

Green Multicore by Codesign of Compiler and Architecture



Prof. Hironori Kasahara, IEEE Life-Fellow, IPSJ Fellow
 Senior Executive Vice President (2018-2022), Waseda Univ.
 IEEE Computer Society President 2018
 ACM/IEEE ISCA'25 General Co-Chair
 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 Executive Vice President, Waseda U. (2018-22)

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,, 2023 IEEE Life Fellow

Reviewed Papers: 243, Invited Talks: 246,
 Granted Patents: 72 (Japan, US, GB, DE, China),
 Articles in News Papers, Web News, TV etc.: 713

Committees in Societies and Government 300
 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
 Development Member (Top 1:NWT, Earth Simulator, K)
 JST SPRING Ph.D Promotion Chair, Boost AI Ph.D.
 Boosting Chair, SBIR Phase 1 Chair, Moonshot Project
 G3 Robot & AI Advisor & Int'l Symp Vice Chair,
 【COCN】 Council of Competitiveness Nippon Ex-Board

ACM/IEEE ISCA'25, DECC Panel, June 21, 2025, Tokyo, Japan

Research on OSCAR Parallelizing Compiler & Co-designed Hardware Since 1984

72 international patents in USA, UK, China, Japan to improve effective
 performance, cost-performance, software productivity and power efficiency

High Performance & Low Power

1) Multigrain Parallelization for Embedded to HPC Homogeneous and Heterogeneous Multicores

Coarse-grain task parallelism among loops,
 subroutines & basic blocks in addition to
 the loop parallelism

2) Data Localization: Optimization of Cache & Local Memory Usage

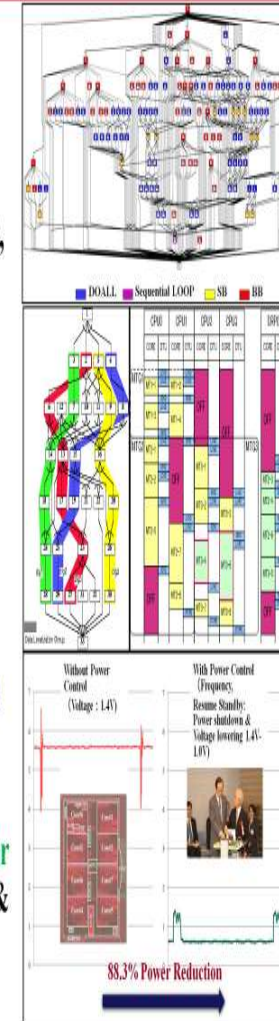
- Automatic data decomposition and data
 reuse control for Distributed shared memory,
 Cache and Local memory
- Data Transfer Control

Overlapping Data Transfer using Data
 Transfer Unit, or DMAC

3) Automatic Power Reduction

- OSCAR Compiler can reduce power
 consumption by using DVFS and Clock- &
 Power-gating with hardware supports.

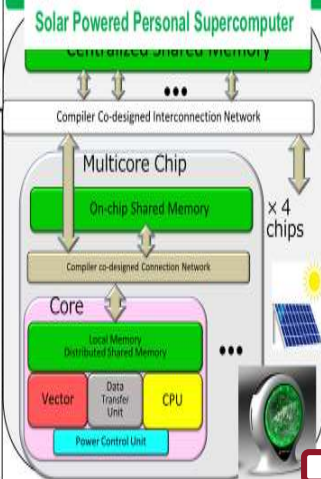
4) Codesigned Accelerator: See right figure



Green Accelerator

can be attached to any processor
 cores, RISCv, arm, Infineon
 Renesas, AMD, Intel, etc, without
 instruction extensions.

