

# 低消費電力高性能コンピュータの将来

— 太陽電池で駆動できるマルチコア・メニーコアとソフトウェア —

## 早稲田大学

理工学術院基幹理工学部情報理工学科 教授  
アドバンスト・マルチコア・プロセッサ研究所 所長

IEEE Computer Society理事

笠原博徳

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

1985年 早稲田大学博士課程了 工学博士  
カリフォルニア大学バークレー客員研究員  
日本学術振興会第1回特別研究員  
1986年 早大理工専任講師, 1988年 助教授  
1997年 教授、現在 理工学術院情報理工学科  
1989年～1990年 イリノイ大学Center for  
Supercomputing R&D客員研究員  
2009年 IEEE Computer Society 理事 (Golden Core  
Member 2010-)

査読付論文 172件, シンポジウム論文 27件, 研究会論文 125件, 全国大会論文 154件, 招待講演 83件, **新聞・Web記事・TV等メディア掲載 400件以上**

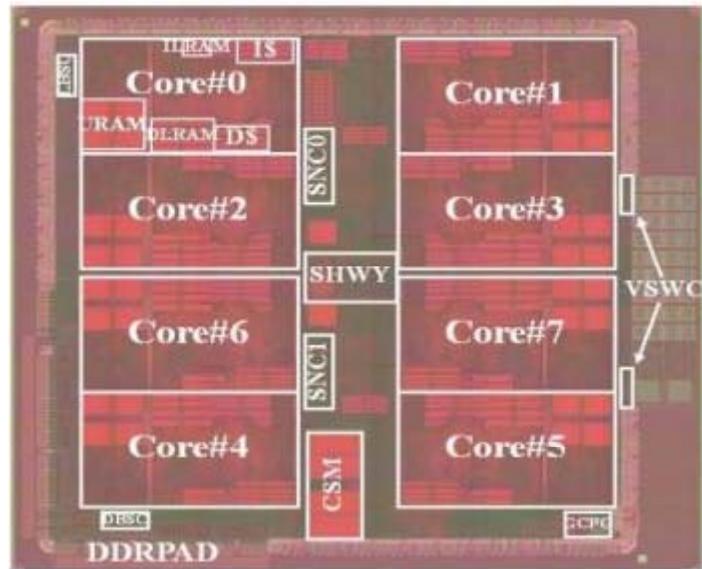
### 受賞

1987年 IFAC World Congress Young Author Prize  
1997年 情報処理学会坂井記念特別賞  
2005年 STARC(半導体理工学研究センタ)共同研究賞  
2008年 LSI・オブ・ザ・イヤー 2008 準グランプリ  
2008年 Intel Asia Academic Forum Best Research Award  
2009年IEEE Computer Society Golden Core Member

### 政府・学会委員等歴任数 201件

【経済産業省・NEDO】 30件 【内閣府】 8件  
【文部科学省・JST・JSPS・JAXA・原子力機構・海洋研】 26件  
IEEE 21件, 情報処理学会 36件, ACM 14件, 国際会議PC等 57  
件, 高校生科学技術チャレンジ審査委員等 他9件

# マルチコアEverywhereの時代



## ■ 組み込みプロセッサからスパコンまで

### ➤ 情報家電用マルチコア

カーナビ, 携帯電話, ゲーム, デジタルTV, DVD

IBM/ Sony/ Toshiba Cell (9 cores), Fujitsu FR1000 (4 cores),

Panasonic Uniphier, NEC/ARM MPCore/MP211/NaviEngine (4 cores)

Renesas, Hitachi & Waseda 4 core RP1, 8 core RP2, 15 core Hetero RPX

Tilera Tile64, SPI Storm-1(16 VLIW cores)

### ➤ PC, サーバ

Intel Quad Xeon, Core 2 Quad, Montvale, Nehalem(8cores), 80 cores,

Larrabee(32cores), SCC: Single Chip Cloud (48 cores)

AMD Quad Core Opteron, Phenom

### ➤ WSs, Deskside & Highend Servers

IBM Power4,5,5+,6, 7 (HPCS 実効性能1PFLOP: 8 cores),

Cyclops64 (80 cores), Sun Niagara(SparcT1,T2), Rock

### ➤ スーパーコンピュータ

地球シミュレータ, 2002年3月 5120ベクトルプロセッサ :40TFLOPS

IBM Blue Gene/L: 360TFLOPS, 2005,低消費電力マルチコアベース

128K processor chips, BG/Q 20PFLOPS (2011)

Blue Waters (2011年7月実効性能1PFLOPS:Power7)

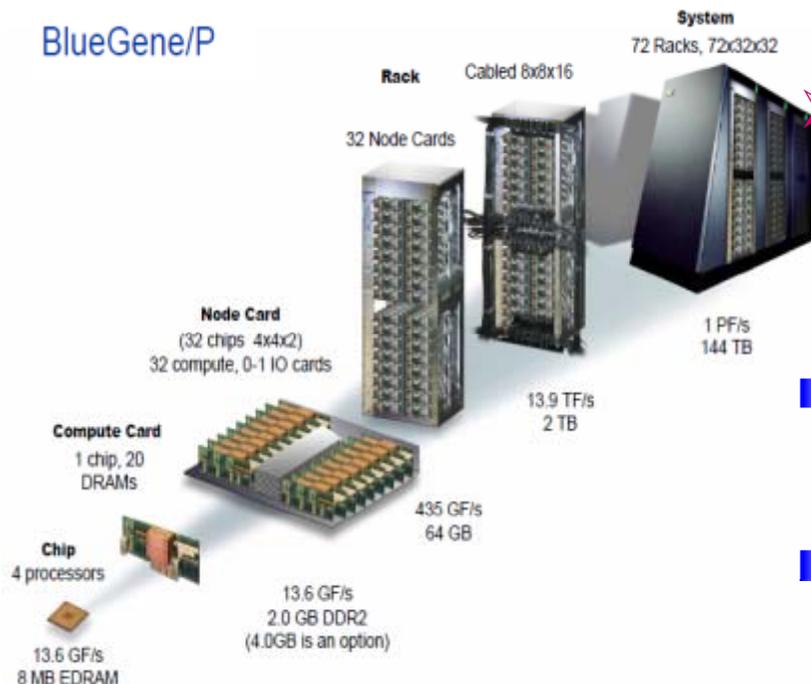
## ■ アプリケーションソフトの充実, 短期間システム開発, 低コスト, 低消費電力, 高機能化が市場競争力決定

<例>携帯電話, ゲーム, 自動車

## ■ 並列化コンパイラ協調型マルチコアプロセッサ必要<sub>2</sub>

NEDOリアルタイム情報家電用マルチコアプロジェクトルネサス試作チップ(OSCAR)標準アーキテクチャ

BlueGene/P



# NEDOリアルタイム情報家電用マルチコア技術

＜目的＞ DVD,デジタルTV,カーナビ,携帯電話等リアルタイム情報家電における優位化製品創出を目指したプロセッサ技術の国際競争力強化

＜研究期間＞平成17年7月～平成20年3月

＜内容＞ 下記特徴をもつマルチコア・アーキテクチャ及びコンパイラ技術の確立

- ・優れた価格性能
- ・短ハードウェア・ソフトウェア開発期間
- ・低消費電力
- ・同一アーキテクチャでローエンドからハイエンド製品まで柔軟に対応
- ・集積度向上に対するスケラブルな性能向上(複数半導体世代に対応)
- ・異なる企業開発のマルチコア間でのアプリケーションの共用

＜API: Application Programming Interface並列化コンパイラと各マルチコアの接続＞

日立,富士通,ルネサス,東芝,松下,NEC

＜委託事業＞ NEDO”アドバンスト並列化コンパイラ“の成果と参加各社のプロセッサ技術を委員会及び研究により発展させ、標準的マルチコア・アーキテクチャと並列化APIを策定し、既存及び新規チップ上での性能評価を通し改良する

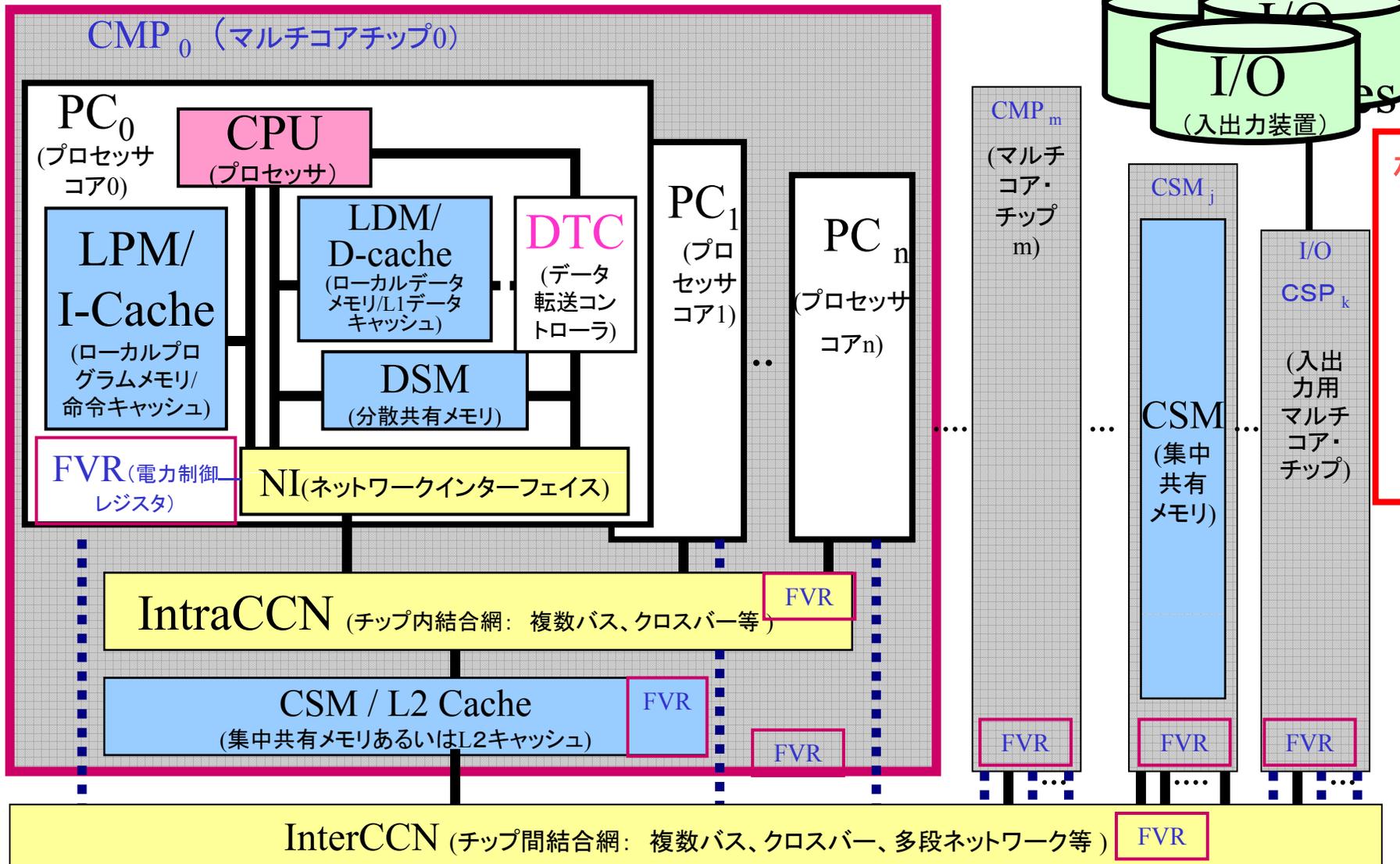
＜助成事業＞ 委託事業にて策定されたアーキテクチャをベースとしたチップ及びその開発環境を開発し委託研究での評価のために供与する

＜PC用プロセッサとの目的の違い＞

リアルタイム処理なアーキテクチャ・自然冷却可能な低消費電力性

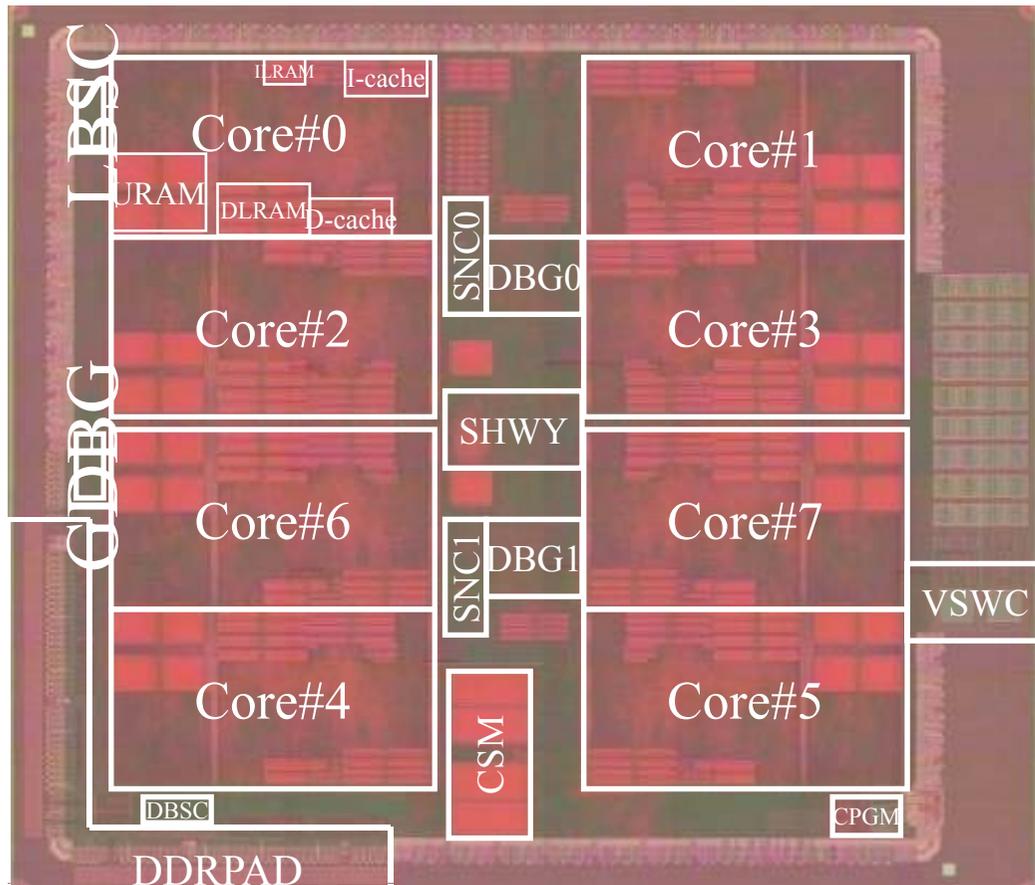
# 標準的メモリアーキテクチャと承認されたOSCARマルチコア

- ローカルメモリ
- 分散共有メモリ
- 集中共有メモリ(チップ内・外)
- DTC(高機能DMAC)
- 電力制御用FVR



# 早稲田OSCARコンパイラ協調型アーキテクチャ

## RP2 8コア搭載マルチコアLSI



8コア集積マルチコアLSIチップ写真

プロセス	90nm CMOS, 8層メタル, 3種Vth
チップサイズ	104.8mm <sup>2</sup> (10.61mm x 9.88mm)
電源電圧	1.0V-1.4V(コア), 1.8/3.3V(I/O)
動作周波数	600MHz
CPU性能	8640 MIPS (Dhrystone 2.1)
FPU性能	33.6 GFLOPS
低電力制御	<ul style="list-style-type: none"> <li>・CPU毎に独立した周波数変更</li> <li>・CPUコアのクロックを停止するスリープモード</li> <li>・CPUコアの一部のクロックを停止するがキャッシュコヒーレンシ維持可能なライトスリープモード</li> <li>・CPUコアの電源供給を停止するフル電源遮断モード</li> <li>・URAM以外のCPUコアの電源供給を停止するレジューム電源遮断モード</li> </ul>

ISSCC08発表: ISSCC08 論文番号4.5, M.ITO, et al., “An 8640 MIPS SoC with Independent Power-off Control of 8 CPUs and 8 RAMs by an Automatic Parallelizing Compiler”

# NEDOリアルタイム情報家電用マルチコアチップ・デモの様子

<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)

# 世界をリードするマルチコア用コンパイラ技術

プロセッサ高速化における3大技術課題の解消

## 1. 半導体集積度向上(使用可能トランジスタ数増大)に対する速度向上率の鈍化

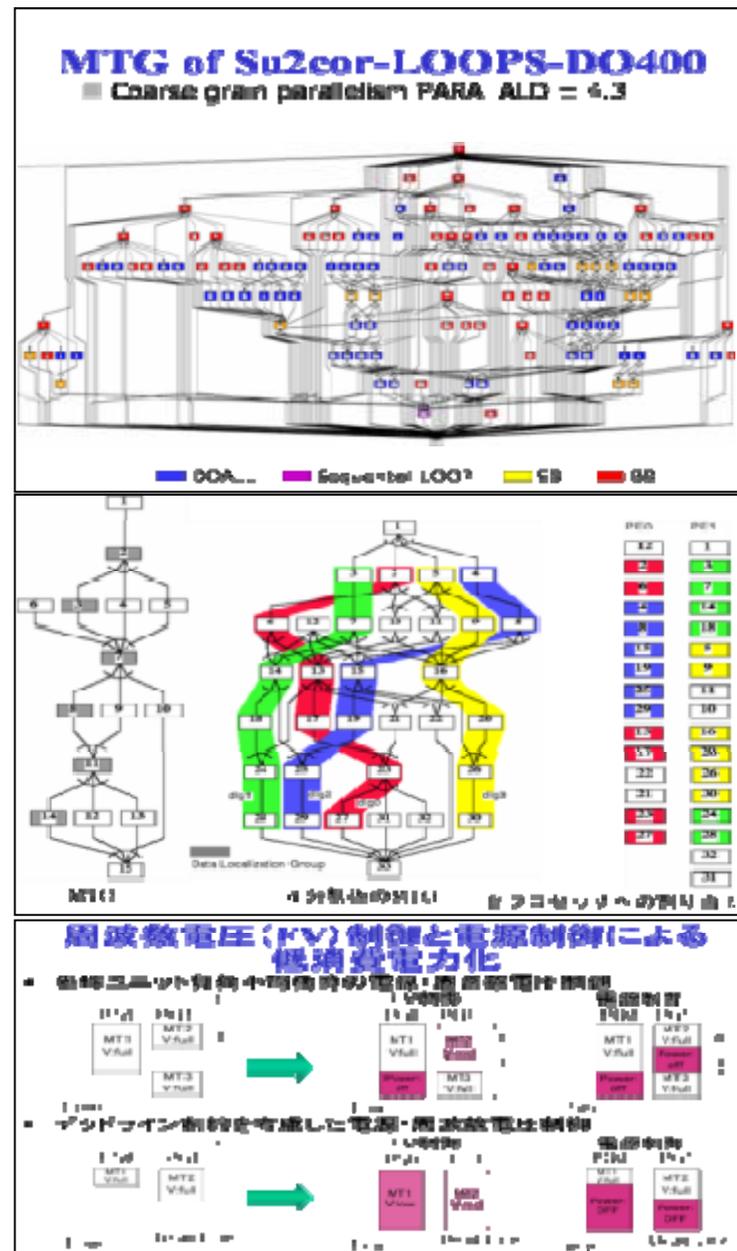
- 粗粒度タスク並列化、ループ並列化、近細粒度並列化によりプログラム全域の並列性を利用するマルチグレイン並列化機能により、従来の命令レベル並列性より大きな並列性を抽出し、複数マルチコアで速度向上

