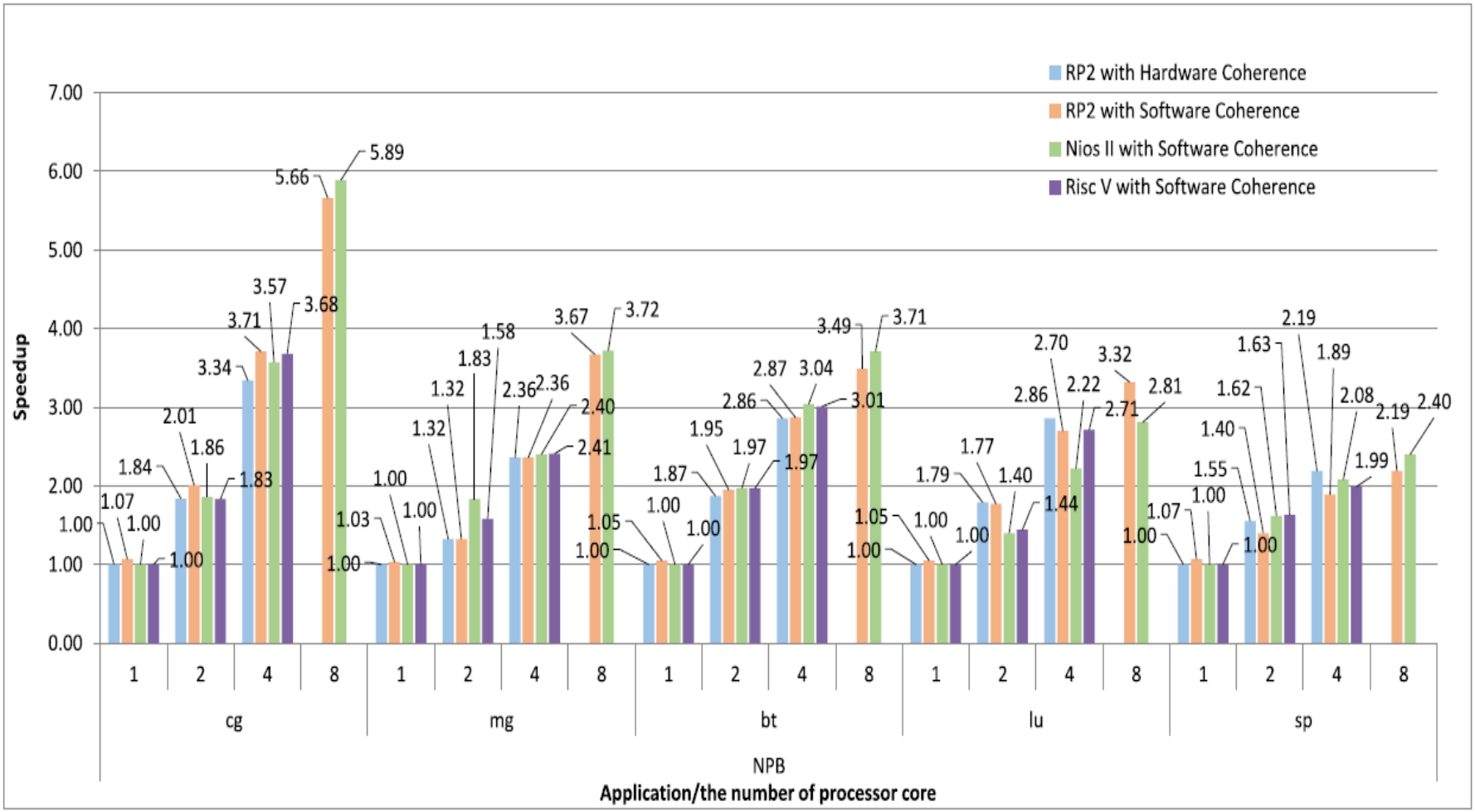
Automatic Software Coherent Control for Manycores

