# Cool Chips, Low Power Multicores, Open the Way to the Future

**Hironori Kasahara**

**President Elect 2017, President 2018**

**IEEE Computer Society**

**IEEE Fellow**

**Professor, Dept. of Computer Science & Engineering**

**Director, Advanced Multicore Processor Research Institute**

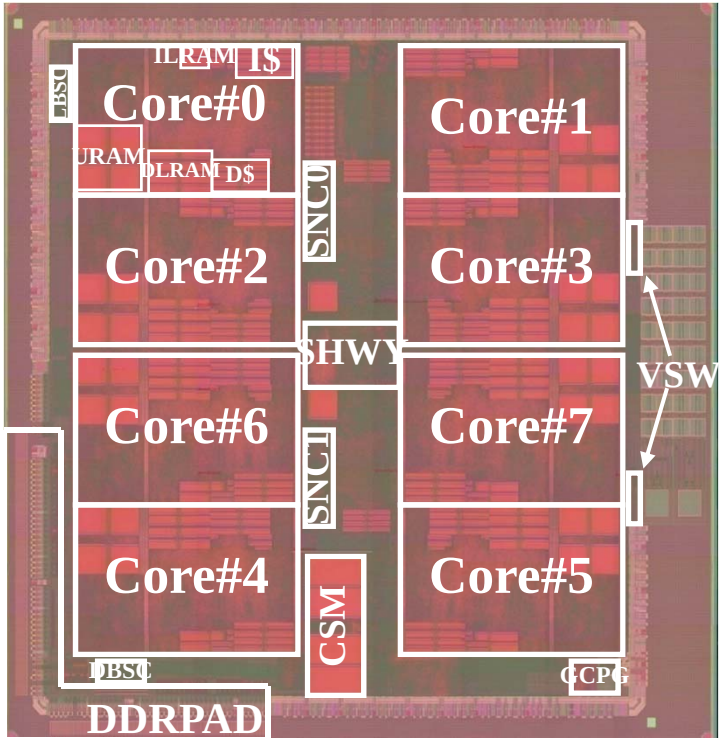
**Waseda University, Tokyo, Japan**

**URL: http://www.kasahara.cs.waseda.ac.jp/**

**Panel at the 20th IEEE COOL Chips, April 20, 2017**

# Multicores for Performance and Low Power

**Power consumption is one of the biggest problems for performance scaling from smartphones to cloud servers and supercomputers (“K” more than 10MW) .**




**IEEE ISSCC08: Paper No. 4.5, M.ITO, … and H. Kasahara, “An 8640 MIPS SoC with Independent Power-off Control of 8 CPUs and 8 RAMs by an Automatic Parallelizing Compiler”**

**Power ∝ Frequency * Voltage$^2$**

**(Voltage ∝ Frequency)**

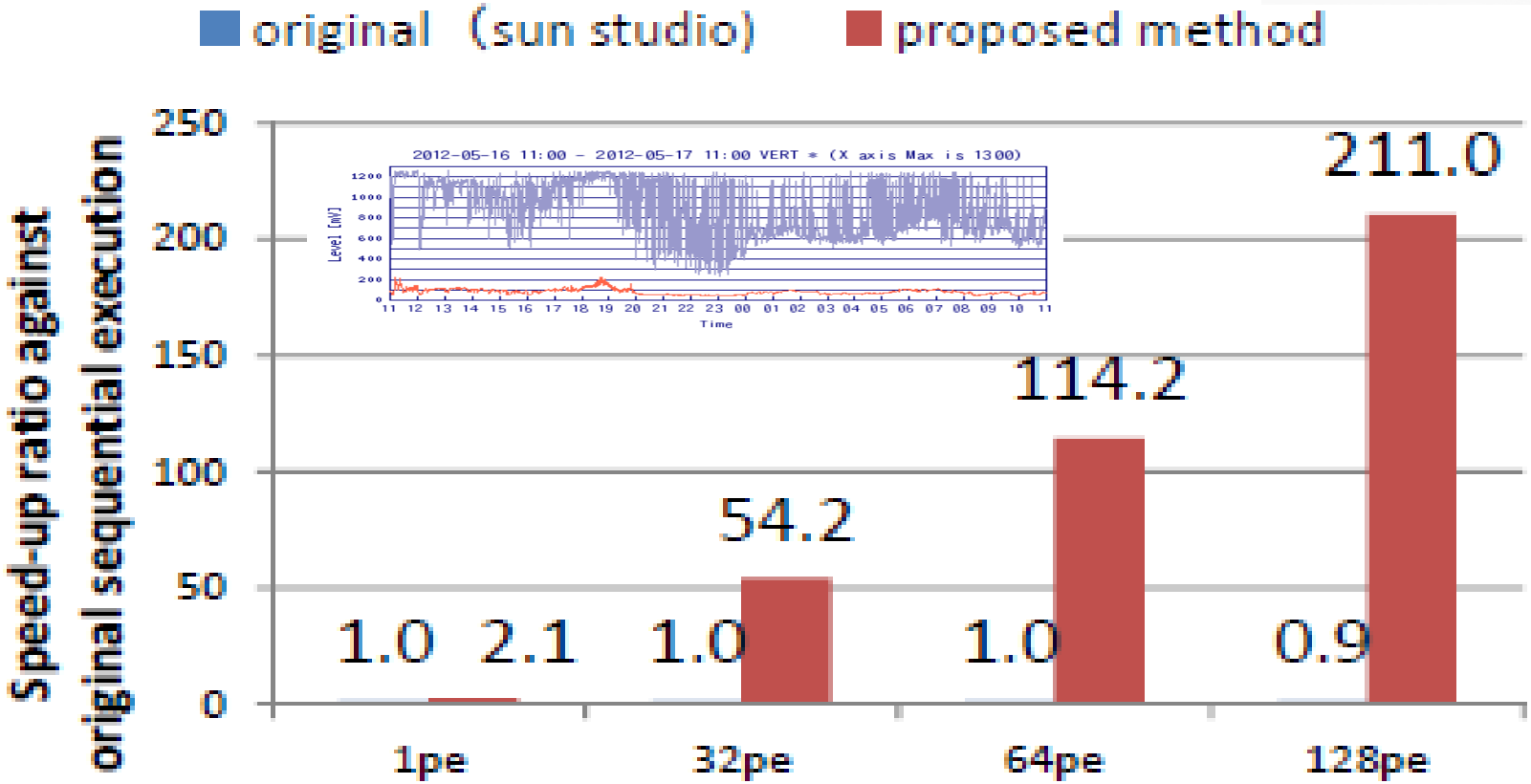
**⇒ Power ∝ Frequency$^3$**

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

**<Multicores>**

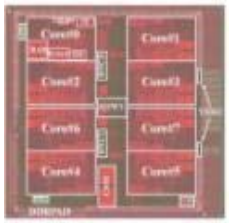
**If 8cores are integrated on a chip, Power is still 1/8 and Performance becomes 2 times.**