## 2. メモリウォール問題

- コンパイラによるローカルメモリへのデータ分割配置、DMAコントローラによるタスク実行とオーバーラップしたデータ転送によりメモリアクセス・データ転送オーバーヘッド最小化

## 3. 消費電力増大による速度向上の鈍化

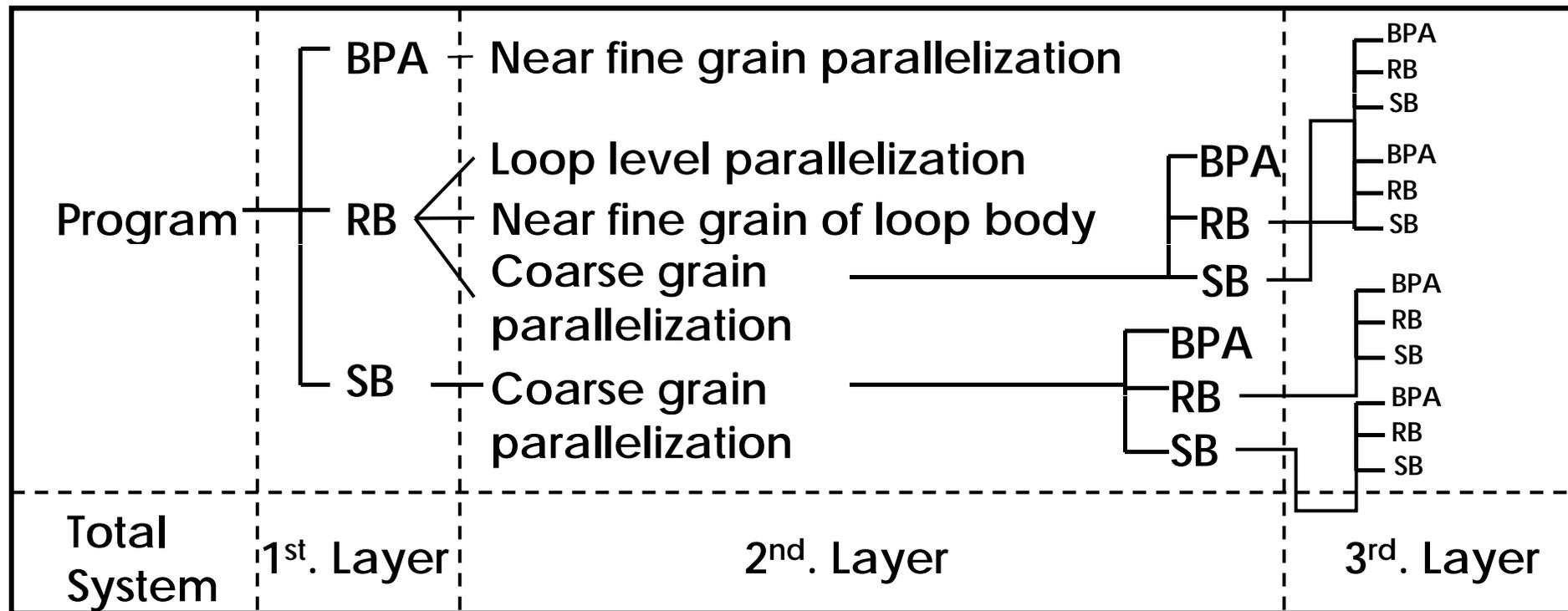
- コンパイラによる低消費電力制御機能を用いたアプリケーション内でのきめ細かい周波数・電圧制御・電源遮断により消費電力低減



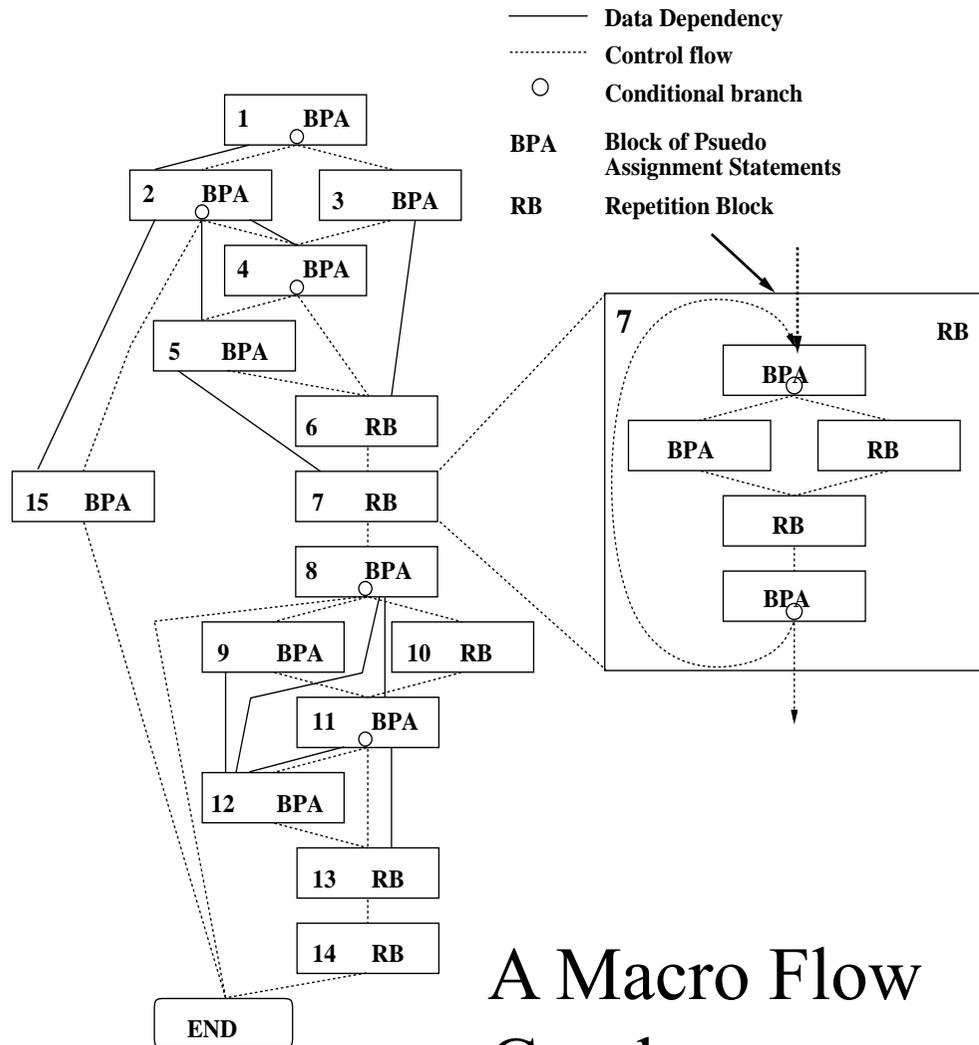
# Generation of coarse grain tasks

## ■ Macro-tasks (MTs)

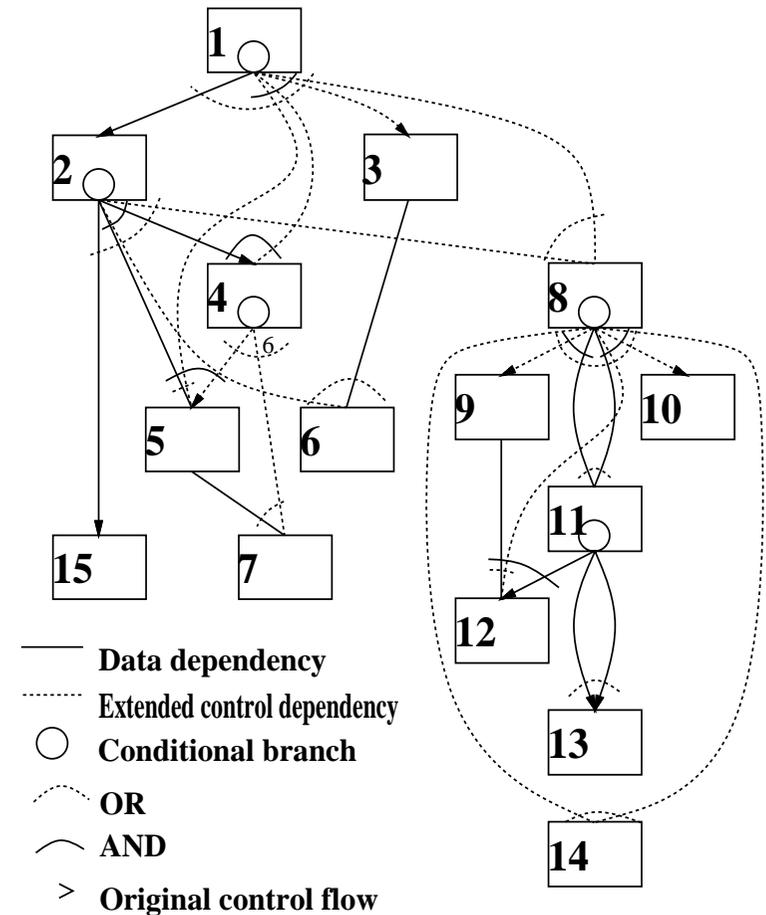
- Block of Pseudo Assignments (BPA): Basic Block (BB)
- Repetition Block (RB) : natural loop
- Subroutine Block (SB): subroutine