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



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

3.57x Speedups for NIOS and 3.68x for RISCV using 4 cores for NPB CG

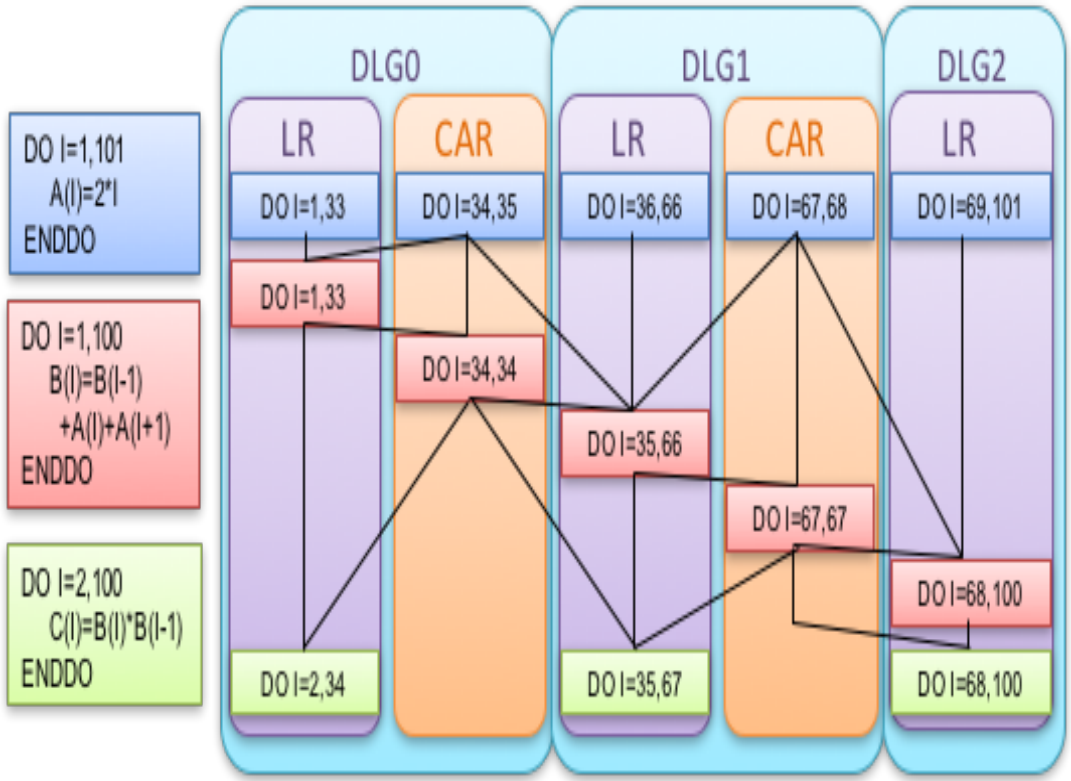


Automatic Local Memory Management

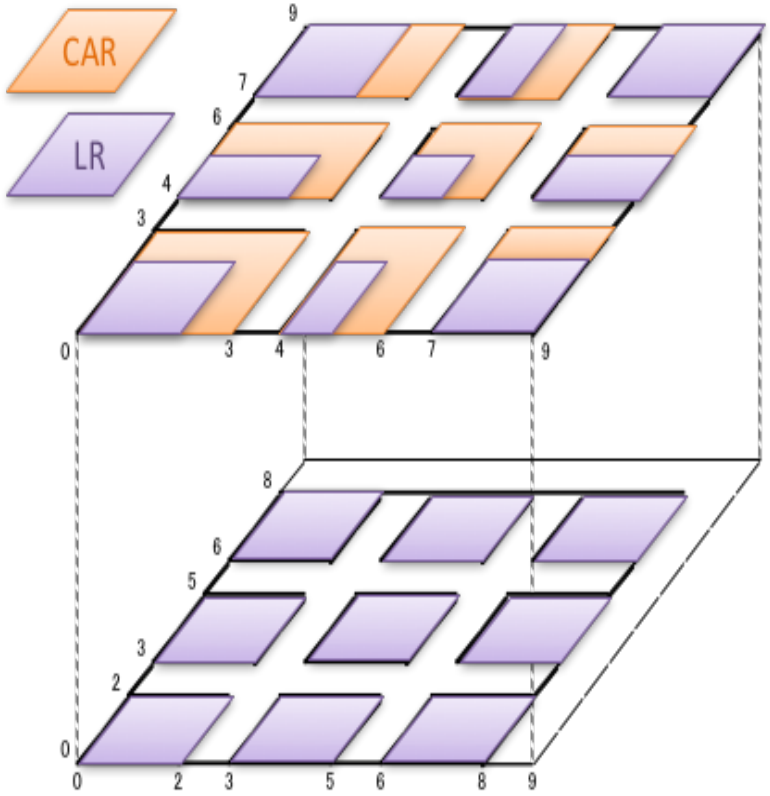
Data Localization: Loop Aligned Decomposition

