Automatic Parallelization of Automobile Engine Control Programs on Multicores

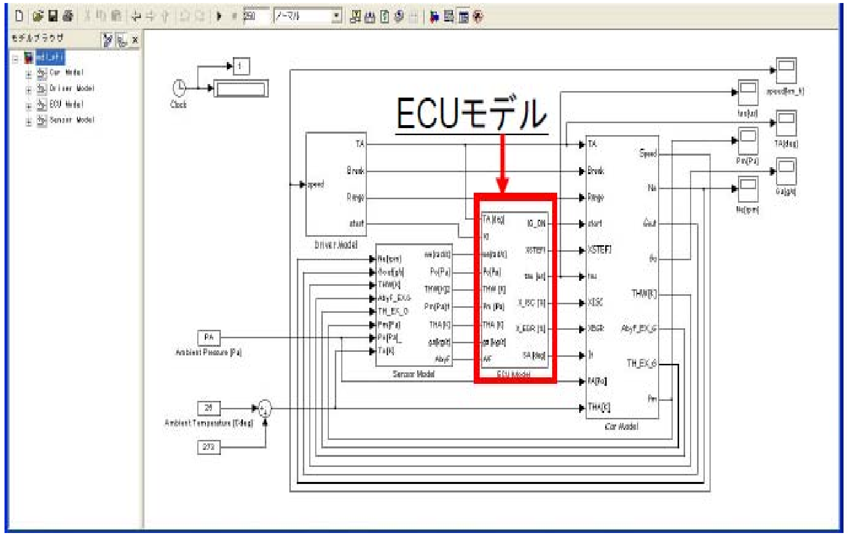
Hironori Kasahara
Professor, Dept. of Computer Science & Engineering
Director, Advanced Multicore Processor Research Institute
Waseda University, Tokyo, Japan
IEEE Computer Society Multicore STC Chair
URL: http://www.kasahara.cs.waseda.ac.jp/

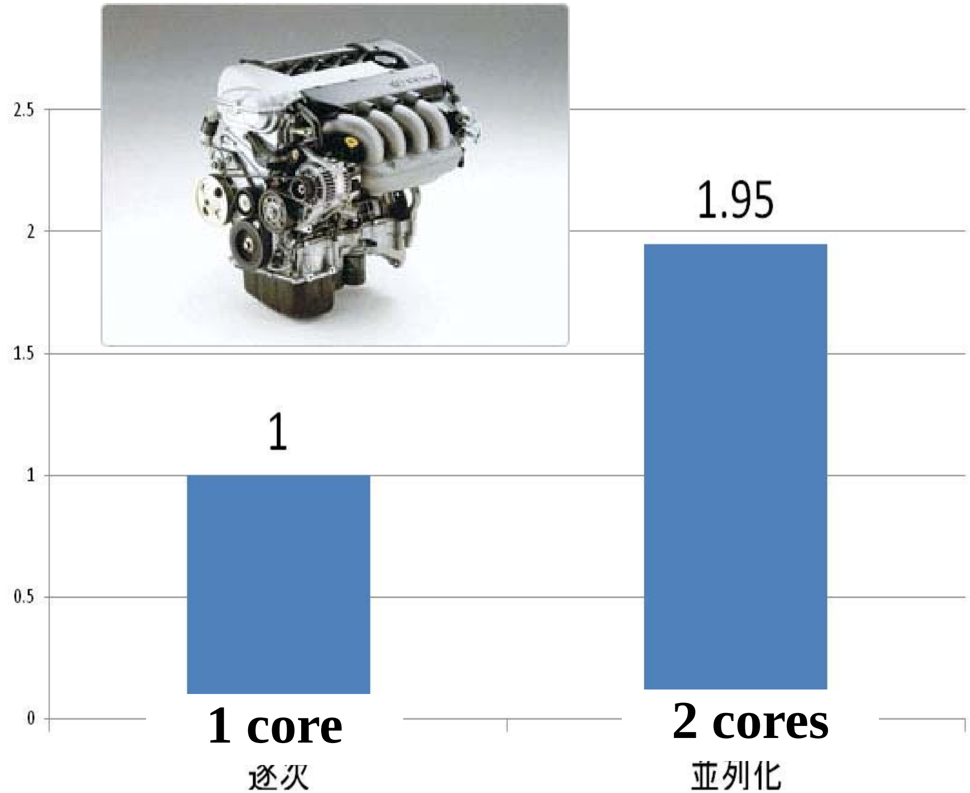
Engine Control by Multicores

Parallel processing of the engine control on multicore has been very difficult because of

- **the hard real time control using local memory**
- **programs with conditional branches, basic blocks, and no loop .**

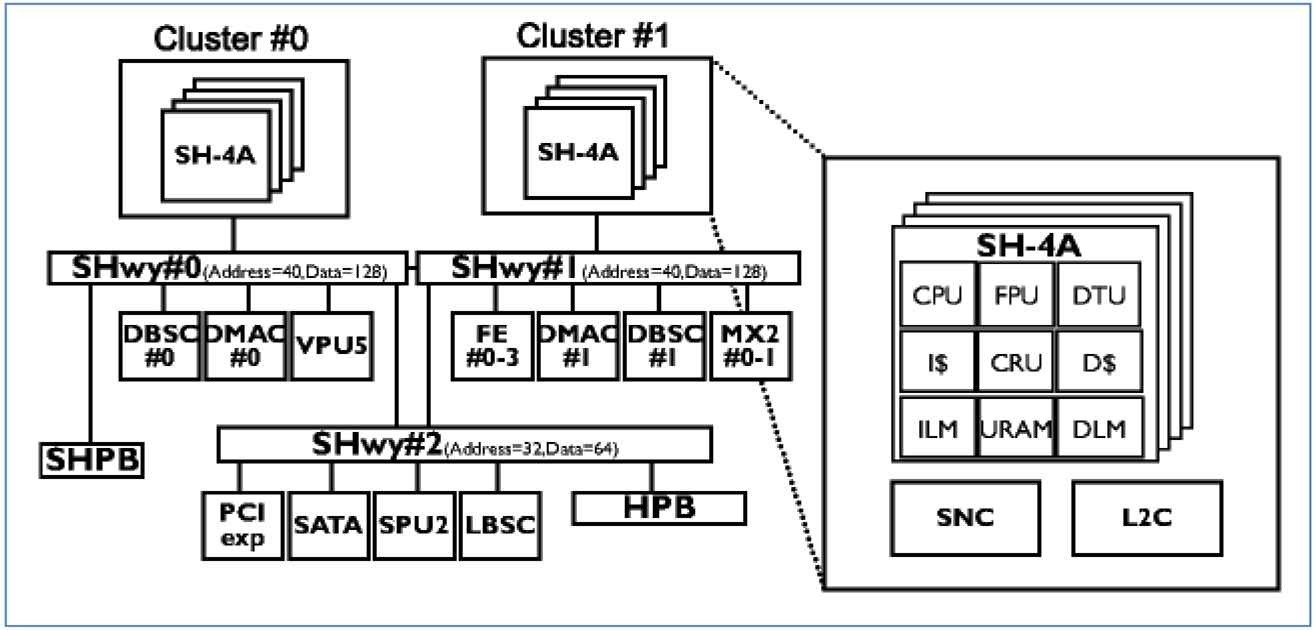
The developed method can be applied both for hand-written codes and model based designed codes.