It works with automatic local
 memory management and
 power reduction control by
 OSCAR Automatic vectorizing
 & parallelizing compiler



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>

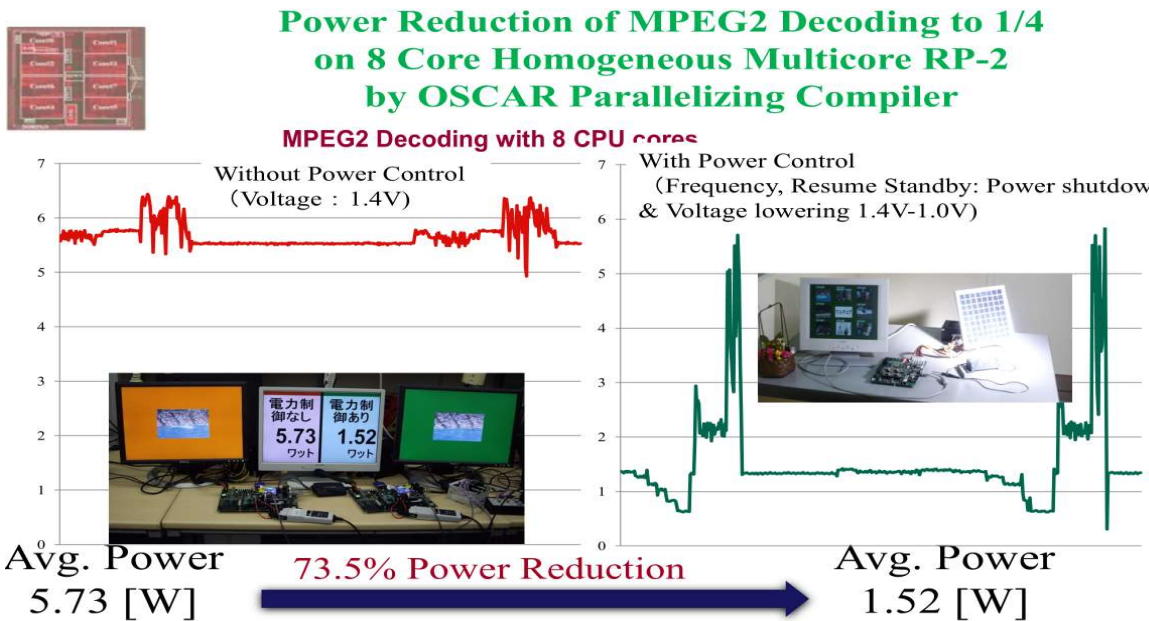


Codesign of Compiler and Multiprocessor Architecture since 1985

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), 1.803.3V (I/O)	1.0-1.4V (internal), 1.803.3V (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	640MHz, 13.7GIPS, 115GFOPS, 36.2GFLOPS
11.4 GOPS/W (32n換算)	18.3 GOPS/W (32n換算)	37.3 GOPS/W (32n換算)

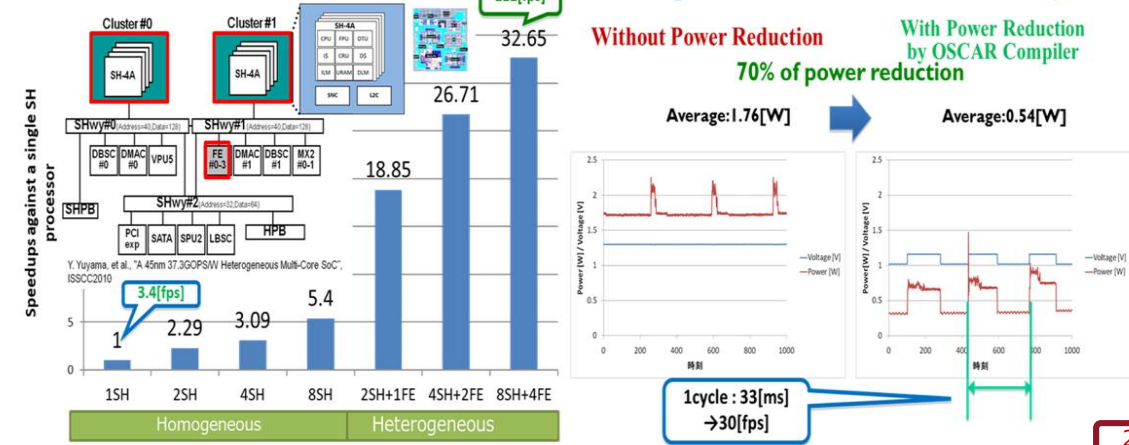
Prime Minister FUKUDA is touching our multicore chip during execution.



