SEOUL NATIONAL UNIVERSITY

DAL Design Automation Laboratory

Reconfigurable MP-SoC Architecture & Application Mapping

Kiyoung Choi

Design Automation Lab
Dept. of EECS
Seoul National University

Outline

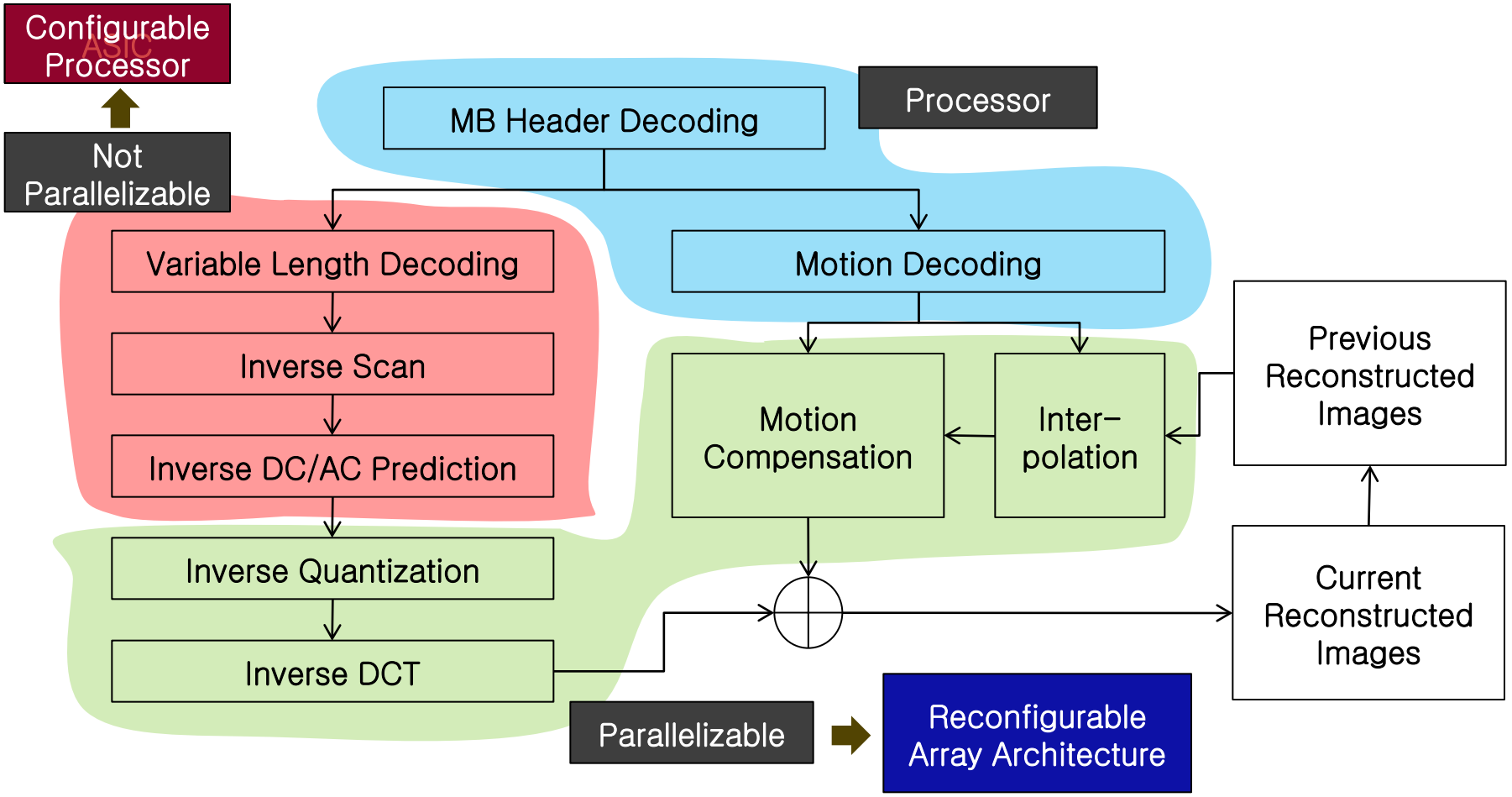
- Introduction
- FloRA Architecture
  - Reconfigurable Computing Module
- Application Mapping
  - Kernel Mapping onto RCM
- Conclusion