1.95 times speedup on 2core V850 multicore processor

Embedded Multi-core Processor RPX developed by Hitachi, Renesas & Waseda



- 15 cores heterogeneous multicore
- SH-4A 648MHz * 8
- 3 types of accelerator cores

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

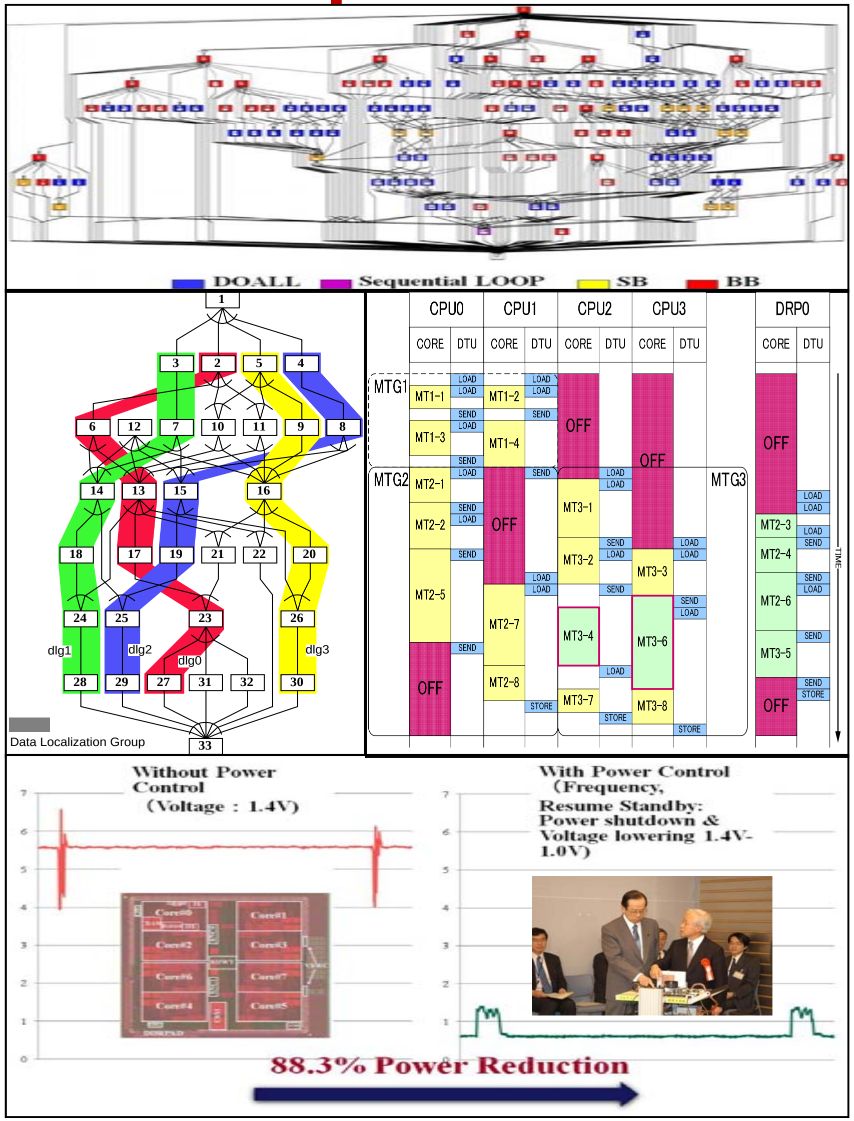
Automatic data management for distributed shared memory, cache and local memory

Data Transfer Overlapping

Data transfer overlapping using Data Transfer Controllers (DMAs)