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)

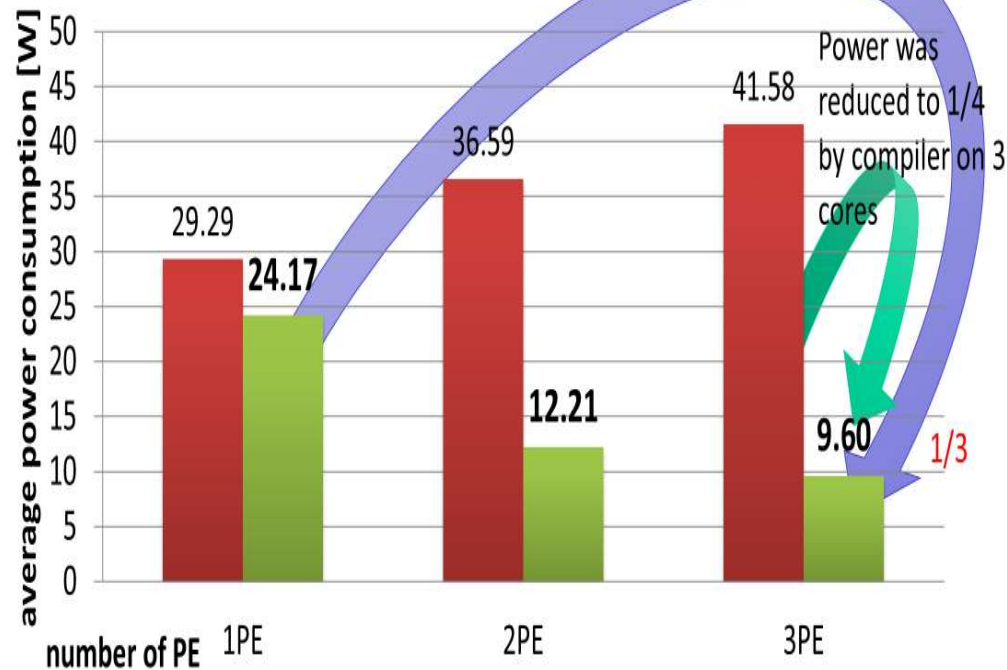
Power Reduction in a real-time execution controlled by OSCAR Compiler and OSCAR API on RP-X (Optical Flow with a hand-tuned library)



Power Reduction on Intel Haswell for Real-time Optical Flow

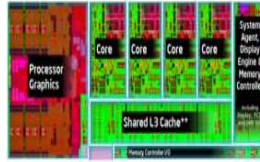
Intel CPU Core i7 4770K

■ without power control ■ with power control



Power was reduced to 1/4 (9.6W) by the compiler power optimization on the same 3 cores (41.6W).