Introduction

- Hardware-like performance and software-like flexibility
  - Quickly adapt to the fast changing market
  - Self-adaptation to user environment change
  - Performance scaling
- Parallel architecture with reconfigurability
  - Configurable processor
  - Coarse-grained reconfigurable array
- Challenges
  - Communication
  - Programming

Introduction

- MPEG-4 example



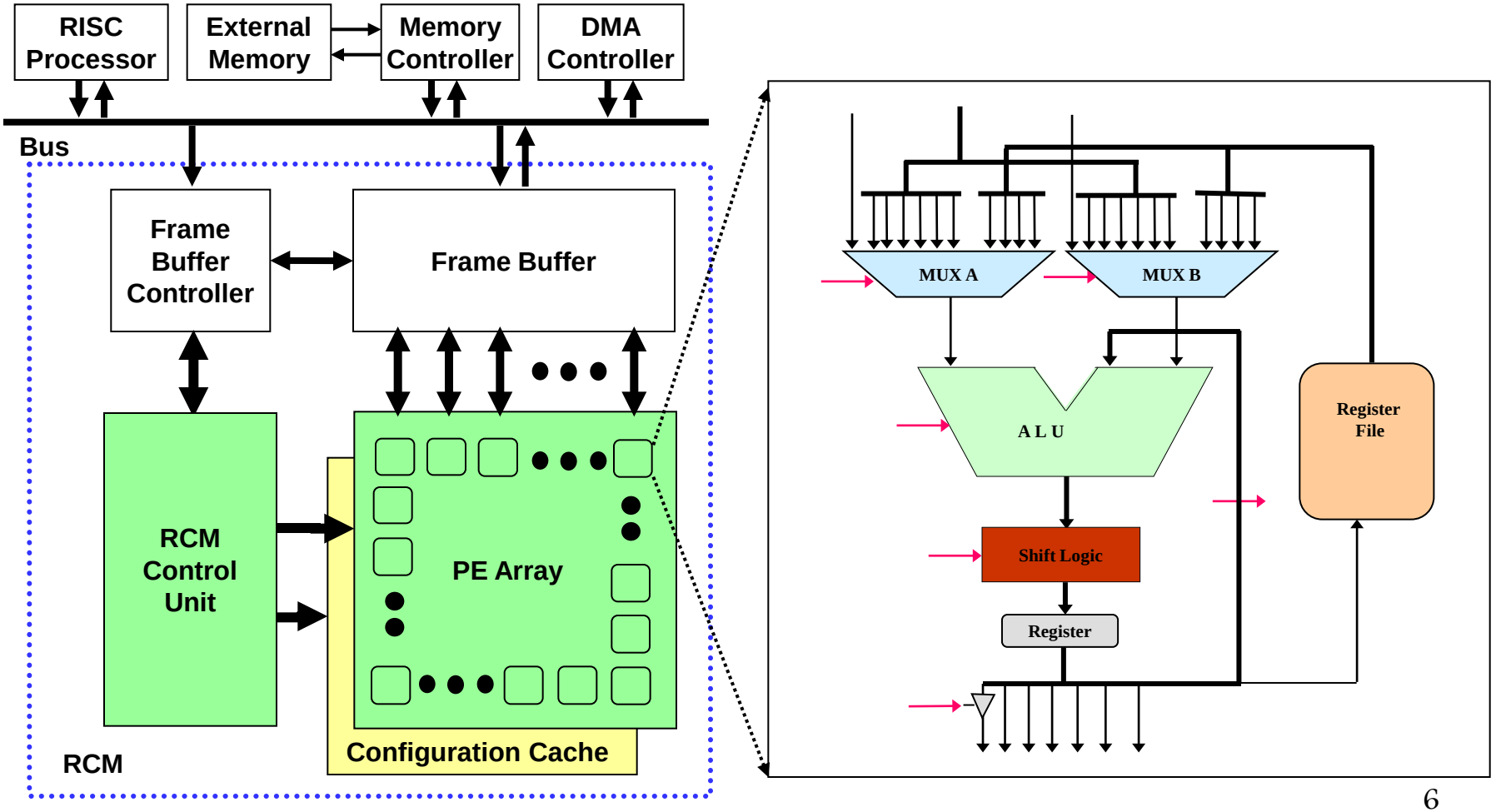
➡ The whole system is flexible now!