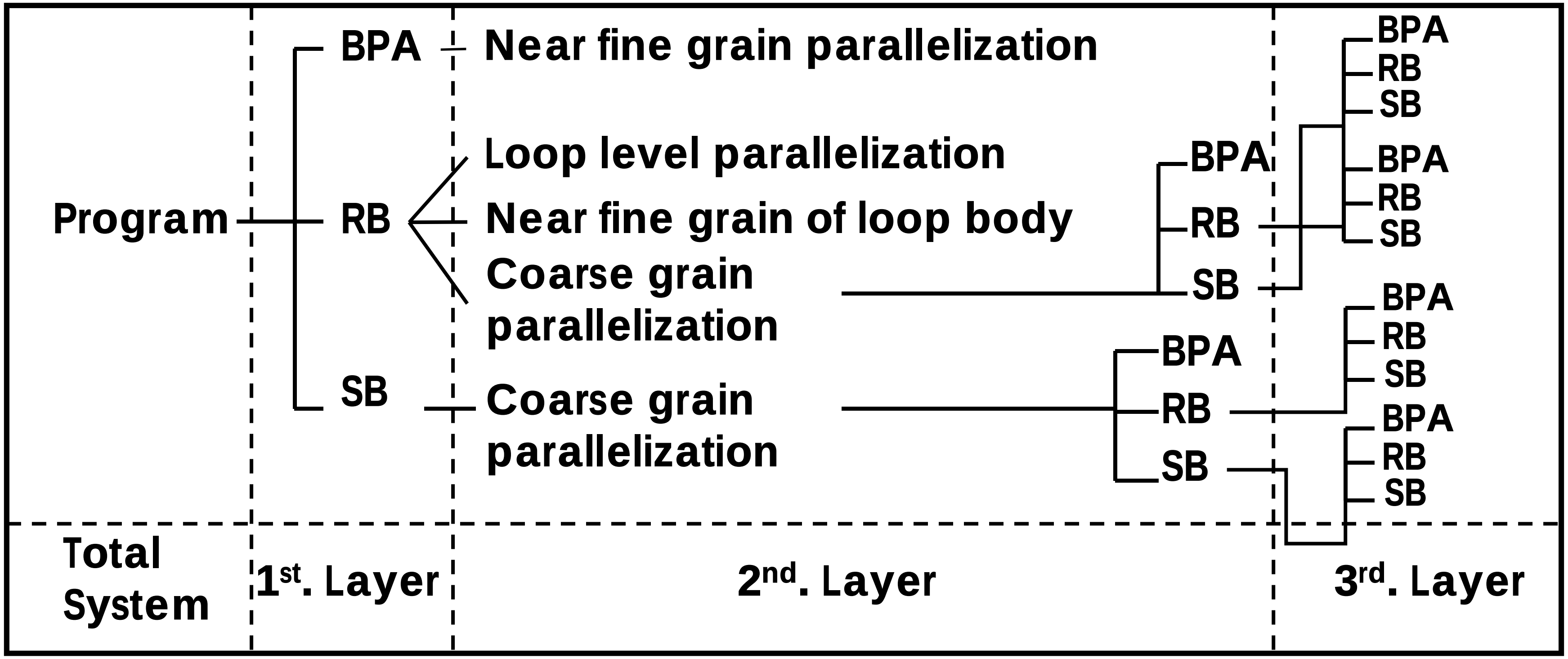
Power Reduction

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

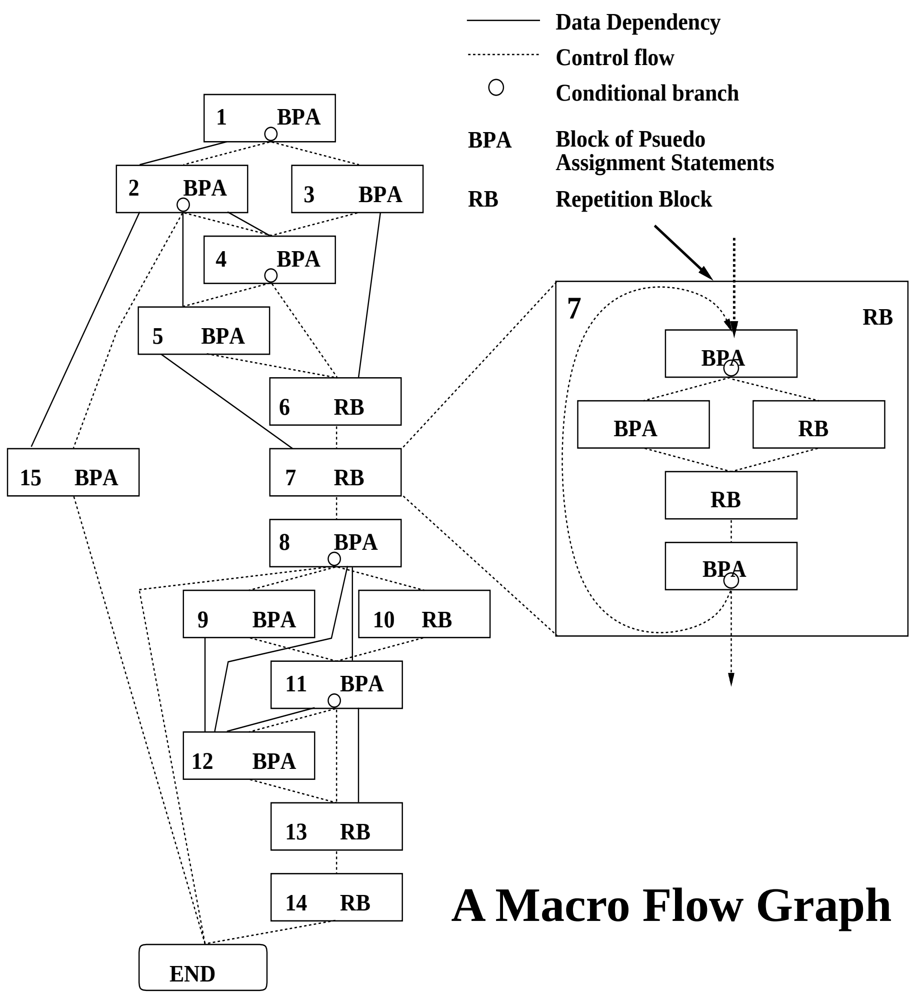


Generation of Coarse Grain Tasks

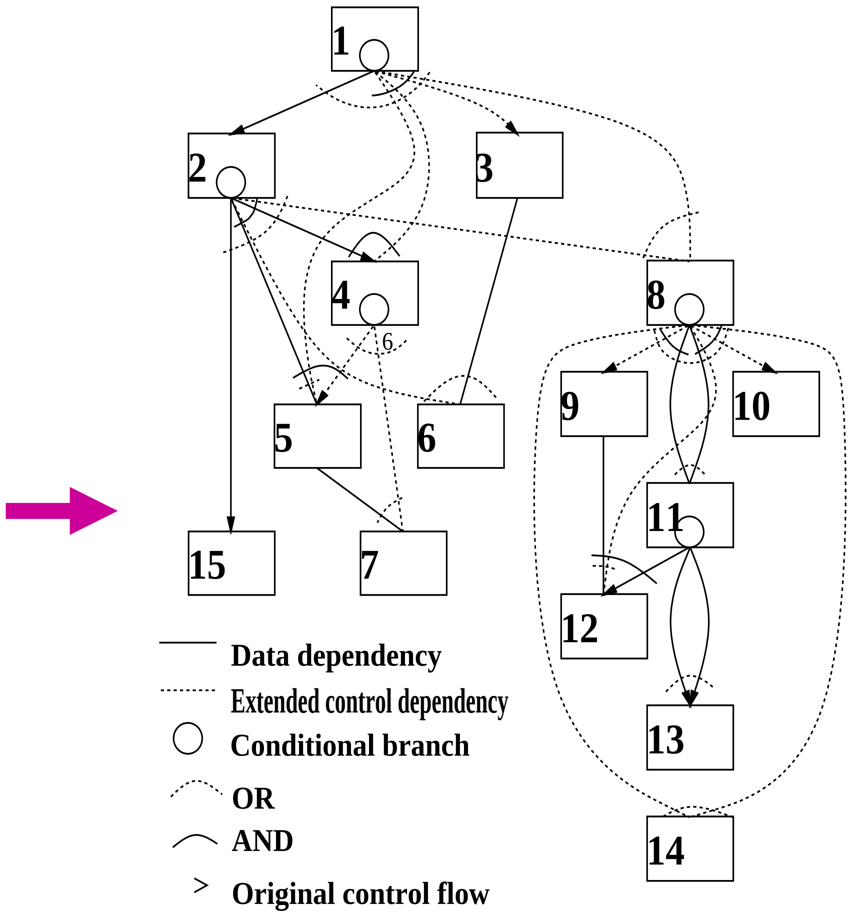
- **Macro-tasks (MTs)**
  - **Block of Pseudo Assignments (BPA): Basic Block (BB)**
  - **Repetition Block (RB) : natural loop**
  - **Subroutine Block (SB): subroutine**



