



# Reliable multiprocessor system for low voltage MPSoC

Yoshinori TAKEUCHI, Jun KAWABE  
Graduate School of Information Science and  
Technology  
Osaka University



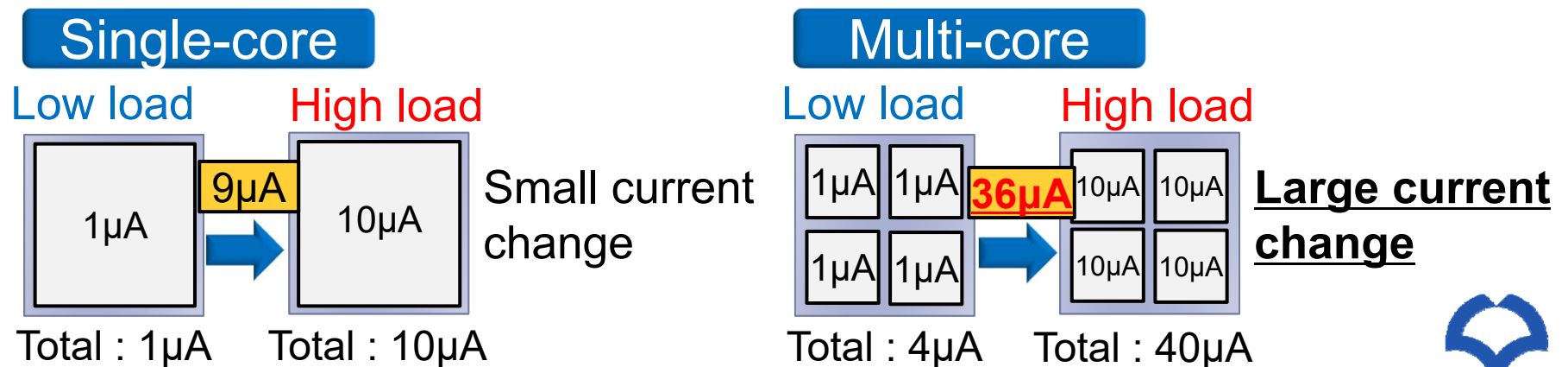
# Introduction

---

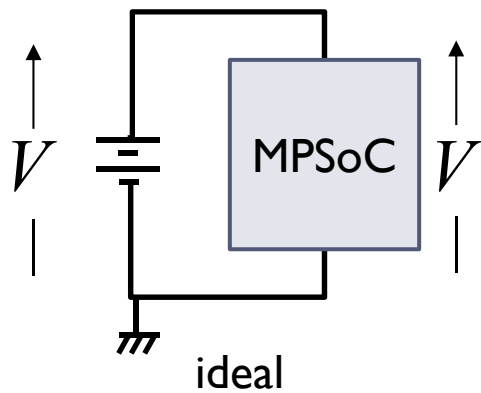
- ▶ Requirement for embedded systems
  - ▶ Low power
  - ▶ High performance
- ▶ MPSoC is expected for one of solutions as software defined hardware
  - ▶ High-performance by multi-function PE and multicore
  - ▶ Operating supply voltage is shrinking in order to save power consumption
  - ▶ Large current consumption by many functional units

