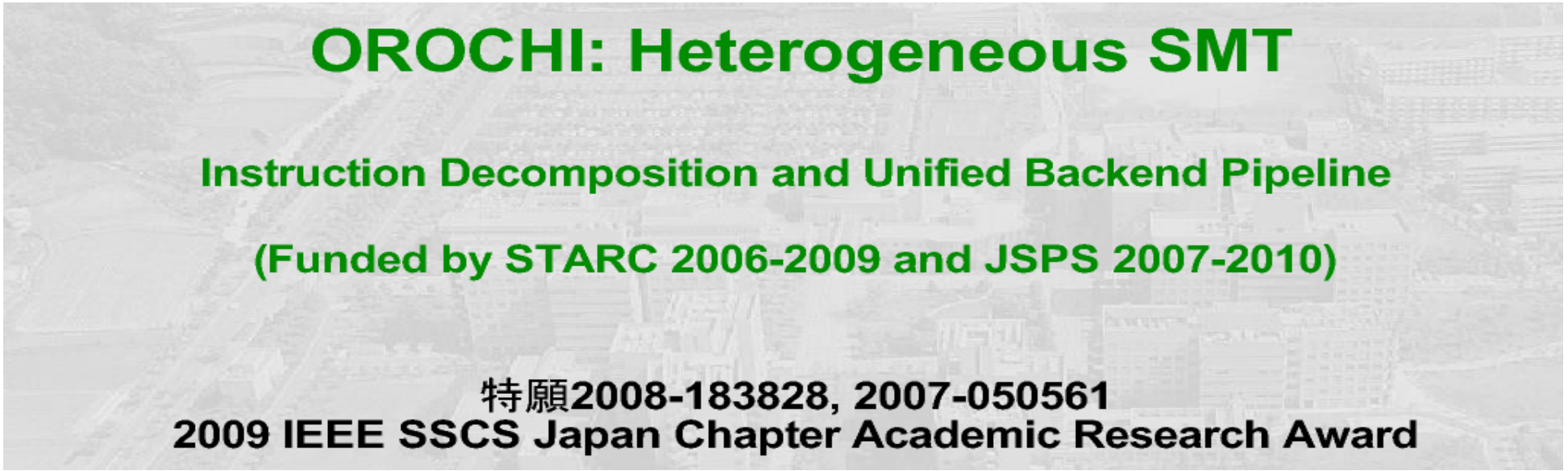
# 高性能計算基盤

## 自習 APDX09:Heterogenious Multithreading

http://archlab.naist.jp/Lectures/ARCH/x09/apdx09j.pdf

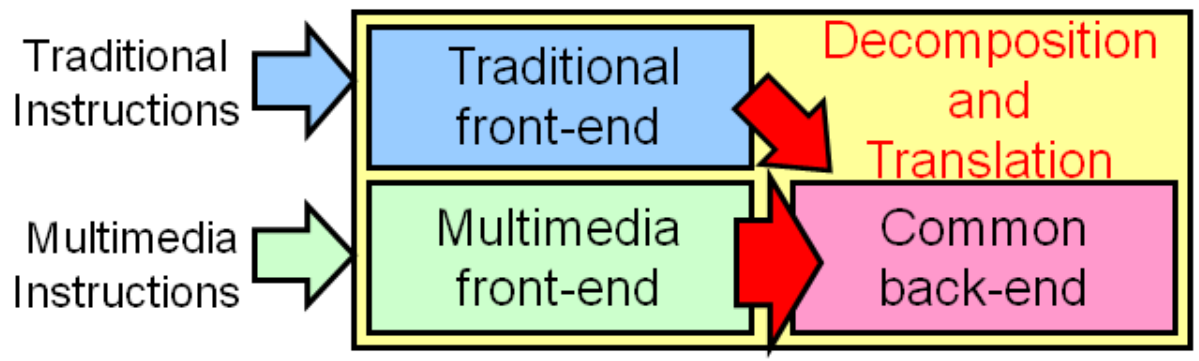





ナレータ VOICEVOX:もち子(cv 明日葉よもぎ)

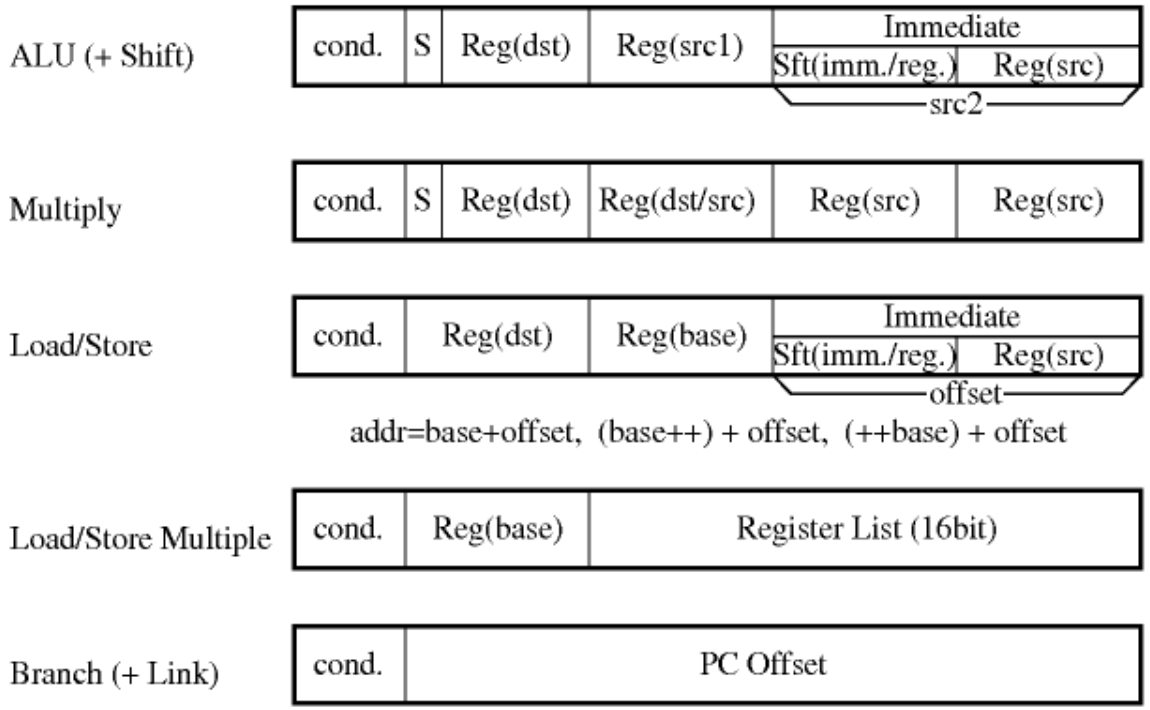
# Outline of OROCHI

- Heterogeneous SMT processor
  - For Small chip area and low power consumption
  - VLIW-based processor for execution of multimedia instruction set
  - Traditional instructions decomposed and translated into internal instructions




[1] Hajime Shimada, Takashi Shimada, Takekazu Tabata, Tomoya Kojima, Kenji Kise, Yasuhiko Nakashima, Toshiaki Kitamura: "Outline of OROCHI: A Multiple Instruction Set executable SMT Processor", IWIA, pp. 110–117, 2007.
[2] Takashi Nakada, Yasuhiko Nakashima, Hajime Shimada, Kenji Kise, Toshiaki Kitamura: "OROCHI: A Multiple Instruction Set SMT Processor", HipHaC, pp. 1–8, 2008.