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

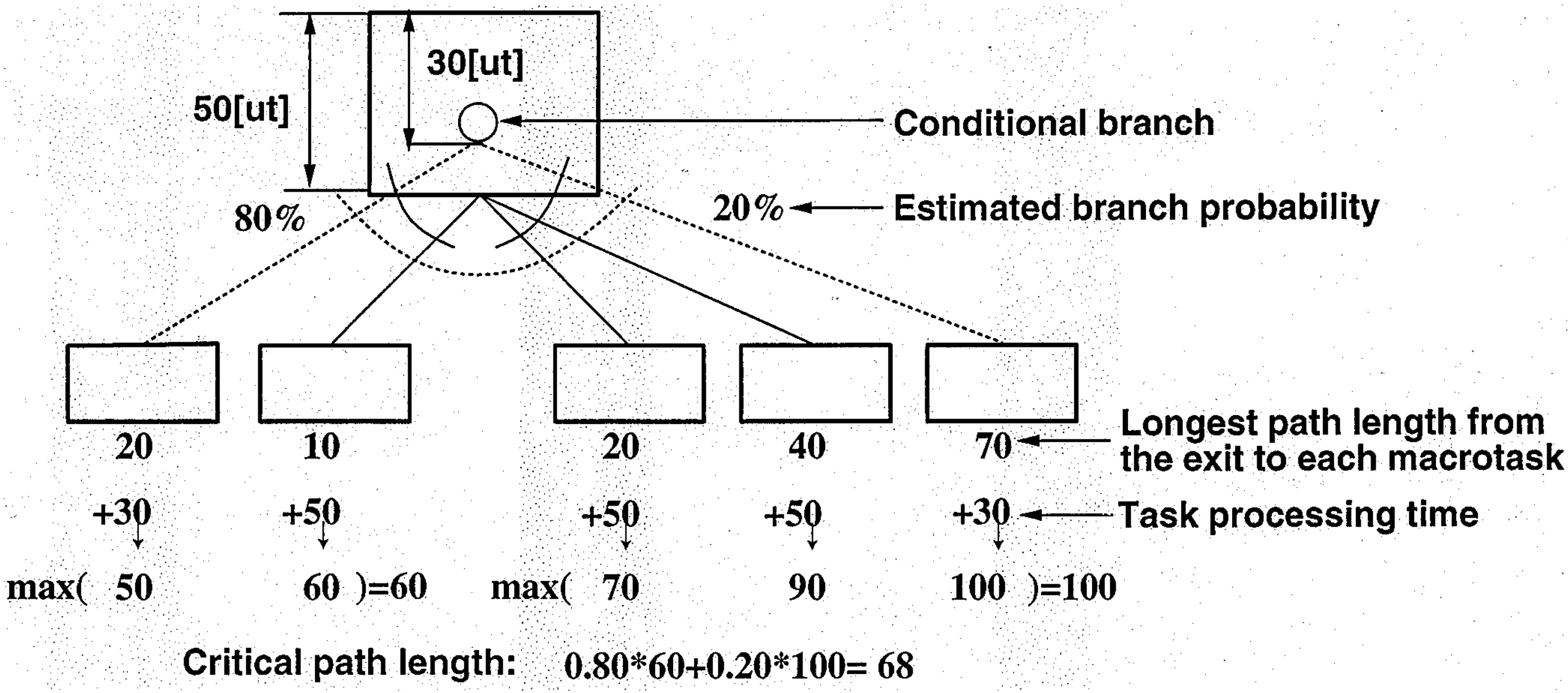


A Macro Flow Graph



A Macro Task Graph

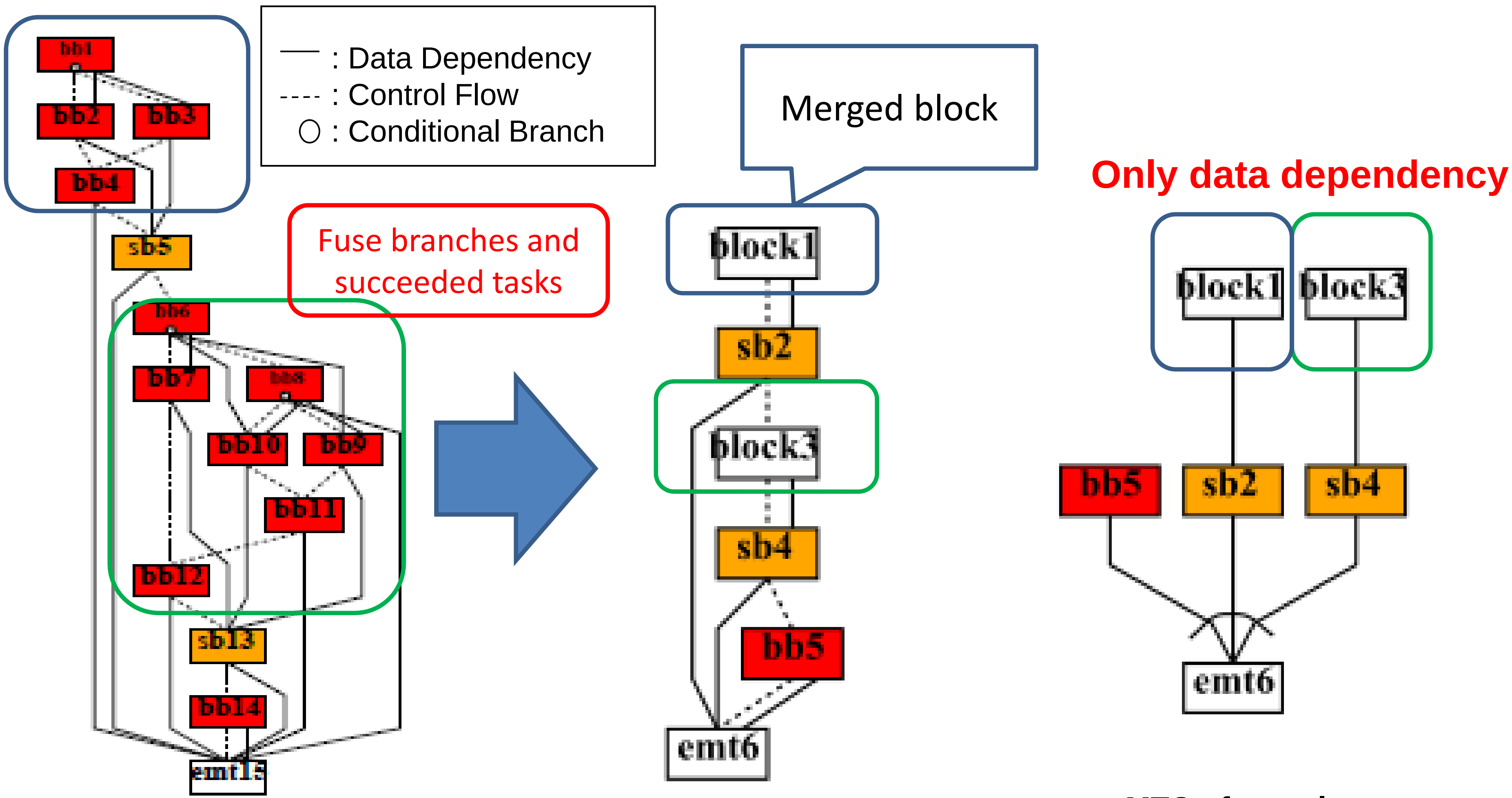
PRIORITY DETERMINATION IN DYNAMIC CP METHOD



Earliest Executable Conditions

Macrotask No.	Earliest Executable Condition
1	
2	1_{2}
3	$(1)_{3}$
4	2_{4} OR $(1)_{3}$
5	$(4)_{5}$ AND [2_{4} OR $(1)_{3}$]
6	3 OR $(2)_{4}$
7	5 OR $(4)_{6}$
8	$(2)_{4}$ OR $(1)_{3}$
9	$(8)_{9}$