FloRA Architecture

- Configurable processor
  - High-speed clock for sequential operations
  - Instruction-set extension for irregular parallel operations
- Coarse-grained reconfigurable array
  - For regular parallel operations
  - Resource sharing for area
  - Pipelining for throughput
  - Configuration pipelining for low power
  - Floating-point computation for applicability
- Memory-centric communication
  - To reduce communication overhead

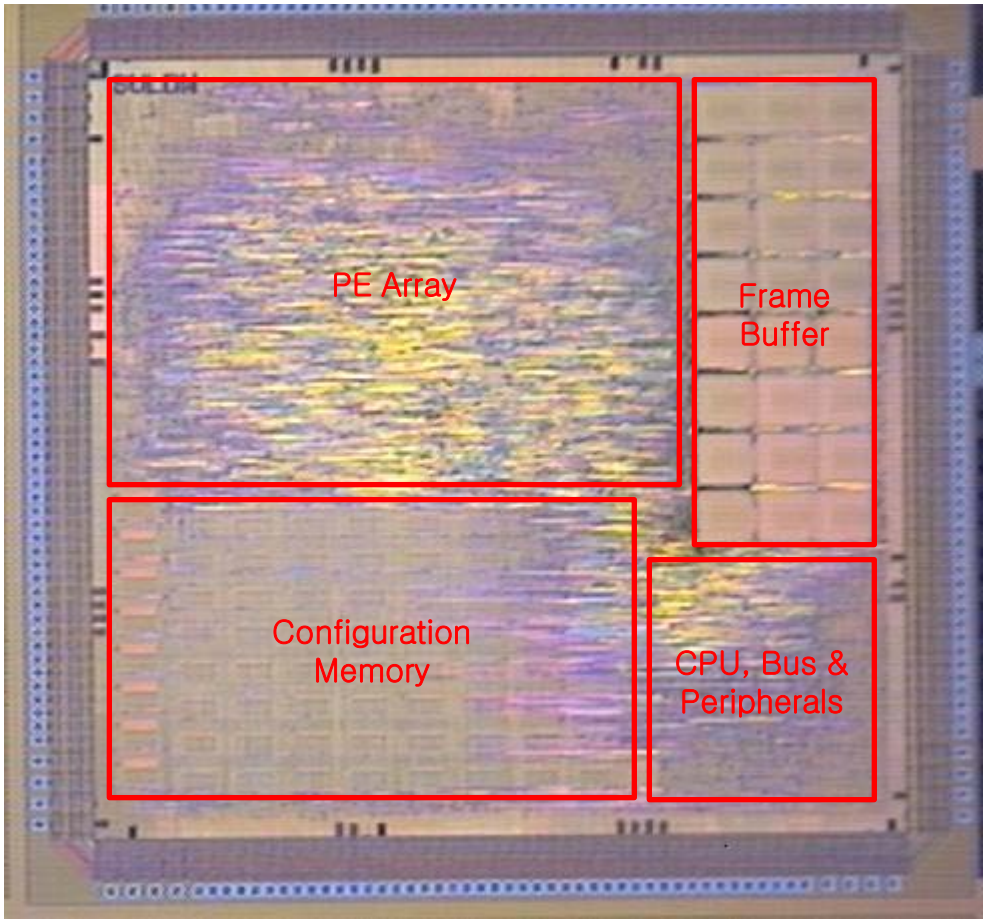
FloRA Architecture

- Bus-centric communication architecture



FloRA Architecture

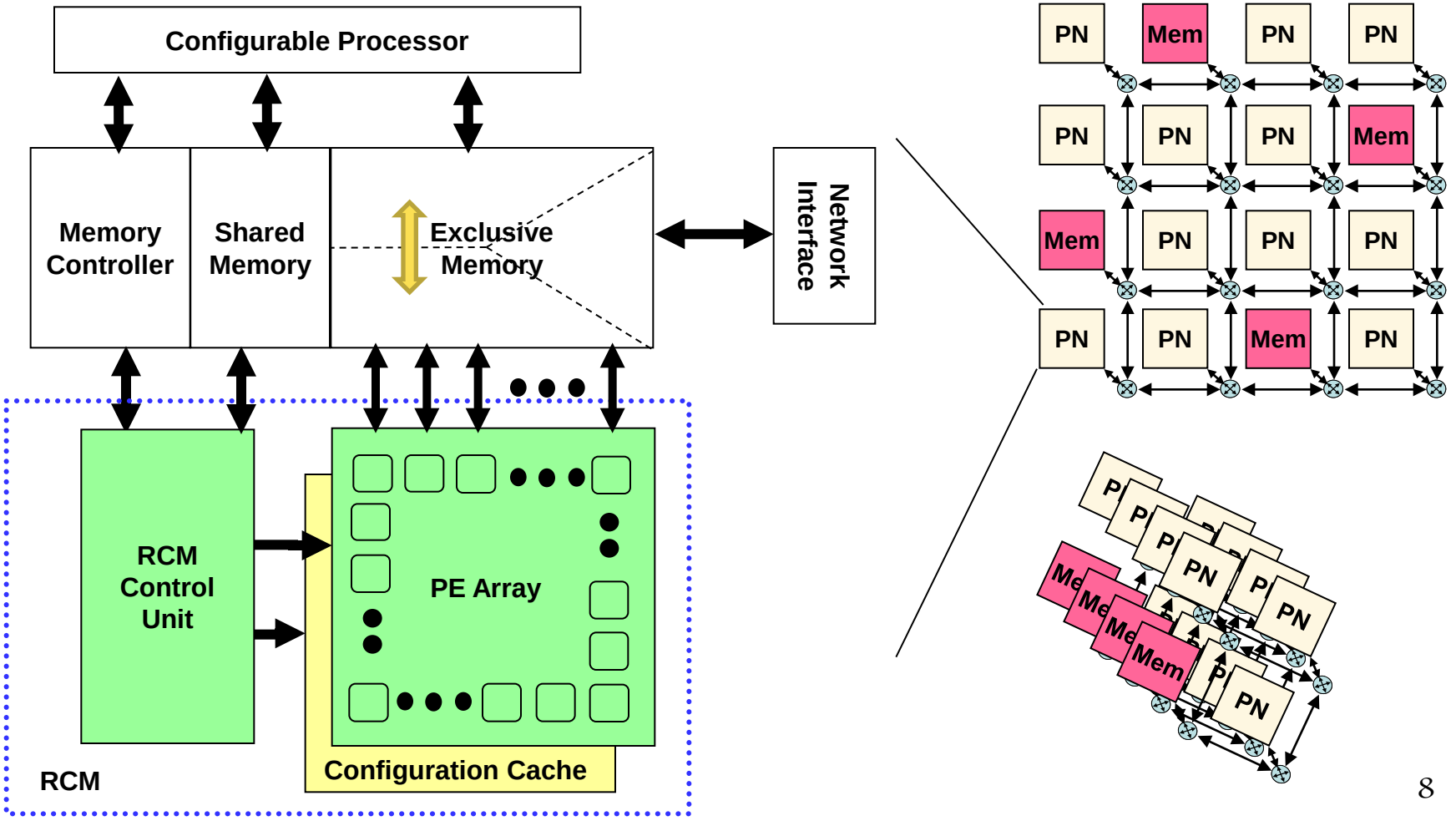
- Physical chip design



- PE Array size: 8x8