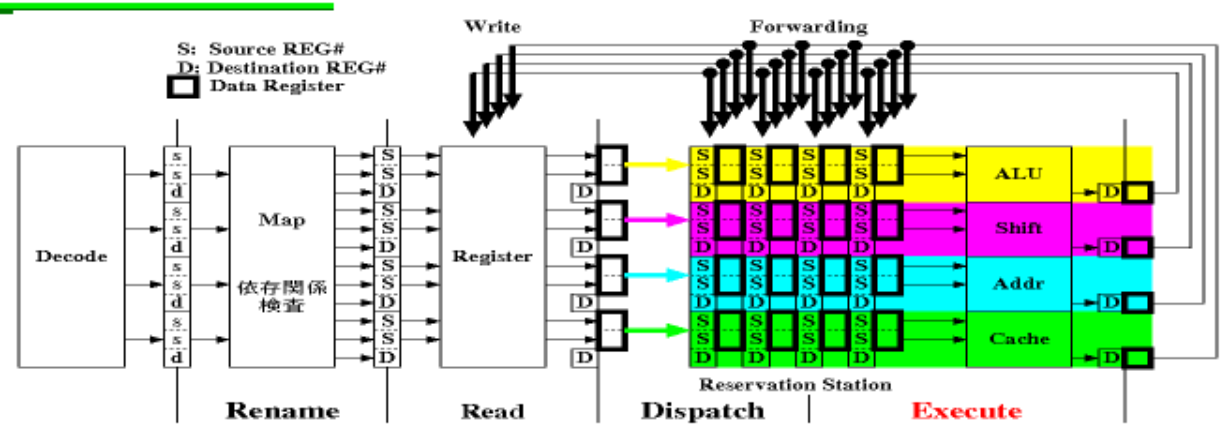
# Decomposition of ARM instructions



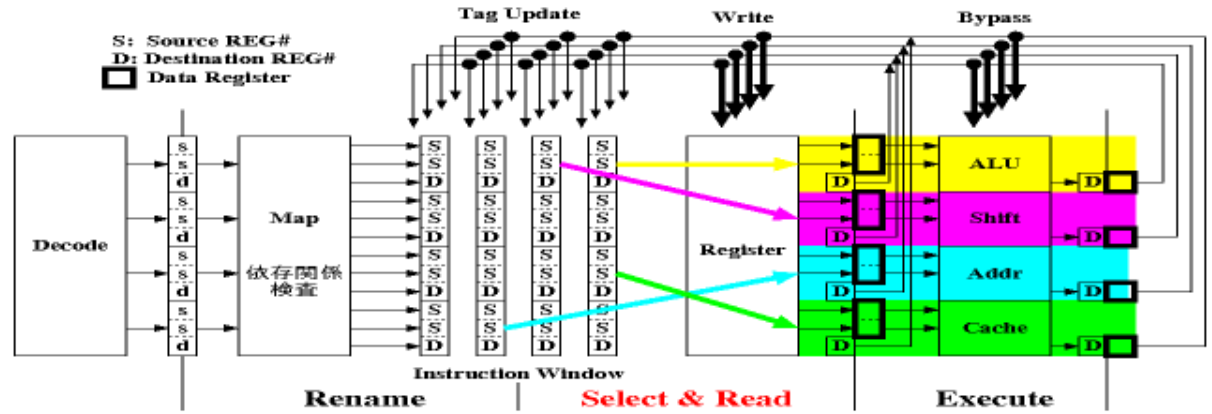

| ARM | Internal | # of Insn. |
|---|---|---|
| ALU(w/ shift) | SE | 2 |
| MULT(32*32+32 -> 32bit) | [Mm]*4 | 8 |
| ULT(32*32+64 -> 64bit) | [Mm]*8 m | 17 |
| MULT(Signed32*32 -> 64bit) | mm[Mm]*8 E m*3 | 22 |
| LD/ST(w/ shift,base-reg upd) | LSa | 3 |
| LD/ST(base-reg upd,w/ shift) | SaL | 3 |
| Multiple-LD/ST | aa[aL]*N a | 2*N+3 |
| Branch | B | 1 |

