Panel
Software Challenges in Multi-Core Chip Era

笠原博徳
Hironori Kasahara
Professor, Department of Computer Science
Director, Advanced Chip-Multiprocessor Research Institute
早稲田大学
Waseda University
http://www.kasahara.cs.waseda.ac.jp

Prof. Gao's Questions (1/3)

Q1: From software angle: do you expect that the chip level multi-core architectures will soon be converged to 1-2 style

(like single-core microprocessors did in the history: e.g. VLIW vs. superscalar) ?

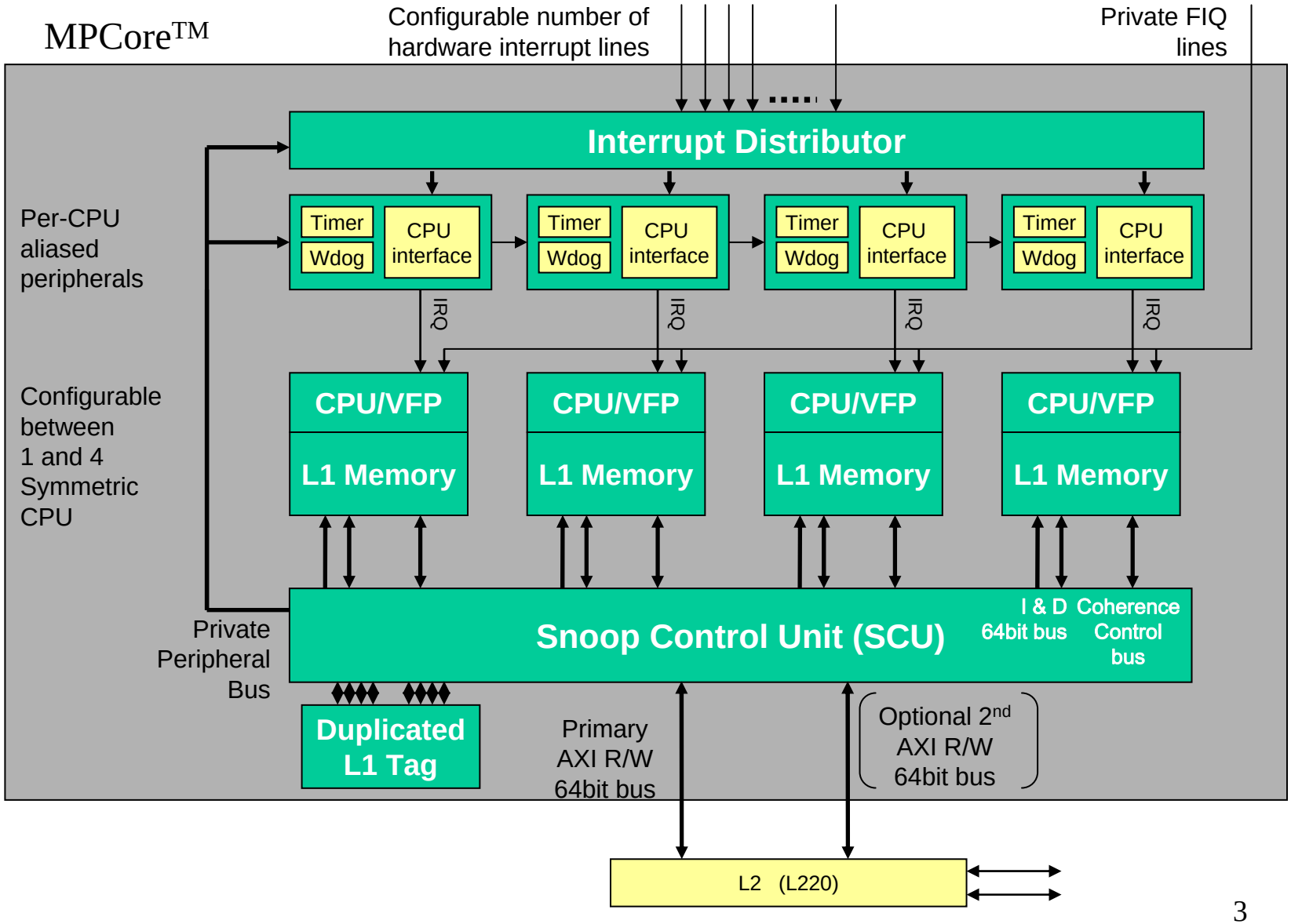
If not, why not ? If yes, what would be the 1-2 style in your assumptions from software angle ?