# Earliest Executable Condition Analysis for coarse grain tasks (Macro-tasks)



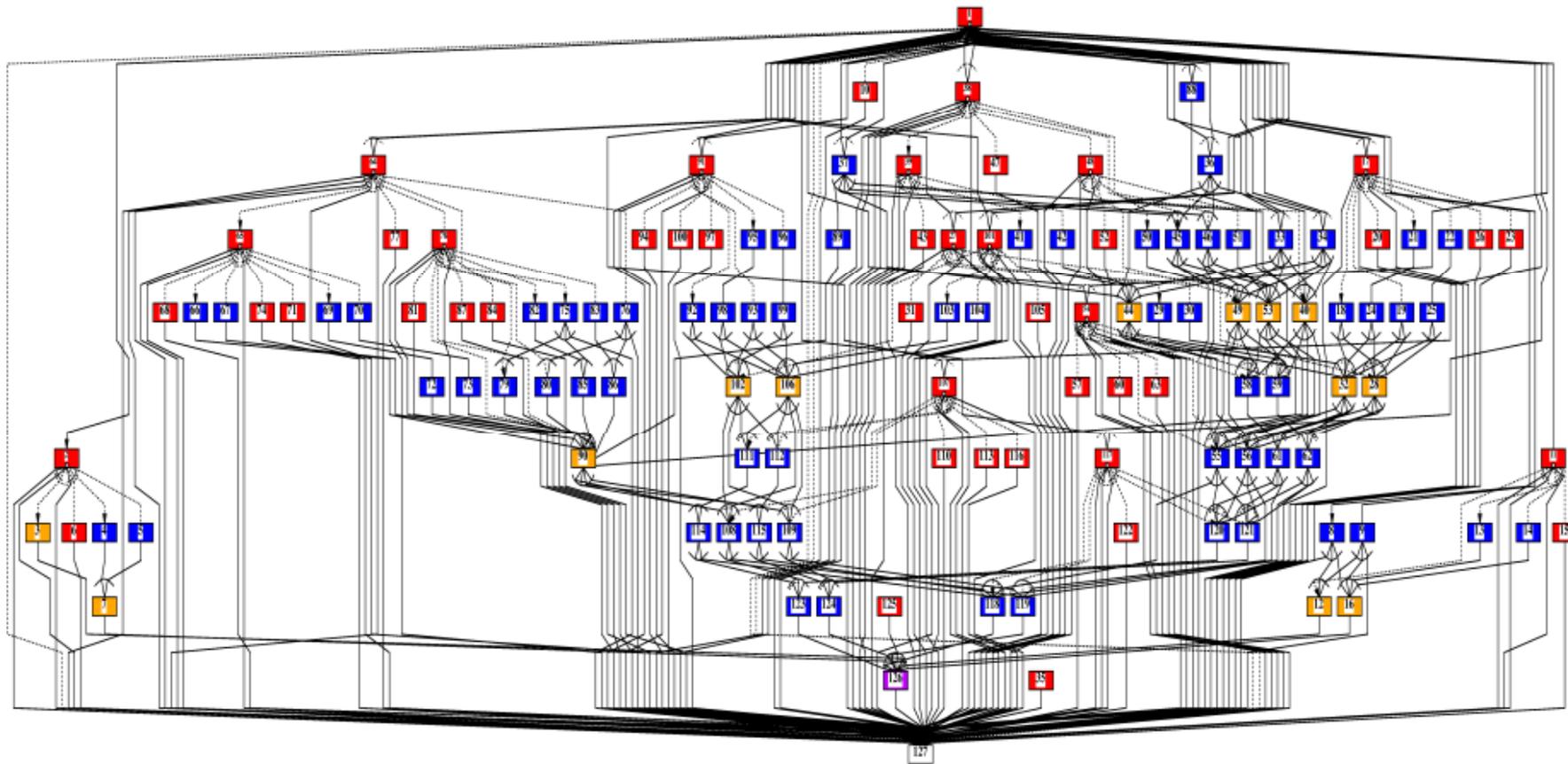
A Macro Flow Graph



A Macro Task Graph

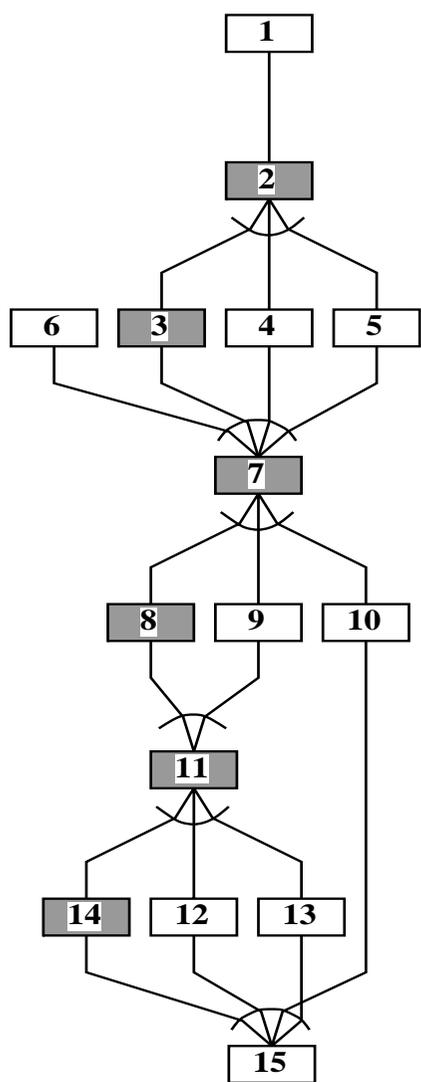
# MTG of Su2cor-LOOPS-DO400

- Coarse grain parallelism  $\text{PARA\_ALD} = 4.3$

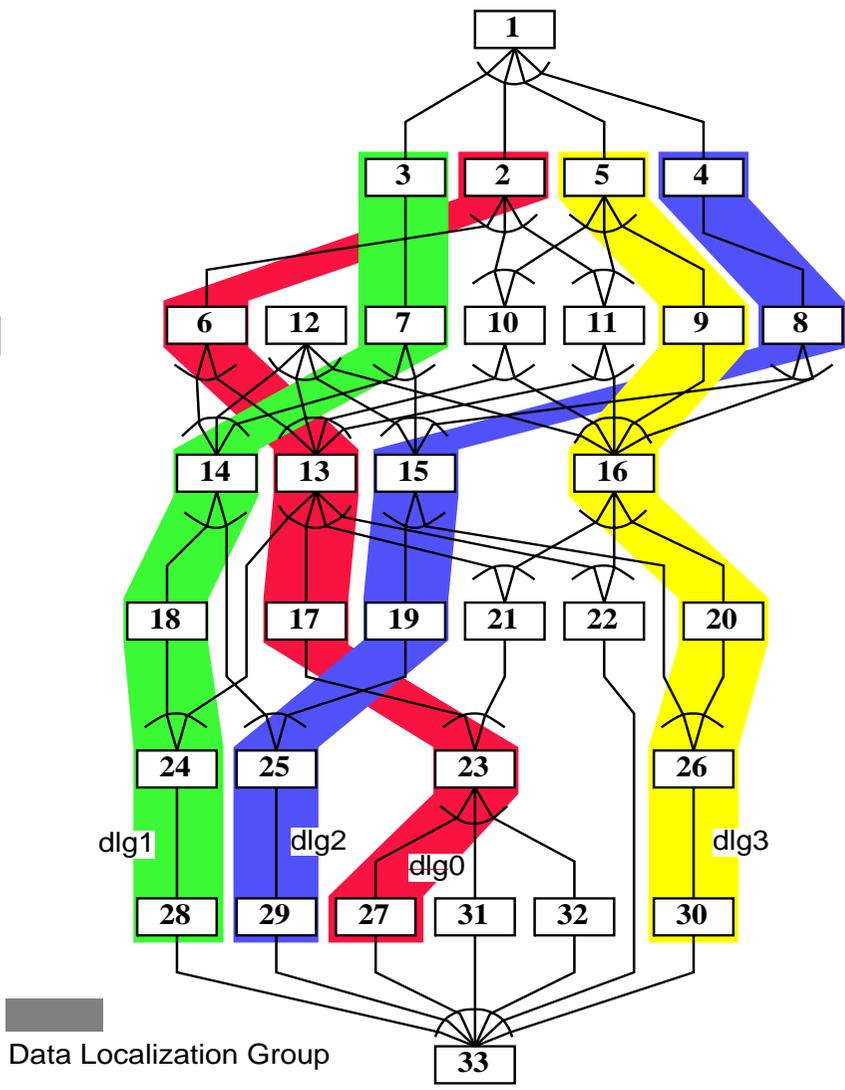


■ DOALL ■ Sequential LOOP ■ SB ■ BB

# データローカライゼーションのイメージ



MTG

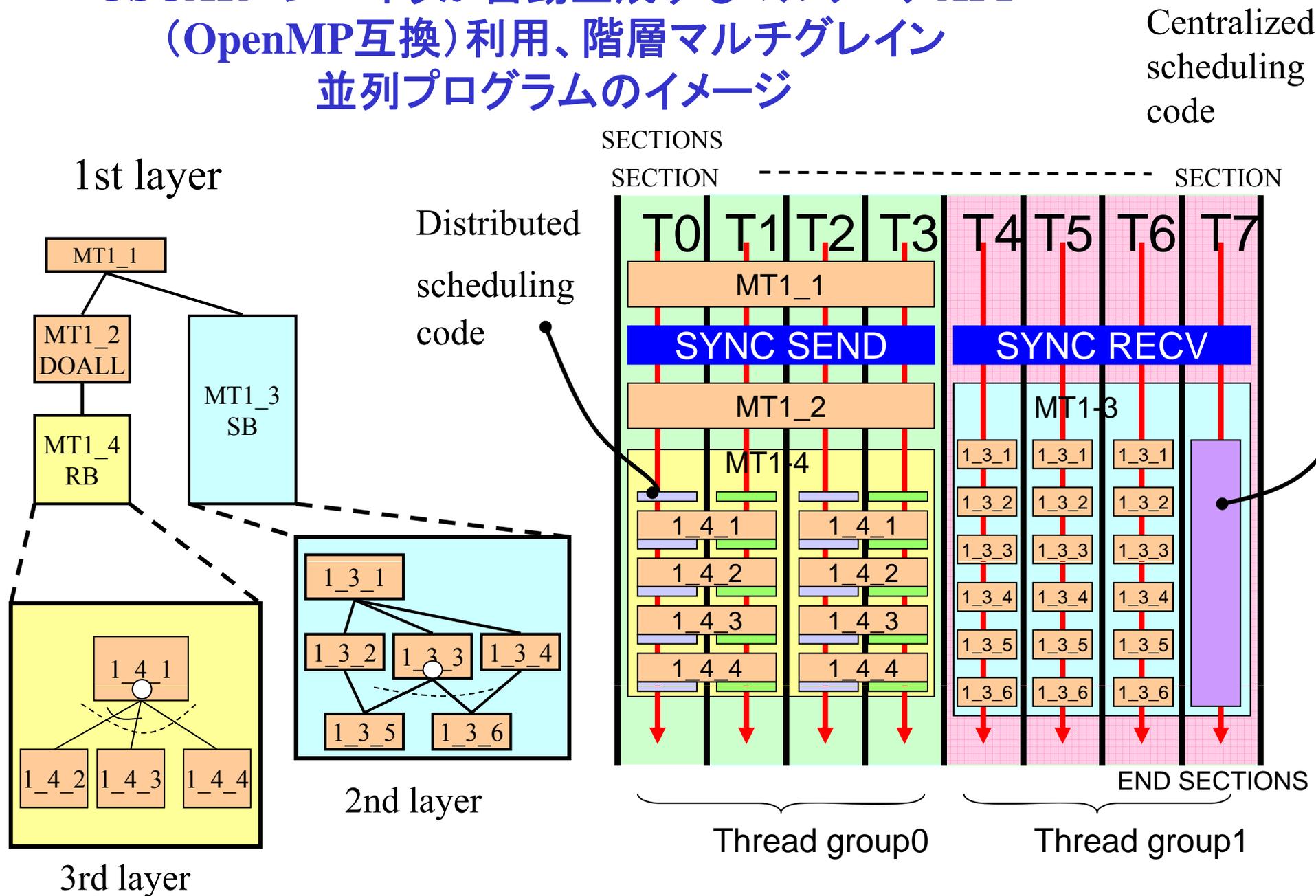


4 分割後のMTG

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

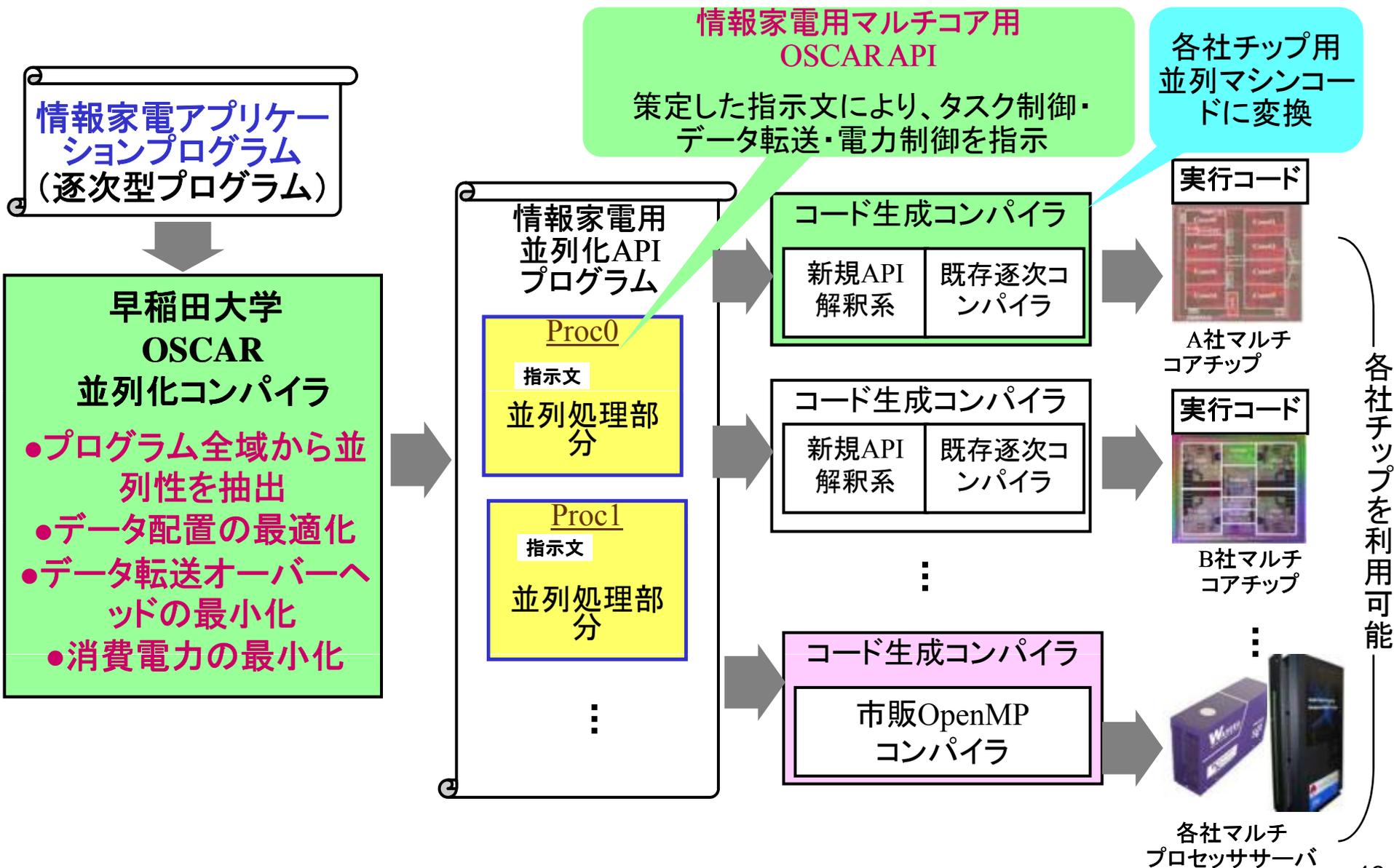
2 プロセッサへの割り当て

# OSCARコンパイラが自動生成するマルチコアAPI (OpenMP互換)利用、階層マルチグレイン 並列プログラムのイメージ



# OSCAR APIを用いたコンパイル・実行の流れ

API: Application Programming Interface



# 低消費電力リアルタイム並列処理を可能とするAPI OSCAR API概要

## <主な特徴>

- ユーザプログラムからの低消費電力制御を可能とする指示文
- リアルタイム並列処理を可能とするメモリ管理・時間管理用指示文
- プロセッサによる計算実行と並列してメモリからのデータ転送を可能とするDMA転送指示文
- 異なる企業のマルチコア間でのプログラム移植が容易
- 15種の指示文のみに抑えたコンパクトな規格で、指示文を理解する小規模ソフトウェアを開発するだけで、既存の逐次コンパイラを用いて並列マシンコードを作成できる。初期導入コストが小
- 早稲田OSCAR並列化コンパイラの利用により、逐次プログラムからAPI入り並列化プログラムの自動生成可能。人手では数ヶ月を要する低消費電力並列化プログラムを数分で自動作成可
- 新規指示文はプログラム中のコメント文の形でユーザプログラムに挿入されるため、市販OpenMPコンパイラを用い、マルチコアPC、サーバ上でも実行可能

# 低消費電力リアルタイム並列処理を可能とするAPI

## OSCAR API概要

指示文リスト 詳細は <http://www.kasahara.cs.waseda.ac.jp/index.ja.html> 参照

- 共有メモリ並列処理用 OpenMP 指示文 (4 種)
  - 並列スレッドの生成指示文
  - 排他同期制御用クリティカルセクション指示文
  - メモリー貫性制御指示文
  - スレッドプライベート変数指示文
- <新規>組込及びリアルタイム処理用メモリ配置指示文 (3 種)
  - オンチップ集中共有メモリへのデータ配置を指定する指示文
  - ローカルデータメモリにデータを配置する指示文 (OpenMP 指示文の拡張利用)
  - 各プロセッサコア上の分散共有メモリへのデータ配置を指定する指示文
- <新規>組込及び高性能計算用データ転送指示文 (4 種)
  - データ転送コントローラを用いた転送指定する指示文 (詳細指示は下記指示文を用いて指定)
  - 連続したデータ領域の転送を指定する指示文
  - とびとびのデータを転送するストライド転送を指定する指示文
  - データ転送コントローラによる同期フラグセット及びチェックを行う指示文
- <新規>低消費電力制御用指示文 (2 種)
  - CPU 等の各種モジュールの周波数・電圧・電源遮断制御を下記 100 分率で指定する指示文  
100 : 最大周波数で動作、50 : 周波数を 1/2 にスローダウン、0 : クロックオフ等。各動作周波数に対応した電圧レベルの指示可能。-1 : リーク電力を抑える電源遮断。
  - 指定したモジュールの周波数・電源状態を取得する指示文
- <新規>メニーコア対応ネスト並列処理用グループバリア同期指示文 (1 種)
  - 任意の CPU グループに対するバリア同期を指定する指示文
- <新規>リアルタイム処理用 タイマー指示文 (1 種)
  - デッドライン管理のため経過時間を取得するタイマー指示文

# OSCAR API v1.0の指示文

- ▶ 並列実行API
  - ▶ `parallel sections (*)`
  - ▶ `flush (*)`
  - ▶ `critical (*)`
  - ▶ `execution`
- ▶ メモリ配置API
  - ▶ `threadprivate (*)`
  - ▶ `distributedshared`
  - ▶ `onchipshared`
- ▶ 同期API
  - ▶ `groupbarrier`
- ▶ データ転送API
  - ▶ `dma_transfer`
  - ▶ `dma_contiguous_parameter`
  - ▶ `dma_stride_parameter`
  - ▶ `dma_flag_check`
  - ▶ `dma_flag_send`
- ▶ 電力制御API
  - ▶ `fvcontrol`
  - ▶ `get_fvstatus`
- ▶ タイマーAPI
  - ▶ `get_current_time`

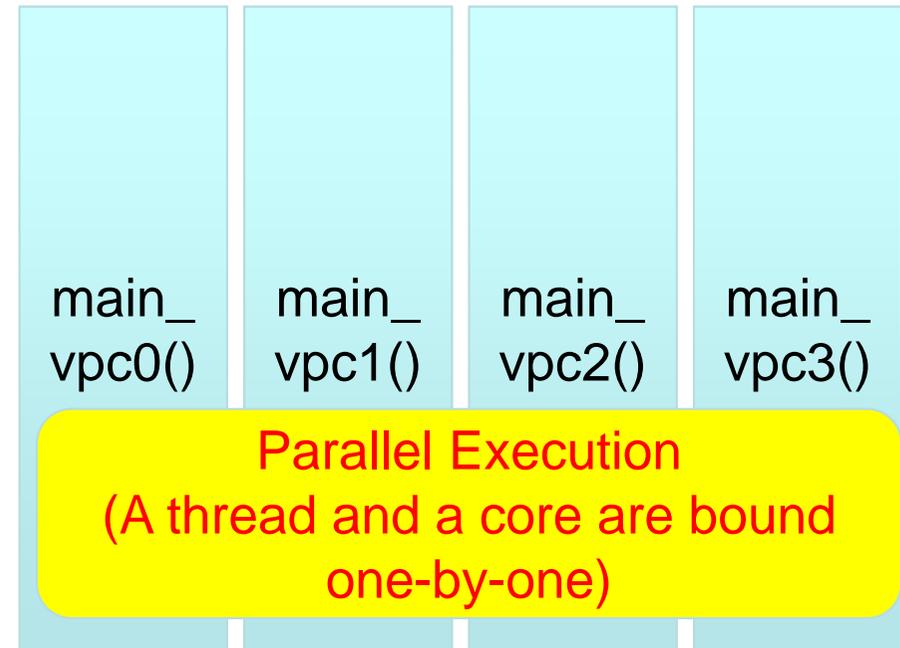
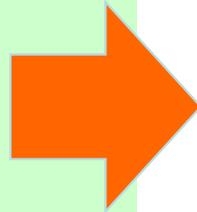
(\* OpenMPからの指示文)

# Parallel Execution

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

# Thread Execution Model

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



VPC: Virtual Processor Core

# Memory Mapping

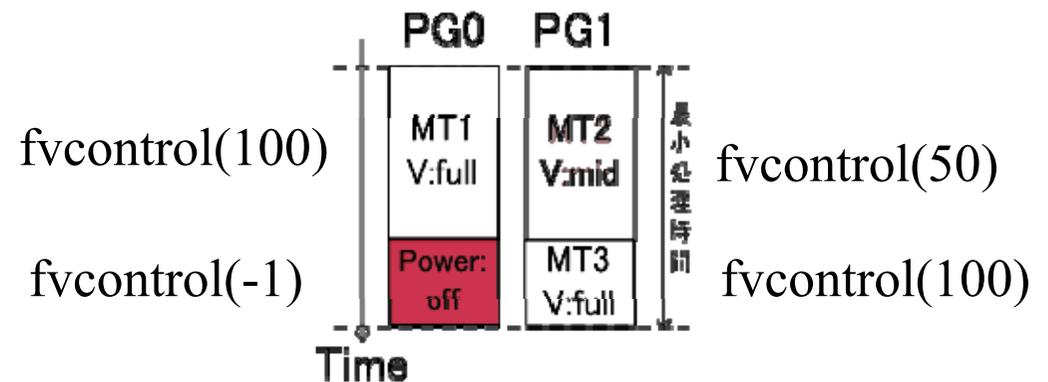
- **Placing variables on an onchip centralized shared memory (onchipCSM)**
  - `#pragma oscar onchipshared (C)`
  - `!$oscar onchipshared (Fortran)`
- **Placing variables on a local data memory (LDM)**
  - `#pragma omp threadprivate (C)`
  - `!$omp threadprivate (Fortran)`
  - This directive is an extension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