# Reliability considering power supply noise

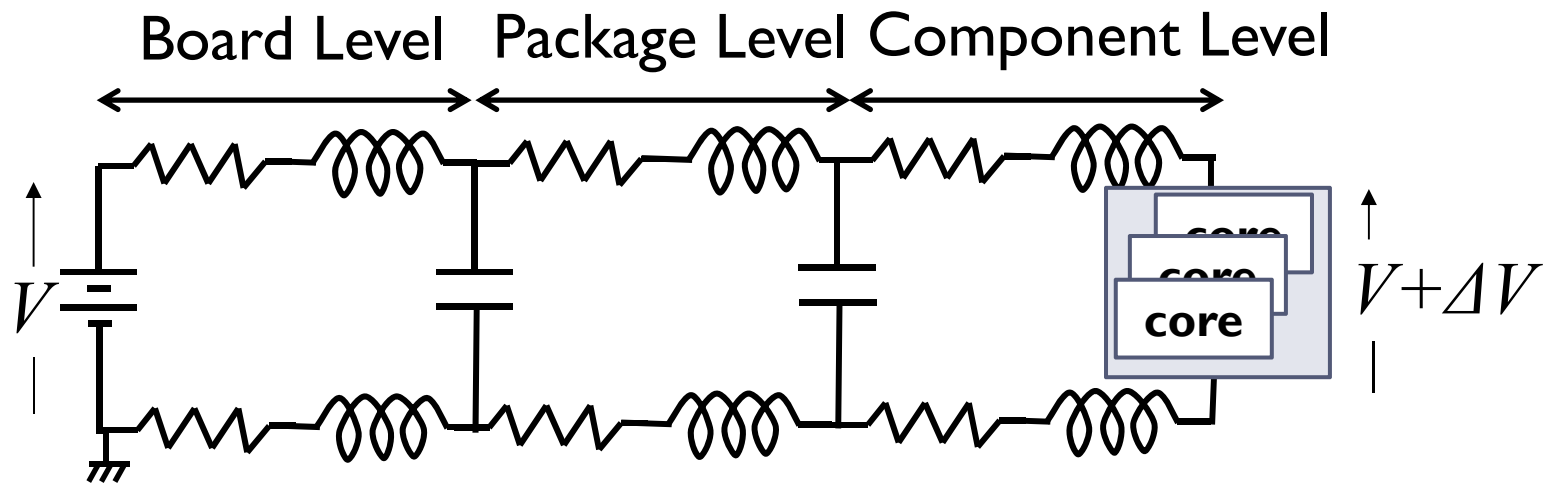
- ▶ Reason
  - ▶ Low operating voltage by advancement of technology
  - ▶ Near or sub-threshold operations on low power operation
- ▶ Noise from power supply swing becomes relatively-large
  - ▶ This effect becomes large when clock frequency becomes higher and current change becomes larger
  - ▶ Noise by electrical resonance on power line