Answer:

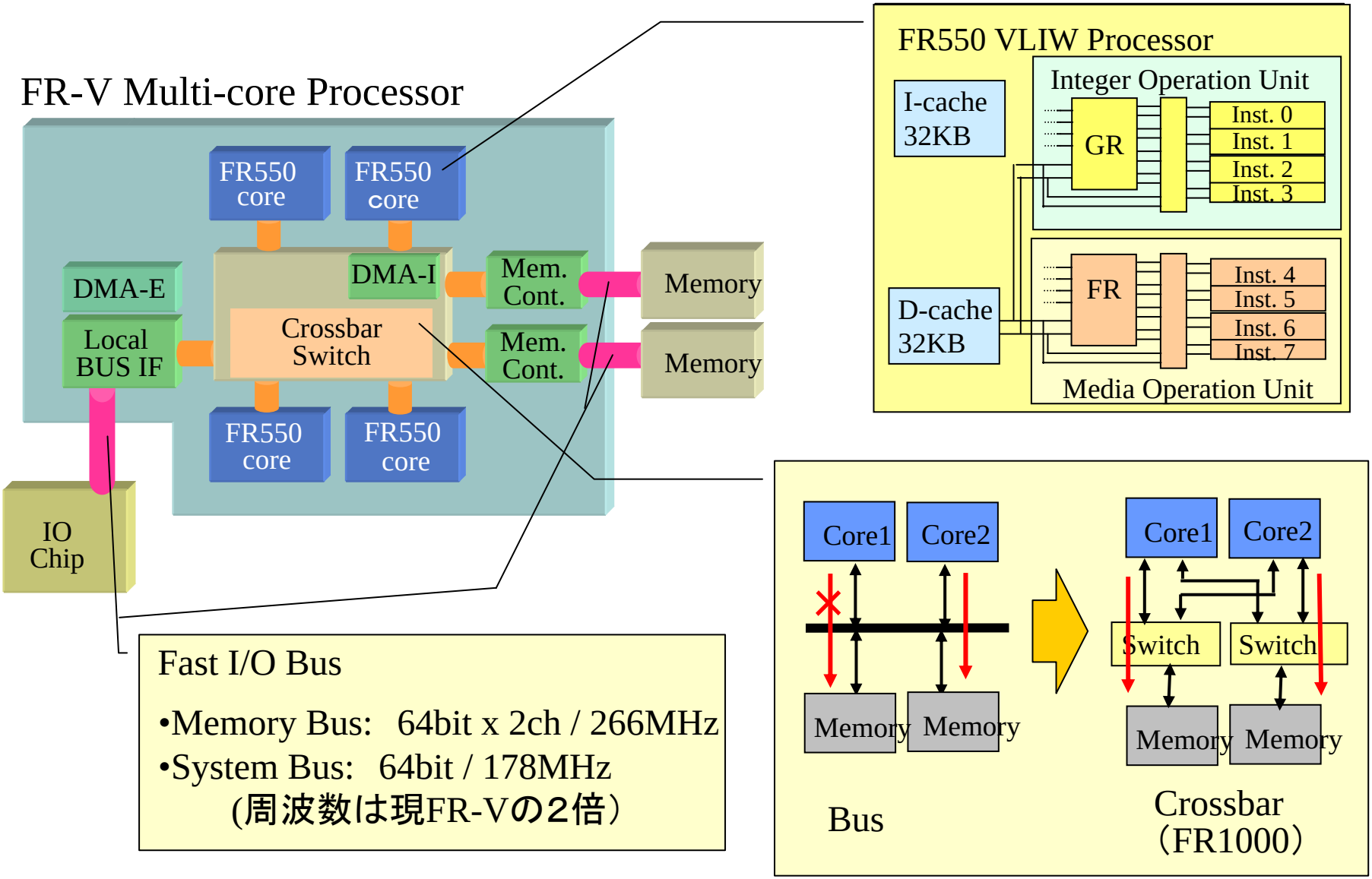
Yes, I think multi-core architecture will converge to SMP for small non-real time systems and OSCAR like software and hardware collaborative architecture with local, distributed shared and centralized memories with DMA controller for real-time embedded systems.