# Earthquake Simulation "GMS" on Fujitsu M9000 Sparc CC-NUMA Server



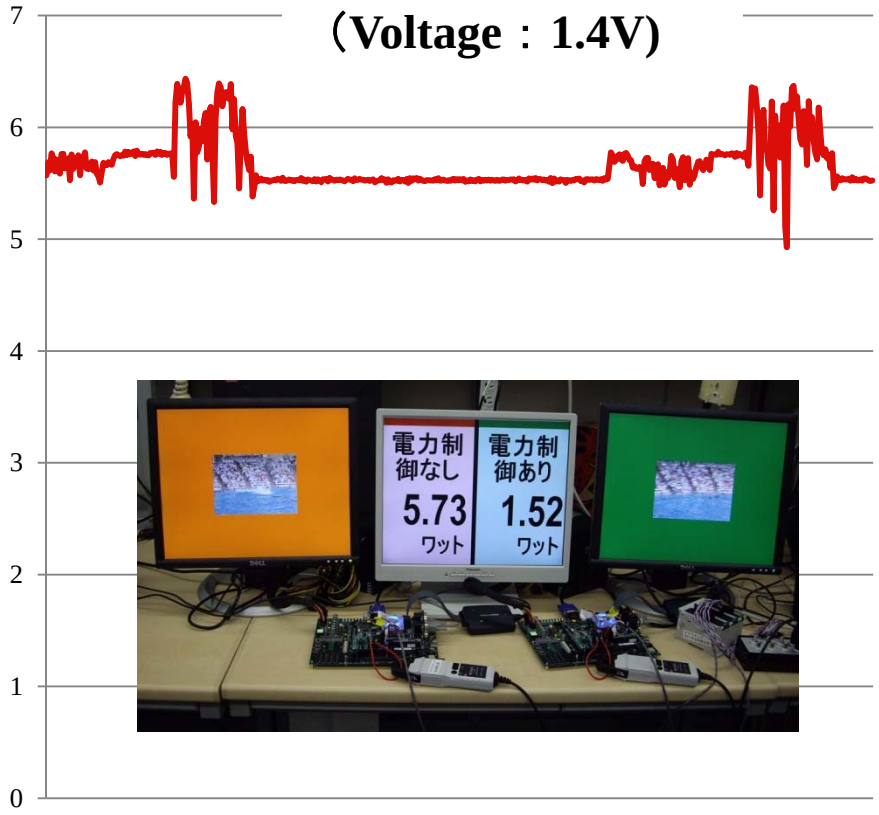

**With 128 cores, OSCAR compiler gave us 100 times speedup against 1 core execution and 211 times speedup against 1 core using Sun (Oracle) Studio compiler.**

# 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

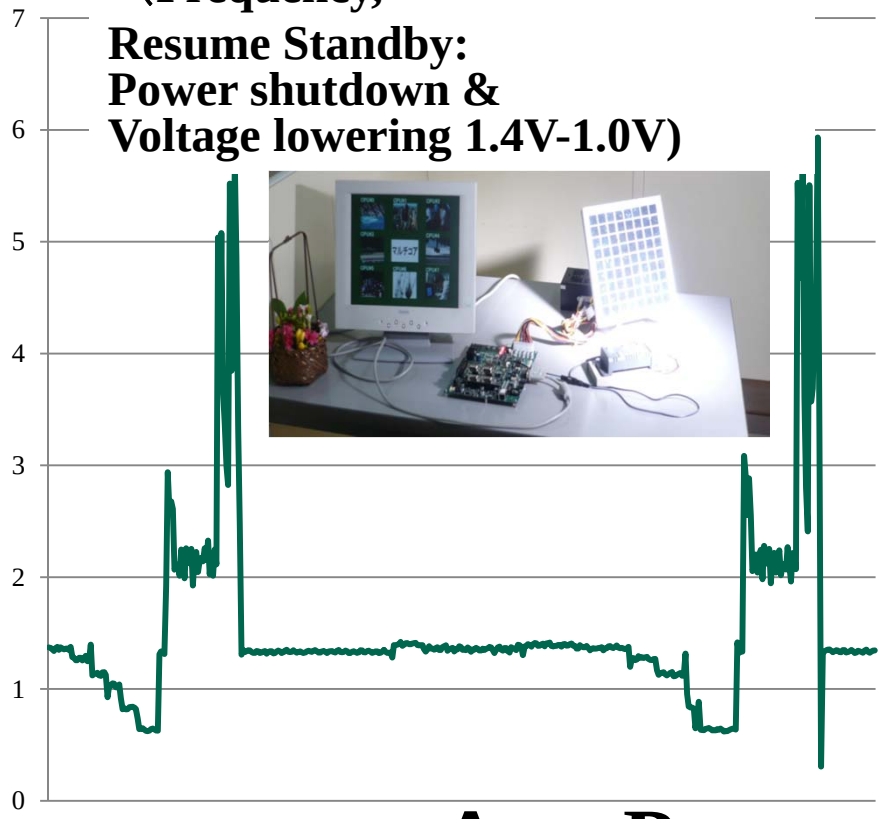
**Without Power Control （Voltage：1.4V)**