# Data Transfer

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

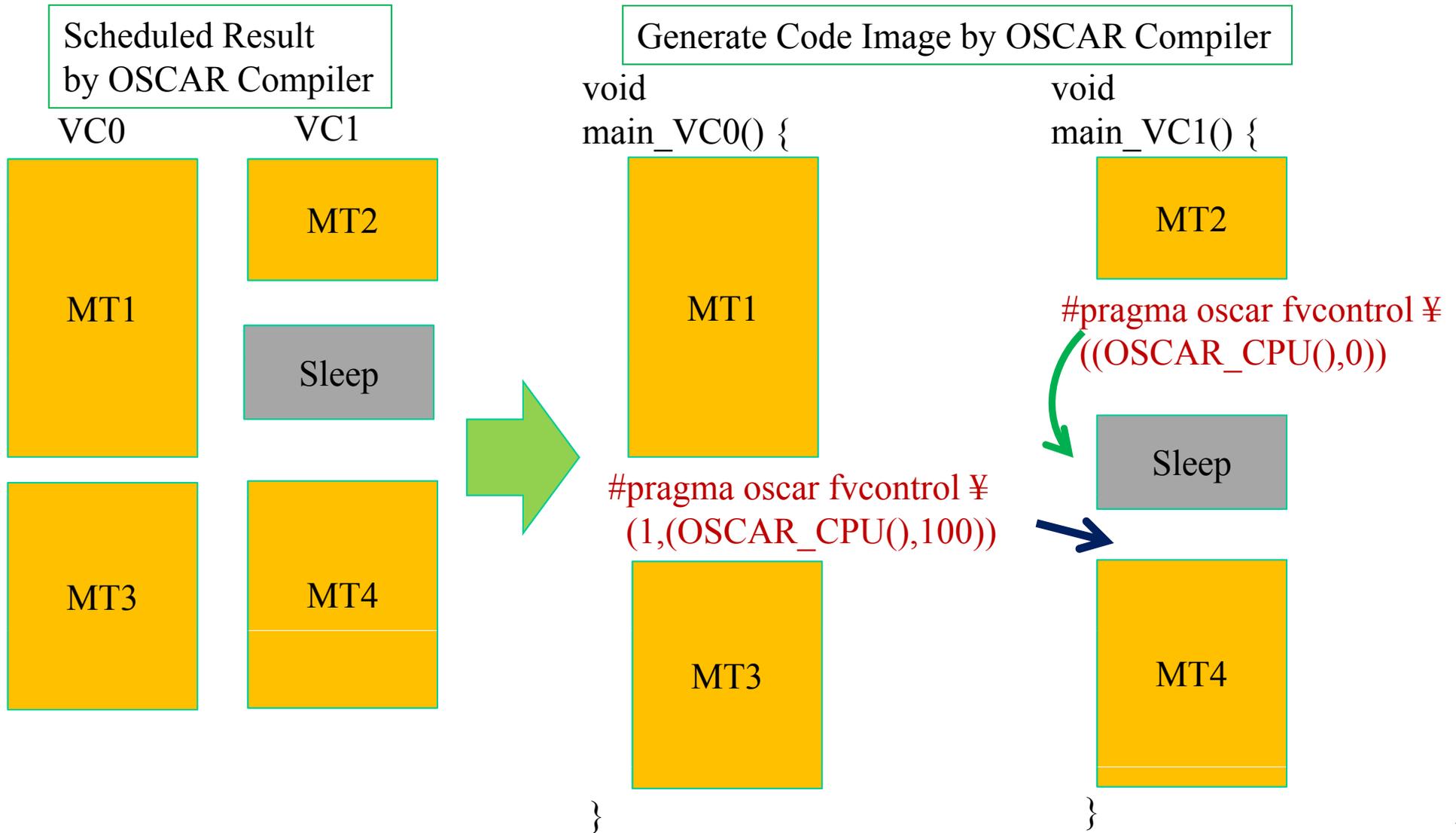
# Power Control

- Making a module into specifying frequency and voltage state
  - `#pragma oscar fvcontrol (C)`
  - `!$oscar fvcontrol (Fortran)`
  - state examples
    - **100: max frequency**
    - **50: half frequency**
    - **0: clock off**
    - **-1: power off**



- Getting a frequency and voltage state of a module
  - `#pragma oscar get_fvstatus (C)`
  - `!$oscar get_fvstatus (Fortran)`

# Low-Power Optimization with OSCAR API

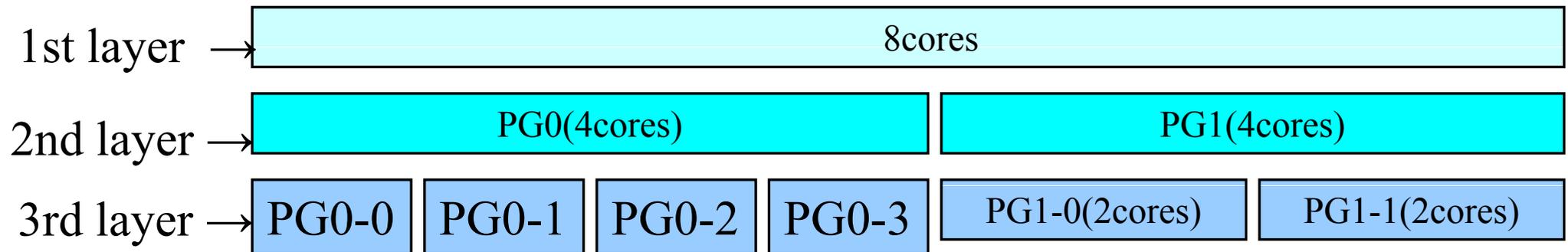


# Timer

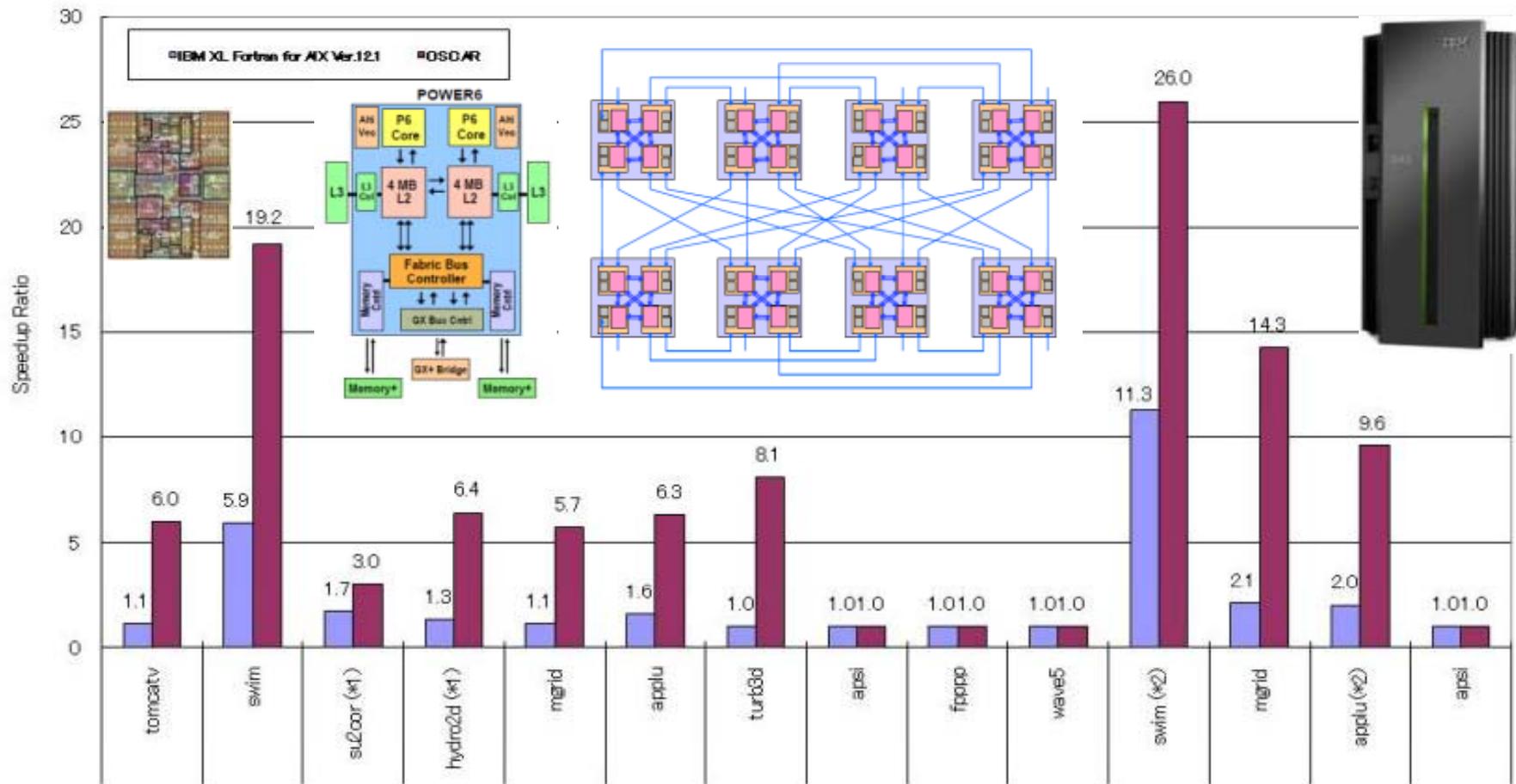
- **Getting an elapsed wall clock time in microseconds**
  - **#pragma oscar get\_current\_time (C)**
  - **!\$oscar get\_current\_time (Fortran)**
- **For realtime execution**

# Hierarchical Barrier Synchronization

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



# Performance of OSCAR Compiler on IBM p6 595 Power6 (4.2GHz) based 32-core SMP Server



OpenMP codes generated by OSCAR compiler accelerate IBM XL Fortran for AIX Ver.12.1 about **3.3 times** on the average

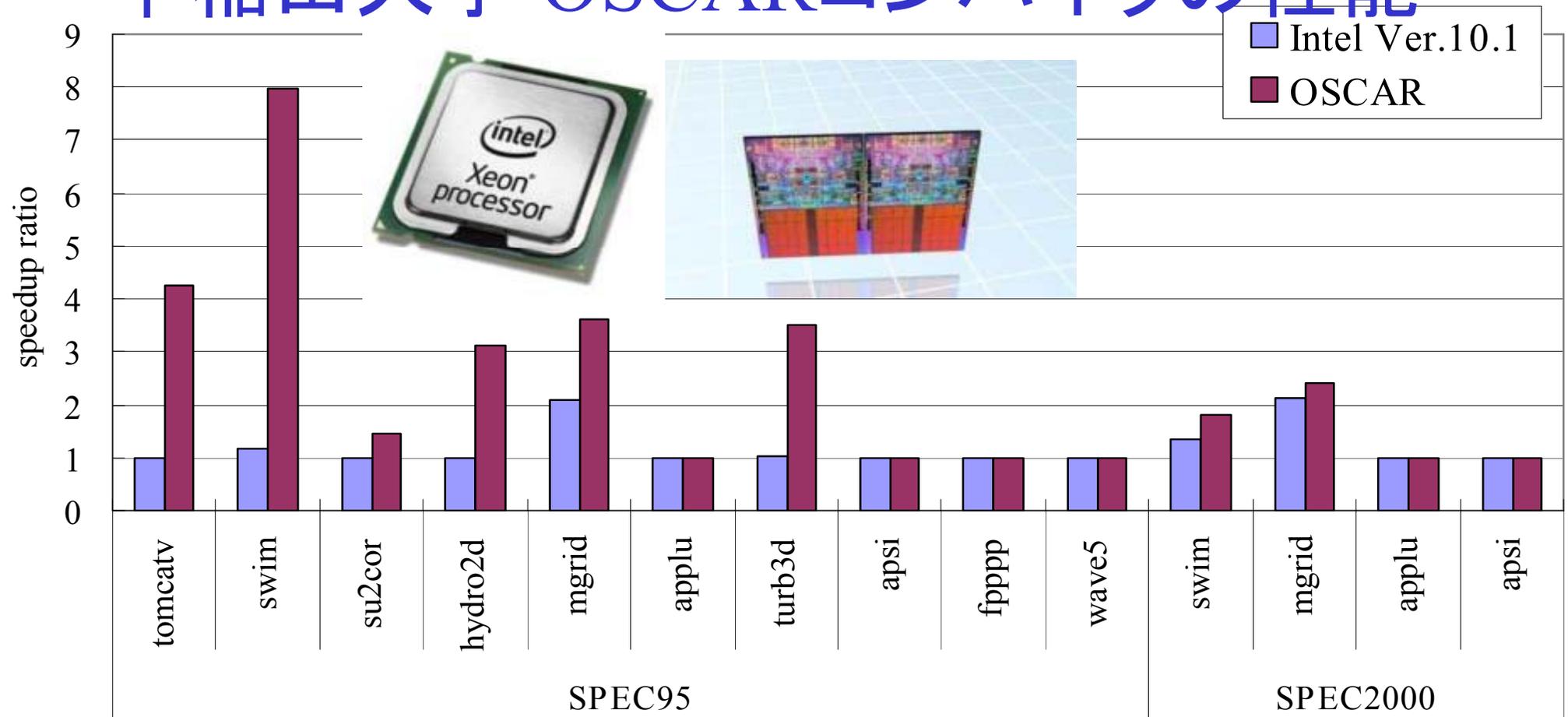
Compile Option:

(\*1) Sequential: -O3 -qarch=pwr6, XLF: -O3 -qarch=pwr6 -qsmp=auto, OSCAR: -O3 -qarch=pwr6 -qsmp=noauto

(\*2) Sequential: -O5 -q64 -qarch=pwr6, XLF: -O5 -q64 -qarch=pwr6 -qsmp=auto, OSCAR: -O5 -q64 -qarch=pwr6 -qsmp=noauto

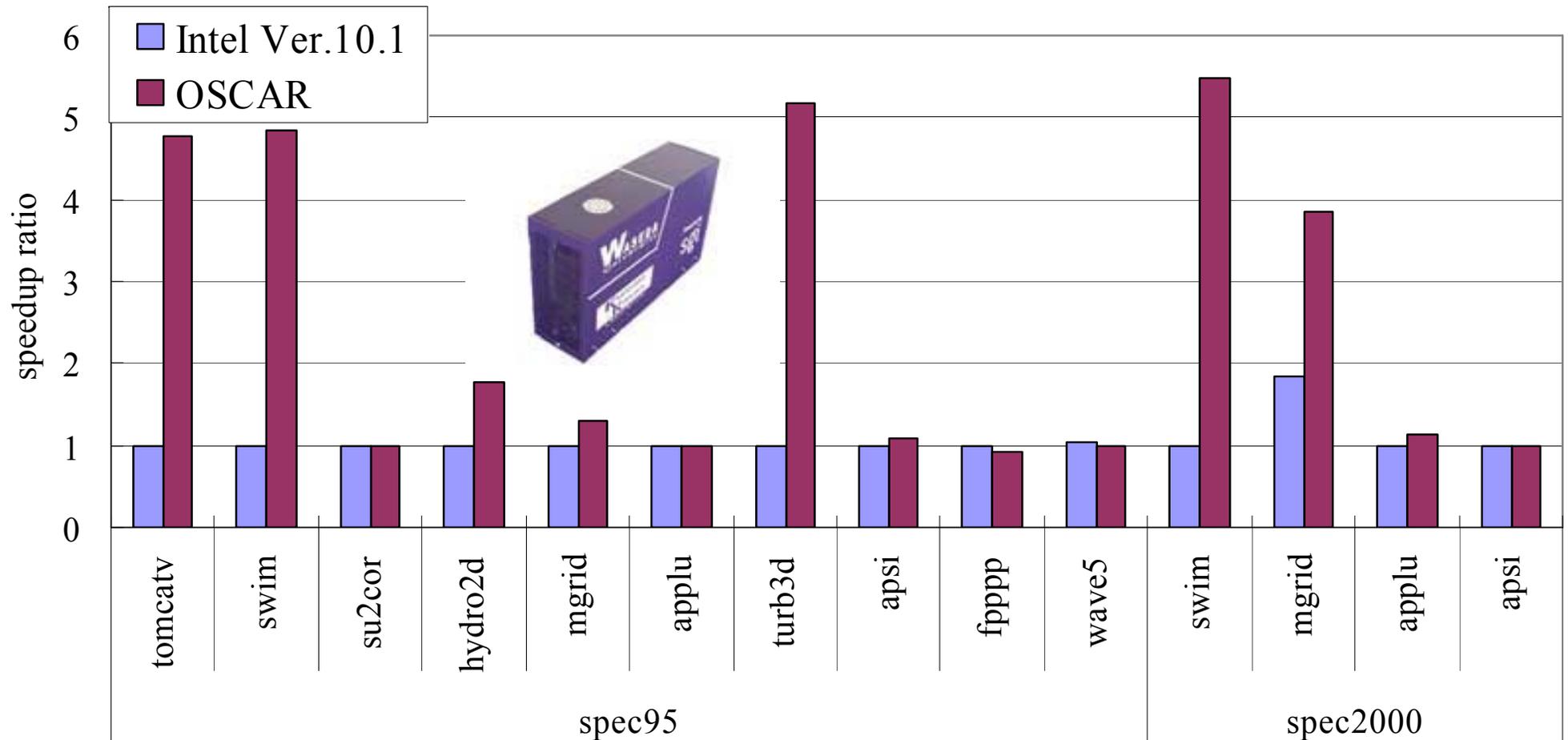
(Others) Sequential: -O5 -qarch=pwr6, XLF: -O5 -qarch=pwr6 -qsmp=auto, OSCAR: -O5 -qarch=pwr6 -qsmp=noauto

# インテル クアッドコア Xeon プロセッサ上での 早稲田大学 OSCAR コンパイラの性能



OSCAR コンパイラが生成するOpenMP並列化コードはインテルコンパイラ ver.10.1の性能をインテル・マルチコア上で2.09 倍向上

# インテルMontvaleマルチコアを用いた 16 コアSGI Altix 450 デスクサイドサーバ上での OSCAR コンパイラの性能

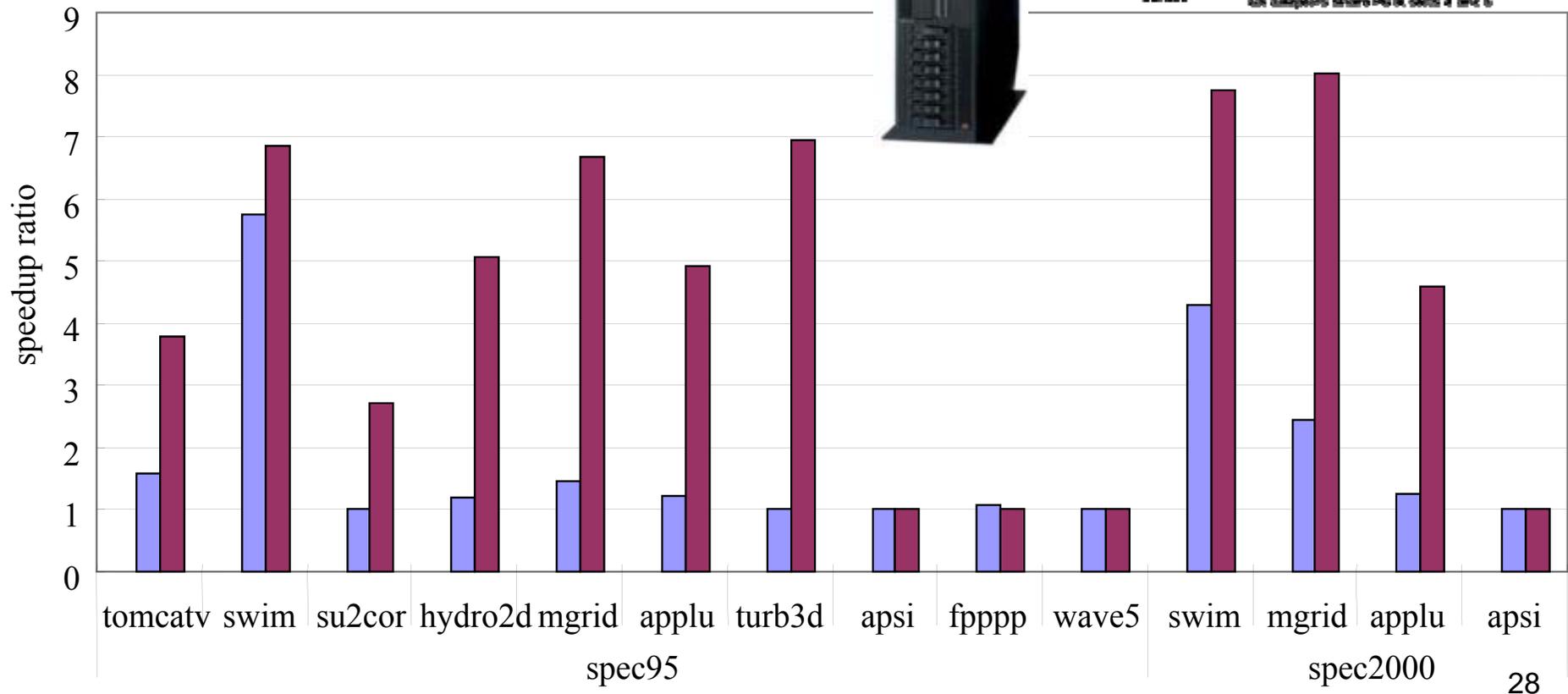
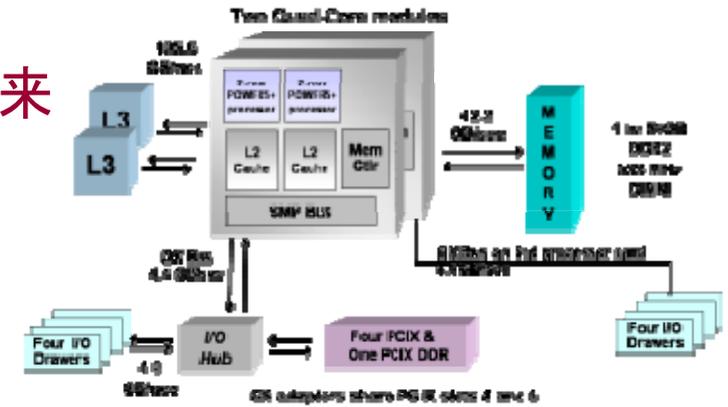