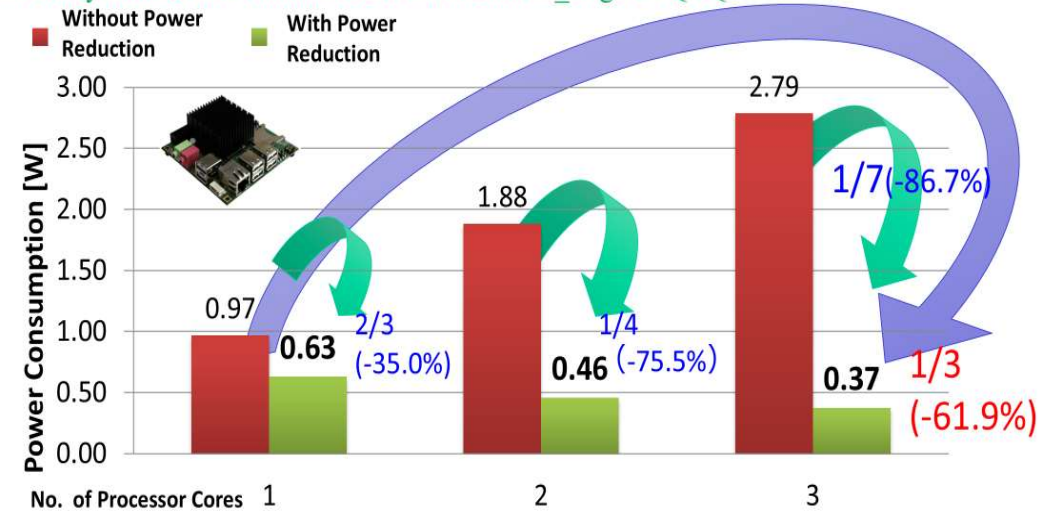
Power with 3 core was reduced to 1/3 (9.6W) against 1 core (29.3W).



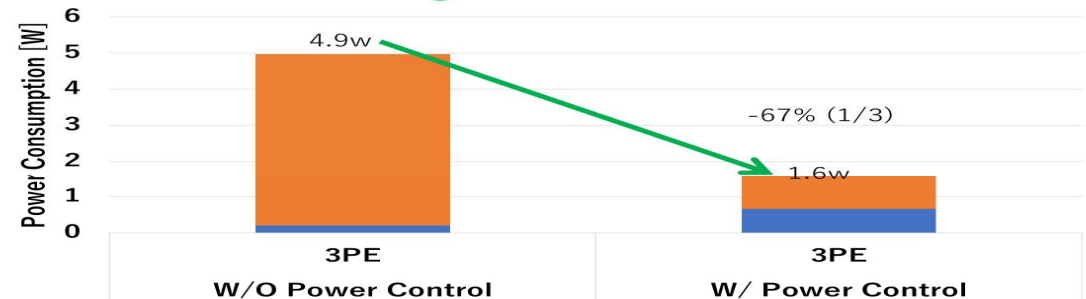
Automatic Power Reduction for MPEG2 Decode on Android Multicore