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



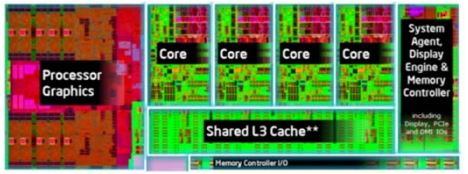



**Avg. Power 5.73 [W]**

**73.5% Power Reduction**

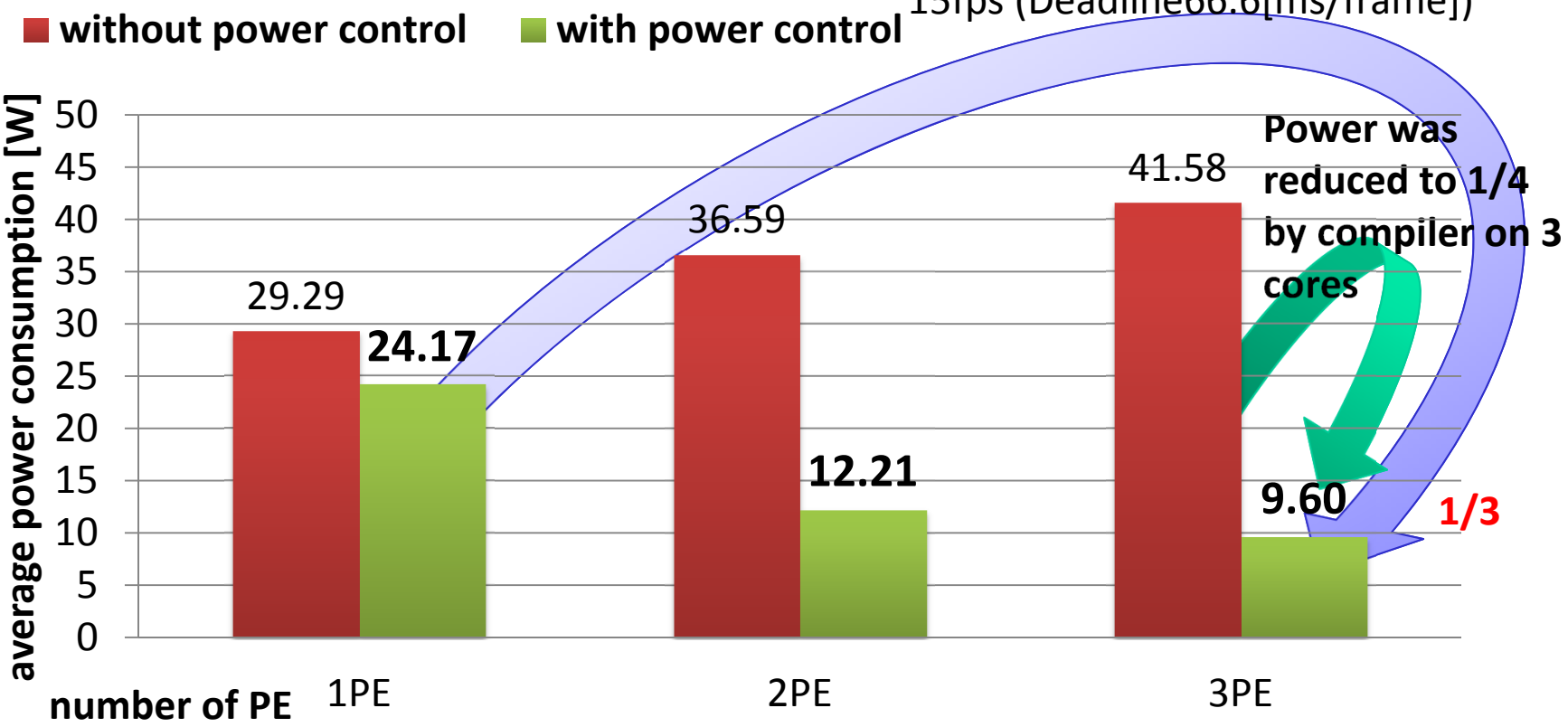
**Avg. Power 1.52 [W]**

# Power Reduction on Intel Haswell for Real-time Optical Flow




**Intel CPU Core i7 4770K**

For HD 720p(1280x720) moving pictures
15fps (Deadline66.6[ms/frame])




**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) .**

# OSCAR Parallelizing Compiler

**To improve effective performance, cost-performance and software productivity and reduce power**

## Multigrain Parallelization

coarse-grain parallelism among loops and subroutines, near fine grain parallelism among statements in addition to loop parallelism

## Data Localization

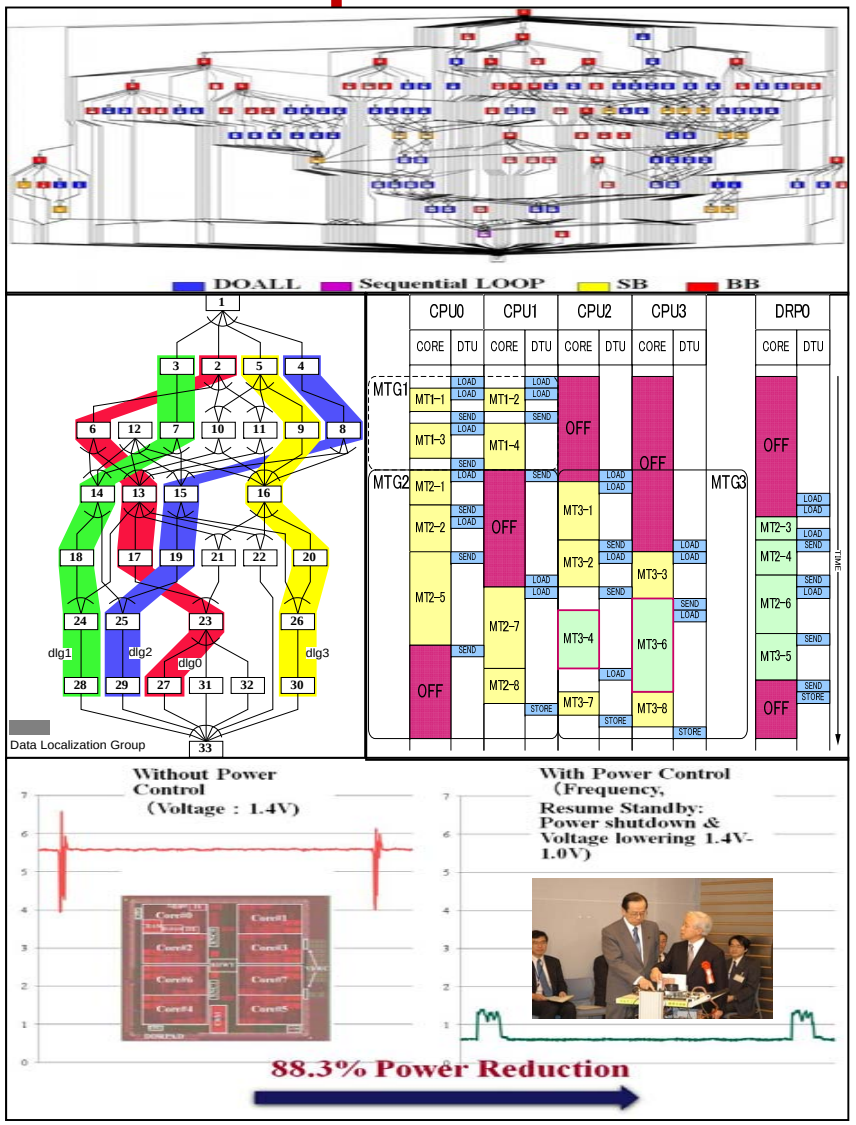
Automatic data management for distributed shared memory, cache and local memory

## Data Transfer Overlapping

Data transfer overlapping using Data Transfer Controllers (DMAs)

## Power Reduction

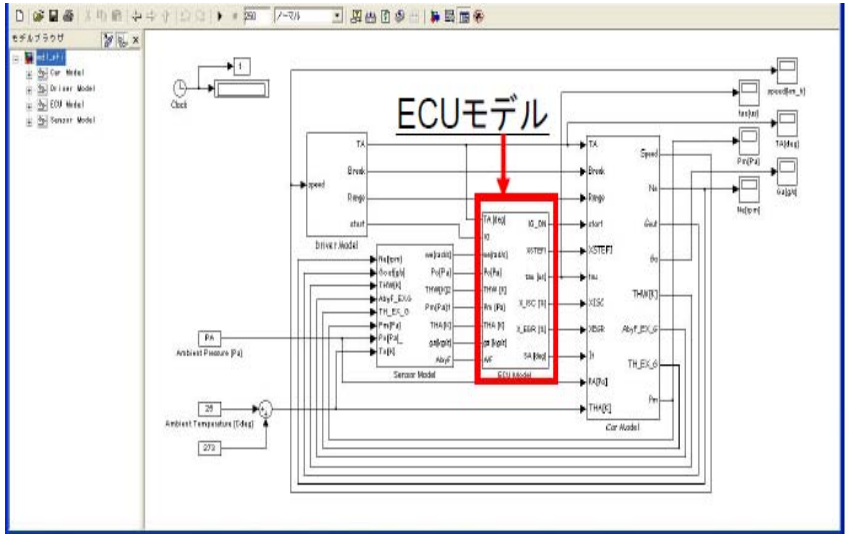
Reduction of consumed power by compiler control DVFS and Power gating with hardware supports.