MPCore™

ARM and NEC Collaboration



Fujitsu FR-1000Multicore Processor

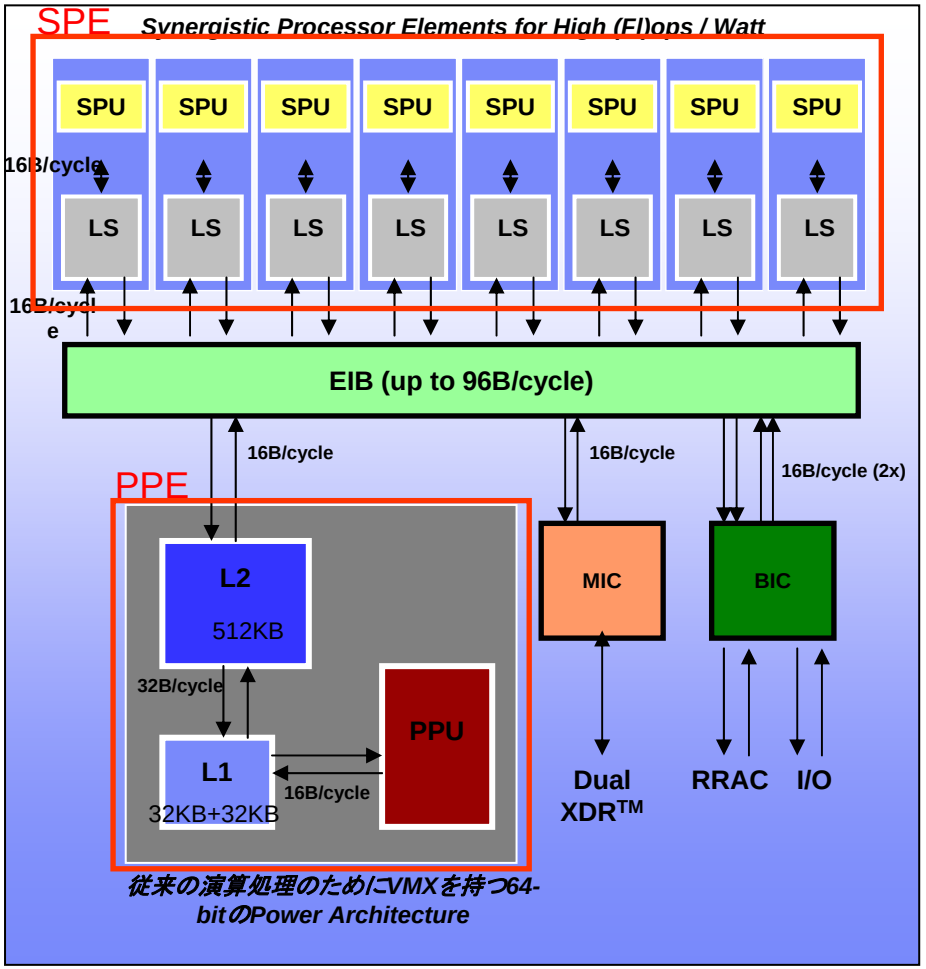


Fast I/O Bus

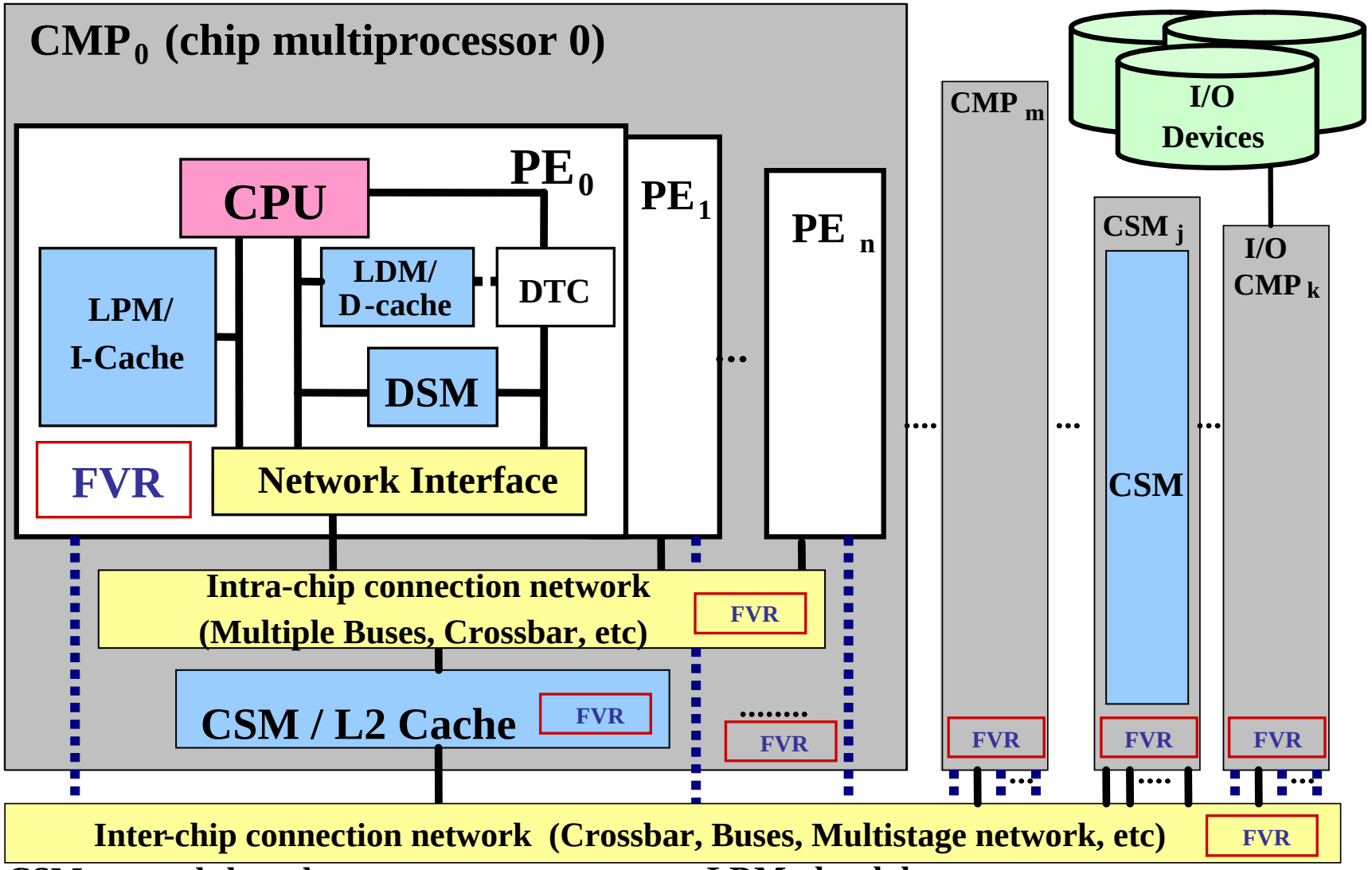
- Memory Bus:　64bit x 2ch / 266MHz
- System Bus:　64bit / 178MHz
  (周波数は現FR-Vの2倍)

CELL Processor Overview

- **Power Processor Element (PPE)**
  - **PowerCore processes OS and Control tasks**
  - **2-way Multi-threaded**
- **Synergistic Processor Element (SPE)**
  - **8 SPE offers high performance**
  - **Dual issue RISC Architecture**
  - **128bit SIMD(16-way)**
  - **128 x 128bit General Registers**
  - **256KB Local Store**
  - **DedicatedDMA engines**



OSCAR Multi-Core Architecture



CSM: central shared mem.

DSM: distributed shared mem.

DTC: Data Transfer Controller

LDM : local data mem.

LPM : local program mem.

FVR: frequency / voltage control register

Prof. Gao's Questions (2/3)

Q2: Automatic compilation for parallel machine did not succeed in general - as proven in the past history. What do you expect this time for multi-core revolution ?
Will it succeed this time ?
If yes, why do you think we may succeed this time ?
If not, what other software technology (if any) you predict may have a chance succeeding ?

Answer:

Yes, I think we will succeed this time because we have continued the compiler research for twenty years and finally could develop multigrain parallelization, local memory management, data transfer control and frequency, voltage and power-off control.

I believe the long long time efforts and the real needs for the compiler will change the situation.

Prof. Gao's Questions (3/3)

Q3: What is your favorite parallel programming model (if any) ? Why ?

Do you believe that the so-called general purpose parallel programming models should be the way to go - for the new multi-core era ? Why or why not ?

Answer:

My favorite model is OpenMP because vendors have supported (just we use Parallel, Section, Flush, Critical), especially section directives for coarse grain task parallel processing. Also, we will add some additional directives for OSCAR type architecture with local, distributed shared, and on-chip and off-chip centralized shared memories, DMA controllers and power control functions.

API and Parallelizing Compiler in METI/NEDO Advanced Multicore for Realtime Consumer Electronics Project