OSCAR コンパイラが生成するOpenMPコードはインテル Itanium  
Compiler revision 10.1の性能を 2.32 倍向上

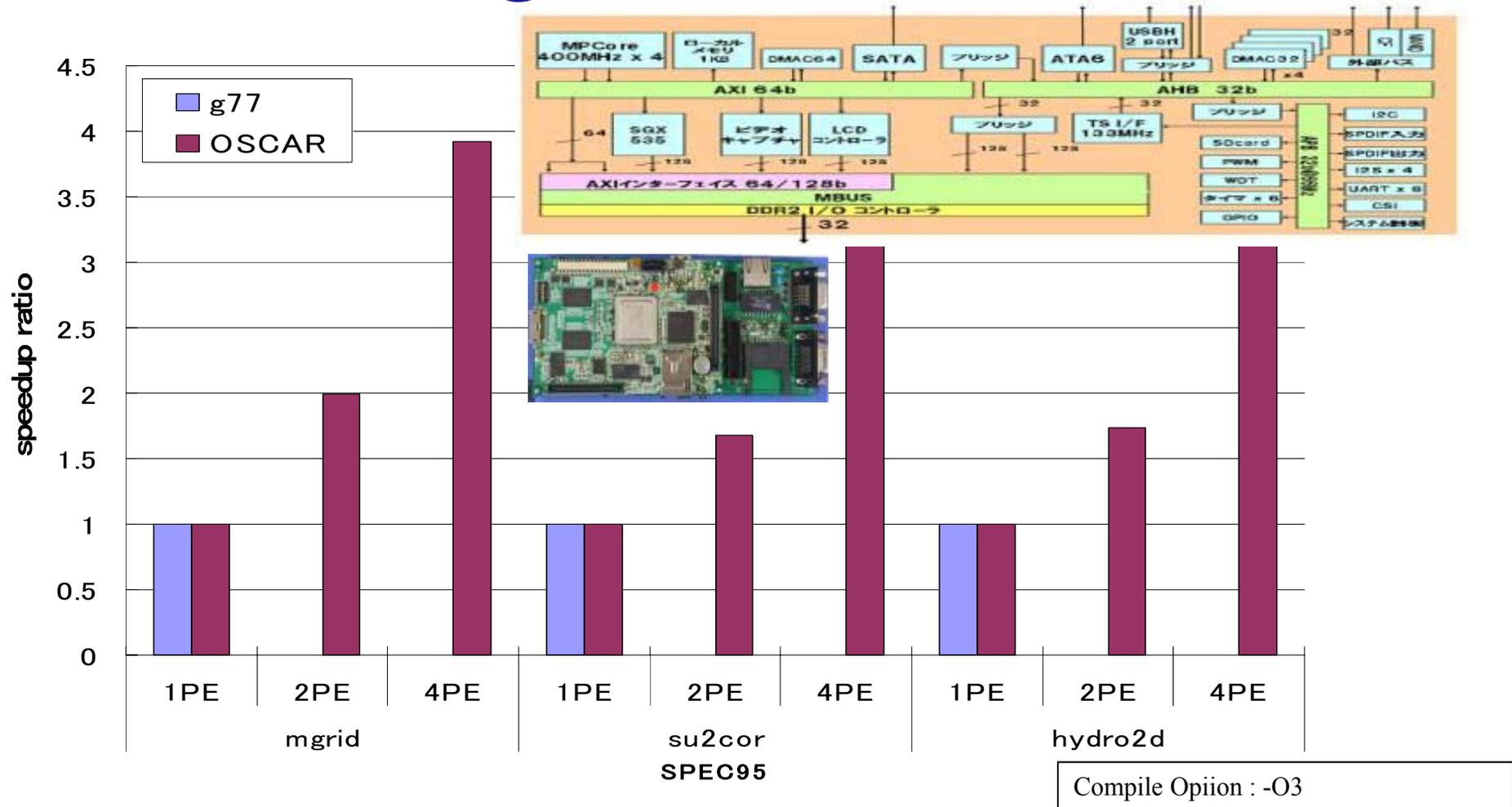
# IBMサーバp550q上でのOSCAR並列化コンパイラの性能

8プロセッサコア・マルチプロセッサシステム上で従来の  
ループ並列化に対し、2.7倍の高速化

- ループ並列化
- マルチグレイン並列化

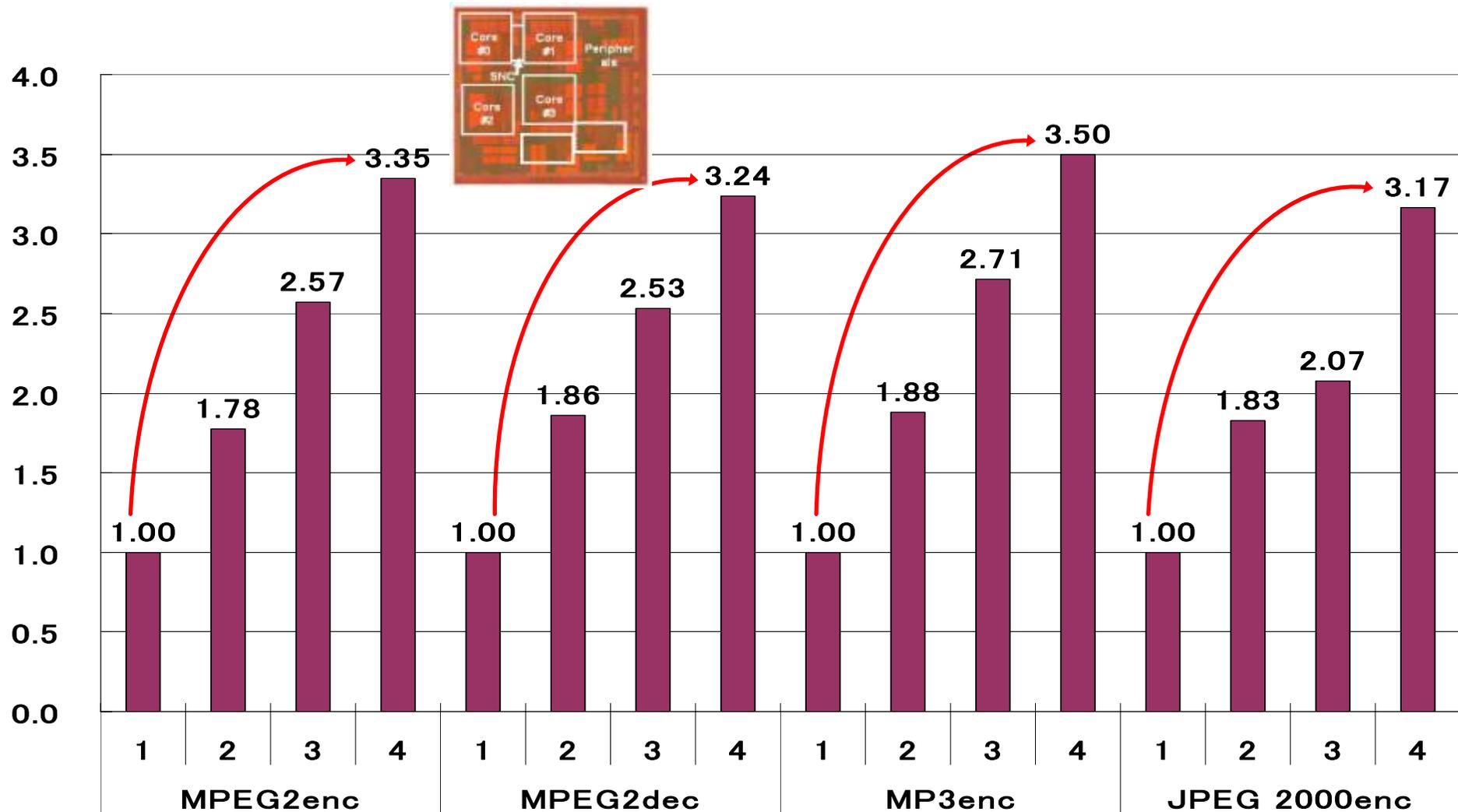


# Performance of OSCAR compiler on NEC NaviEngine(ARM-NEC MPcore)



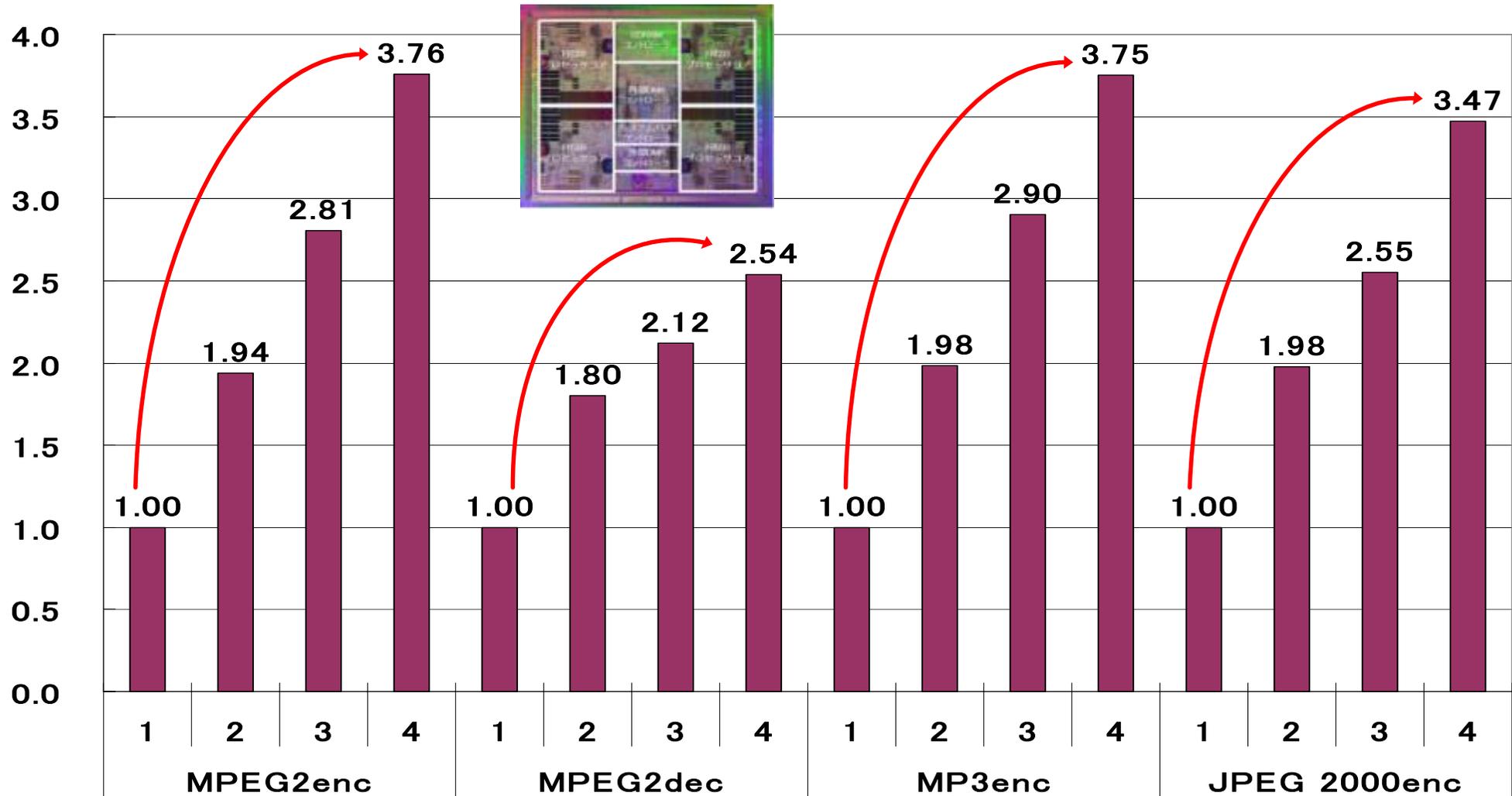
- OSCAR compiler gave us 3.43 times speedup against 1 core on ARM/NEC MPCore with 4 ARM 400MHz cores

# Performance of OSCAR Compiler Using the Developed API on 4 core (SH4A) OSCAR Type Multicore



3.31 times speedup on the average for 4cores against 1core

# Performance of OSCAR Compiler Using the multicore API on Fujitsu FR1000 Multicore



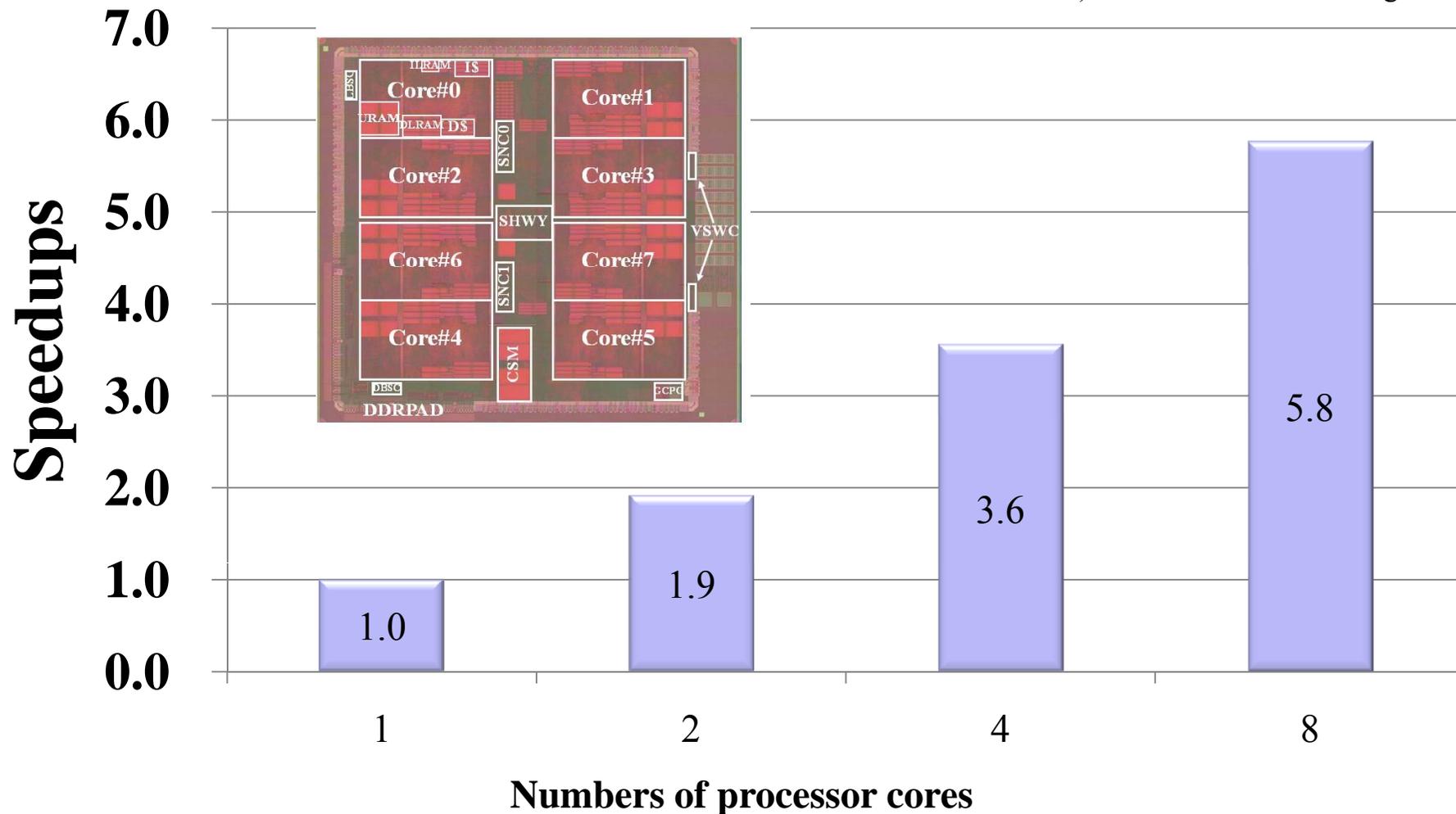
3.38 times speedup on the average for 4 cores against a single core execution