ODROID X2 ARM Cortex-A9 4 cores

http://www.youtube.com/channel/UCS43INYEIkC8i_KIgFZYQBQ



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

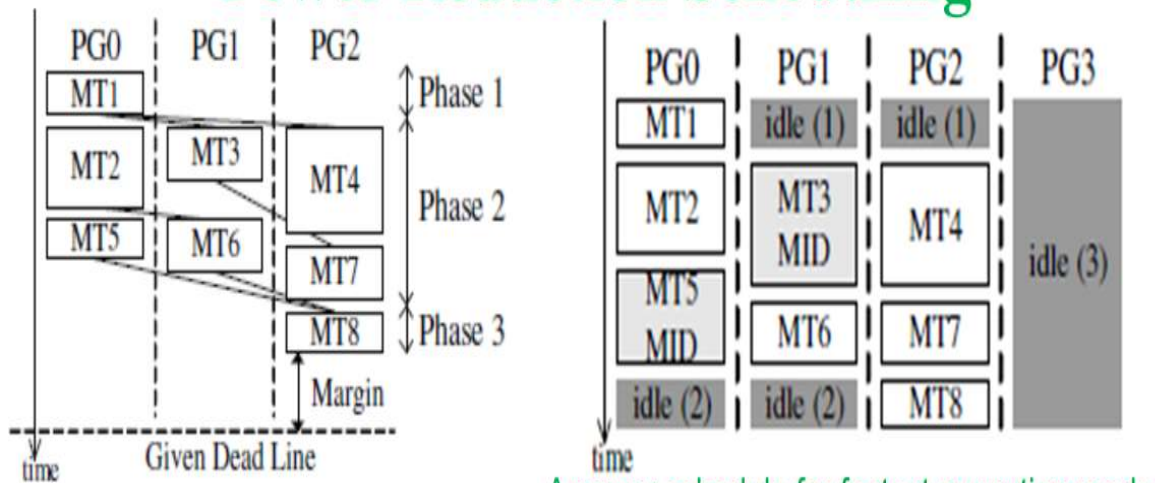


• ODROID-XU3

• 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

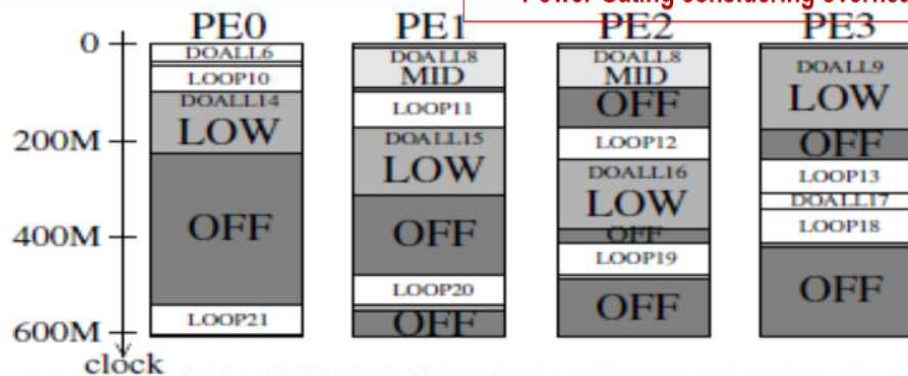
Power Reduction Scheduling



A power schedule for fastest execution mode

- 1) Reduce frequencies (F_s) of MTs on CP considering dead line.
- 2) Reduce F_s of MTs not on CP. Idle: Clock or Power Gating considering overheads.

Realtime scheduling mode
MTs 1,4,7,8 are on Critical Path (CP)