# Power supply model for SoC

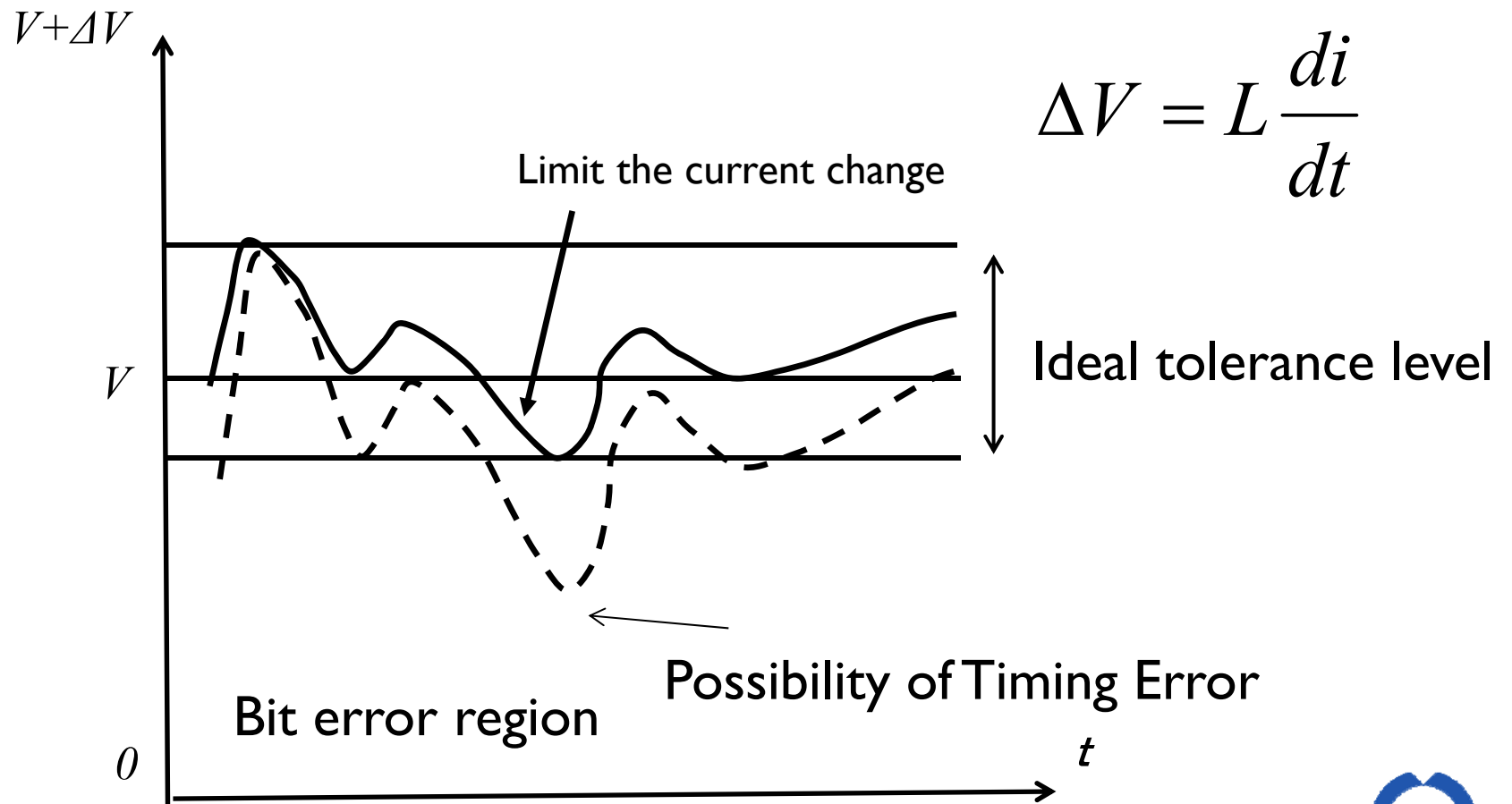


Long wires  
Large capacitors  
+Parasitic elements



Real circuit

# Power supply swing

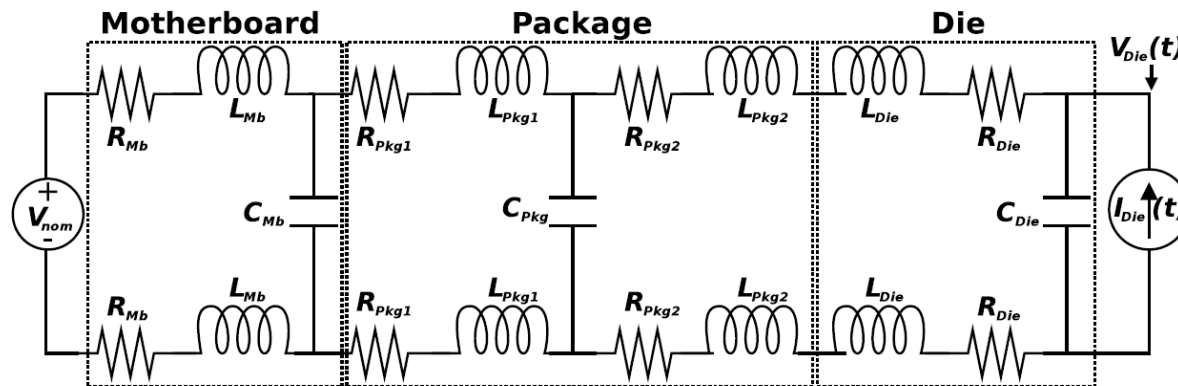


# Power Delivery Network

- ▶ Power Delivery Network (PDN) is composed of RLC

Long wires  
Large capacitors

+Parasitic  
elements



R. Bertran, A. Buyuktosunoglu, P. Bose, T. J. Slegel, G. Salem, S. Carey, R. F. Rizzolo, and T. Stra, "Voltage noise in multi-core processors: Empirical characterization and optimization opportunities," in 2014 47th Annual IEEE/chACM International Symposium on Microarchitecture, Dec 2014, pp. 368–380.