# Superscalar execution of internal instruction

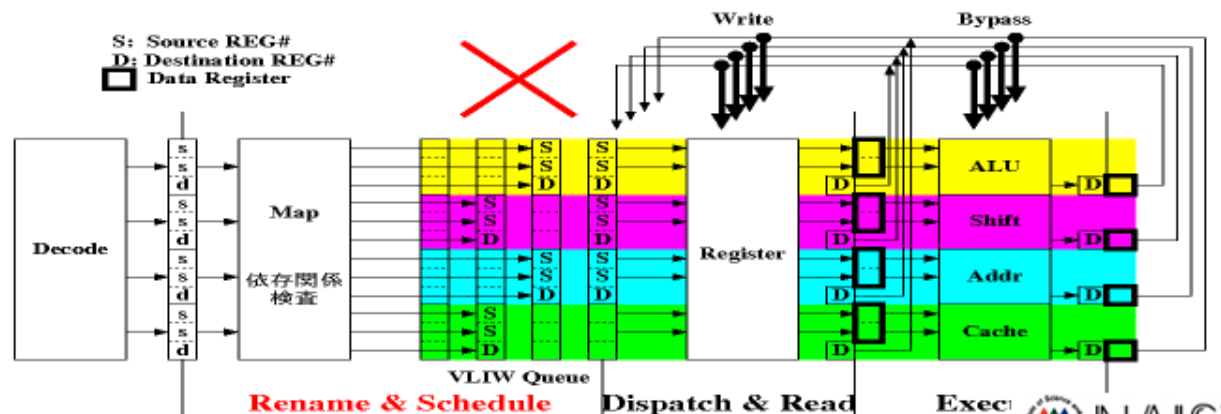
**Well Known Super Scalar Processor**




**Another Structure**



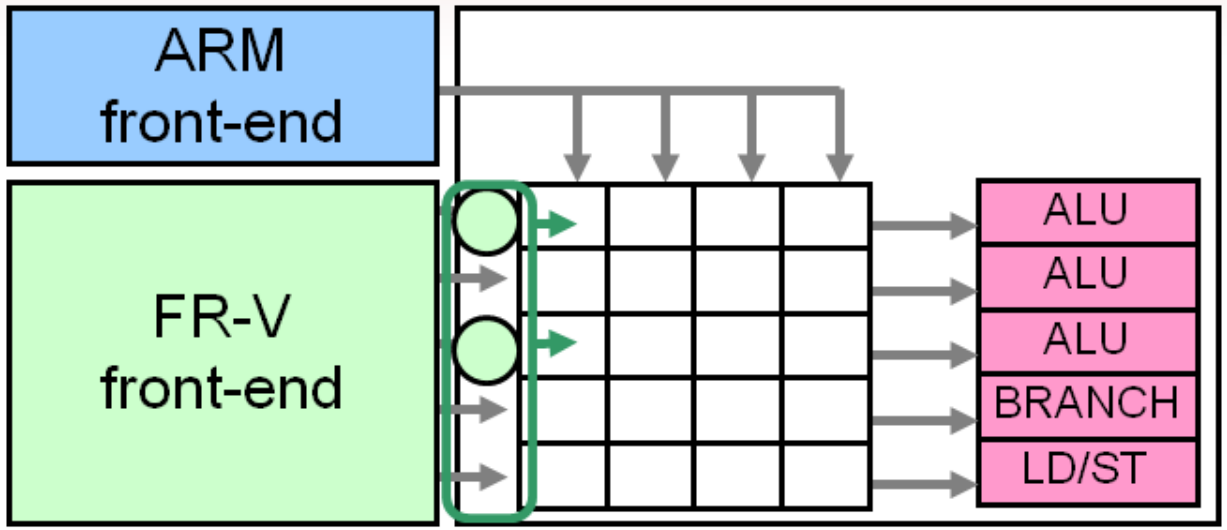

**Our Approach Schedules Internal Instructions into Empty Slots in VLIW**





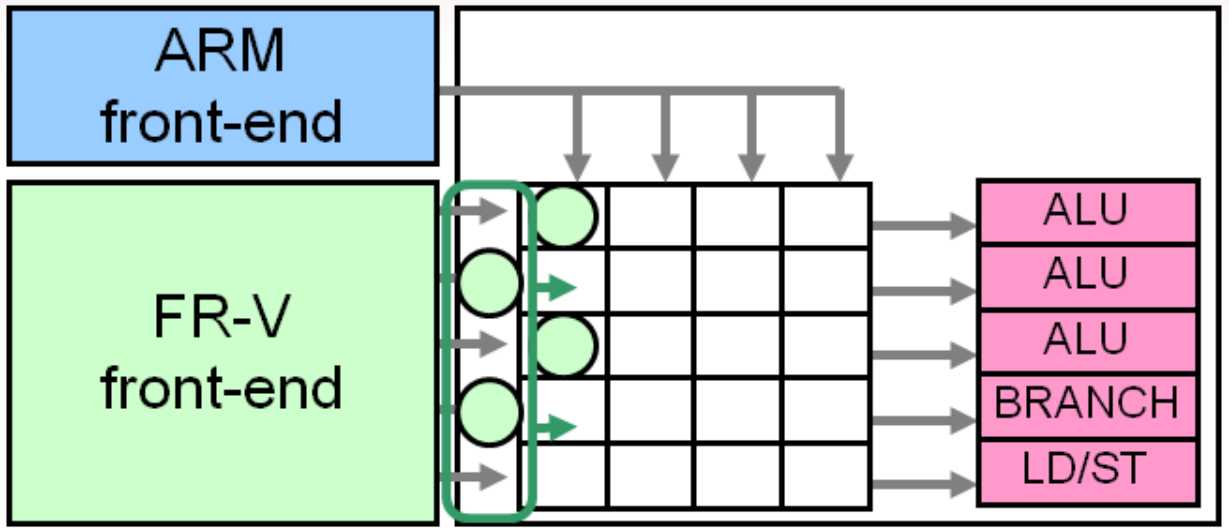

NAIST

# VLIW instruction queue



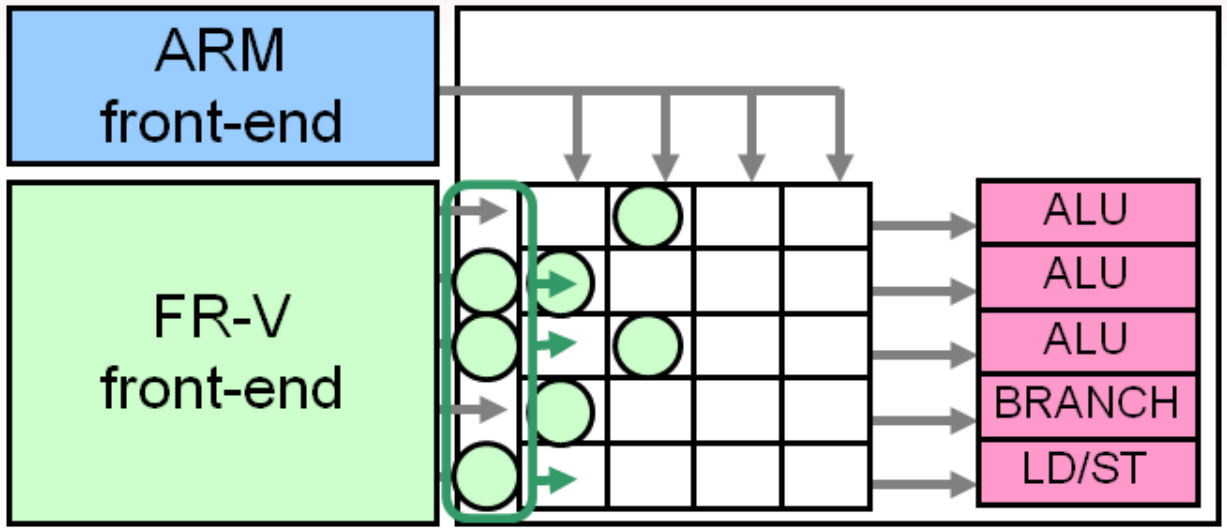

1. FR-V instructions queued and shifted

# VLIW instruction queue



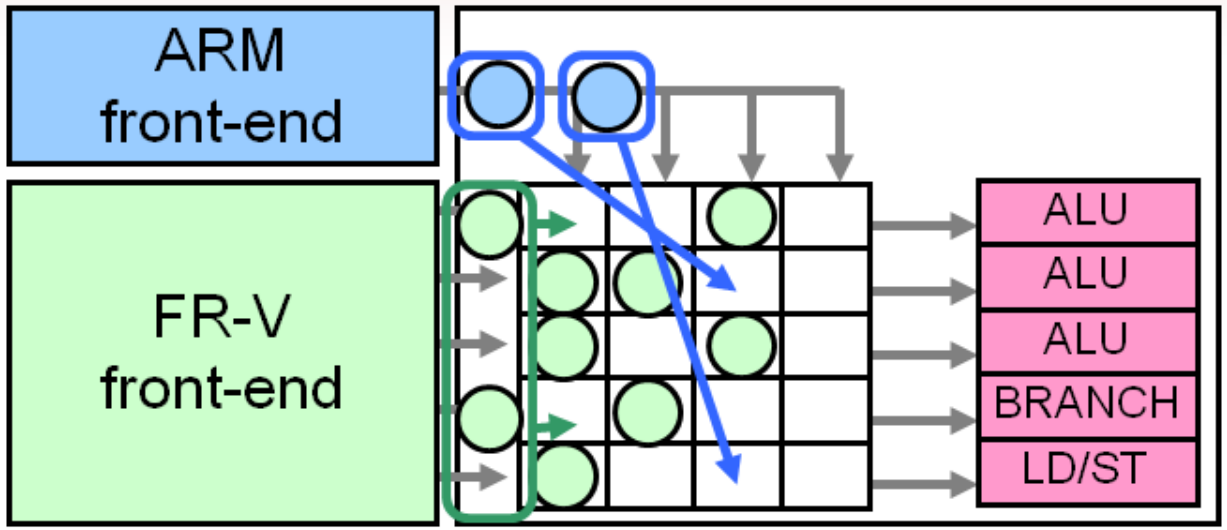

1. FR-V instructions queued and shifted

# VLIW instruction queue



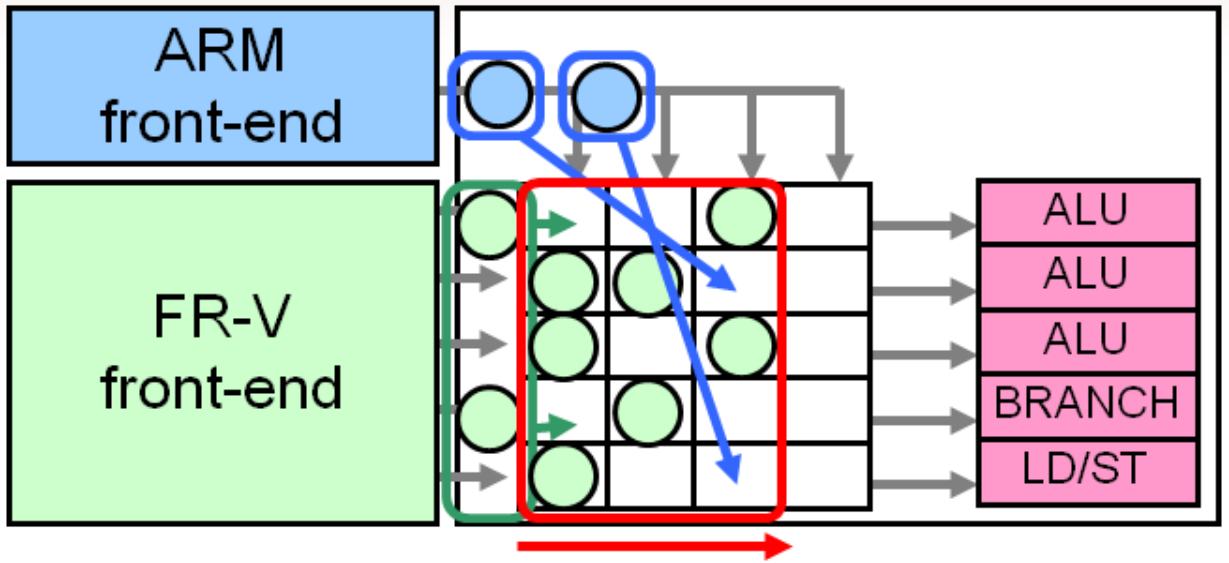

1. FR-V instructions queued and shifted