# Processing Performance on the Developed Multicore Using Automatic Parallelizing Compiler

Speedup against single core execution for audio AAC encoding

32

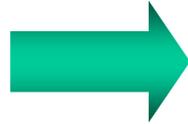
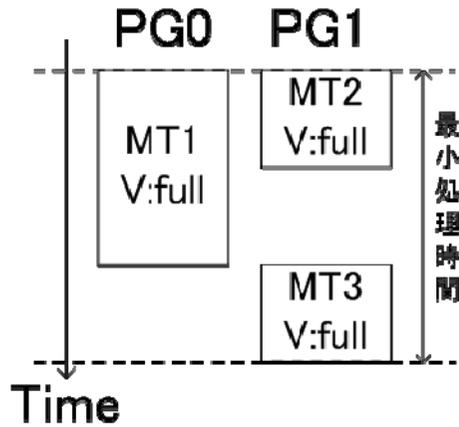
\*) Advanced Audio Coding



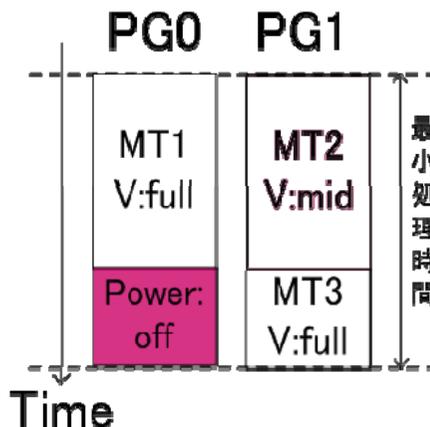
# 周波数電圧 (FV) 制御と電源制御による低消費電力化

## 処理ユニット負荷不均衡時の電源・周波数電圧制御

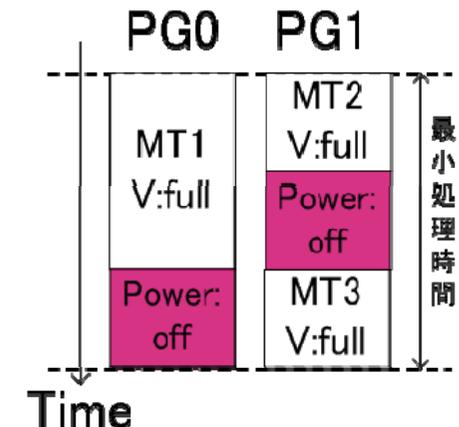
スケジューリング結果



FV制御

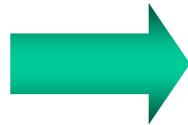
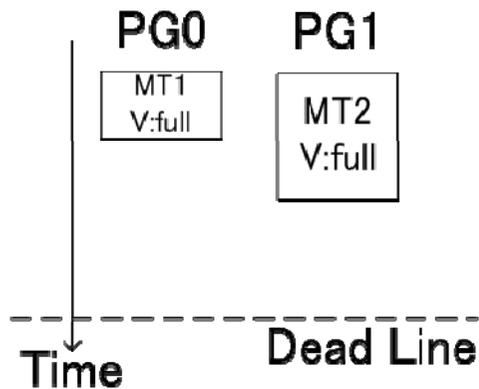


電源制御

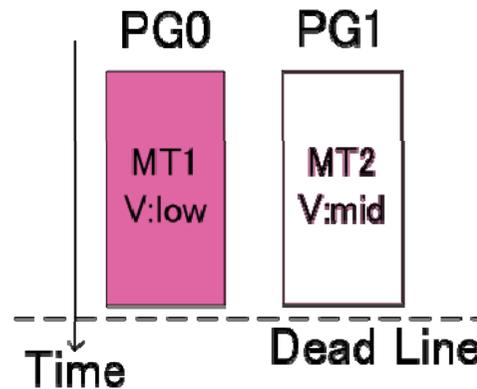


## デッドライン制約を考慮した電源・周波数電圧制御

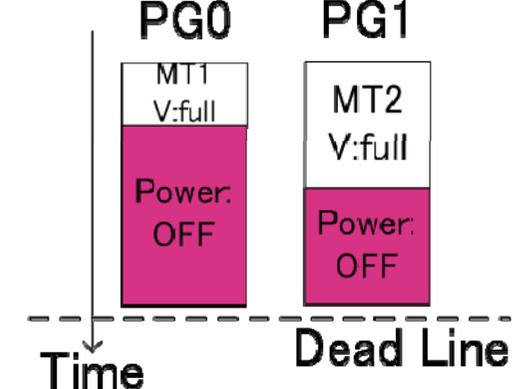
スケジューリング結果



FV制御

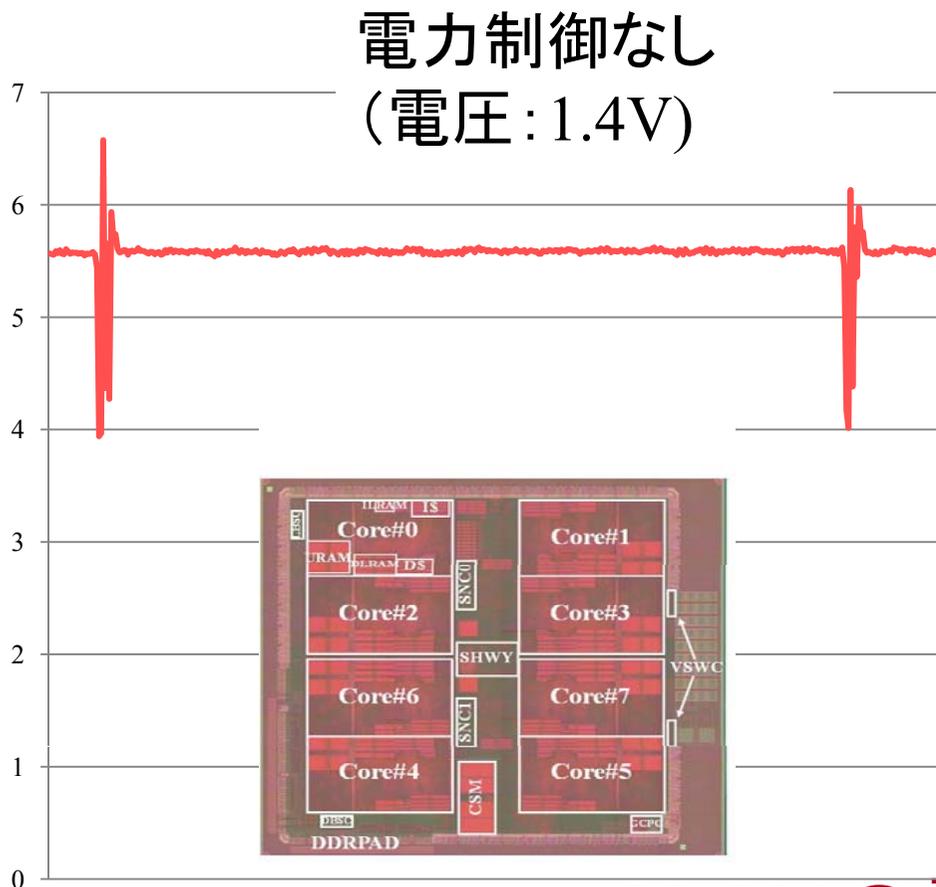


電源制御



# 音楽圧縮におけるコンパイラ周波数/電圧・電源制御 による開発チップ上での消費電力削減効果

セキュアオーディオ圧縮(AACエンコード+AES暗号化)処理を8コアで実行時の消費電力



周波数/電圧・電源制御あり  
(電圧:1.0V~1.4V、  
レジューム電源遮断モード使用)



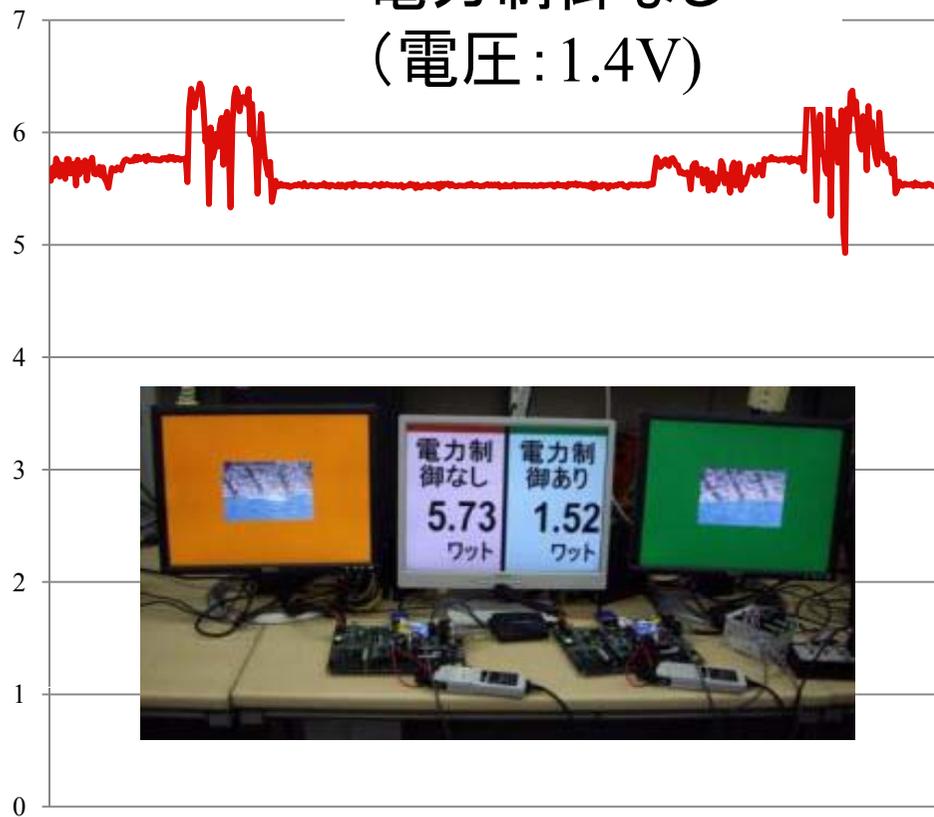
88.3%の電力削減



# 画像表示におけるコンパイラ周波数/電圧・電源制御 による開発チップ上での消費電力削減効果

動画表示(MPEG2デコード処理)を8コアで実行時の消費電力

電力制御なし  
(電圧:1.4V)



平均電力  
5.73 [W]

周波数/電圧・電源制御あり  
(電圧:1.0V~1.4V、  
レジューム電源遮断モード  
使用)



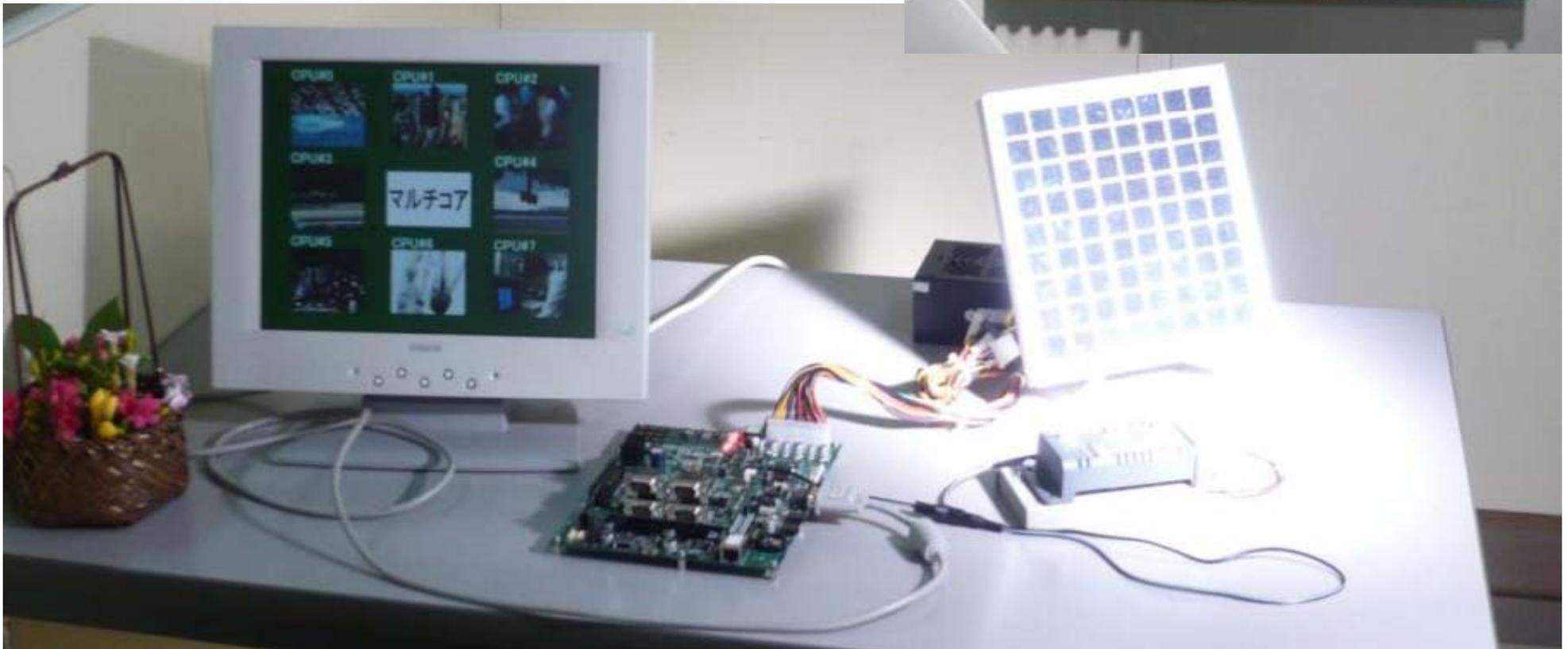
平均電力  
1.52 [W]

73.5%の電力削減



# ソーラパネル駆動可能な 低消費電力高性能マルチコア RP2(SH4A 8コア集積)

- クリーンエネルギーで駆動可
  - 電力供給が困難な場所での使用可能
  - 災害時でも使用可能



# マルチコアからメニーコアへ

## ■ 組込からスパコンまで高性能・低消費電力化

### ➤ 情報家電用マルチコア

カーナビ, 携帯電話, ゲーム, デジタルTV, DVD

IBM/ Sony/ Toshiba Cell, Fujitsu FR1000, P  
anasonic Uniphier, NEC/ARM MPCore/MP211/NaviEngine  
Renesas 4 core RP1, 8 core RP2, 15core Hetero RP-X

**Tilera Tile64, SPI Storm-1(16 VLIW cores),  
Plurality HAL (64-256 cores)**

### ➤ PC,サーバ

**Intel** Quad Xeon, Core 2 Quad, Nehalem(8core), 80 core,

**Larrabee(32core), SCC (48 core)**

AMD Quad Core Opteron, Phenom

### ➤ WSs, Deskside & Highend Servers

**IBM** Power 7(8cores): BlueWaters(HPCS)16 PFLOP(2011)

**Cycrops64 (160 cores),** Sun Rock (16 cores), Rainbow  
Falls (16 cores), Fujitsu SPARC64 VIII fx (8 cores)

### ➤ スーパーコンピュータ

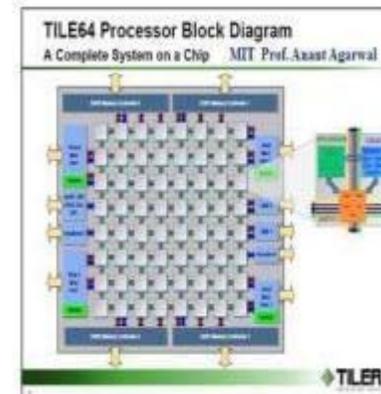
IBM 低消費電力マルチコアベースBG/P PowerPC450 (4 cores), BG/Q (8-  
16cores) 20PFLOPS, 6MW (2011-12)

## ■ 低消費電力,アプリケーションソフトの充実,短期間 システム開発,低コスト,高機能化が市場競争力決定

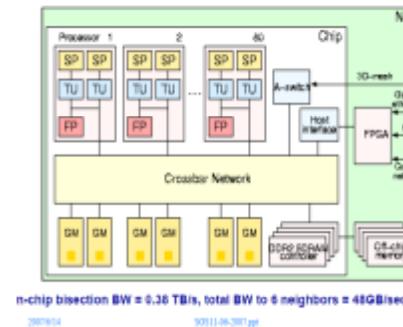
<例>携帯電話,ゲーム,自動車,サーバ

## ■ 自動並列化コンパイラ協調型メニーコアプロセッサ必要

ルネサス, 日立, 東工大  
早大 RP-X



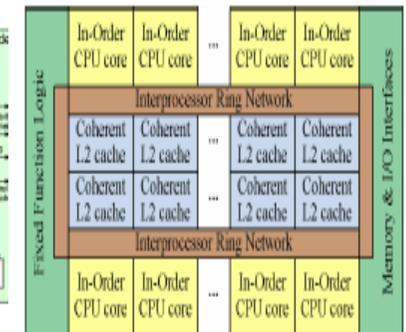
C-64 Chip Architecture



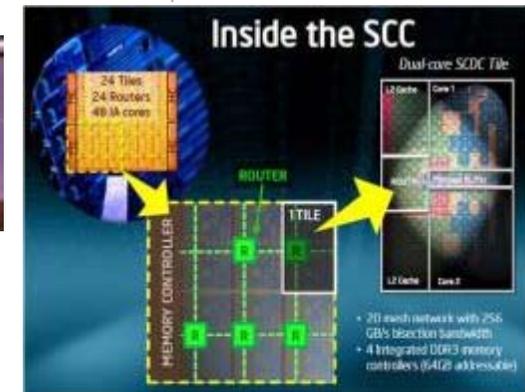
On-chip bisection BW = 0.26 TB/s, total BW to 6 neighbors = 48GB/sec

CAPSL

Intel Larrabee



IBM  
Power7  
1TFLOPS  
Module



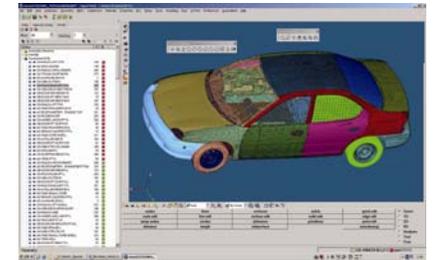
# 低消費電力メニーコアプロセッサ・システム

## <目的>

- CO<sub>2</sub>排出量削減による低炭素社会実現への貢献
- 低消費電力高性能メニーコアプロセッサの導入による高度IT機器の高付加価値化による産業競争力の強化