- **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

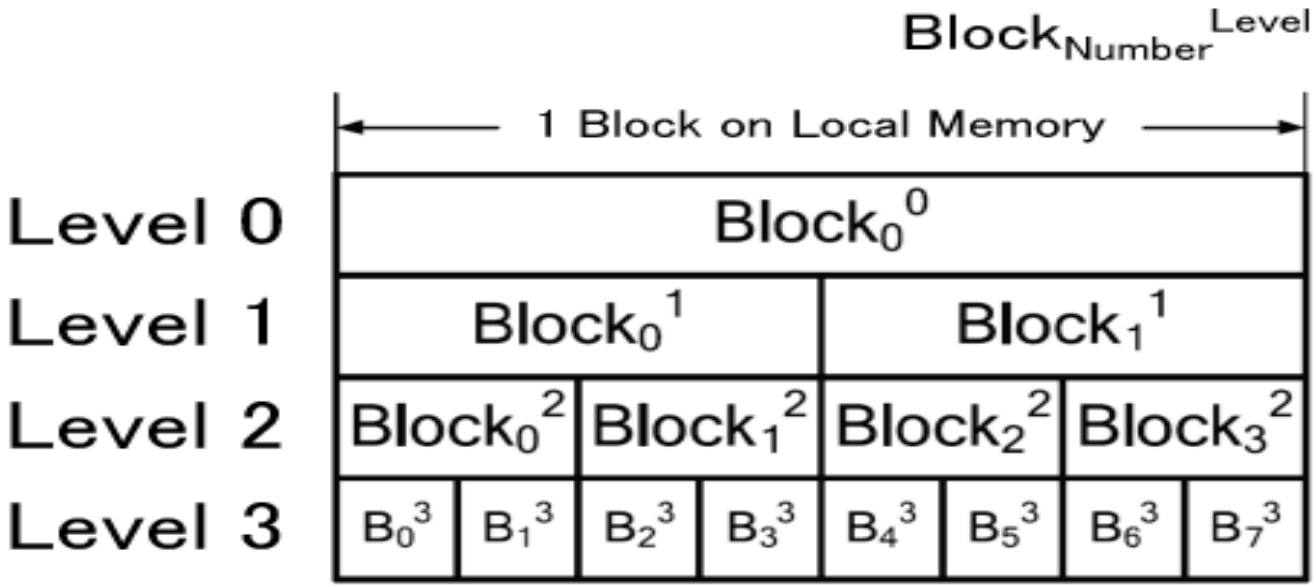


Multi-dimension Decomposition



Adjustable Blocks managed by Compiler

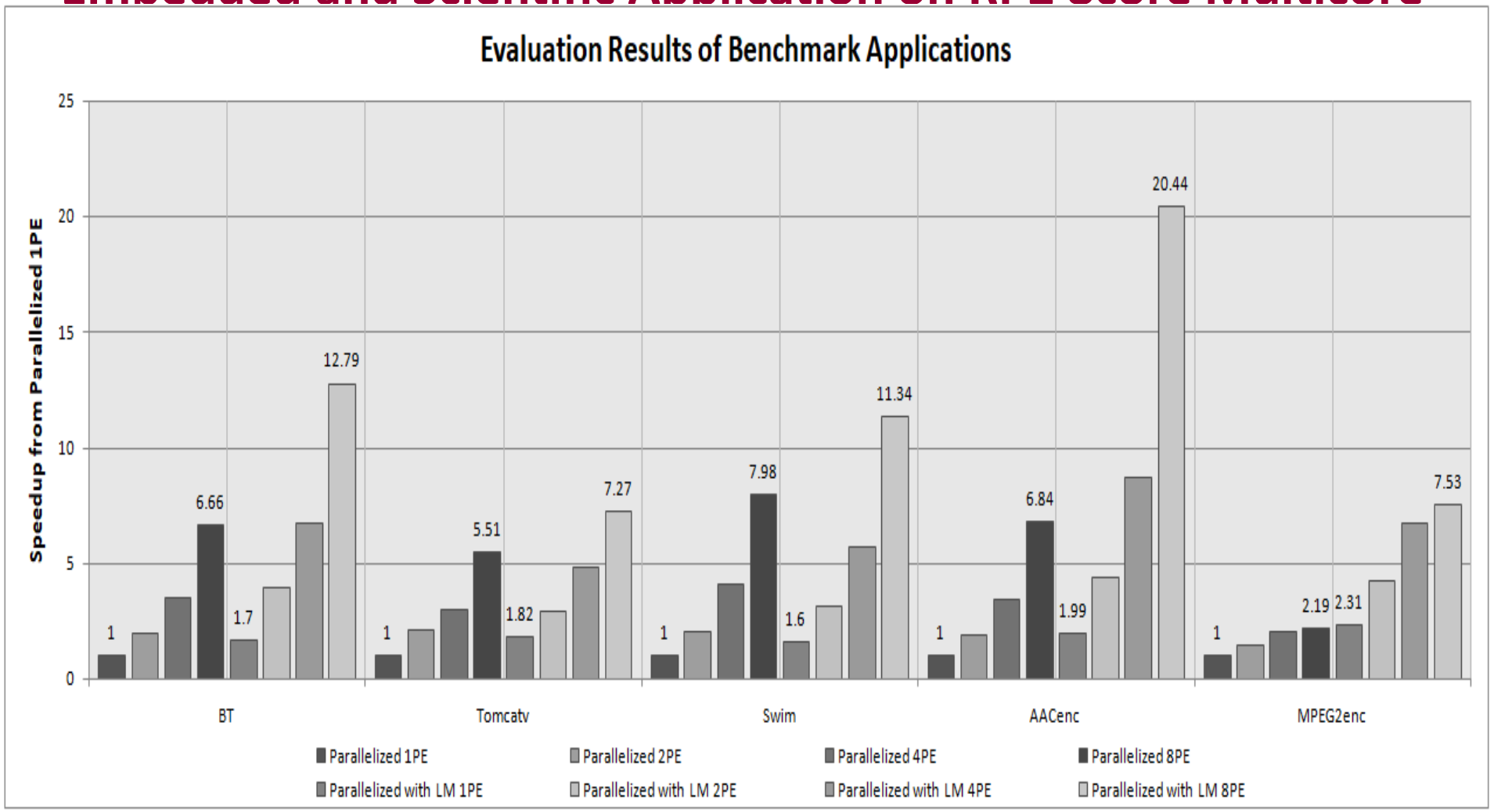
- A suitable block size for each application
  - different from a fixed block size in cache
  - each block can be divided into smaller blocks with integer divisible size to handle small arrays and scalar variables