10	$(8)_{10}$
11	8_{9} OR 8_{10}
12	11_{12} AND [9 OR $(8)_{10}$]
13	11_{13} OR 11_{12}
14	$(8)_{9}$ OR $(8)_{10}$
15	2_{15}

Macro Task Fusion for Static Task Scheduling

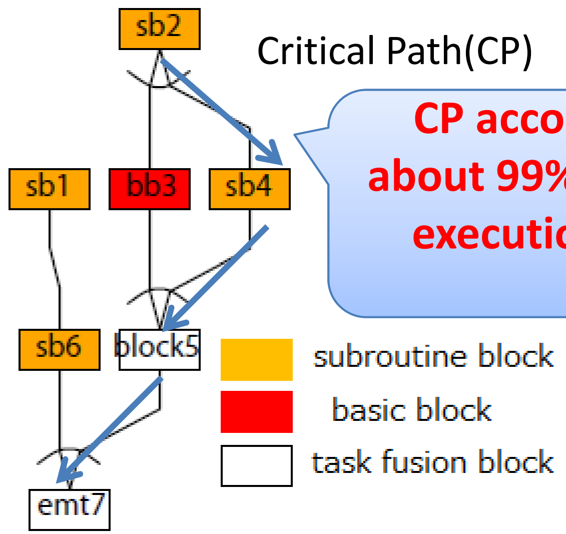


MFG of sample program before maro task fusion

MFG of sample program after macro task fusion

MTG of sample program after macro task fusion

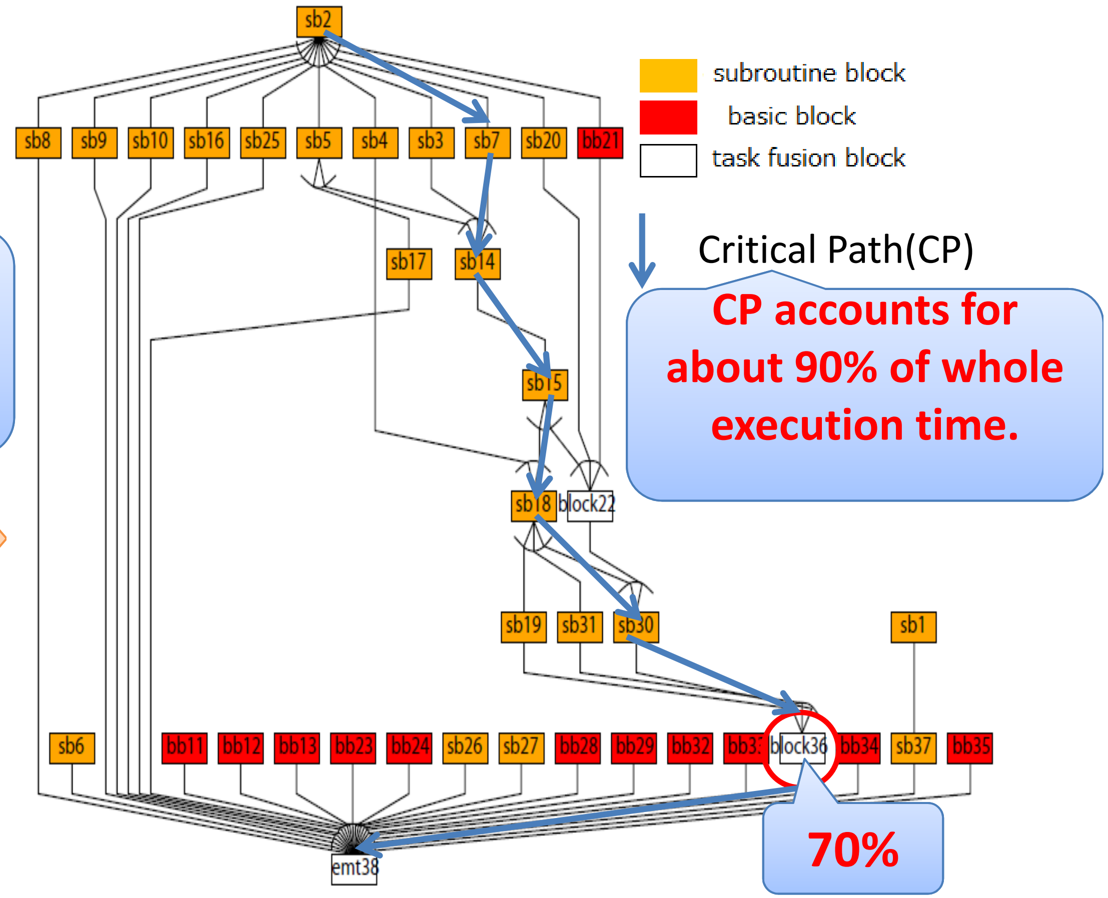
MTG of Crankshaft Program Using Inline Expansion