## <内容>

- 下記項目に関して世界をリードするための目標性能、費用・期間を考慮した実現可能性の検証を行う。
- 太陽電池駆動・自然空冷(ファン無し)  
超低消費電力64-128コア集積メニーコアプロセッサ
  - メニーコアをベースとしたシステム:サーバ・情報家電
    - クールサーバ(静か・ホコリ無し・コンパクト)  
医療現場で使い易い画像処理用サーバ、  
データセンター電力大幅削減可
  - 自動並列化コンパイラ
    - 電力・ソフトウェア開発期間の大幅削減:数分で自動並列化:高い国際競争力
  - 異なる企業開発のマルチコア間でのアプリケーションの共用
    - <OSCAR API: Application Programming Interface並列化コンパイラで各社メニーコアを使用可能に>



RADIOSS (衝撃解析ソルバ、線形ソルバ)  
OptiStruct (構造最適化ソルバ、軽量化ツール)



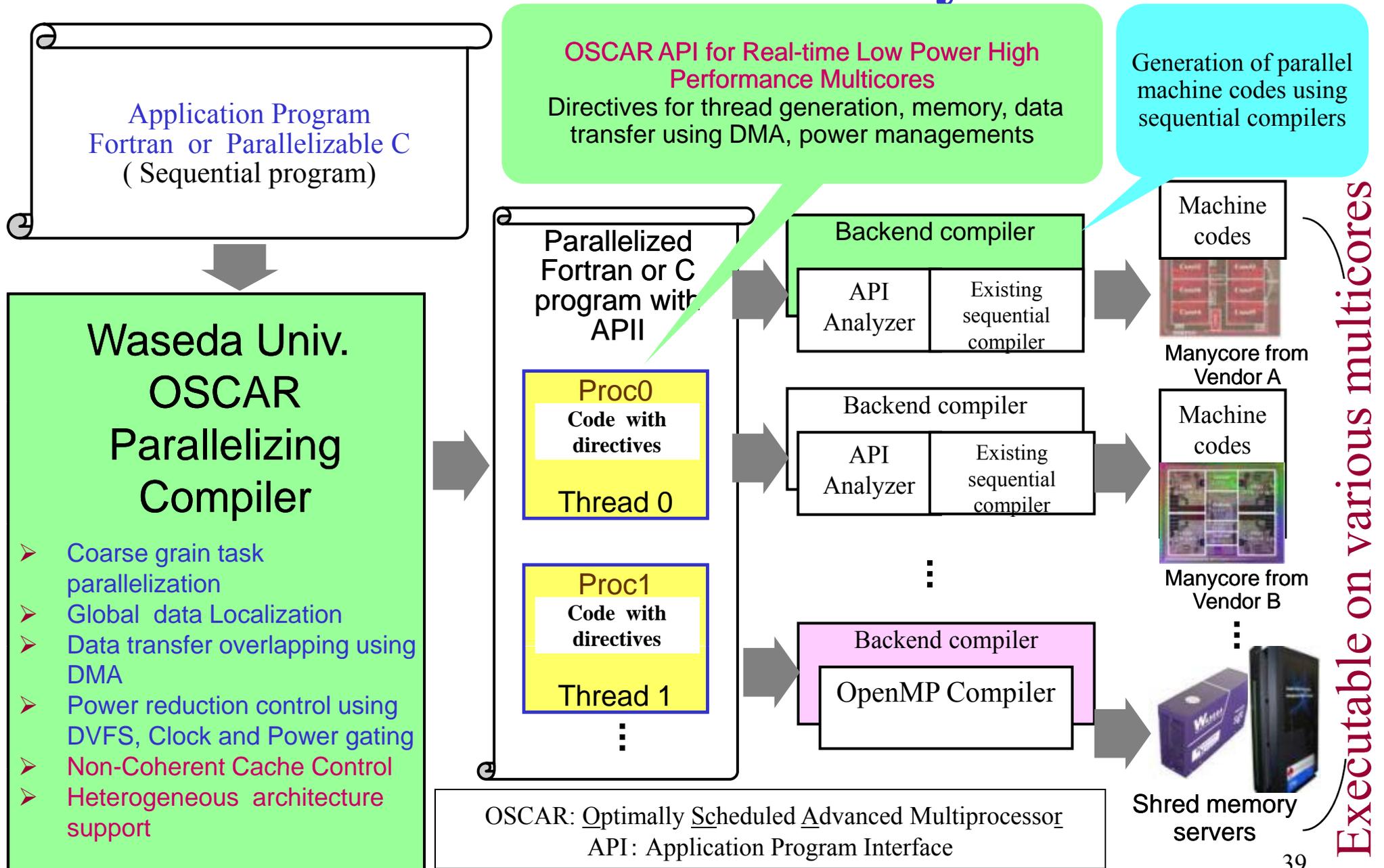
スーパーコンピュータ

Altair PBS  
Green  
Provisioning  
との連携

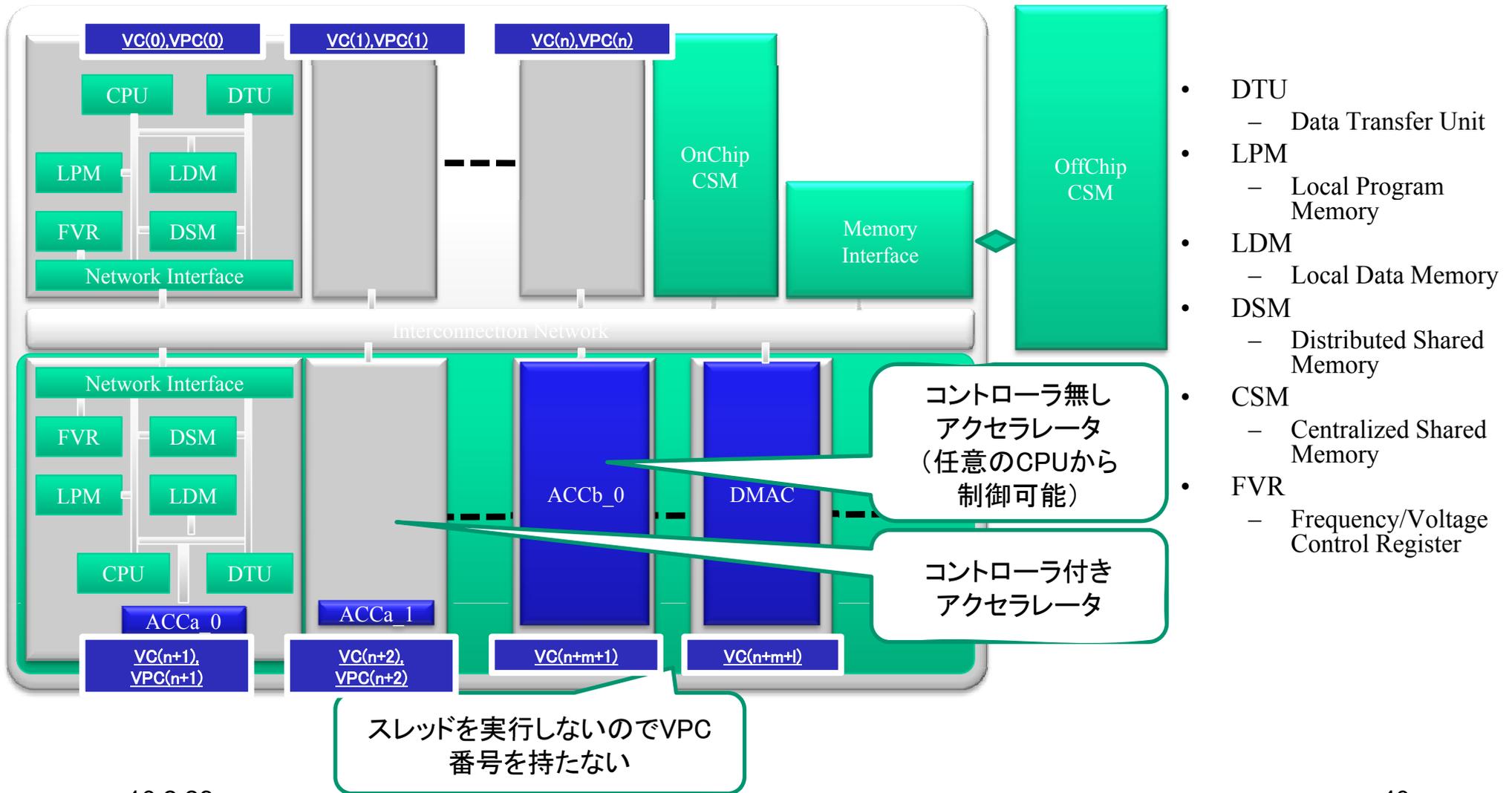
日立,富士通,ルネサス,東芝, NEC

- 次世代オンチップネットワークアーキテクチャ及び標準的なインターフェイス

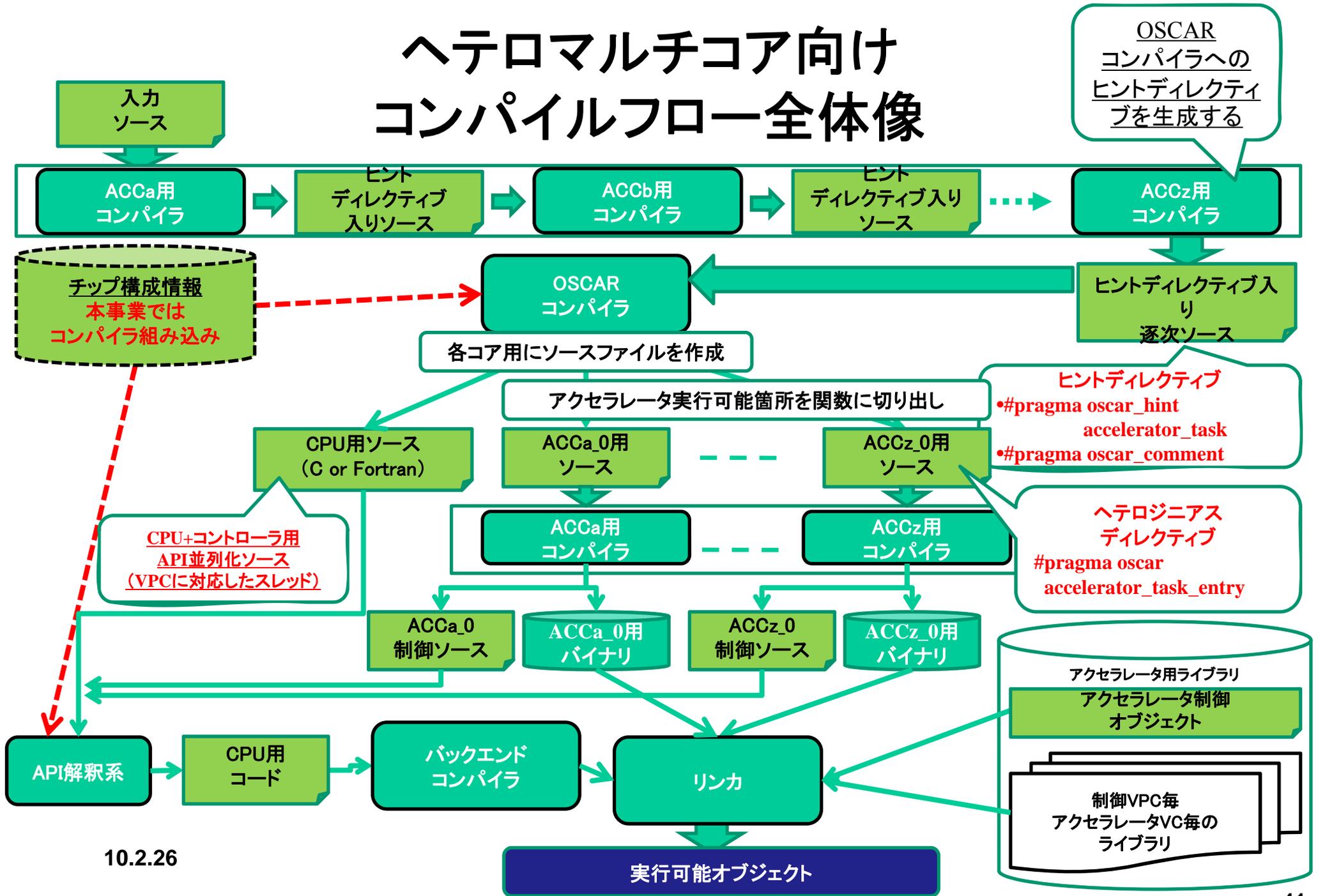
# OSCAR API for Manycores



# OSCAR API-Applicable ヘテロジニアスマルチコアアーキテクチャ



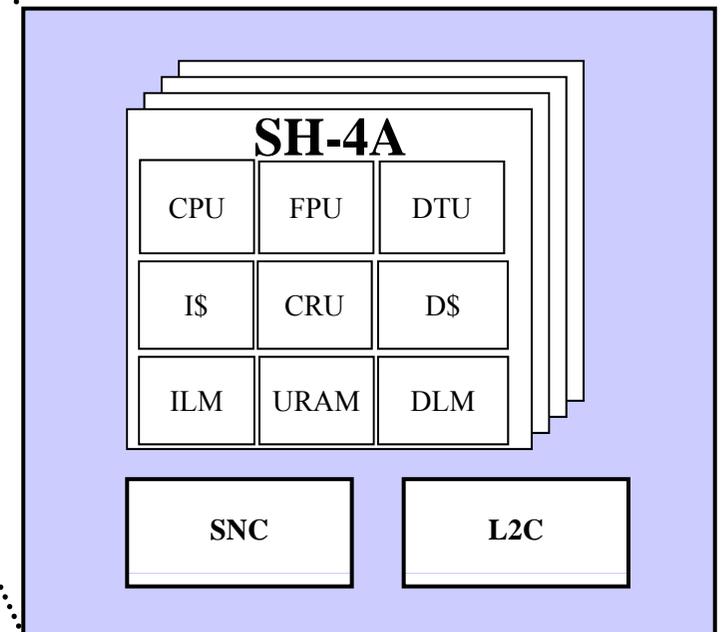
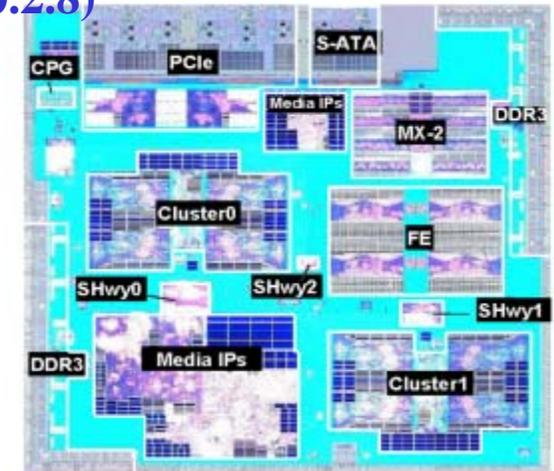
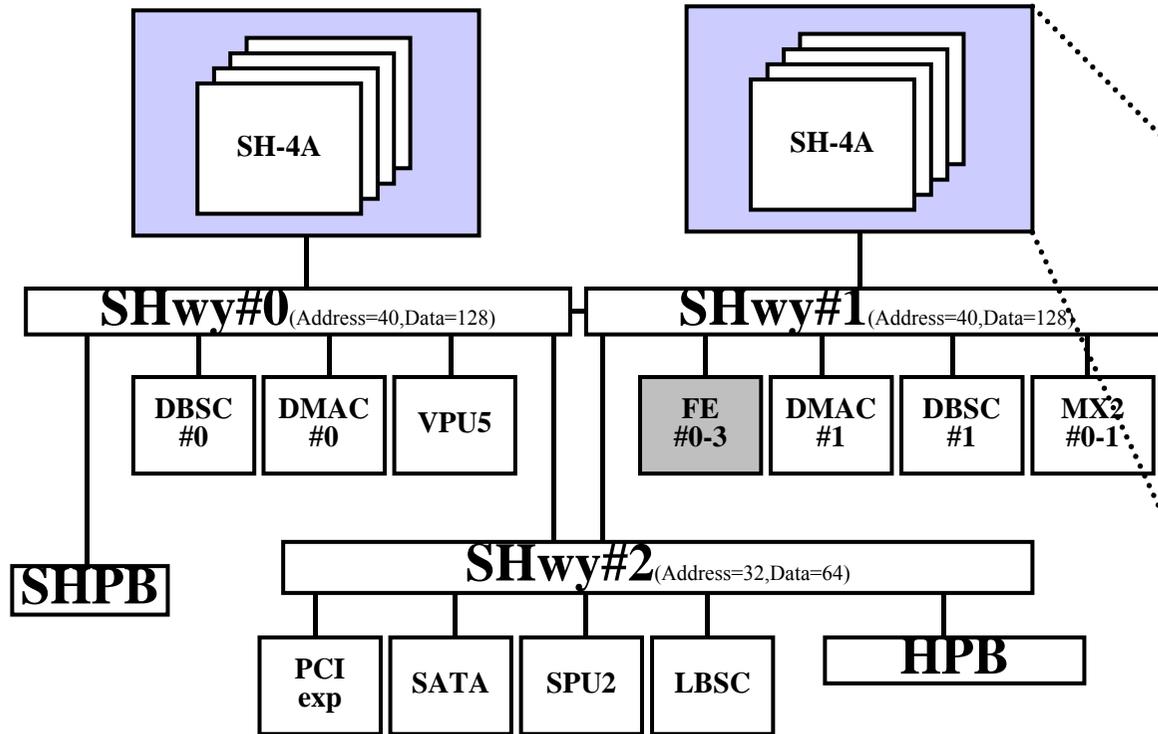
# ヘテロマルチコア向け コンパイルフロー全体像