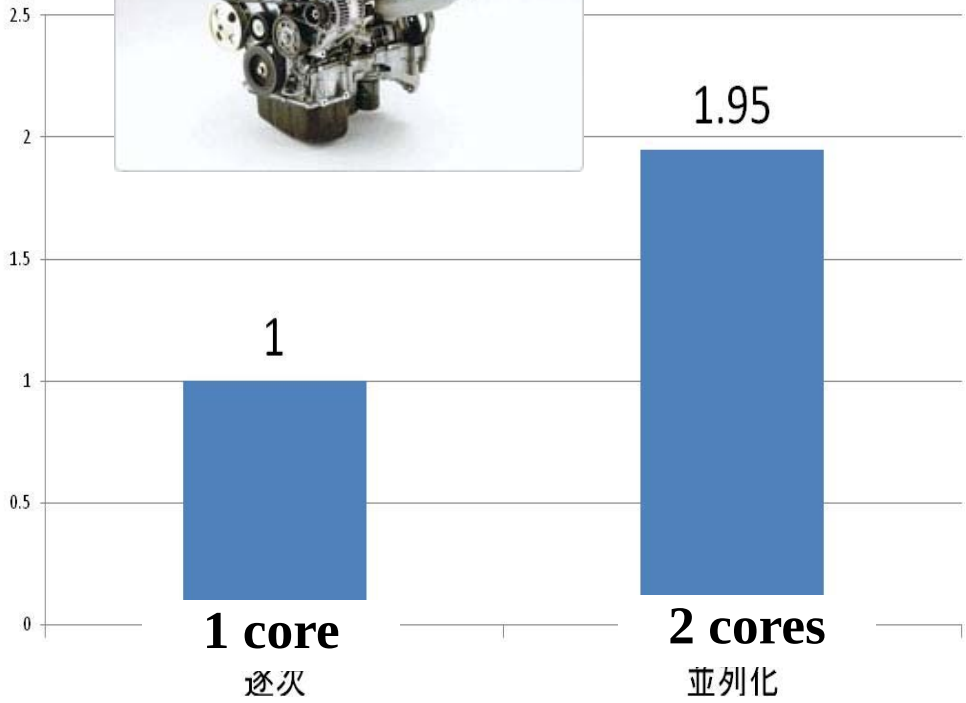


# Engine Control by multicore with Denso

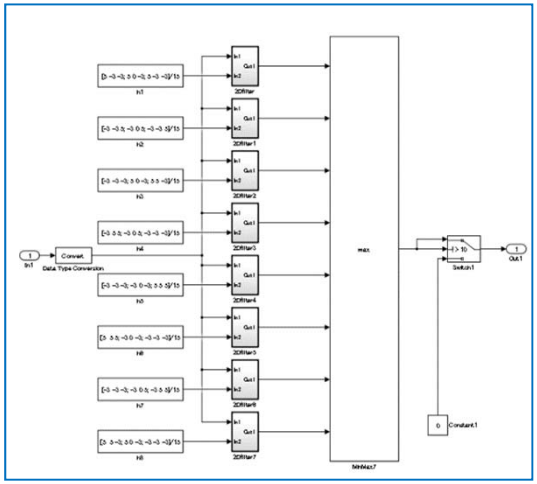
**Though so far parallel processing of the engine control on multicore has been very difficult, Denso and Waseda succeeded 1.95 times speedup on 2core V850 multicore processor.**


Hard real-time automobile engine control by multicore



# OSCAR Compile Flow for Simulink Applications



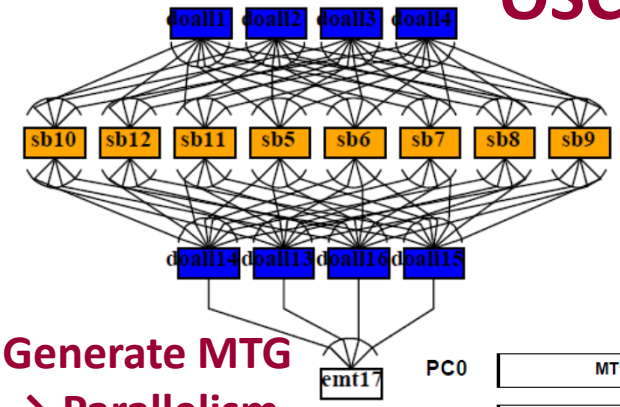
Simulink model

Generate C code
using Embedded Coder

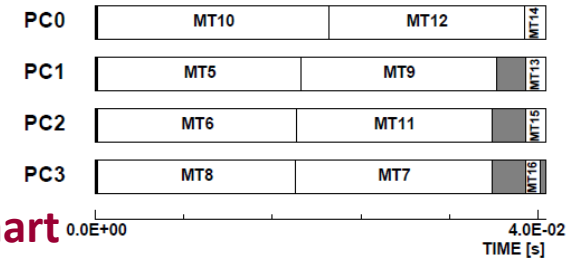
```
/* Model step function */
void VesselExtraction_step(void)
{
  int32_T i;
  real_T u0;

  /* DataTypeConversion: '<S1>/Data Type Conversion' incorporates:
   *  Inport: '<Root>/In1'
   */
  for (i = 0; i < 16384; i++) {
    VesselExtraction_B.DataTypeConversion[i] = VesselExtraction_U.In1[i];
  }

  /* End of DataTypeConversion: '<S1>/Data Type Conversion' */

  /* Outputs for Atomic SubSystem: '<S1>/2Dfilter' */

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

  /* End of Outputs for SubSystem: '<S1>/2Dfilter' */

  /* Outputs for Atomic SubSystem: '<S1>/2Dfilter1' */

  /* Constant: '<S1>/h2' */
  VesselExtraction_Dfilter(VesselExtraction_B.DataTypeConversion,
    VesselExtraction_P.h2_Value, &VesselExtraction_B.Dfilter1,
    (P_Dfilter_VesselExtraction_T *)&VesselExtraction_P.Dfilter1);
```