# VLIW instruction queue



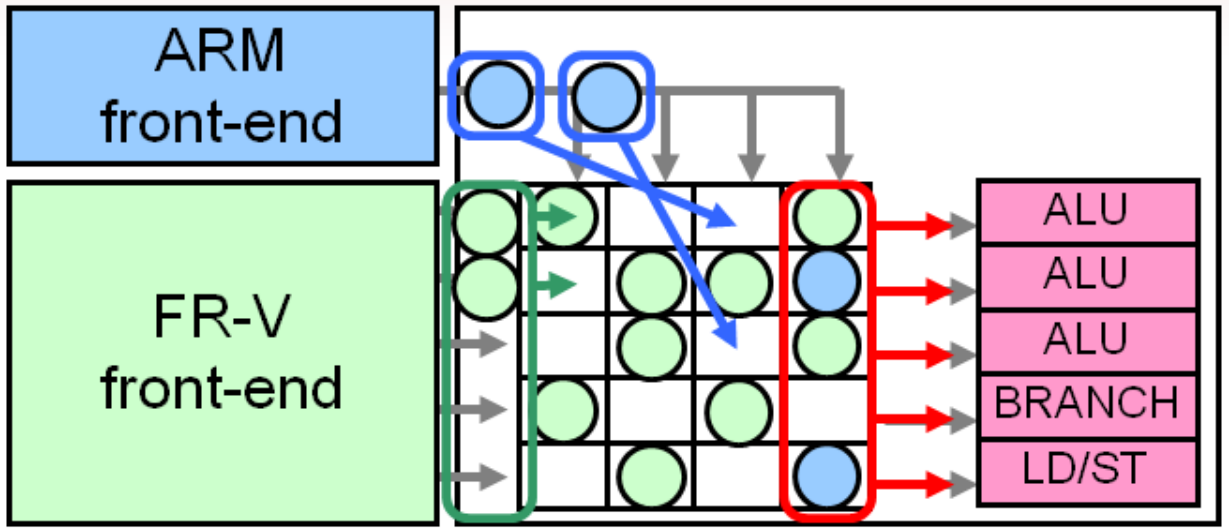

1. FR-V instructions queued and shifted
2. ARM instructions inserted to empty slot

# VLIW instruction queue



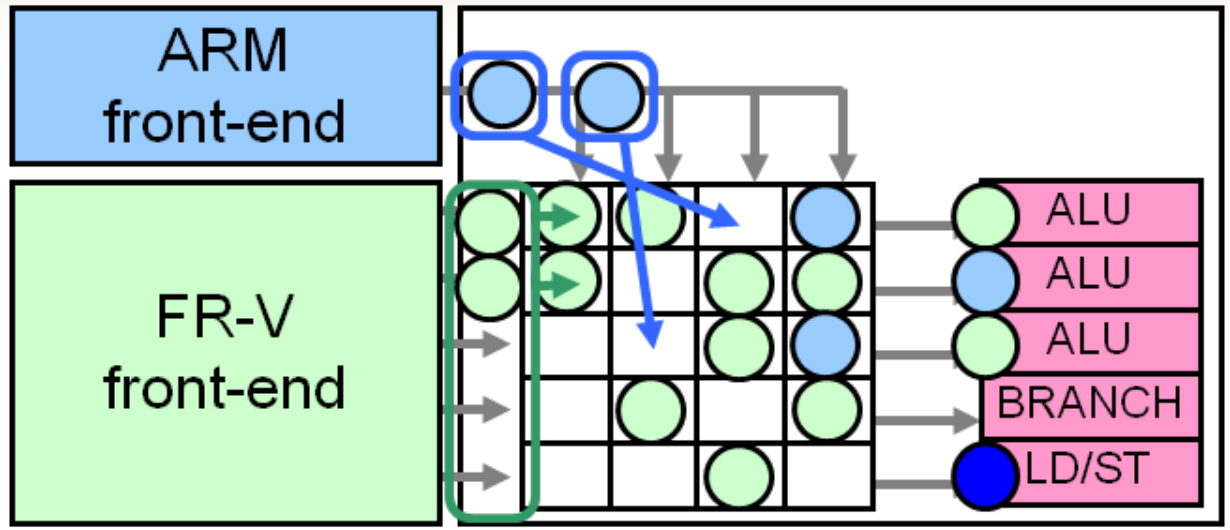

1. FR-V instructions queued and shifted
2. ARM instructions inserted to empty slot
3. ARM and FR-V instructions shifted toward right

# VLIW instruction queue



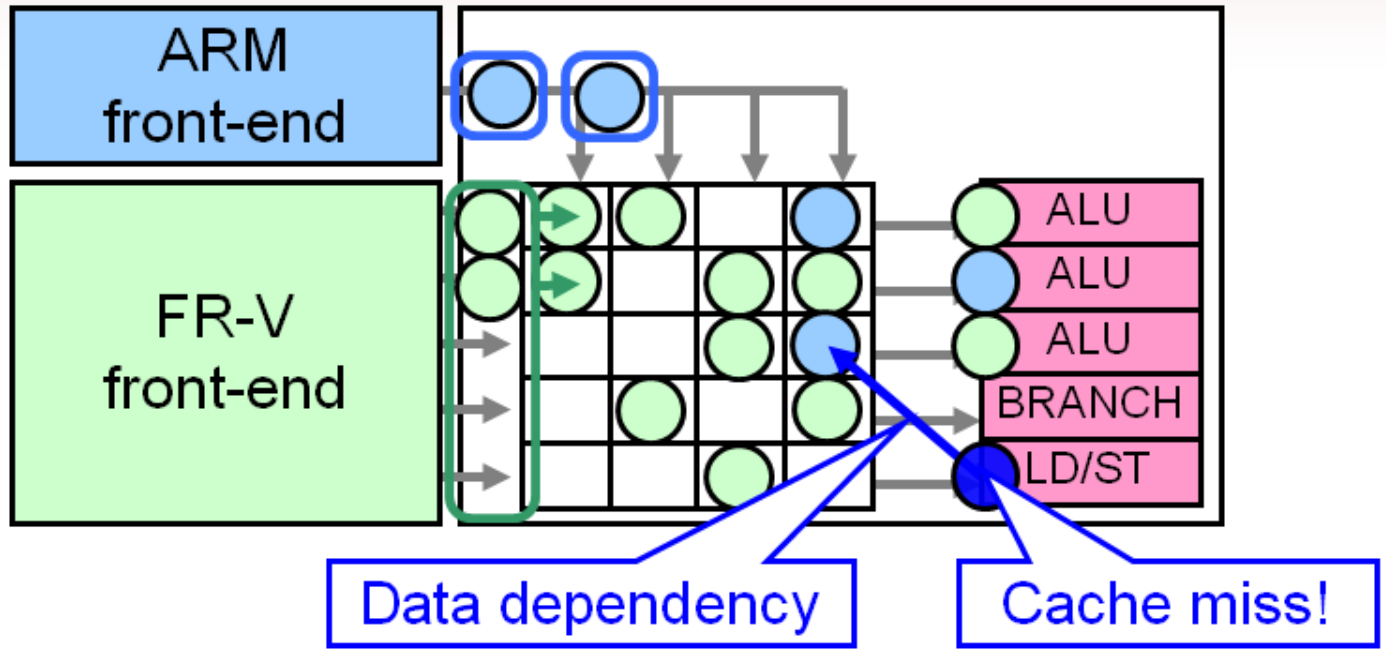

1. FR-V instructions queued and shifted
2. ARM instructions inserted to empty slot
3. ARM and FR-V instructions shifted toward right
4. ARM and FR-V instructions issued from right most column