$$V_{Die} = V_{nom} - V_{droop} - V_{didt} = V_{nom} - V_{noise}$$

- ▶ The  $V_{noise}$  magnitude can be mitigated by adding decoupling capacitance – also known as *decap*
- ▶ There's a report that decap occupies 10% of chip area [1]

## Related Work

---

- ▶ [2],[3] reduced  $dI/dt$  by gradually activation of functional unit
  - ▶ A few cycles are wasted every time the functional unit is used
- ▶ [4] reduced  $dI/dt$  by staggered core activation
  - ▶ [4] only considers start time of cores

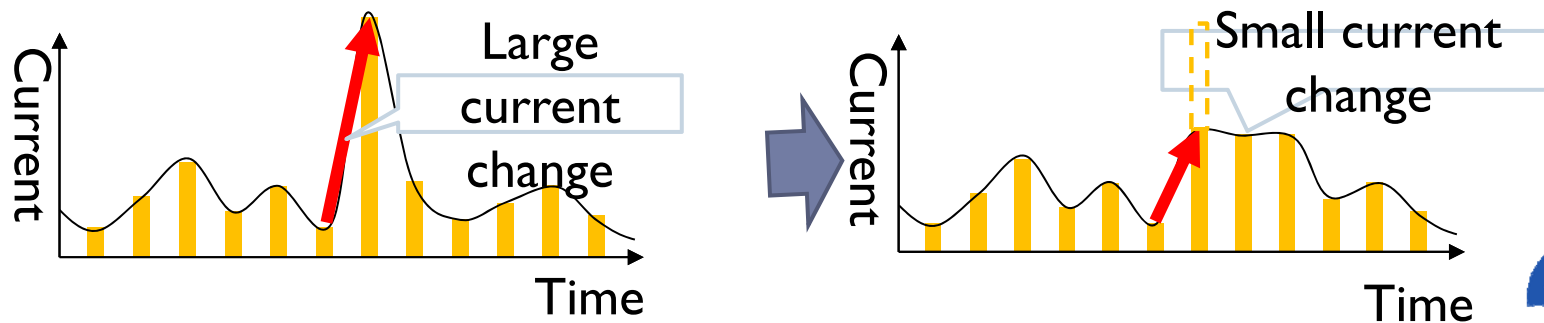
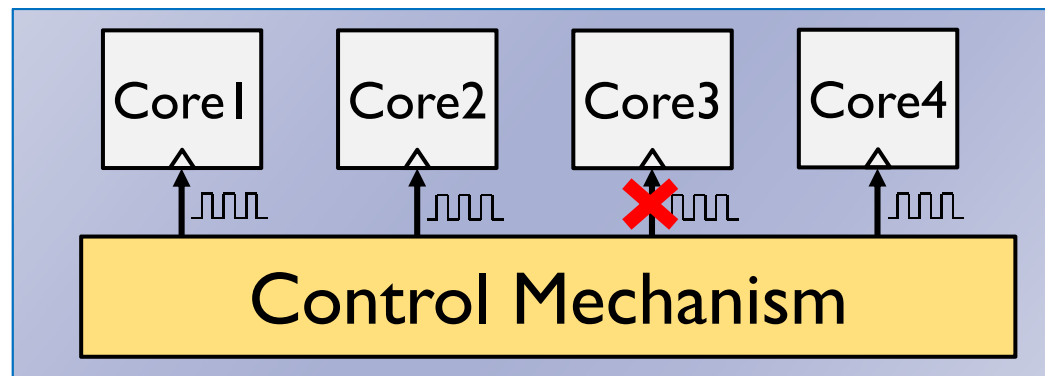
[2] M. Pant, P. Pant, D. Wills, and V. Tiwari, “An architectural solution for the inductive noise problem due to clock-gating,” in Proceedings of ACM/IEEE International Symposium on Low Power Electronics and Design, Aug. 1999, pp. 255–257.