C code

## OSCAR Compiler




(1) Generate MTG
→ Parallelism



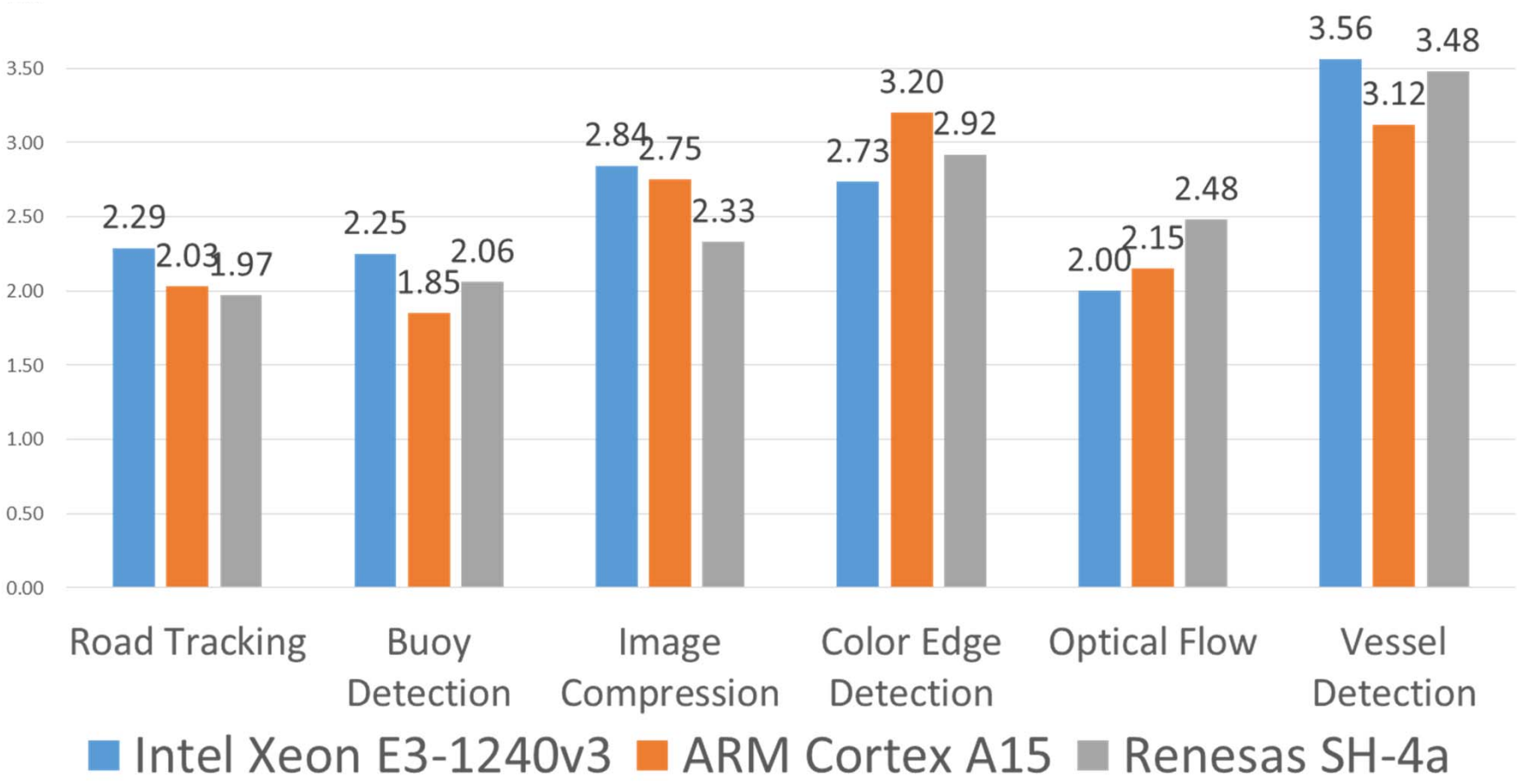

(2) Generate gantt chart
→ Scheduling in a multicore

```
void VesselExtraction_step ( )
{
    int thr1 ;
    int thr2 ;
    int thr3 ;
    {
        oscar_thread_create ( & thr1 ,
            thread_function_001 , (void*)1 ) ;
        oscar_thread_create ( & thr2 ,
            thread_function_002 , (void*)2 ) ;
        oscar_thread_create ( & thr3 ,
            thread_function_003 , (void*)3 ) ;

            VesselExtraction_step_PE0 ( ) ;

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

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

(3) Generate parallelized C code
using the OSCAR API
→ Multiplatform execution
(Intel, ARM and SH etc)