- Configuration Memory:
  (temporal) 2,560 bytes
  (spatial) 3,072 bytes
- Frame Buffer: 6,144 bytes
- Technology: 130nm
  (Dongbu HiTek)
- Clock frequency: 125MHz
  (gate level, typical case)
- Area: 11.2 mm^2

FloRA Architecture

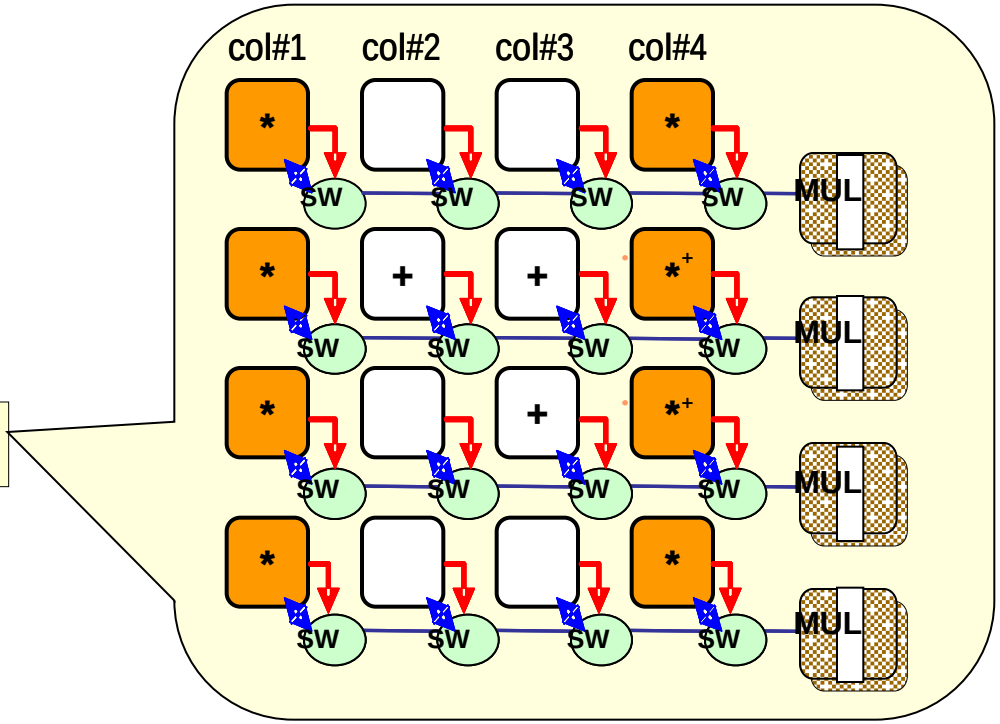
- Memory-centric communication architecture