[3] R. Singh, A. Kim, S. Kim, and S. Kim, “A three-step power-gating turn-on technique for controlling ground bounce noise,” in Proceedings of ACM/IEEE International Symposium on Low Power Electronics and Design, Aug. 2010, pp. 171–176.

[4] A. Paul, M. Amrein, S. Gupta, A. Vinod, A. Arun, S. Sapatnekar, and C. H. Kim, “Staggered core activation: A circuit/architectural approach for mitigating resonant supply noise issues in multi-core multi-power domain processors,” in Proceedings of the IEEE 2012 Custom Integrated Circuits Conference, Sept 2012, pp. 1–4.

# Proposed Method

- ▶ Control mechanism to select activation of cores in order to limit current change below a specified level

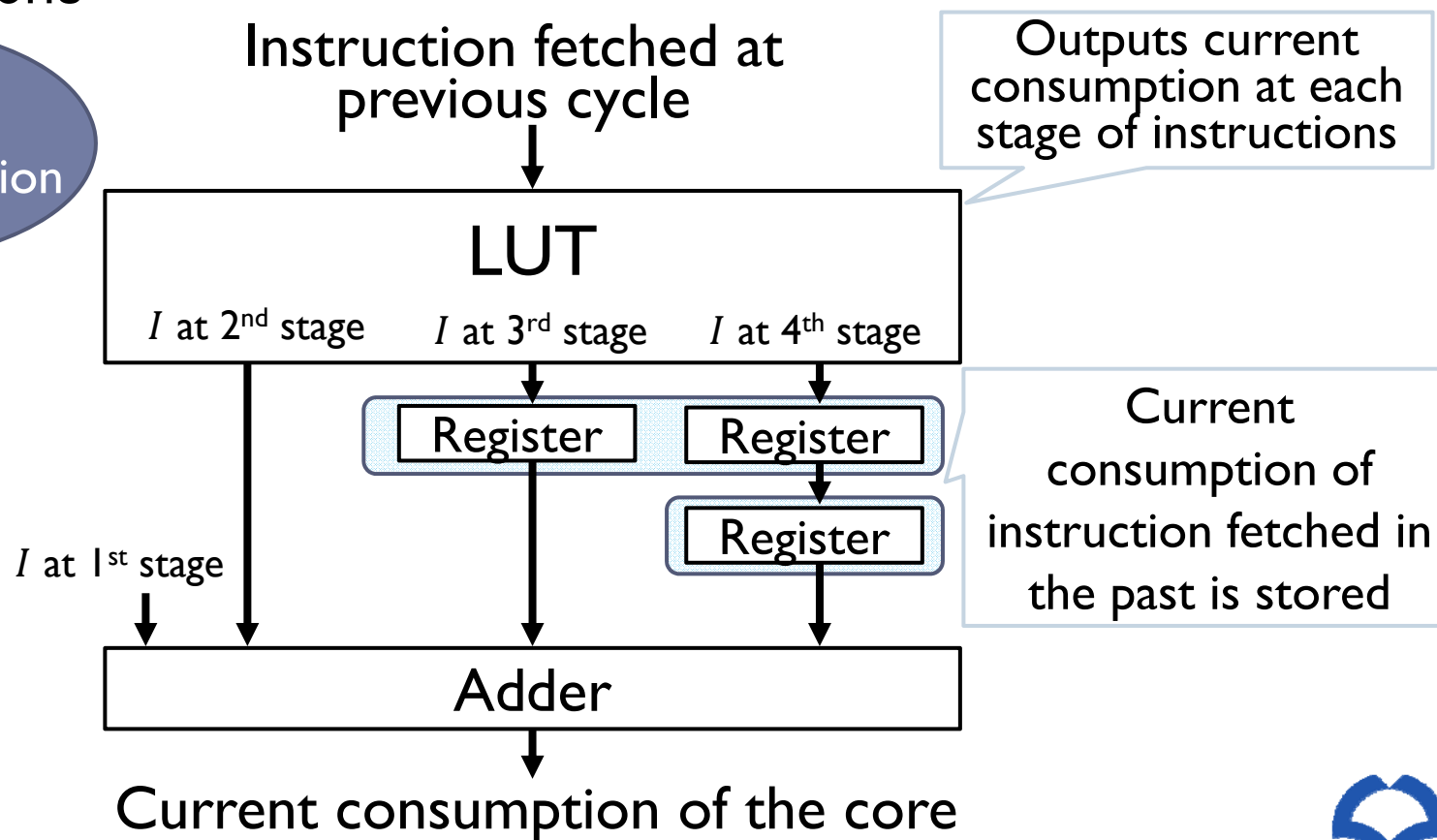




# Current Consumption Calculator

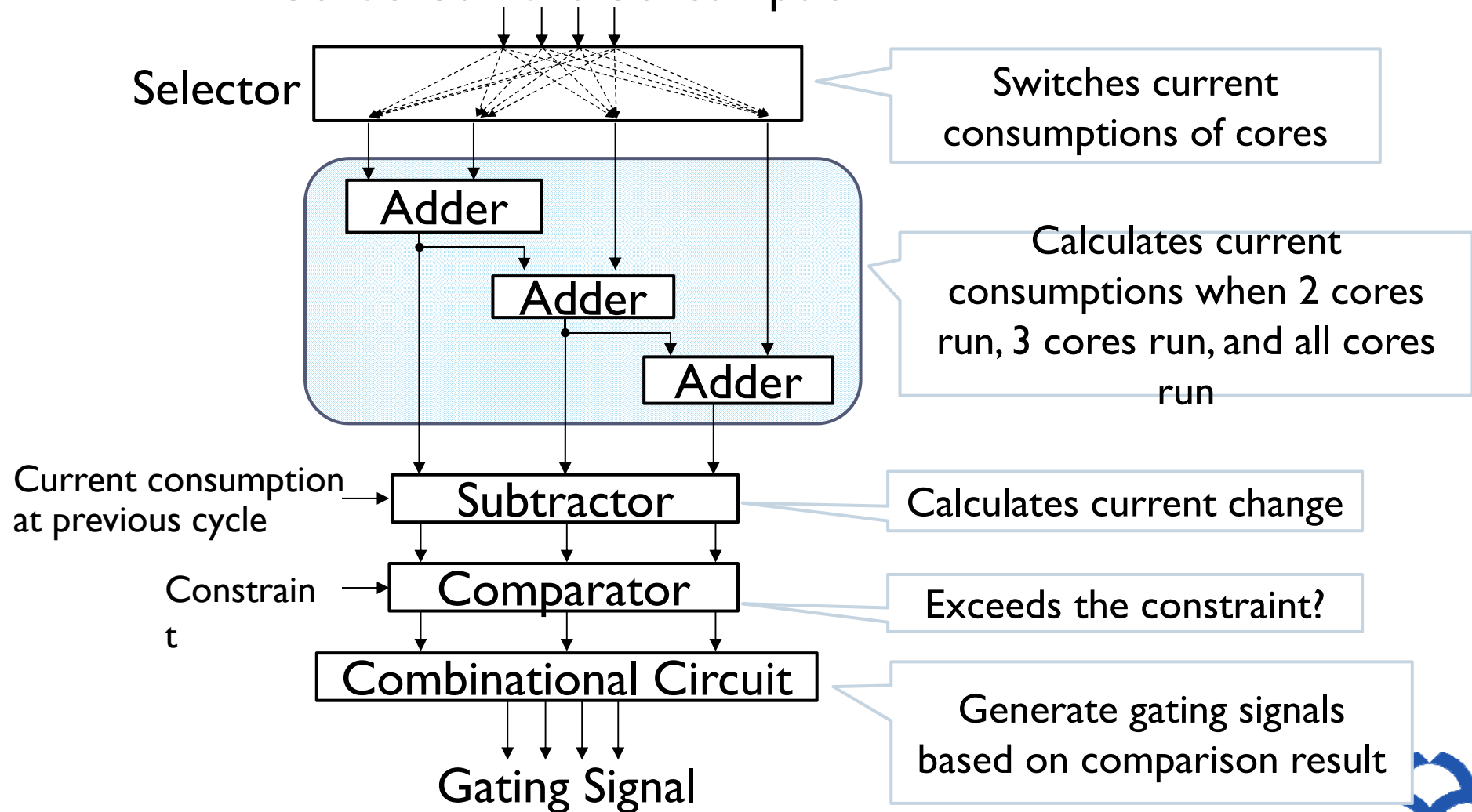
- Estimates current consumption at the next cycle by summing up current consumptions at each stage of instructions