# Speedups of MATLAB/Simulink Image Processing on Various 4core Multicores

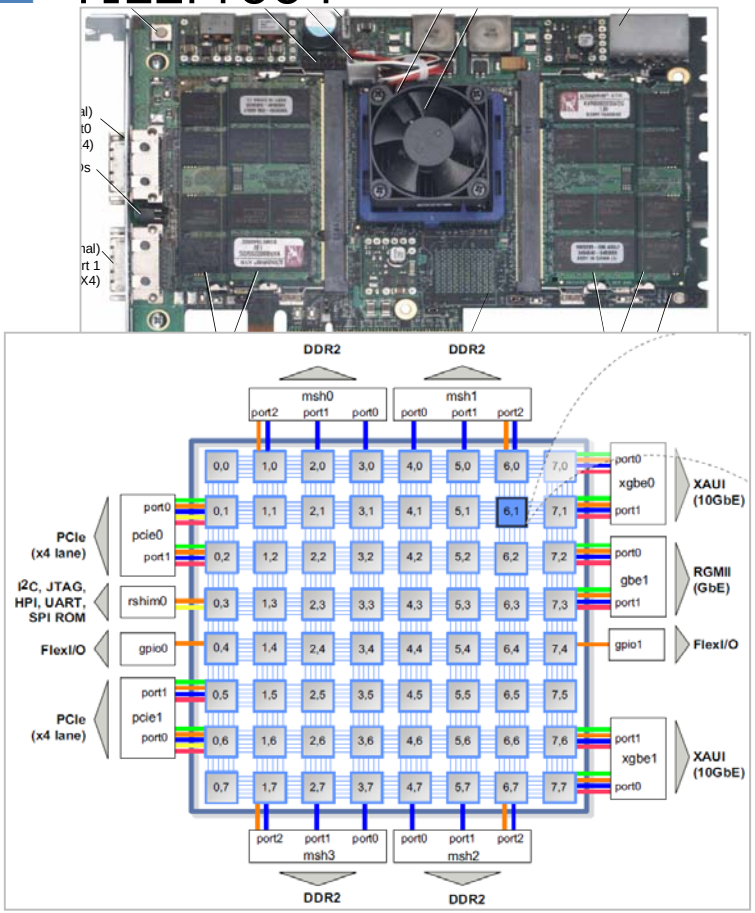
## (Intel Xeon, ARM Cortex A15 and Renesas SH4A)

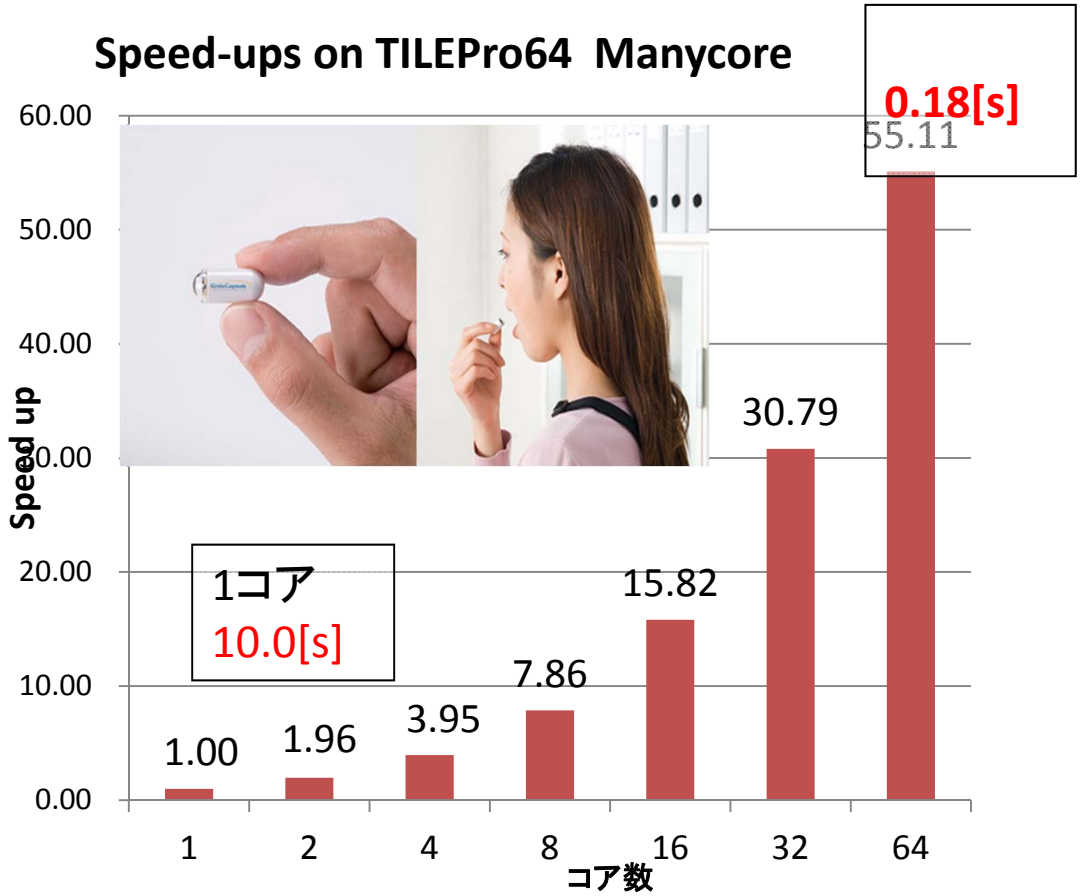



Road Tracking, Image Compression : http://www.mathworks.co.jp/jp/help/vision/examples
Buoy Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/44706-buoy-detection-using-simulink
Color Edge Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/28114-fast-edges-of-a-color-image--actual-color--not-converting-to-grayscale-/
Vessel Detection : http://www.mathworks.co.jp/matlabcentral/fileexchange/24990-retinal-blood-vessel-extraction/

# Automatic Parallelization of Still Image Encoding Using JPEG-XR for the Next Generation Cameras and Drinkable Inner Camera

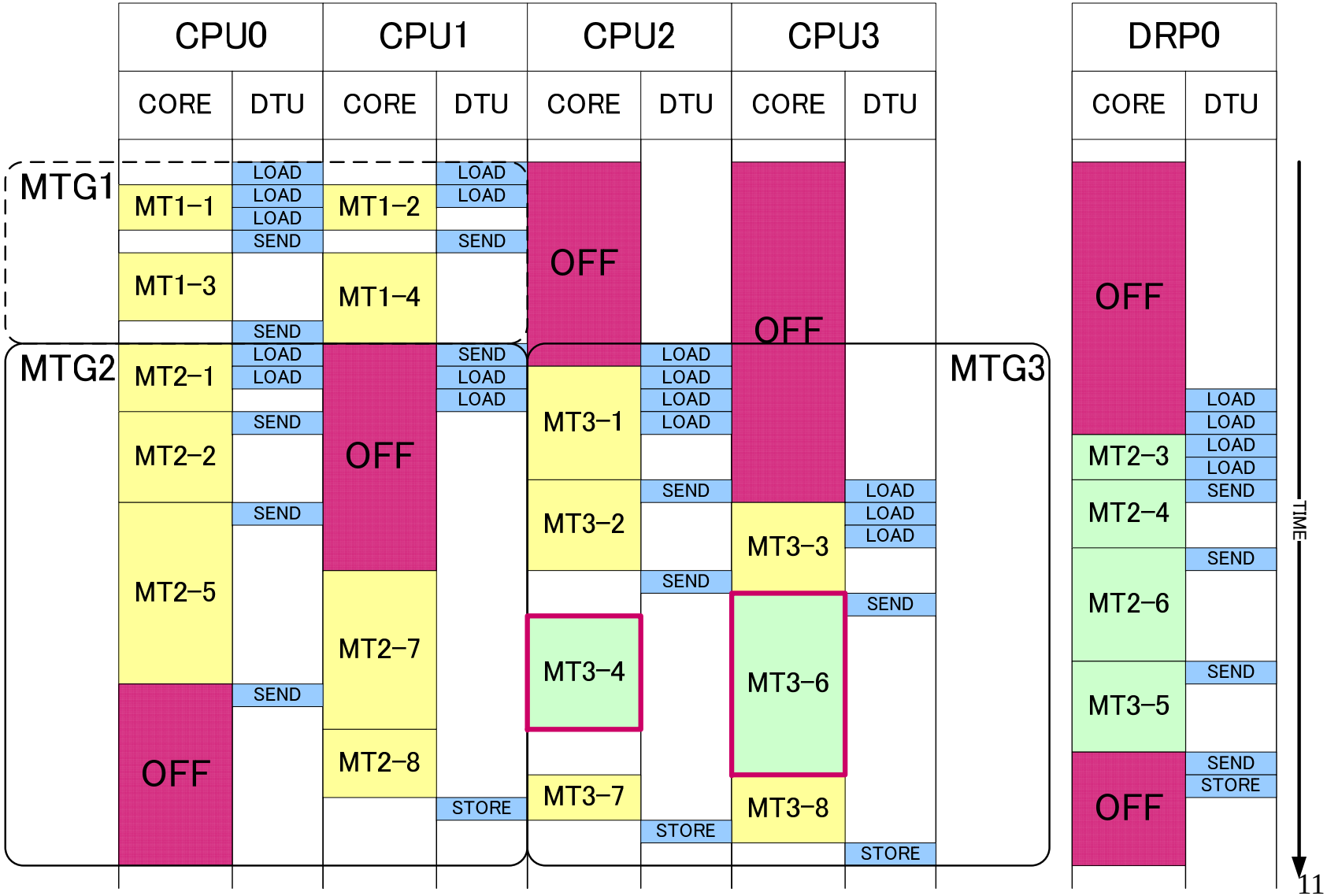