Critical Path(CP)

CP accounts for about 99% of whole execution time.

MTG of crankshaft program before restructuring

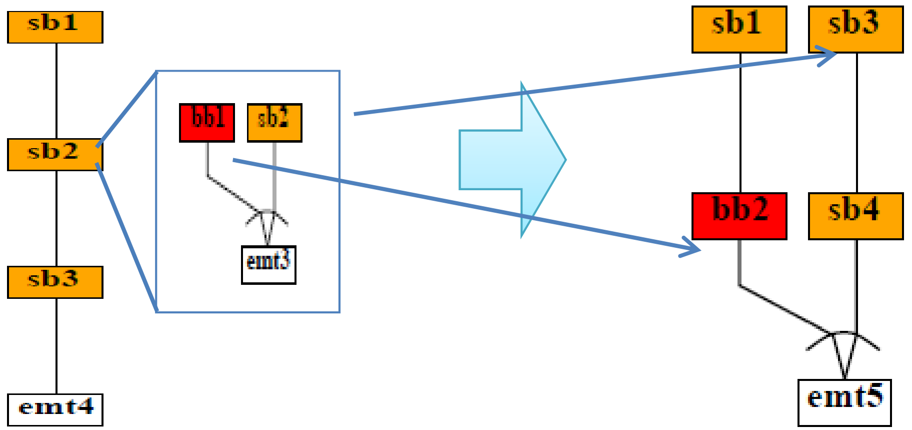


Not enough coarse grain parallelism yet!