10.2.26

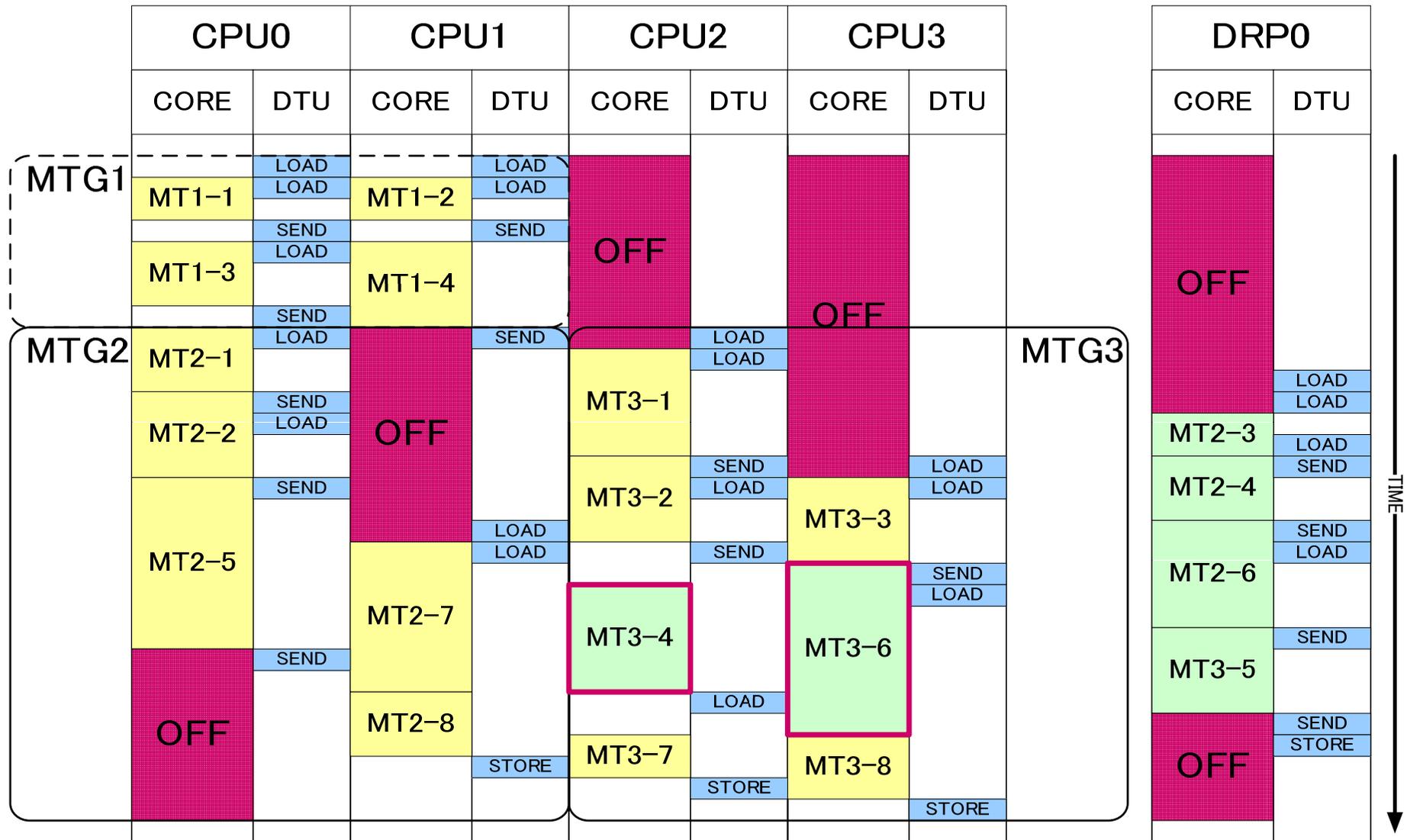
# 新規開発ヘテロジニアスマルチコアRP-X

ISSCC2010 Processorセッションにて発表(2010.2.8)

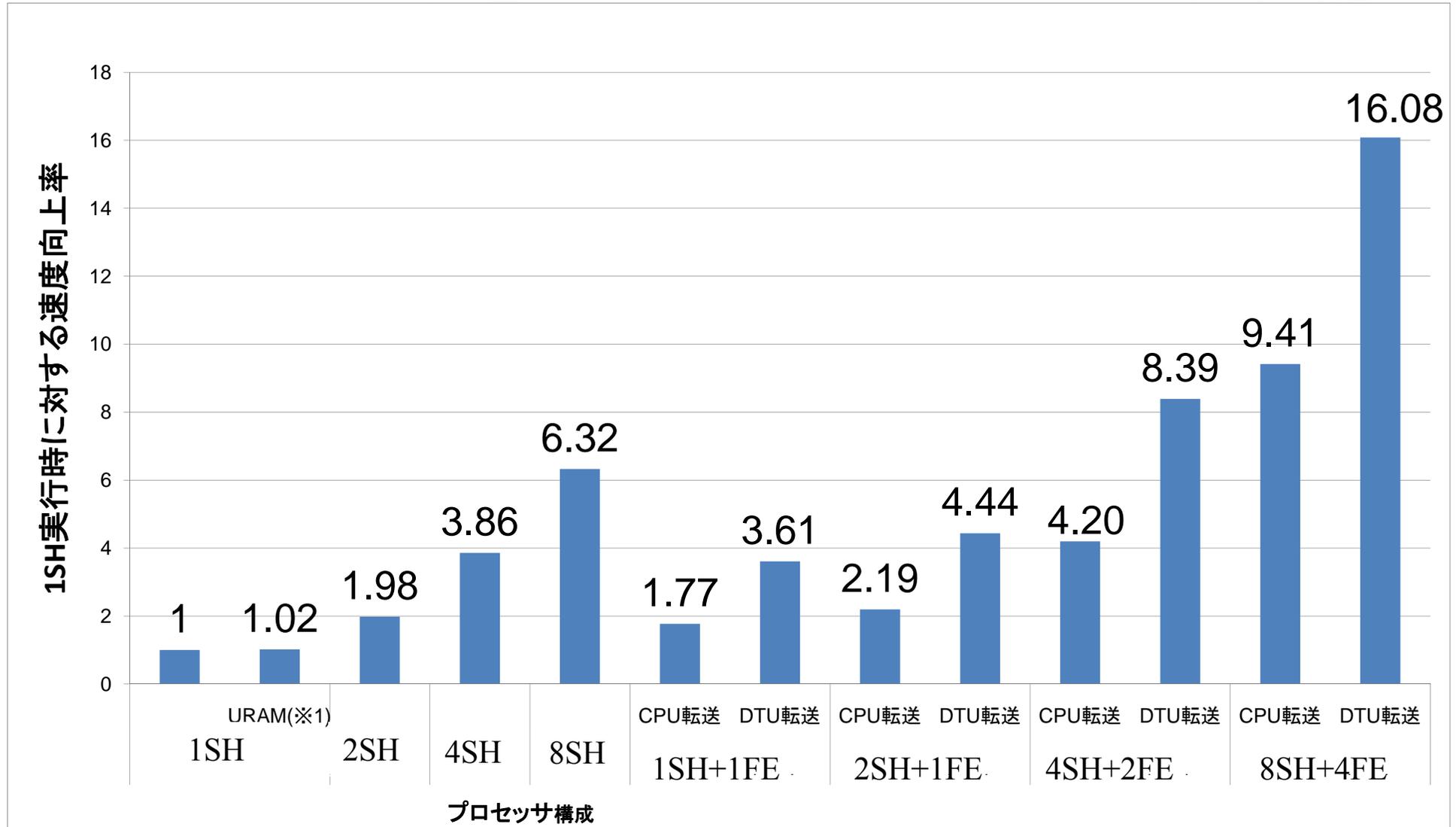


ルネサステクノロジ・日立・東工大・早稲田により開発

# ヘテロジニアスマルチコア用スケジューリングイメージ



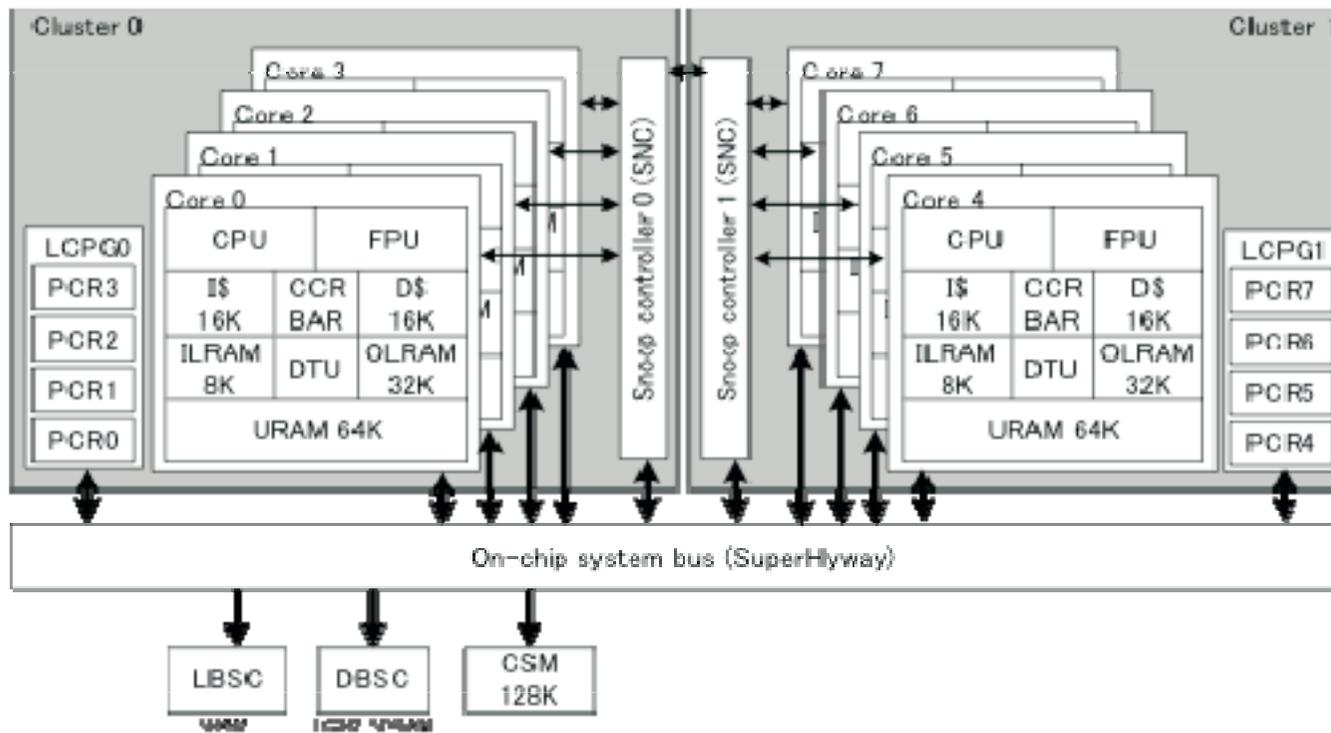
# RP-XでのOSCAR コンパイラ/APIを用いた AACエンコーダを用いたヘテロジニアス並列性能



# ノンコヒーレントキャッシュ向けの OSCAR API指示文

- 新たに5つの指示文を追加
- メモリ配置指示文
  - noncacheable: 変数をノンキャッシュャブルにする
  - aligncache : 変数の先頭をキャッシュラインの境界にアラインメント
- キャッシュ操作指示文
  - cache\_writeback : キャッシュ上のダーティラインの書き戻し
  - cache\_selfinvalidate : キャッシュラインの無効化
- メモリ操作順序保証指示文
  - complete\_memop: メモリ操作の完了

# リアルタイム情報家電用低消費電力マルチコア RP2 (8コア) Renesas/Hitachi/Waseda

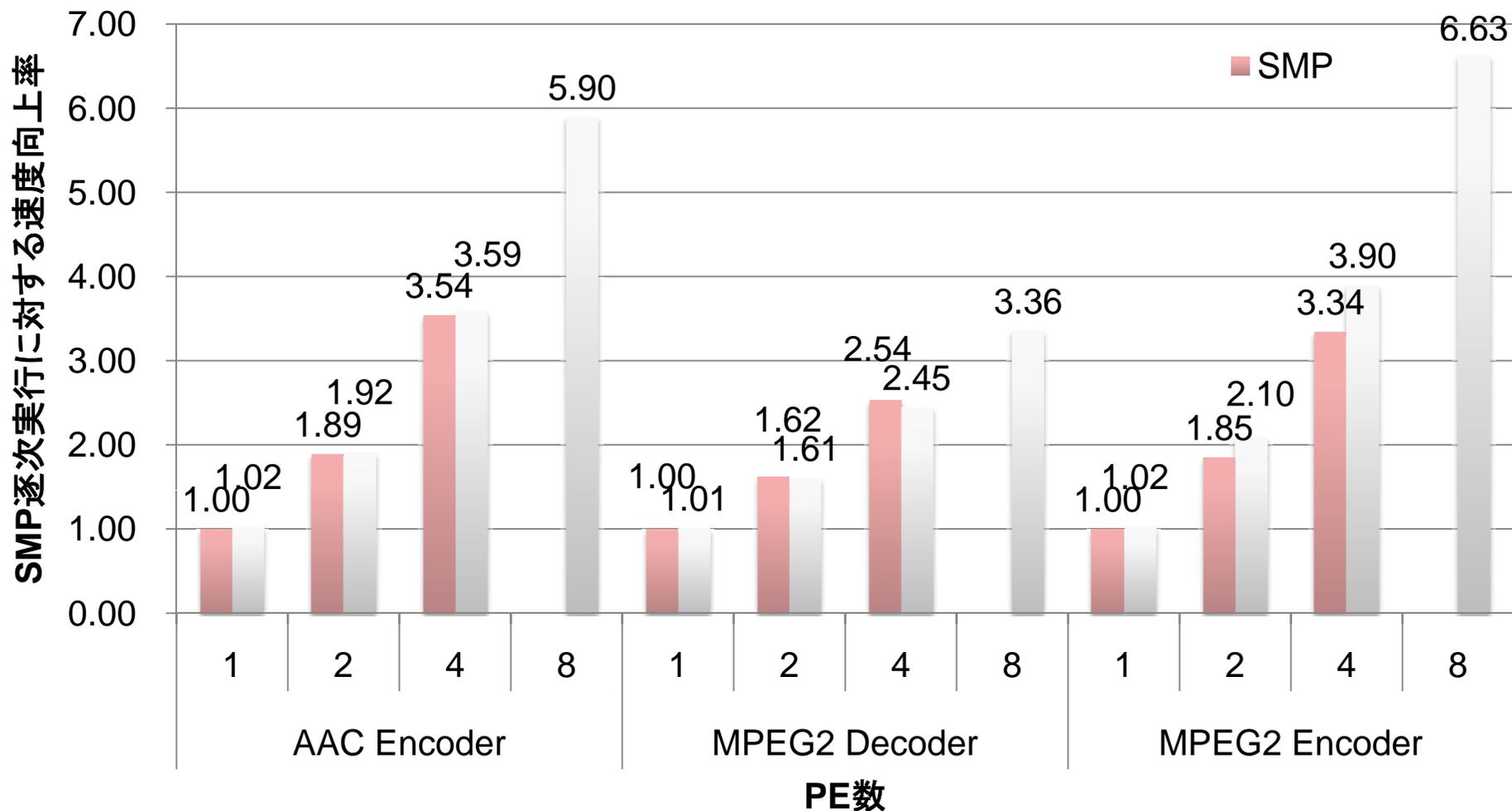


Process Technology	90nm, 8-layer, triple-Vth, CMOS
Chip Size	104.8mm <sup>2</sup> (10.61mm x 9.88mm)
CPU Core Size	6.6mm <sup>2</sup> (3.36mm x 1.96mm)
Supply Voltage	1.0V–1.4V (internal), 1.8/3.3V (I/O)
Clock frequency	600MHz, 300MHz, 150MHz, 75MHz
Power Domains	17 (8 CPUs, 8 URAMs, common)

M. Ito, et al., "An 8640 MIPS SoC with Independent Power-off Control of 8 CPU and 8 RAMS by an Automatic Parallelizing Compiler", ISSCC2008

クラスタ間ではハードウェアはコヒーレンスを維持しない  
 → コンパイラでソフトウェアコヒーレンス制御

# RP2上でのコンパイラによるソフトウェアコピー レンシ制御手法及びメニーコア用APIの検討



# グリーン・コンピューティング・システム研究開発センター 概要

## <目標>

太陽電池で駆動可能で  
冷却ファンが不要な

超低消費電力・高性能

メニーコアプロセッサ\*のハードウェア、  
ソフトウェア、応用技術の研究開発

\*1チップ上に多数のプロセッサコアを集積する  
次世代マルチコアプロセッサ



## <産学連携>

富士通, 日立, ルネサス,  
東芝, NEC 等

## <波及効果>

超低消費電力メニーコア

➤ CO<sub>2</sub>排出量削減

➤ サーバ国際競争力強化

➤ 我が国の産業利益を支える

情報家電, 自動車の高付加価値化





# まとめ

- 世界初のコンパイラ協調型 高性能・低消費電力・リアルタイム情報家電向け4コア(RP1),8CPUコア(RP2), 15コアヘテロ(RPX)マルチコアを試作
- 8コアRP2チップでは、8個のプロセッサとメモリの独立電源遮断・動作周波数制御(1/2,1/4,1/8,0)・電圧制御(1.4V,1.2V,1.0V)可能。
  - 総合科学技術会議:8画面までの動画像を表示し、標準半導体と比べ
    - 8画面表示時 1/14に電力削減 マルチコア: 2.5W,標準半導体:35W
    - 1画面表示時1/50に電力削減 マルチコア: 0.5W, 標準半導体:25W
- 世界最高処理性能かつ世界初の電力制御を実現したOSCAR自動並列化コンパイラの開発に成功(動的電力、今後の微細化で問題となるリーク電力も削減)
  - 自動並列化によりAACエンコーダを8プロセッサで5.8倍の高速化
  - IBM(Power6), Intel(SGI Altix450 最新Itanium2 Montvale16コア, Quad-core Xeon)上で各社コンパイラの性能を2倍以上向上
  - 世界初の電力制御に成功し、オーディオAACエンコードで88%、動画像MPEG2エンコードで74%の電力削減:太陽電池にて駆動可能
- 従来の手動並列化では数週間単位の時間を要した並列アプリケーション作成を、各社のマルチコア用にコンパイラによりAPIを用いて数秒単位で作成
- 今後:情報家電・自動車統合制御系・クールサーバ(グリーンIT)・スパコン