- TILEPro64



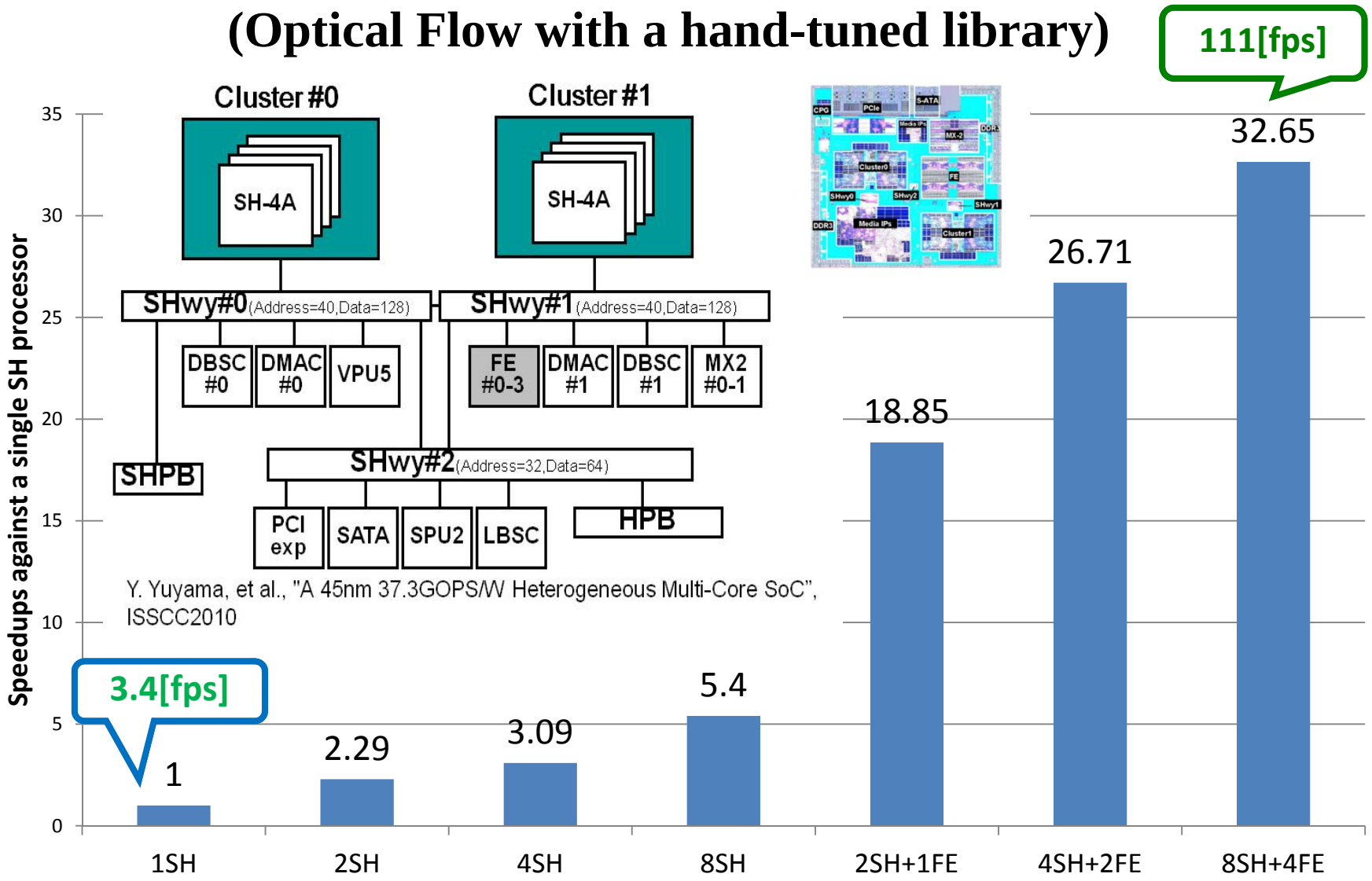

55 times speedup with 64 cores against 1 core

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



# 33 Times Speedup Using OSCAR Compiler and OSCAR API on RP-X

## (Optical Flow with a hand-tuned library)



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

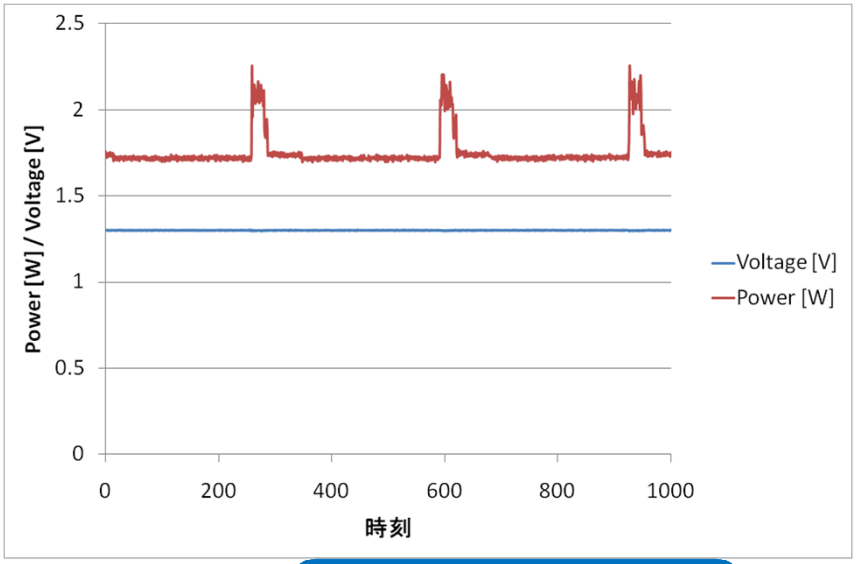
**Without Power Reduction**

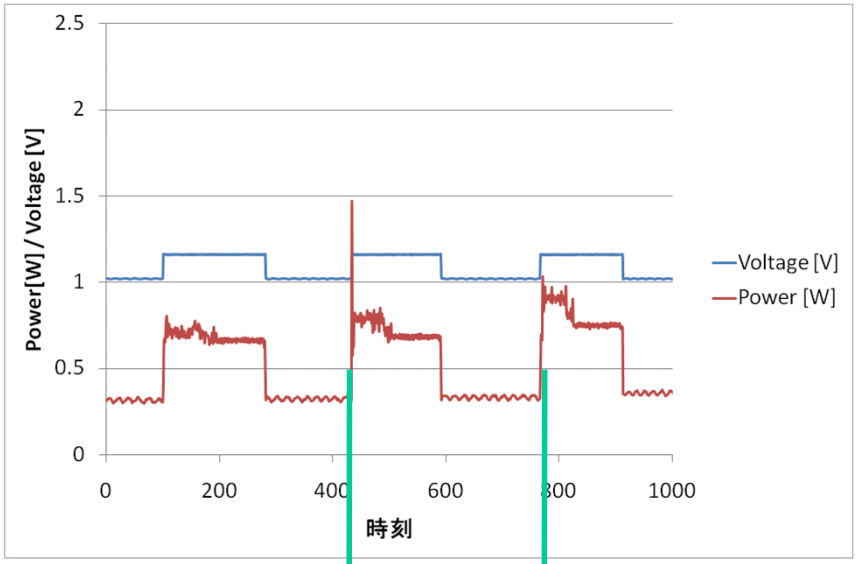
**With Power Reduction by OSCAR Compiler**

**70% of power reduction**

**Average:1.76[W]**

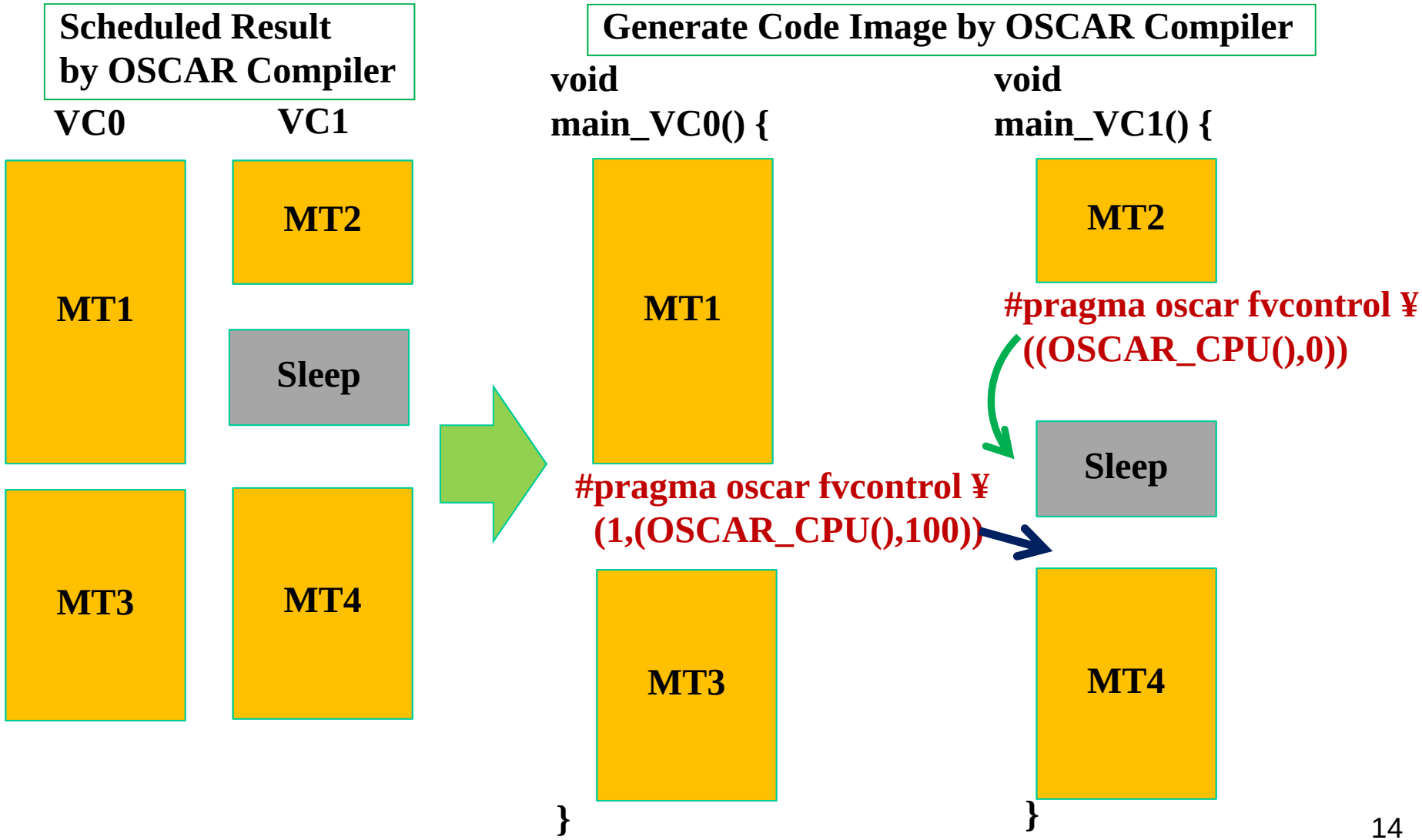
**Average:0.54[W]**



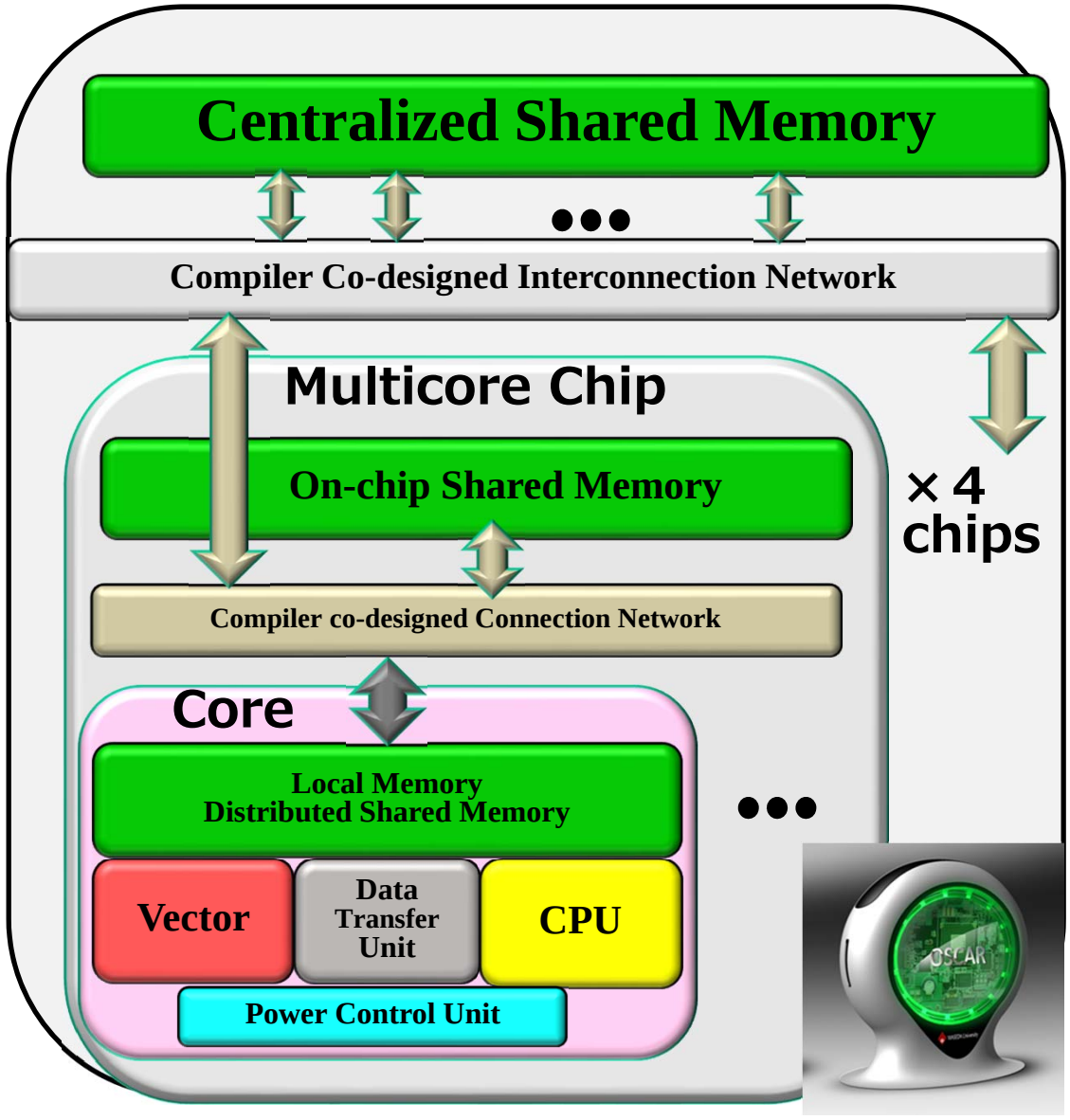




**1cycle : 33[ms]**
**→30[fps]**

# Low-Power Optimization with OSCAR API



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




**Target:**

- **Solar Powered with compiler power reduction.**
- **Fully automatic parallelization and vectorization including local memory management and data transfer.**

## Summary

- Now, we are in the era of Low-Power and High-Speed Chips.
- Software like compiler allows us to get several times more speedup and reduce power to one severalth on the same Low-Power and High-Speed multicore hardware.
- To improve performance and reduce power, collaboration of architecture, software, and application will be more important.
- To invite software, applications and co-design papers, how about changing the subtitle like "COOL Chips: IEEE Symposium on Low-Power and High-Speed Chip Systems: Chips, Software, Applications"