Considers Pipeline Execution



# Run/Stop Core Decider

## Core's Current Consumption



# Run/Stop Core Decision Method

---

- ▶ Run/Stop core decision method is important
  - ▶ The task assigned to not activation core stalls
- ➔ Directly affect processing performance of system



Run/Stop core decision method which suppresses performance degradation is required

- (1) Maximum Current Change Selection Method (MCSM)
- (2) Heuristics for Maximum Current Change Selection

# (1) Maximum Current change Selection Method

- ▶ Select a combination of run/stop core which maximizes current consumption at each cycle

Combinations of run/stop core at a certain cycle

Core1 (7 $\mu$ A)	Core2 (9 $\mu$ A)	Core3 (8 $\mu$ A)	Core4 (9 $\mu$ A)	Total [ $\mu$ A]	Current Change [ $\mu$ A]	Exceeds the limit?
○	×	×	×	7	3	NO
⋮	⋮	⋮	⋮	⋮	⋮	⋮
×	○	×	○	18	14	NO
○	○	○	×	24	20	NO
○	○	○	○	33	29	YES

Calculation for all combinations is required



Implementation cost is LARGE

Current consumption at previous cycle : 4 $\mu$ A  
Maximum current change (limit) : 20 $\mu$ A

## (2) Heuristics for Maximum Current change Selection

- ▶ Reduce combinations of run/stop core
  - ▶ Only calculates consecutive two, three, ..., and all cores from a certain core are selected as run core
- ▶ To avoid unfair selection, the core which starts calculation should be changed every cycle

When calculating from Core1

Combinations of run/stop core at a certain cycle

Core1 (7 $\mu$ A)	Core2 (9 $\mu$ A)	Core3 (8 $\mu$ A)	Core4 (9 $\mu$ A)	Total [ $\mu$ A]	Current Change [ $\mu$ A]	Exceeds the limit?
○	○	×	×	16	12	NO
○	○	○	×	24	20	NO
○	○	○	○	33	29	YES