A power schedule for SPEC95 APPLU for fastest execution mode

Doall6, Loop 10,11,12,13, Doall 17, Loop 18,19,20, 21 are on CP

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

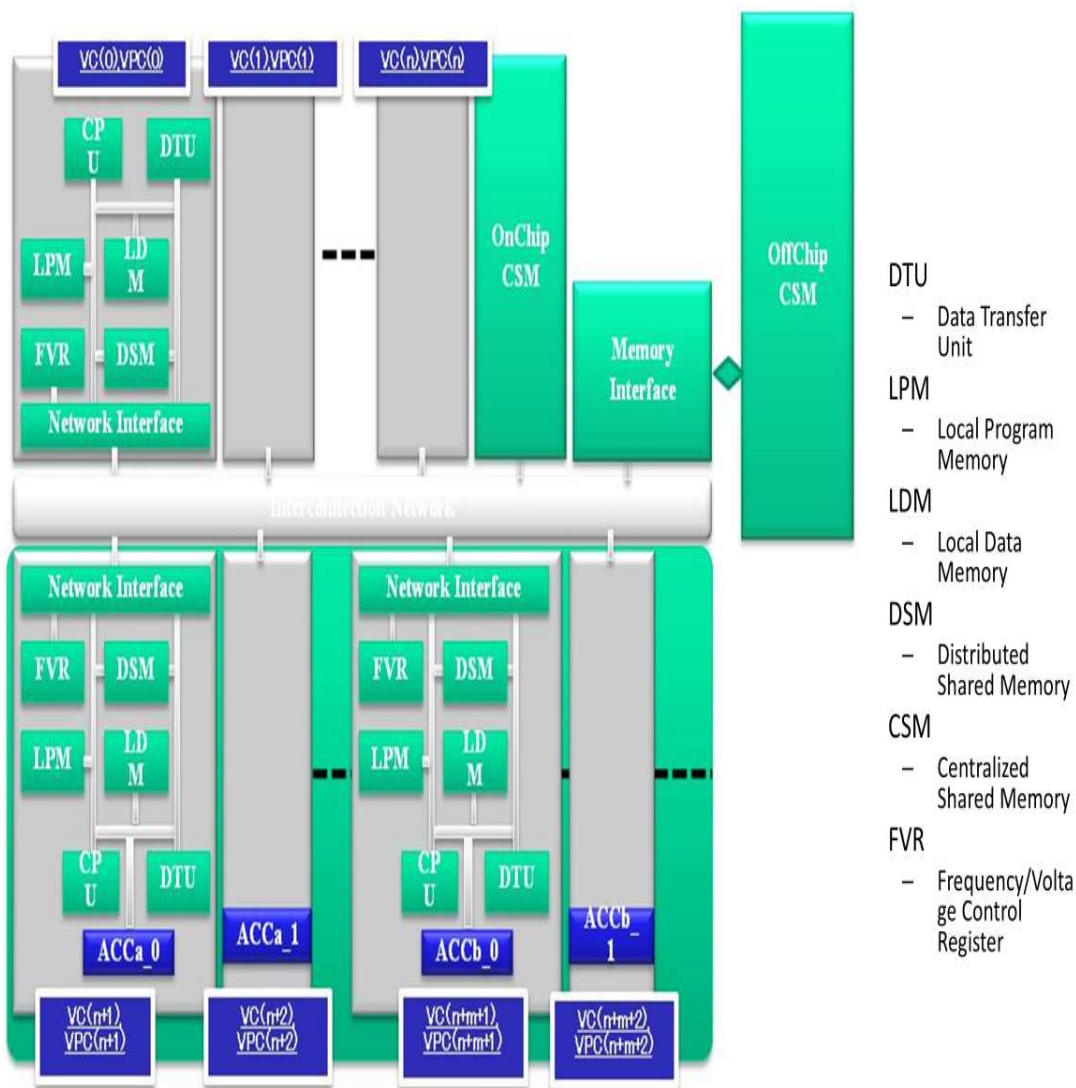
• State transition overhead

state	FULL	MID	LOW	OFF	state	FULL	MID	LOW	OFF
FULL	0	40k	40k	80k	FULL	0	20	20	40
MID	40k	0	40k	80k	MID	20	0	20	40
LOW	40k	40k	0	80k	LOW	20	20	0	40
OFF	80k	80k	80k	0	OFF	40	40	40	0