Block Replacement Policy by OSCAR Compiler

- 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

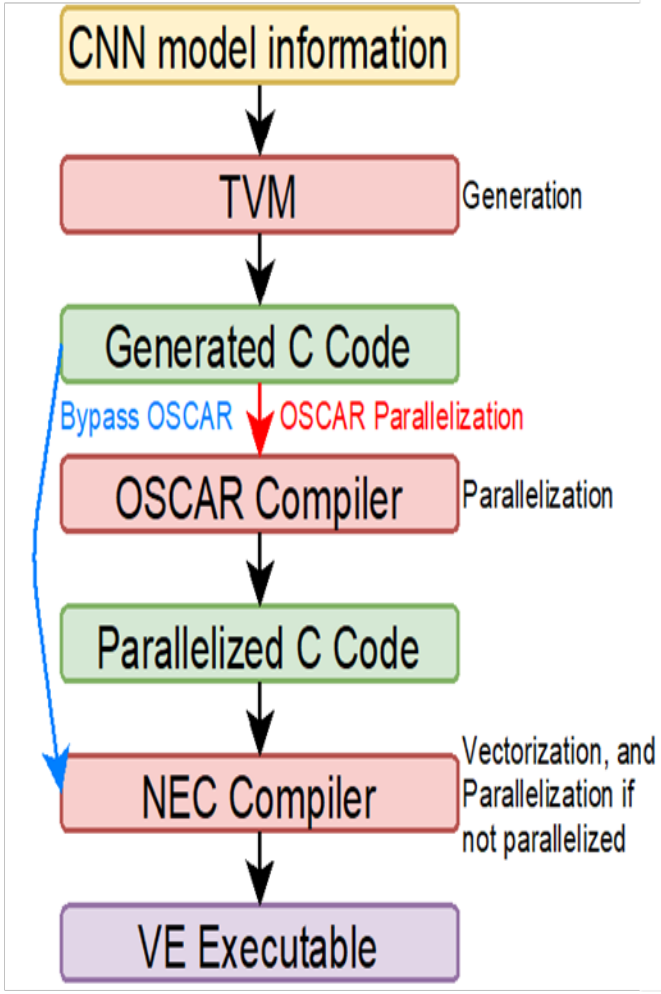


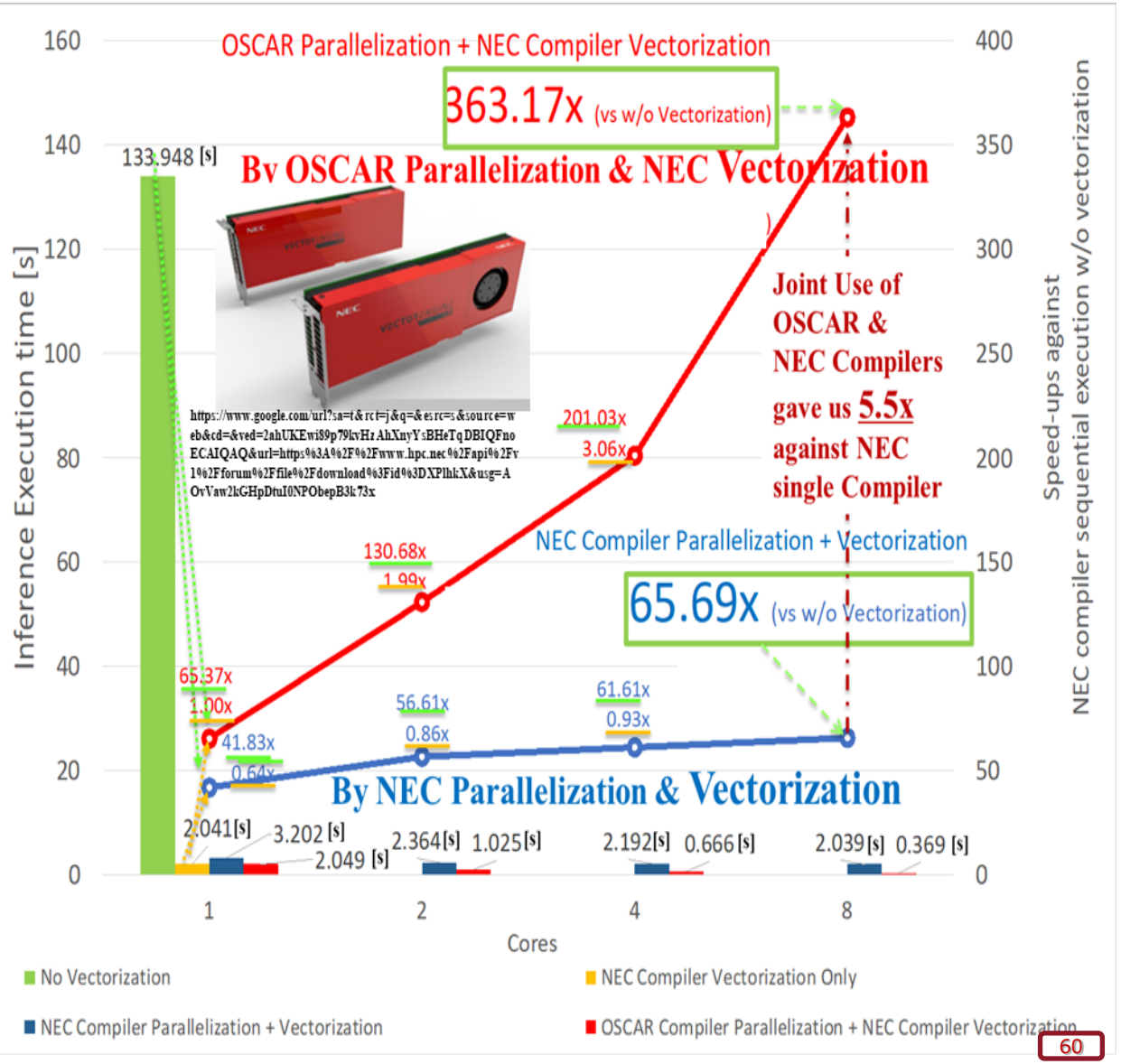
Maximum of 20.44 times speedup on 8 cores using local memory against sequential execution using off-chip shared memory