Reconfigurable Computing Module

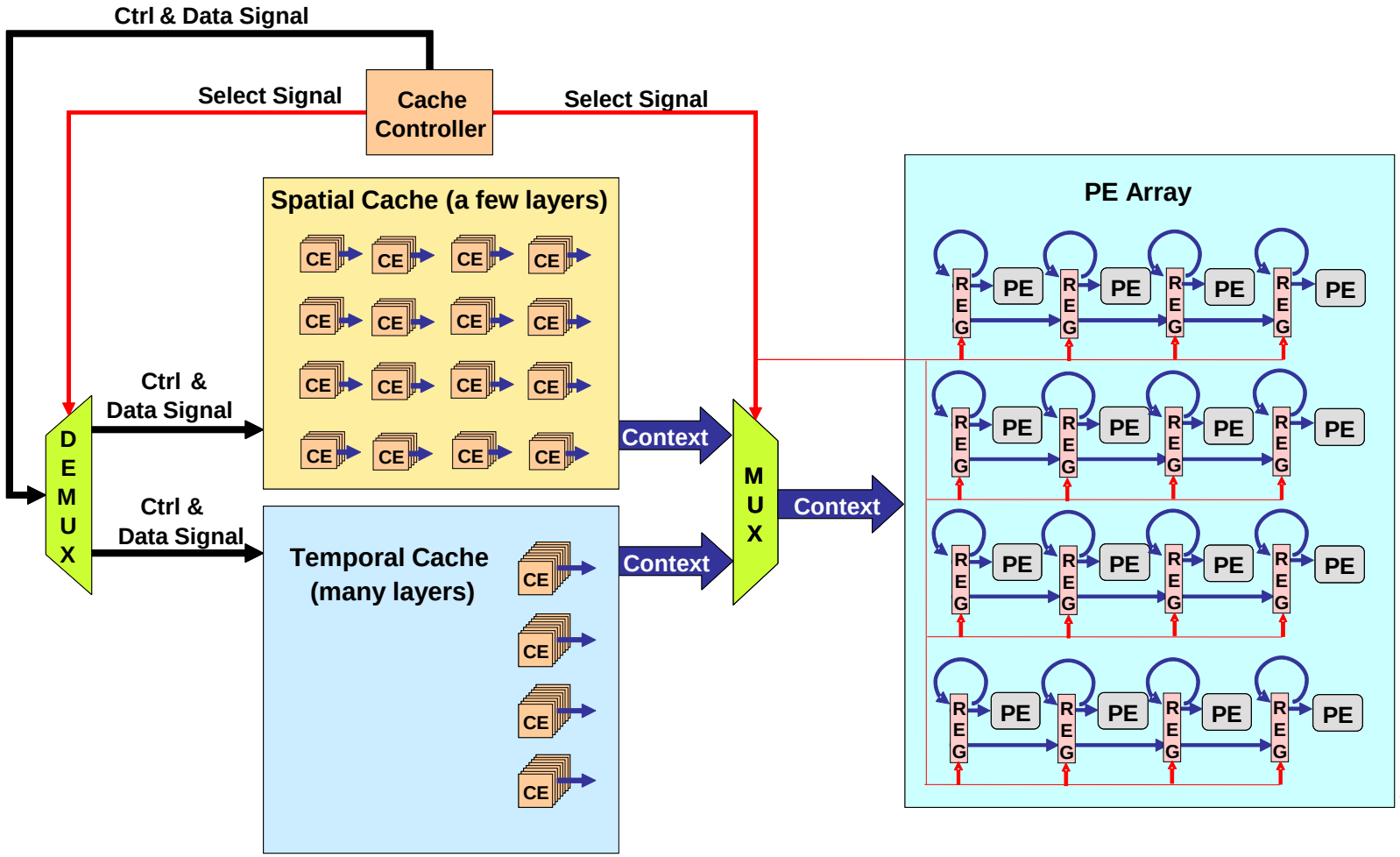
- Resource sharing and pipelining

	Col#1	Col#2	Col#3	Col#4
1	**Ld**			
2	***1**	**Ld**		
3	***2**	***1**	**Ld**	
4	**+**	***2**	***1**	**Ld**
5	**+**	**+**	***2**	***1**
6	***1**	**+**	**+**	***2**
7	***2**	***1**	**+**	**+**
8	**St**	***2**	***1**	**+**