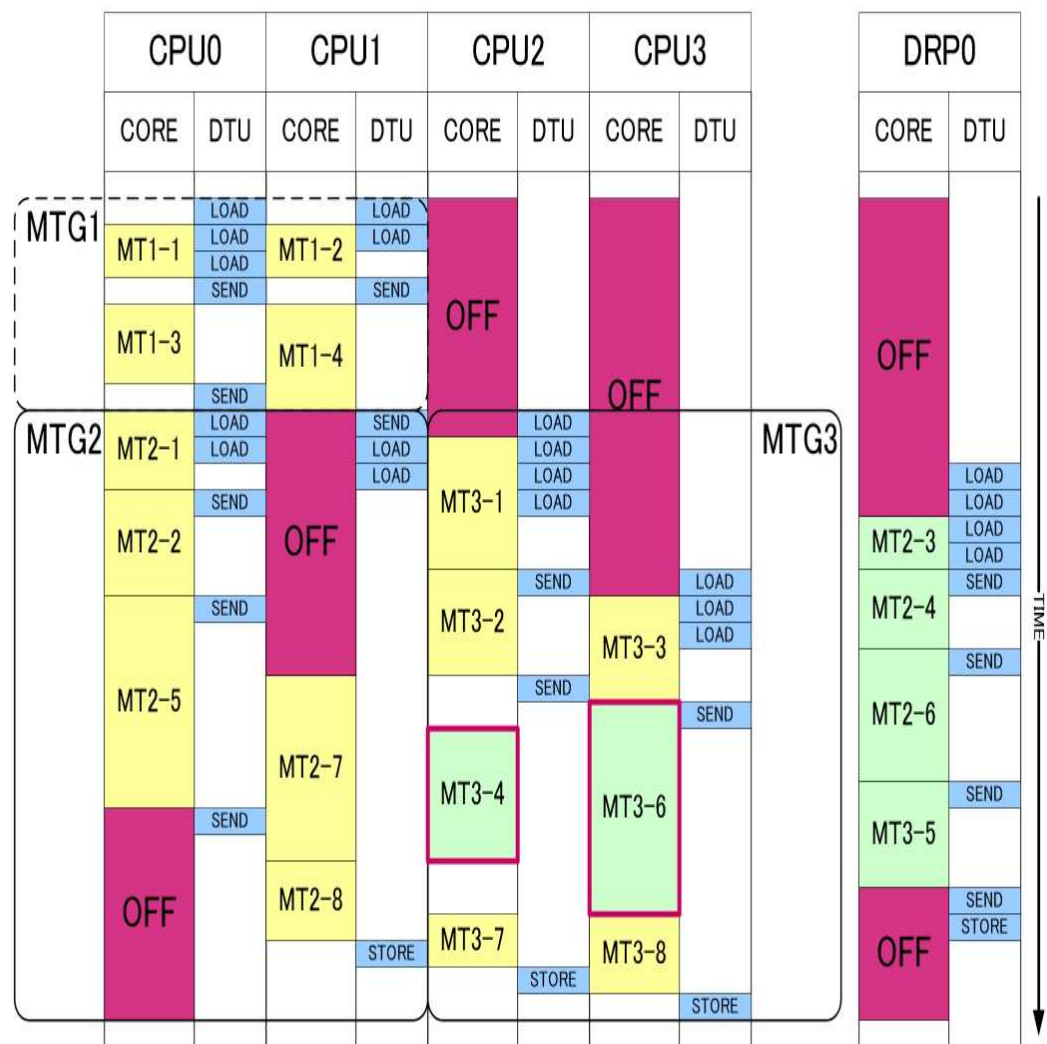
delay time [u.t.]

energy overhead [μ J]

OSCAR Heterogeneous Multicore



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



OSCAR API for Homogeneous/Heterogeneous Multicores with Multigrain Parallelization, Local Memory Management, Software Cache Coherent Control, DMA Data Transfer, Power Reduction

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`
- ▶ Memoary 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`

For systems having no cache coherent control hardware

2 hint directives for OSCAR compiler

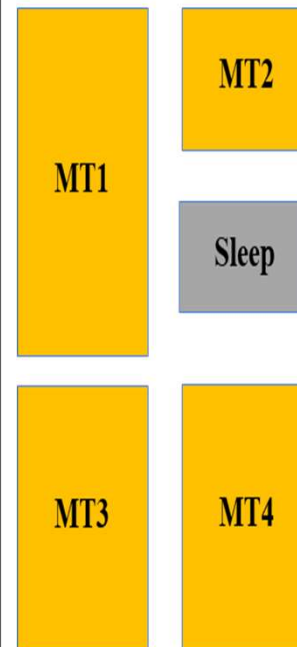
- `accelerator_task`
- `oscar_comment`

from V2.0

Low-Power Optimization with OSCAR API

Scheduled Result
by OSCAR Compiler

VC0 VC1



Generate Code Image by OSCAR Compiler

```
void  
main_VC0() {
```



```
#pragma oscar fvcontrol ¥  
(1,(OSCAR_CPU0,100))
```



```
}
```

```
void  
main_VC1() {
```



```
#pragma oscar fvcontrol ¥  
((OSCAR_CPU0,0))
```



```
}
```