# Problem



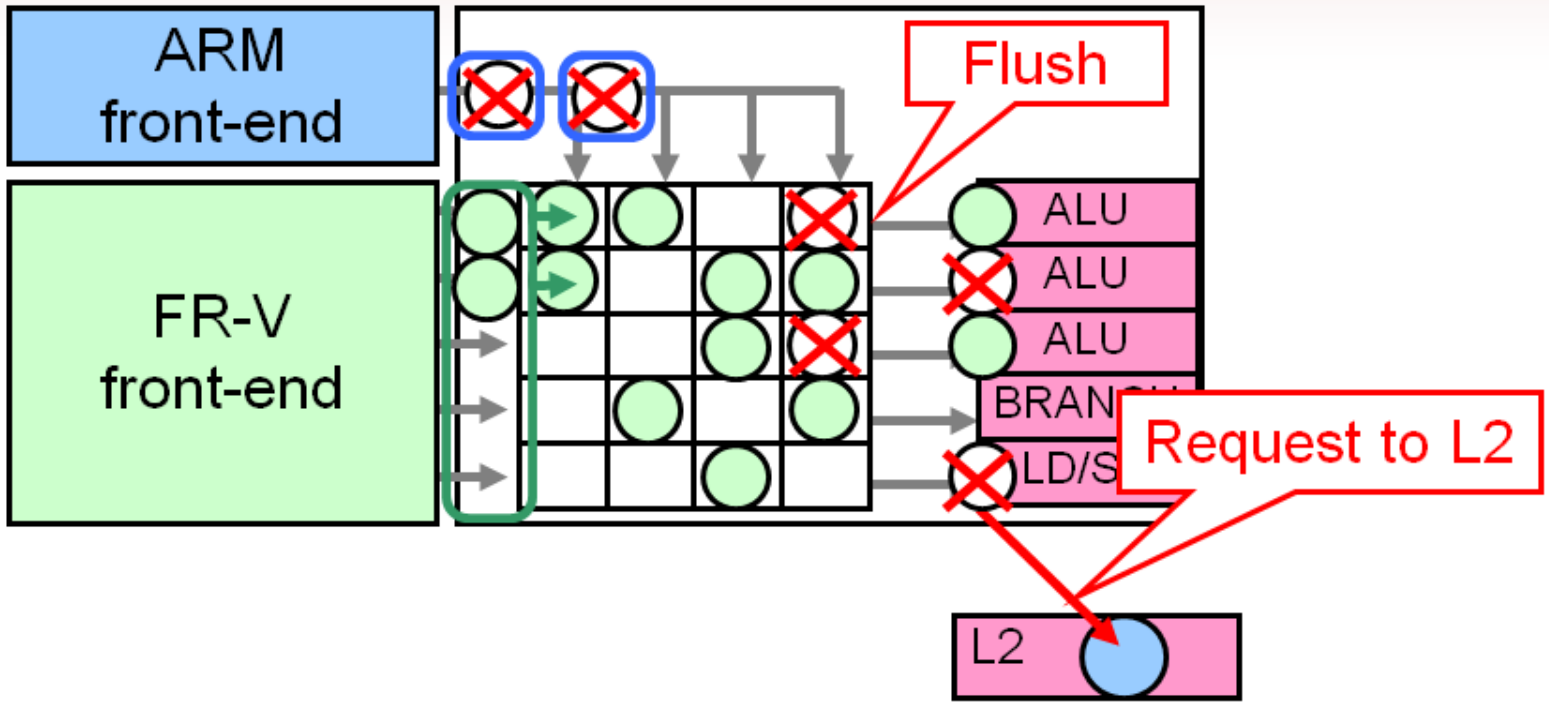

1. Non-priority load instruction issued

# Problem



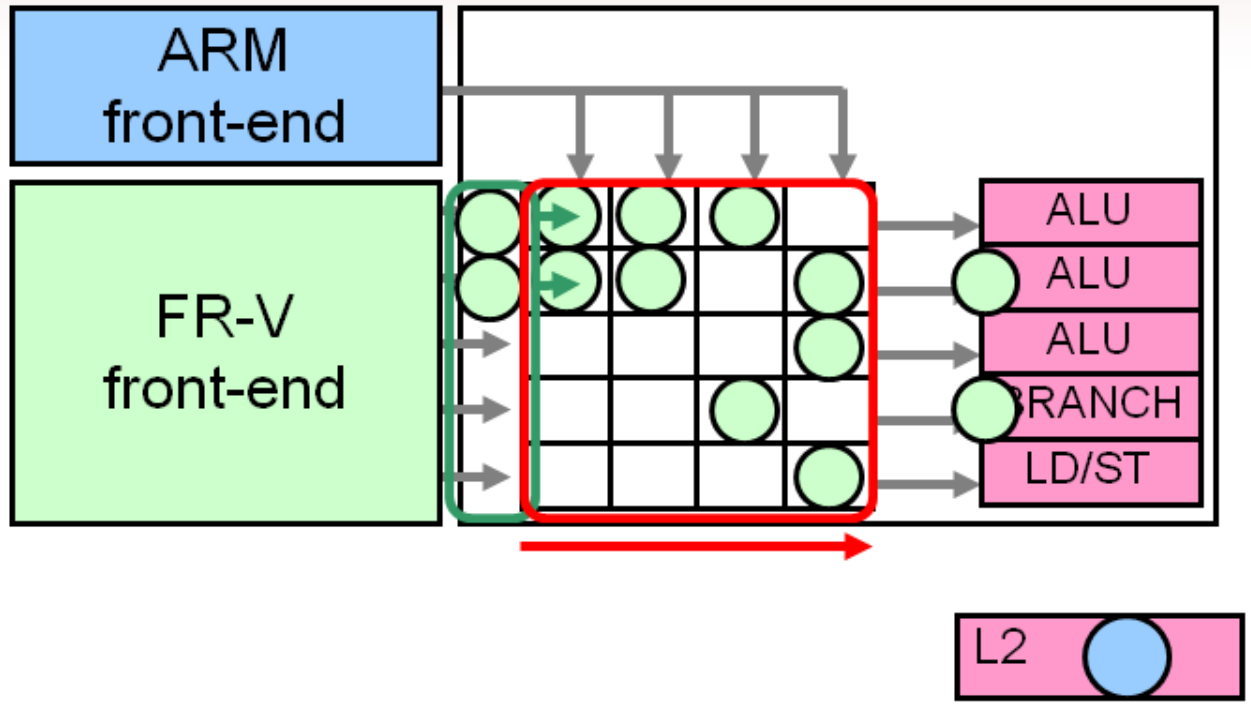

1. Non-priority load instruction issued
2. L1 cache miss occurs and subsequent instruction has data dependency

# Selective flush mechanism




1. When cache miss occurs, non-priority instructions are flushed, but request to L2 cache is enabled