3.1 Restructuring : Inline Expansion

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

Improves coarse grain parallelism



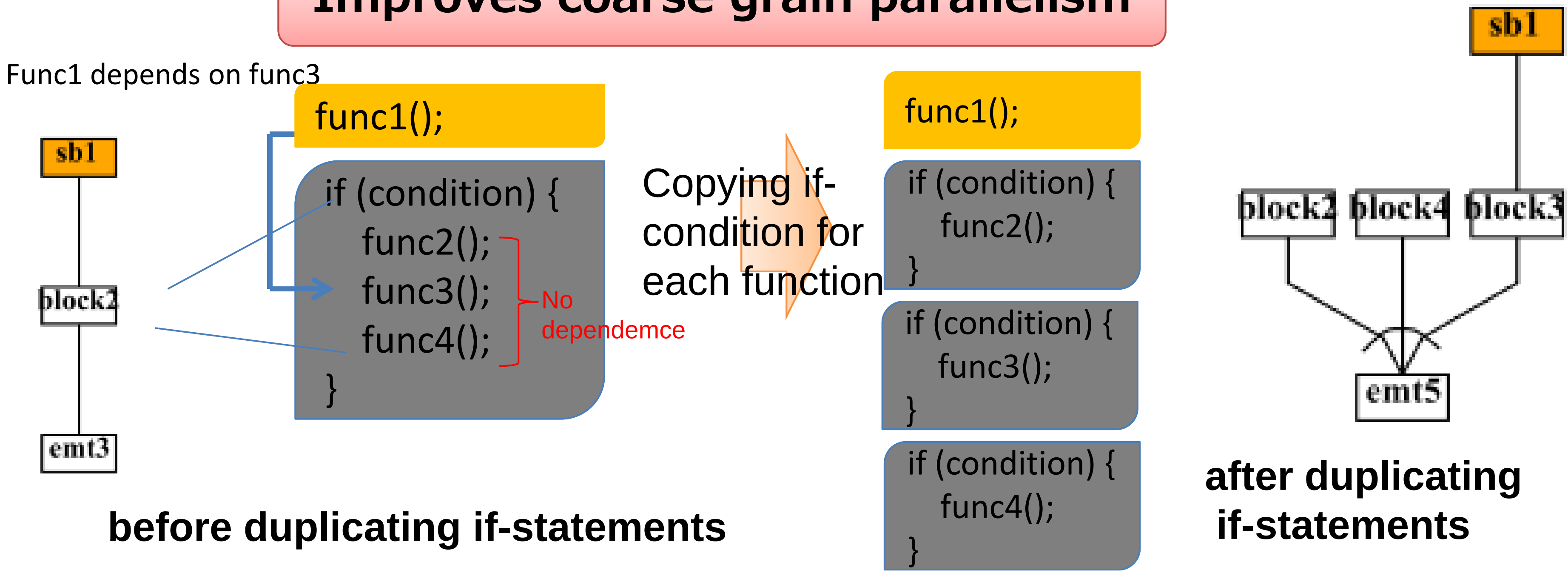
MTG before inline expansion

MTG after inline expansion

3.2 Restructuring: Duplicating If-statements

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

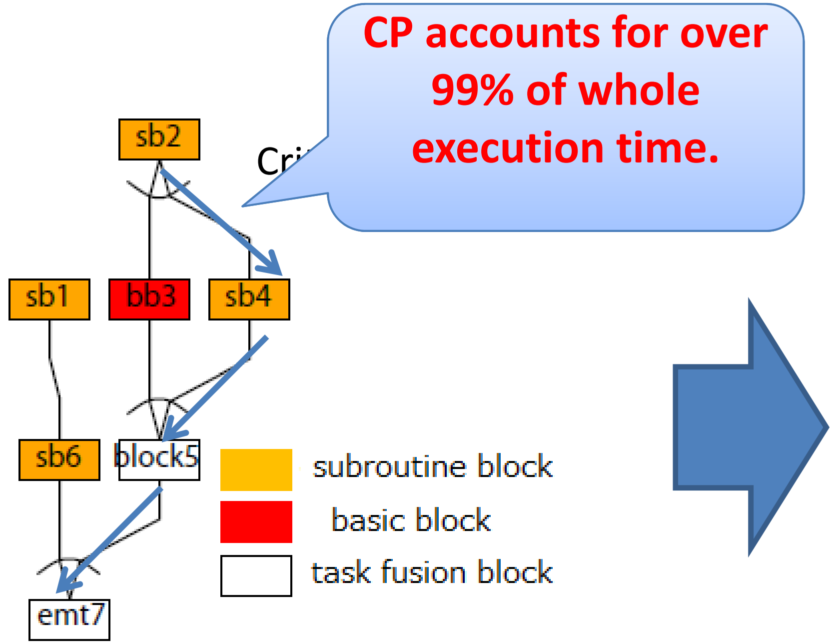
Improves coarse grain parallelism



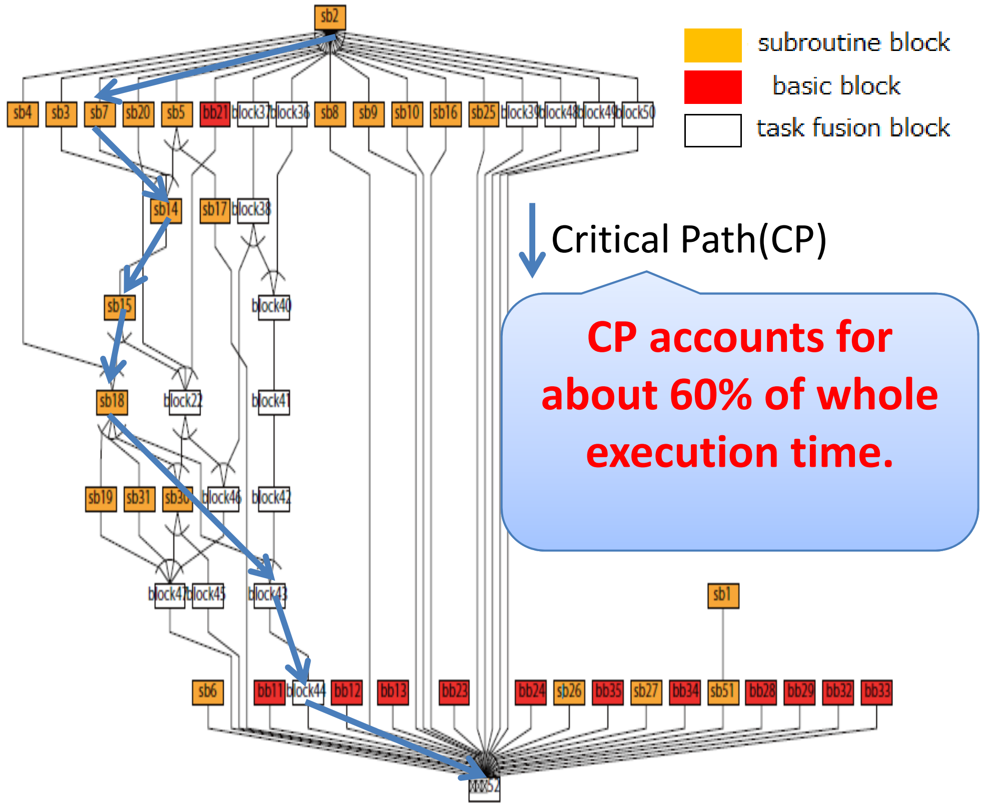
before duplicating if-statements

after duplicating if-statements

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



MTG of crankshaft program before restructuring

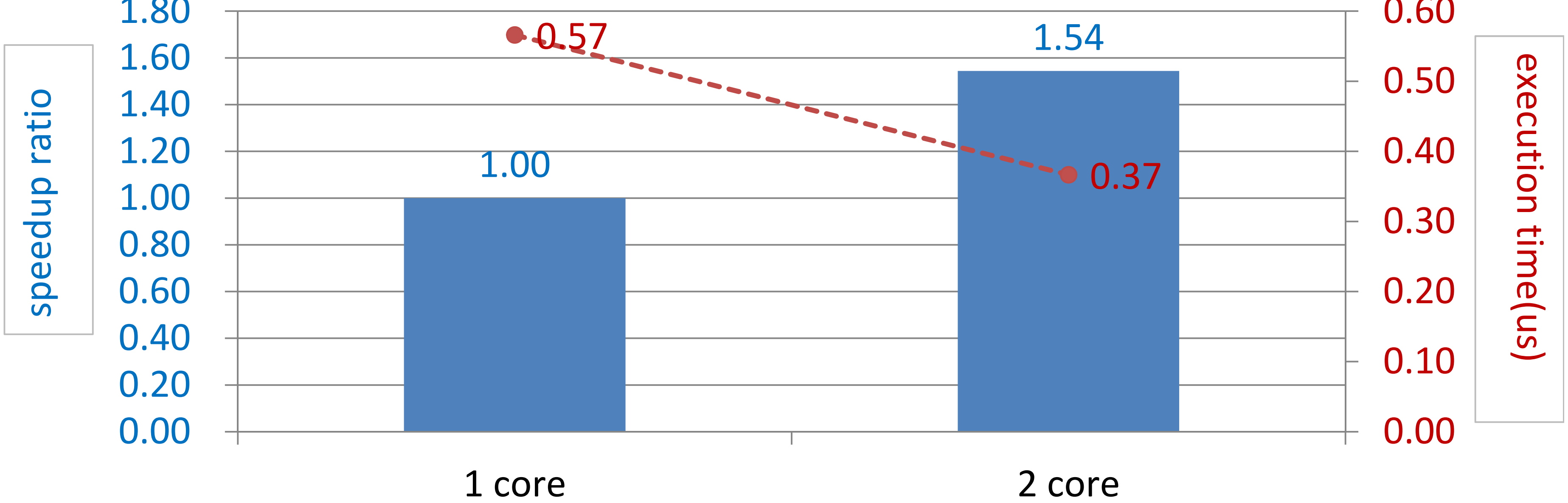


MTG of crankshaft program after restructuring

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

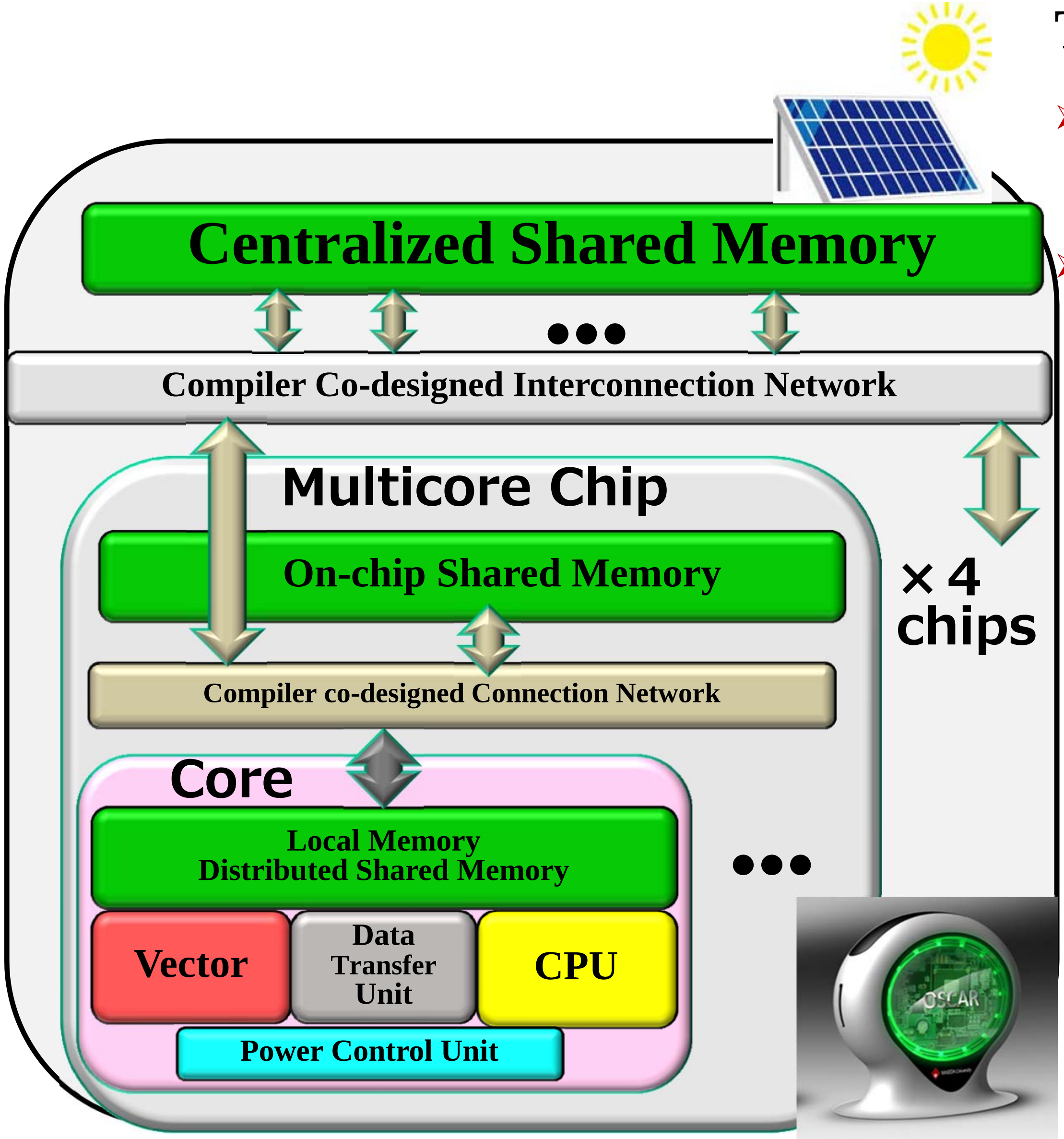
Successfully increased coarse grain parallelism

Evaluation of Crankshaft Program with Multi-core Processors



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

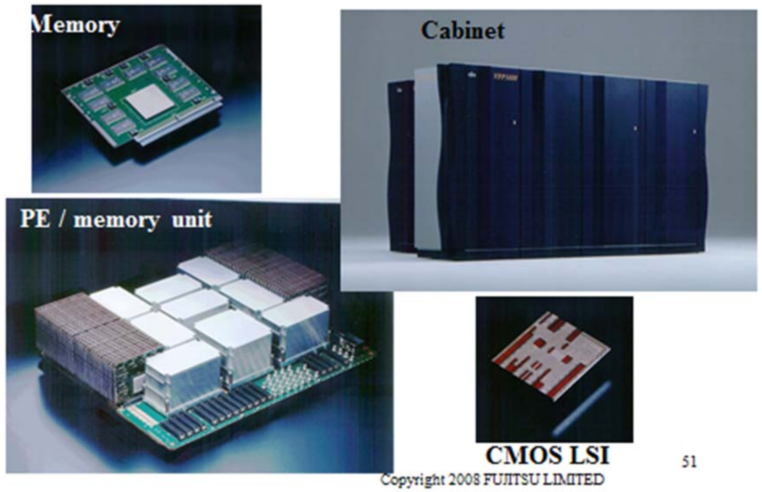
OSCAR Vector Multicore and Compiler for Embedded to Severs with OSCAR Technology



Target:

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

Fujitsu Vector Multiprocessor Supercomputer VPP700



Summary

- **This talk has introduced the automatic parallelization method of automobile engine control programs implemented in OSCAR parallelizing compiler.**
- **It includes the conversion of a macrotask graph with control dependence edges to a macrotask graph with only data dependence edges to apply static task scheduling to avoid the relatively large overhead of dynamic scheduling considering fine grain tasks composed of basic blocks.**
- **It also includes the inline expansion and the duplication of if statements to reduce the critical path length of the macrotask graph to increase the parallelism.**
- **The developed method can be applied both for hand-written codes and model based designed codes.**
- **The compiler can support various multicores such as 1.95 times speedup for a basic control code on two cores V850 multicore and 1.54 times speedup for a Crankshaft Program on two cores of SH based RPX.**
- **The compiler will be available from OSCAR Technology.**