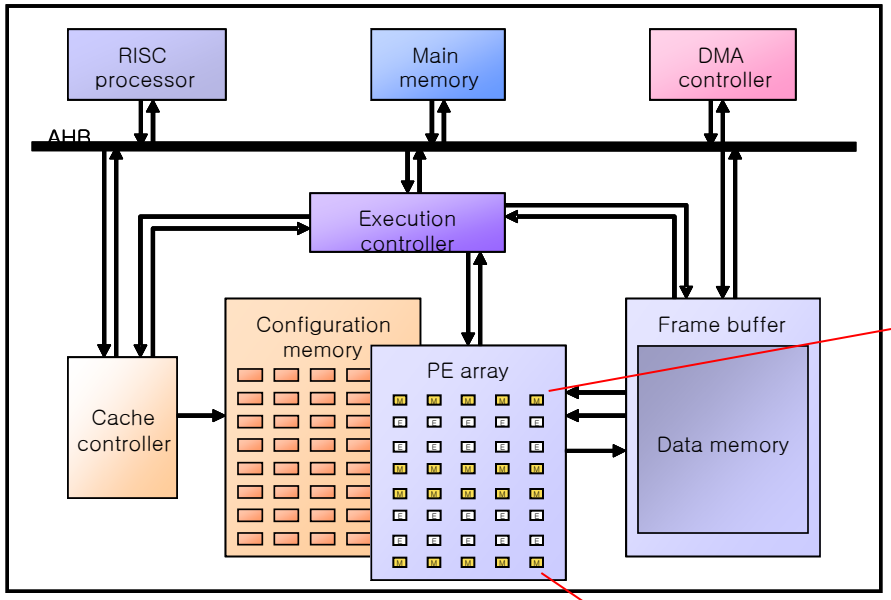
Reconfigurable Computing Module

- Hybrid configuration of cache structure

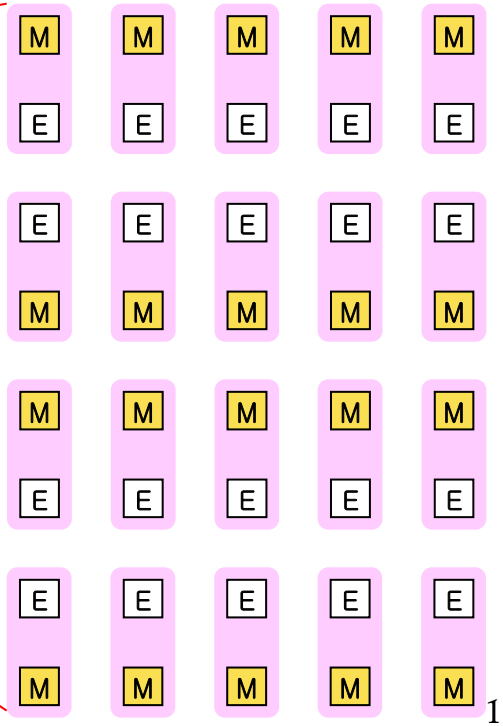