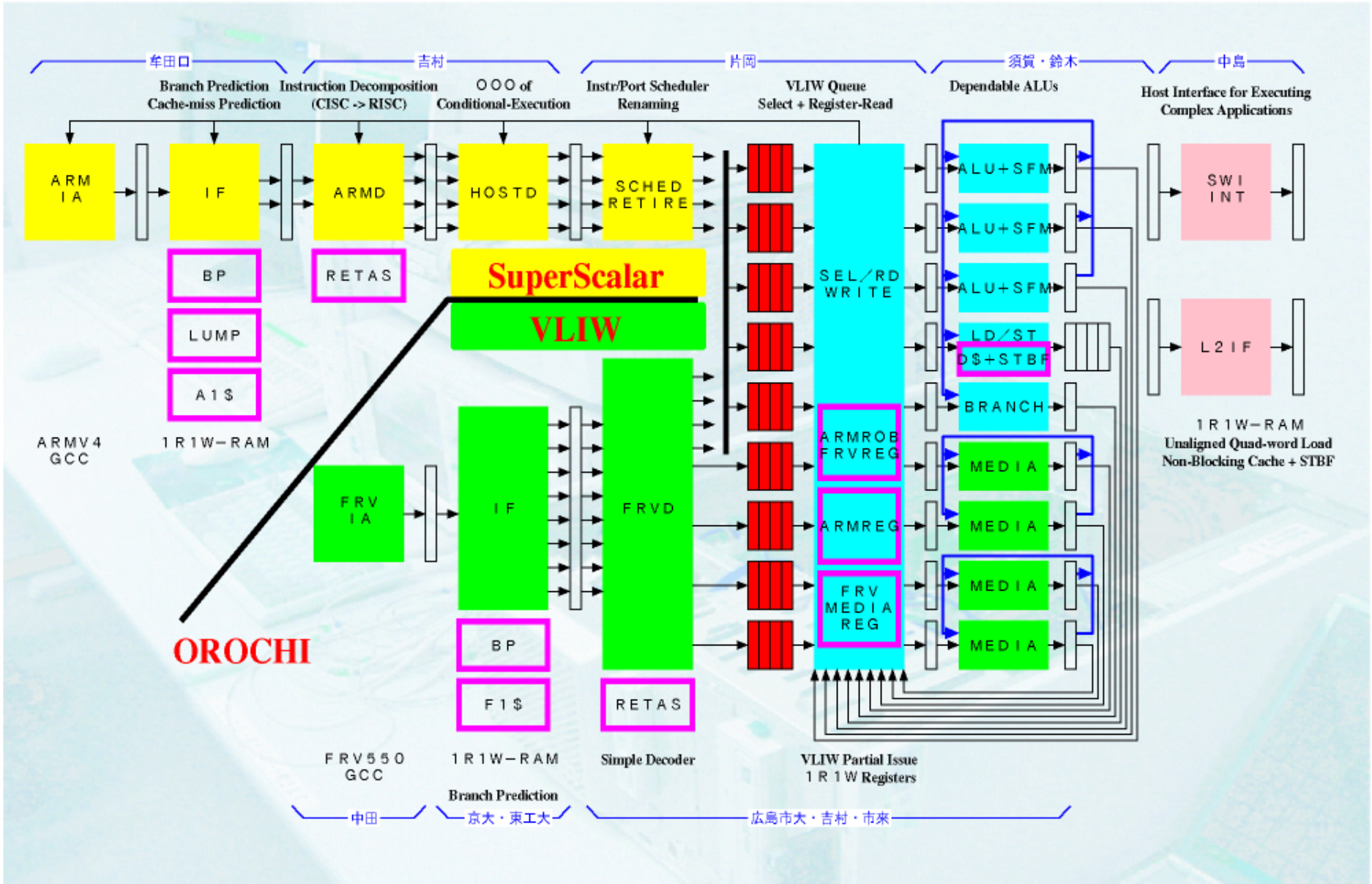
# Selective flush mechanism




1. When a cache miss occurs, non-priority instructions are flushed, but a request to L2 cache is enabled

→ Priority instructions can be executed

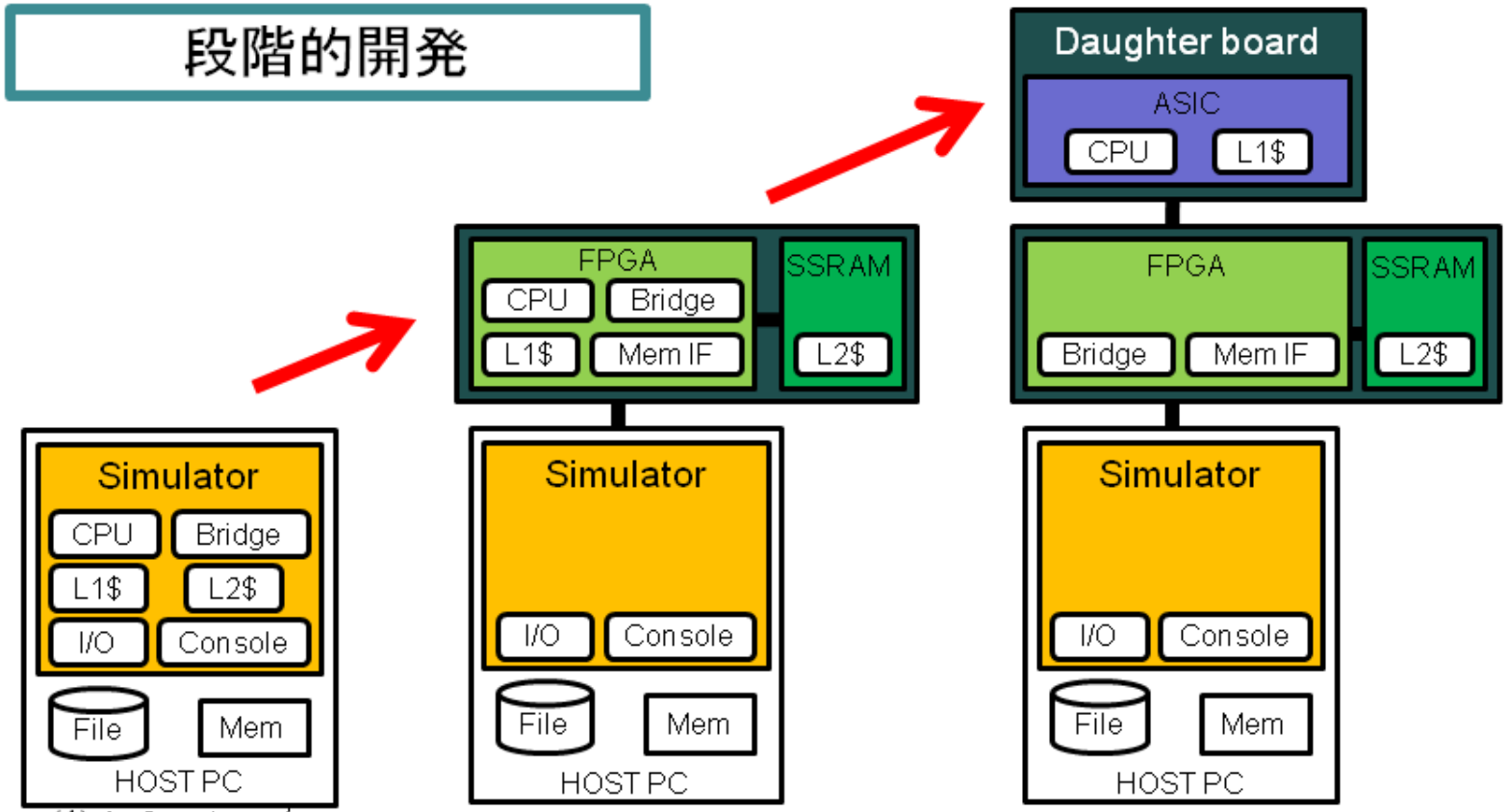
# Simultaneous Execution of ARM and VLIW