Speedups of Deep Learning Winograd 2D-Convolution generated by TVM on NEC Personal Vector Supercomputer SX-Aurora TSUBASA 8 Core Type 10C

OSCAR Parallelization and NEC Vectorization gave us 363x Speedup against a Scalar Core

Parallelization of Deep Leaning C Code generated by TVM

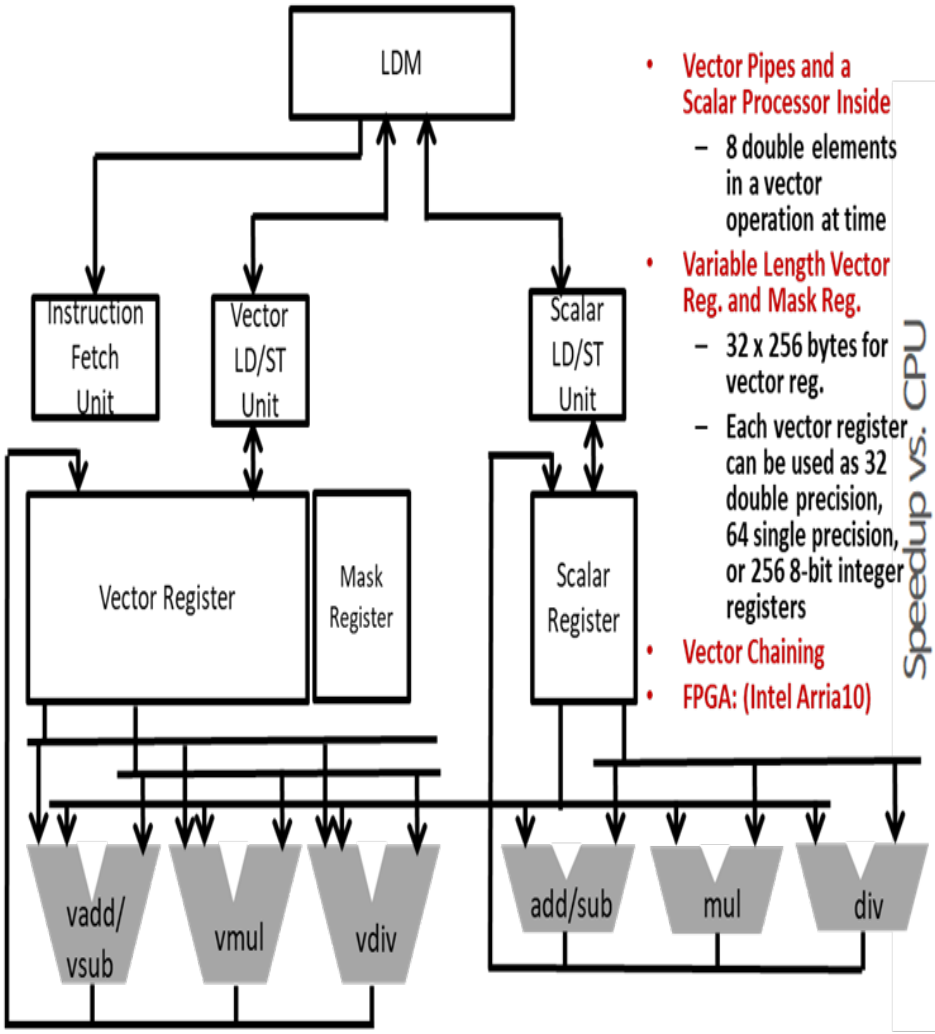




Performance for Multimedia & Scientific Applications on OSCAR Vector Accelerator

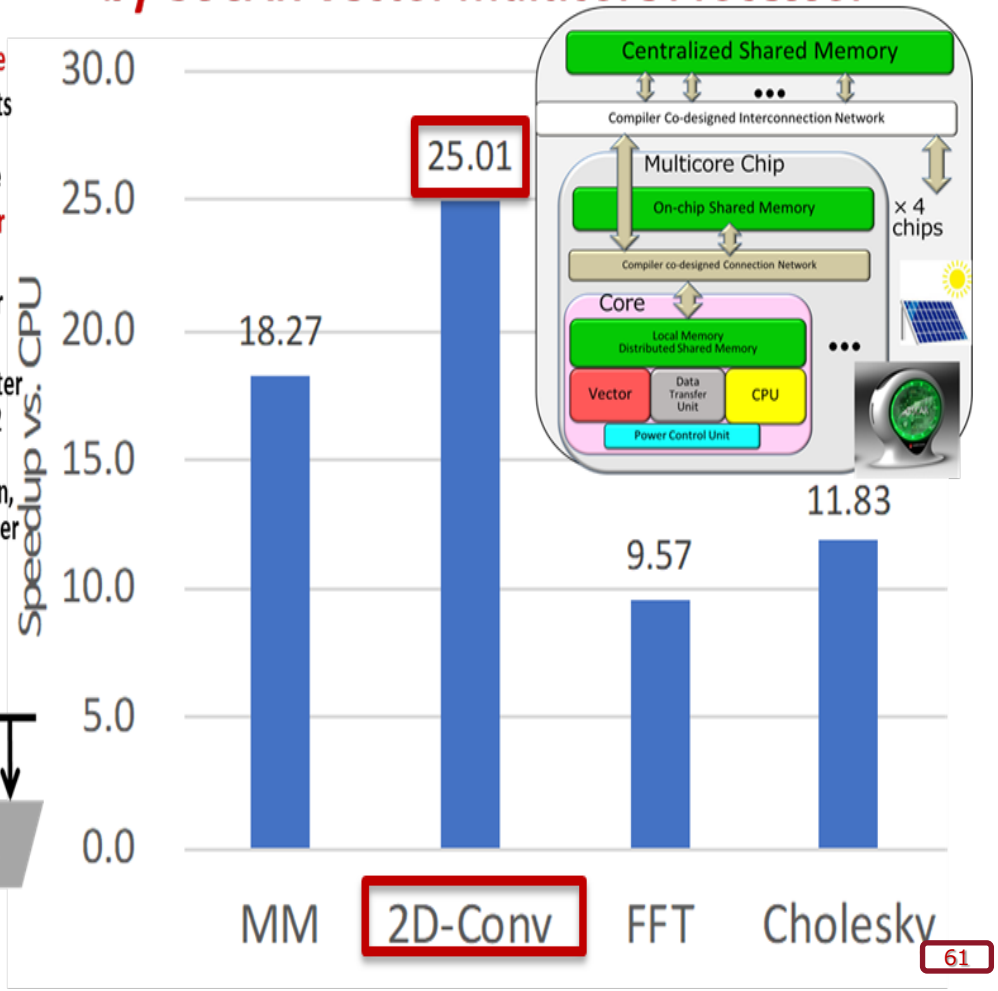
(A Vector Processor with Local Memory or Distributed Shared Memory and DMA Controller Managed by the OSCAR Compiler Redesigned Improving Japanese Supercomputer Technology in 1980-2000)

OSCAR Vector Accelerator on FPGA



- Vector Pipes and a Scalar Processor Inside
  - 8 double elements in a vector operation at time
- Variable Length Vector Reg. and Mask Reg.
  - 32 x 256 bytes for vector reg.
  - Each vector register can be used as 32 double precision, 64 single precision, or 256 8-bit integer registers
- Vector Chaining
- FPGA: (Intel Arria10)

Speedups against General Purpose Processor by OSCAR Vector Multicore Processor



Future Multicore Products with Automatic Parallelizing Compiler

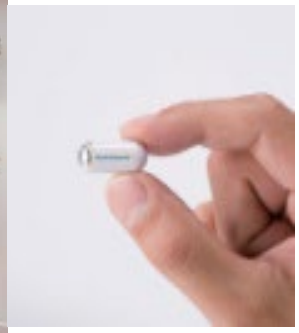
Next Generation Automobiles

- 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

ISCA2025, June 21-25, 2025, Waseda University, Tokyo, Japan

ACM SIGARCH

General Co-Chairs: Jean-Luc Gaudiot (Prof. UCI)
Hironori Kasahara (Prof. Waseda)

IEEE COMPUTER SOCIETY TCCA
Technical Committee on Computer Architecture