Number of combinations is small



Implementation  
cost is small

Current consumption at previous cycle: 4 $\mu$ A

Maximum current change (limit): 20 $\mu$ A

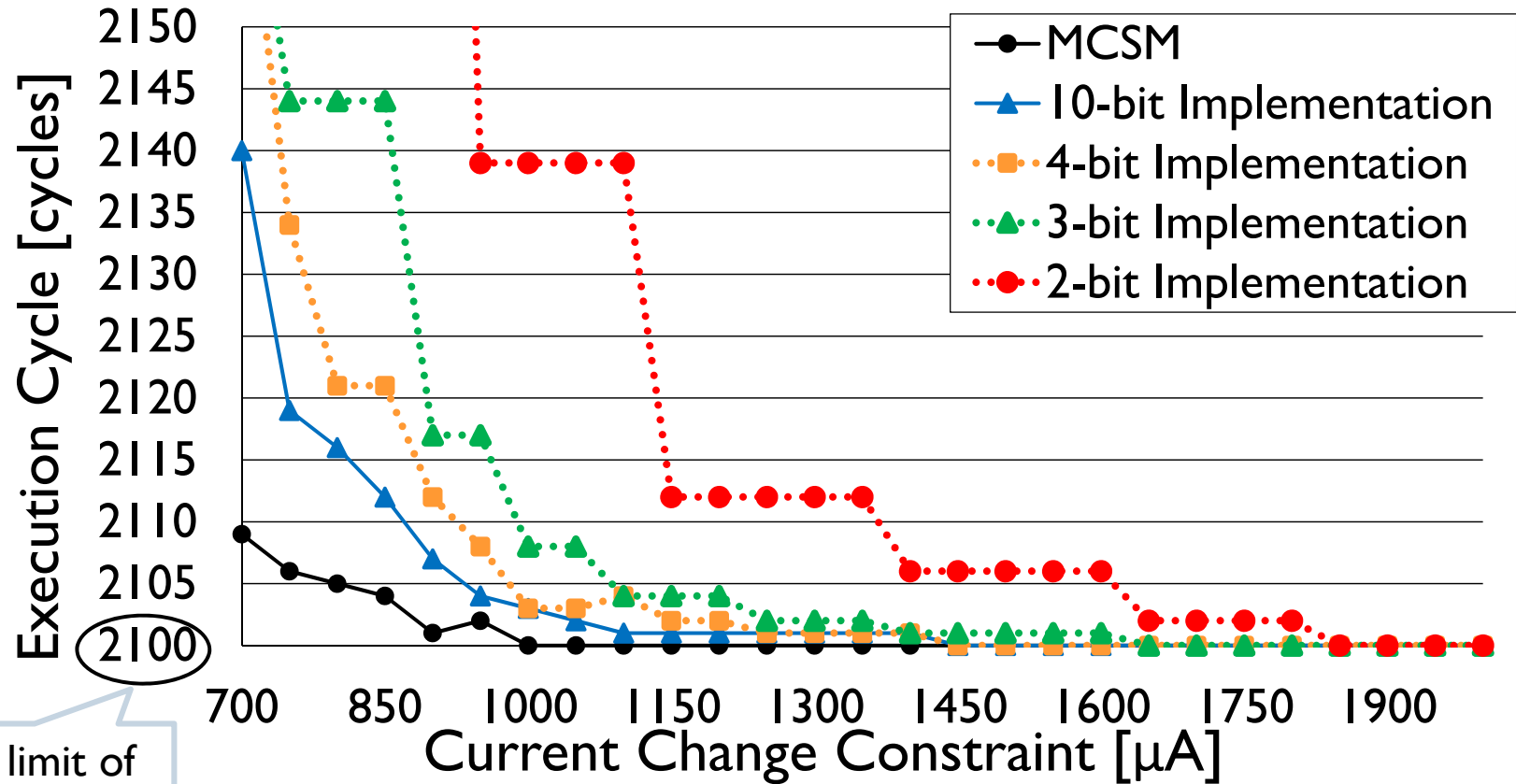
# Experimental Setup

---

- ▶ Evaluation condition
  - ▶ Core processor: BrownieSTD32
    - ▶ 32-bit architecture, 4 stage pipeline, RISC processor
  - ▶ Number of cores: 4
  - ▶ 100 programs were randomly chosen from DSPstone
- ▶ Proposed mechanism was implemented using 10-bit, 4-bit, 3-bit, and 2-bit quantization of instruction current values

# Experimental Result

- Execution cycle increases as the current change constraint decreases because clock-gating frequently occurs



Lower limit of execution cycle



# Conclusion and Future Work

---

## ▶ Conclusion

- ▶ Proposes noise reduction method by limiting current change by using dynamic instruction scheduling
- ▶ Proposes a method considering pipeline architecture behaviors of MP cores

## ▶ Future Work

- ▶ Evaluate total effect of proposed method
- ▶ Reduce electrical resonance effect by instruction scheduling for MPSoC





# Acknowledgement

---

- ▶ This work was partly supported by JSPS KAKENHI  
Grant Number JP26330064



---

# Thank you for your attention!