# Development with only 5 students and 2 years

## ASIC開発の順序

段階的開発

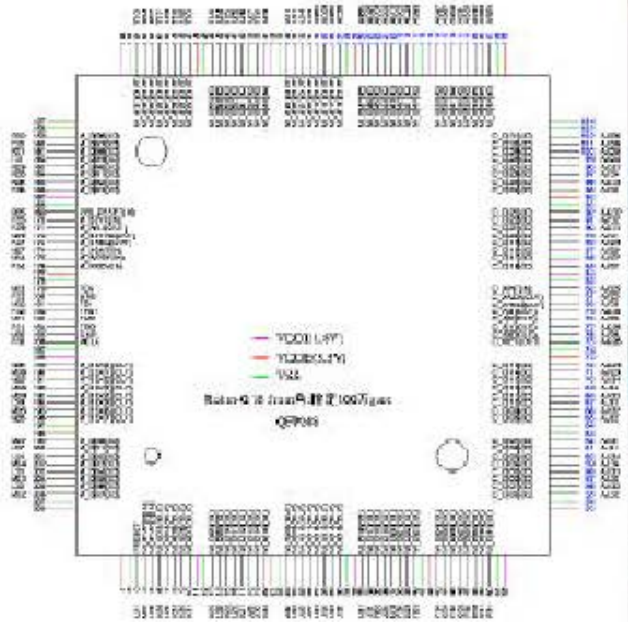


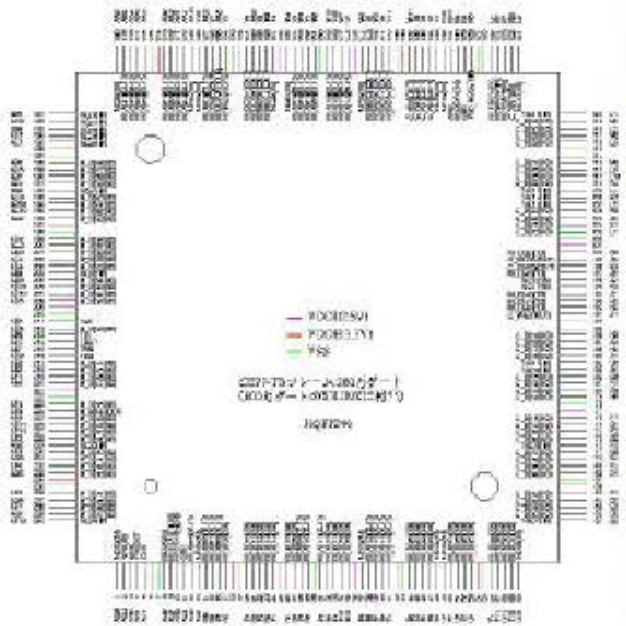

(1) シミュレータを用いた設計

(2) FPGAボードを用いたデバッグ

(3) ドータボードを用いた測定・検証

# Development with only 5 students and 2 years



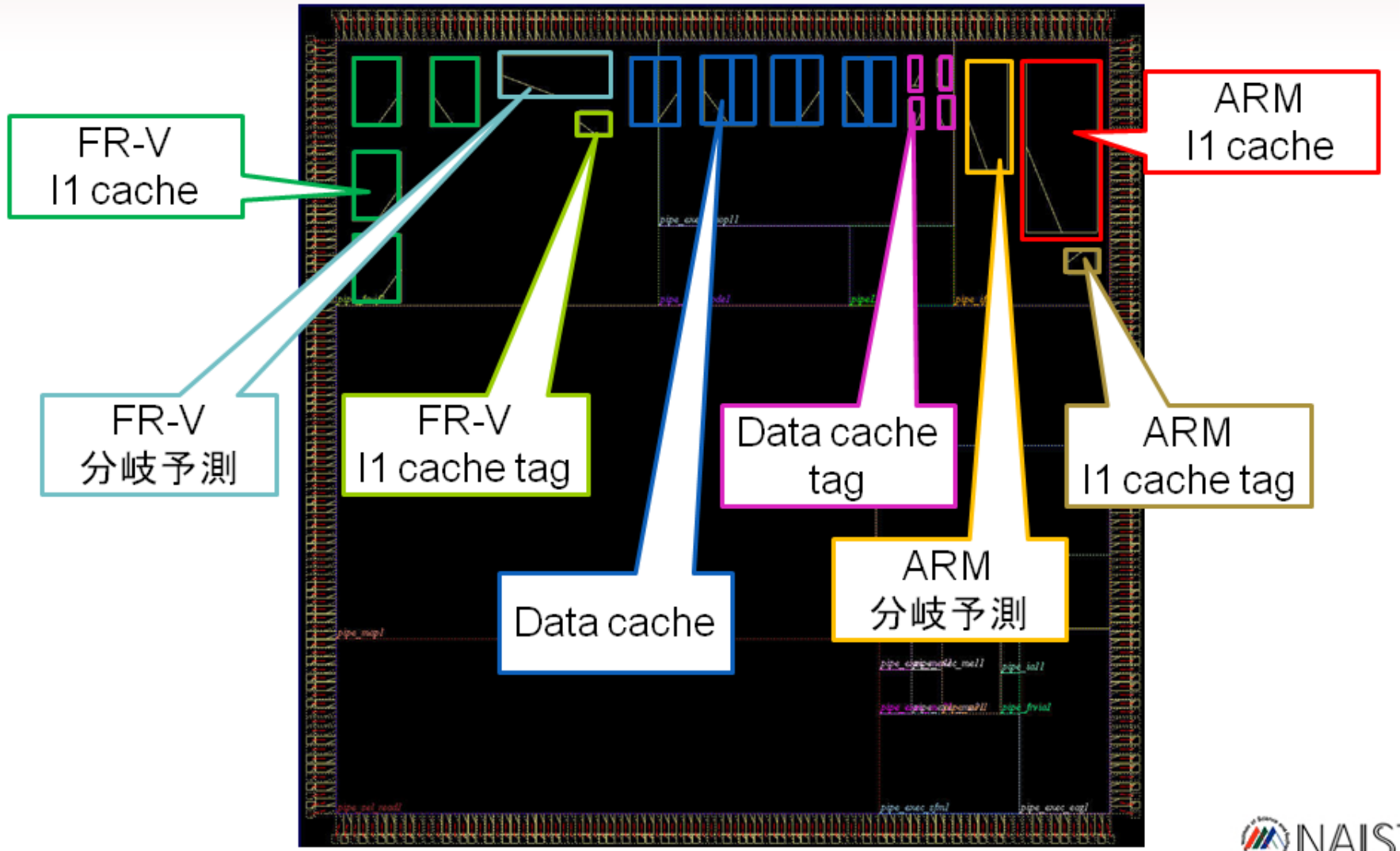




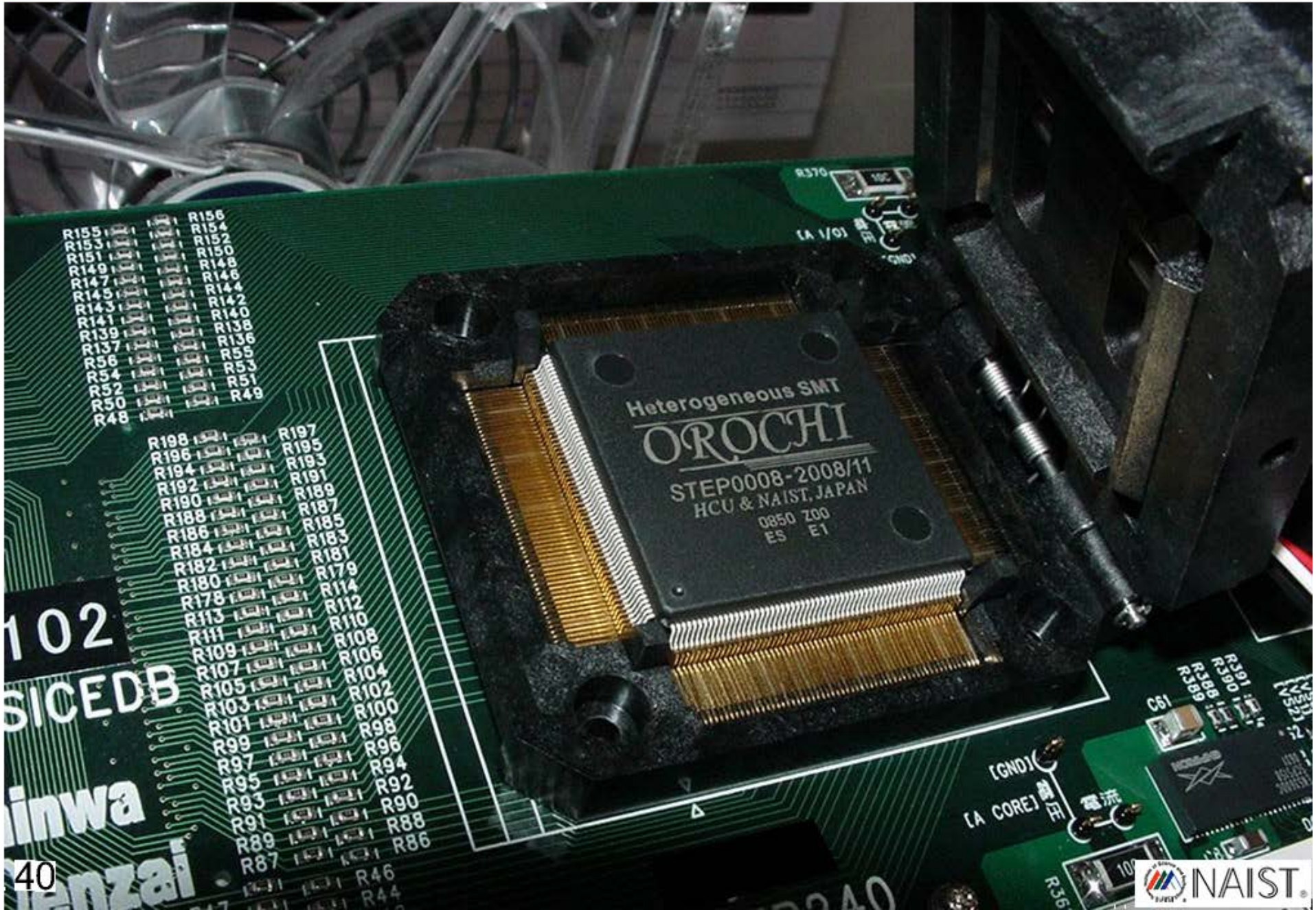
NAIST

# Development with only 5 students and 2 years

## レイアウト画像



# Development with only 5 students and 2 years



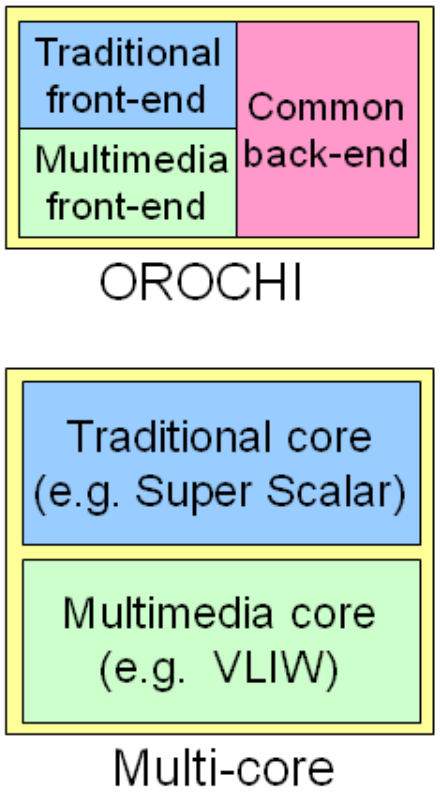



NAIST

# Implementation of OROCHI

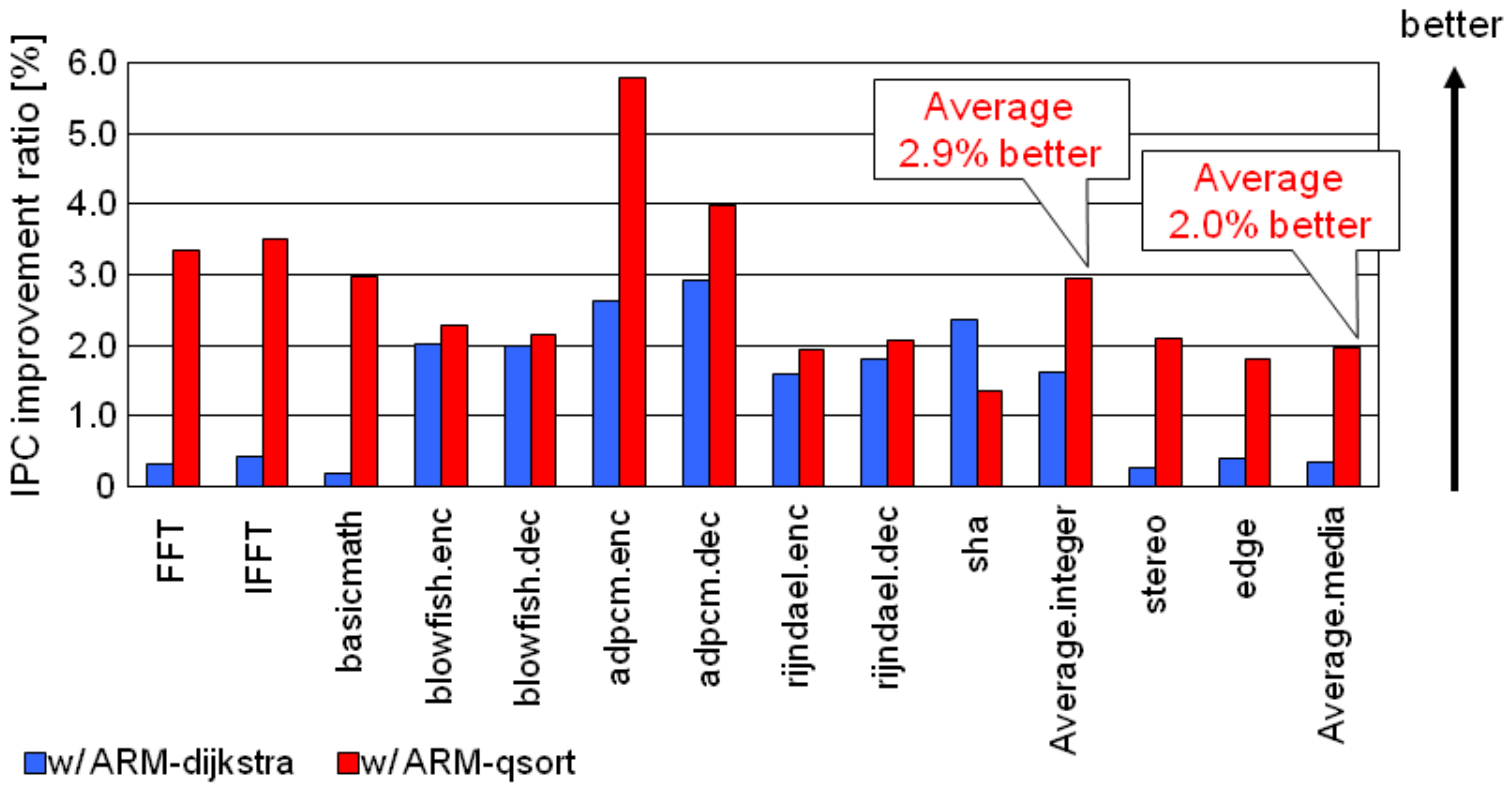
- Implementation of OROCHI with 0.25μm rule
  - Chip area: 1.2 million gates
    - 76% multi-core processor
  - Power consumption: 0.7W
    - 79% multi-core processor
  - Clock frequency: 98.7 MHz
    - 99% multi-core processor

**Comparison**



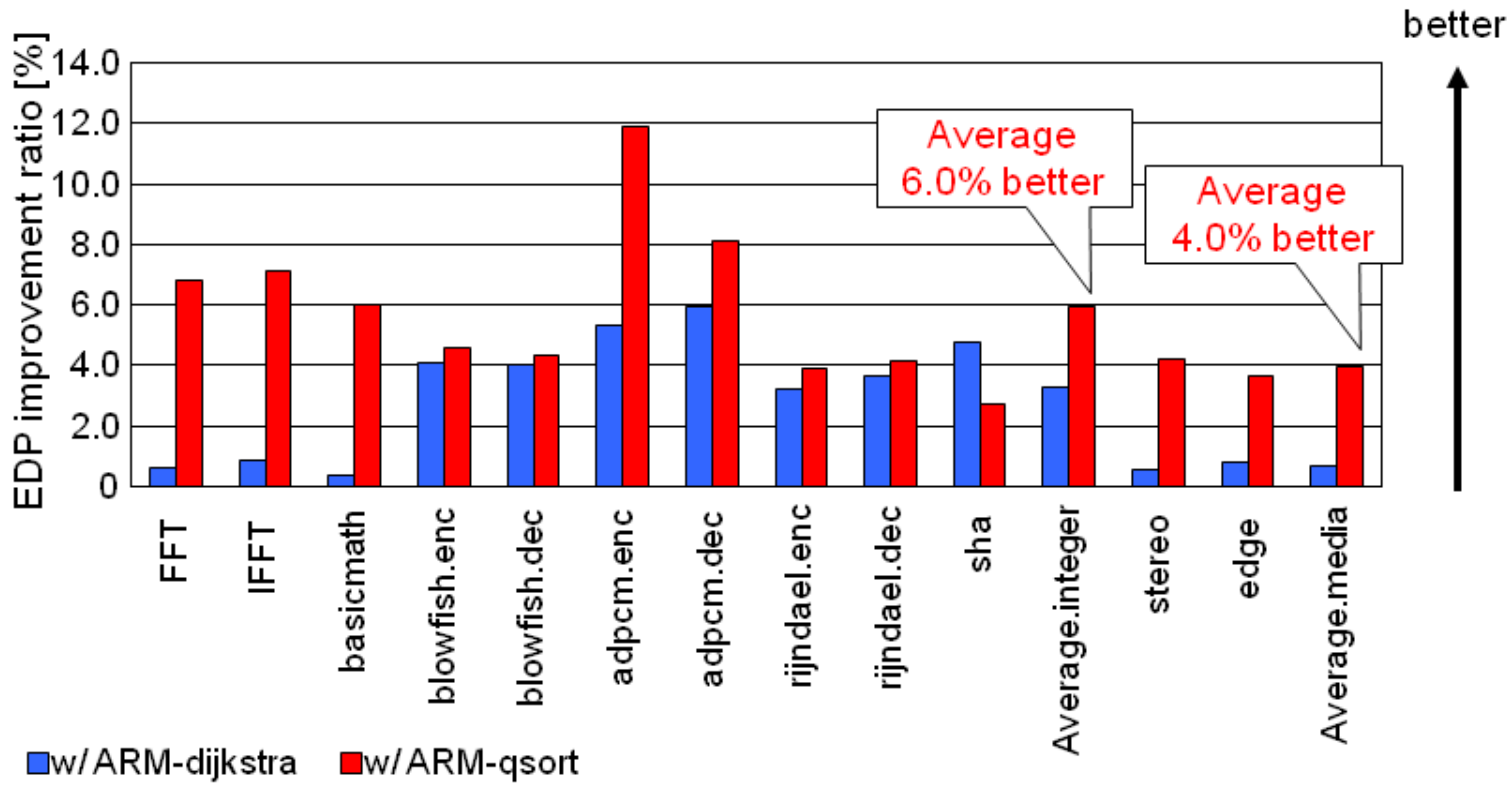
# Improvement of performance

- Selective flush mechanism
  - effective for program with high frequency of cache misses



# Energy efficiency of our mechanism

- Energy delay product (EDP)
  - EDP = energy x delay



# Emulator oriented Minimal InstructioN set CPU EMIN