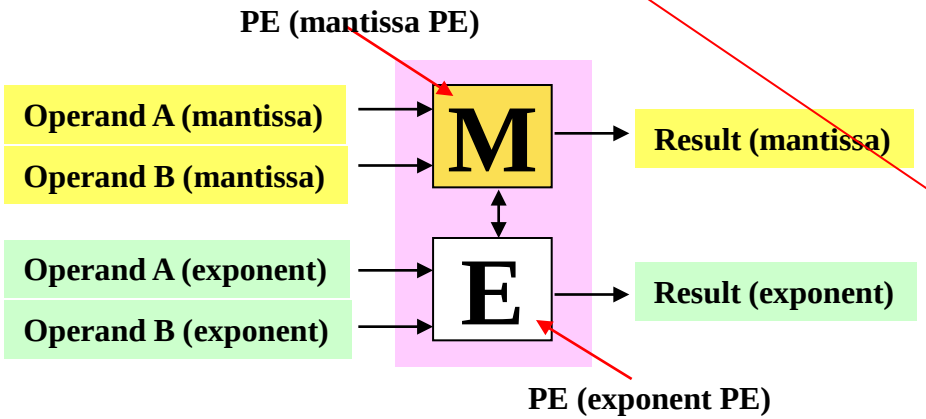
Reconfigurable Computing Module

- Floating point operations



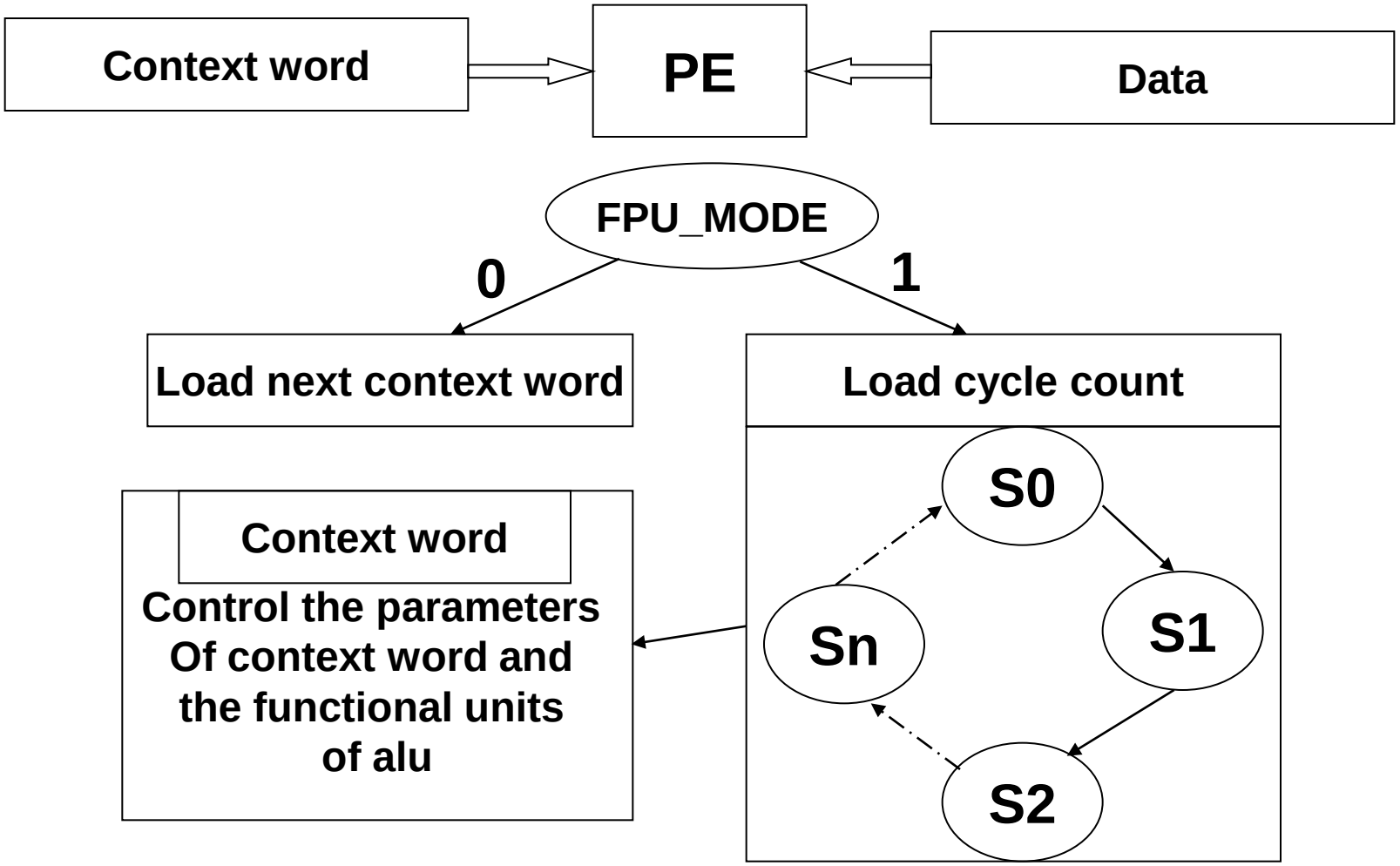
32/16 floating-point operations in parallel





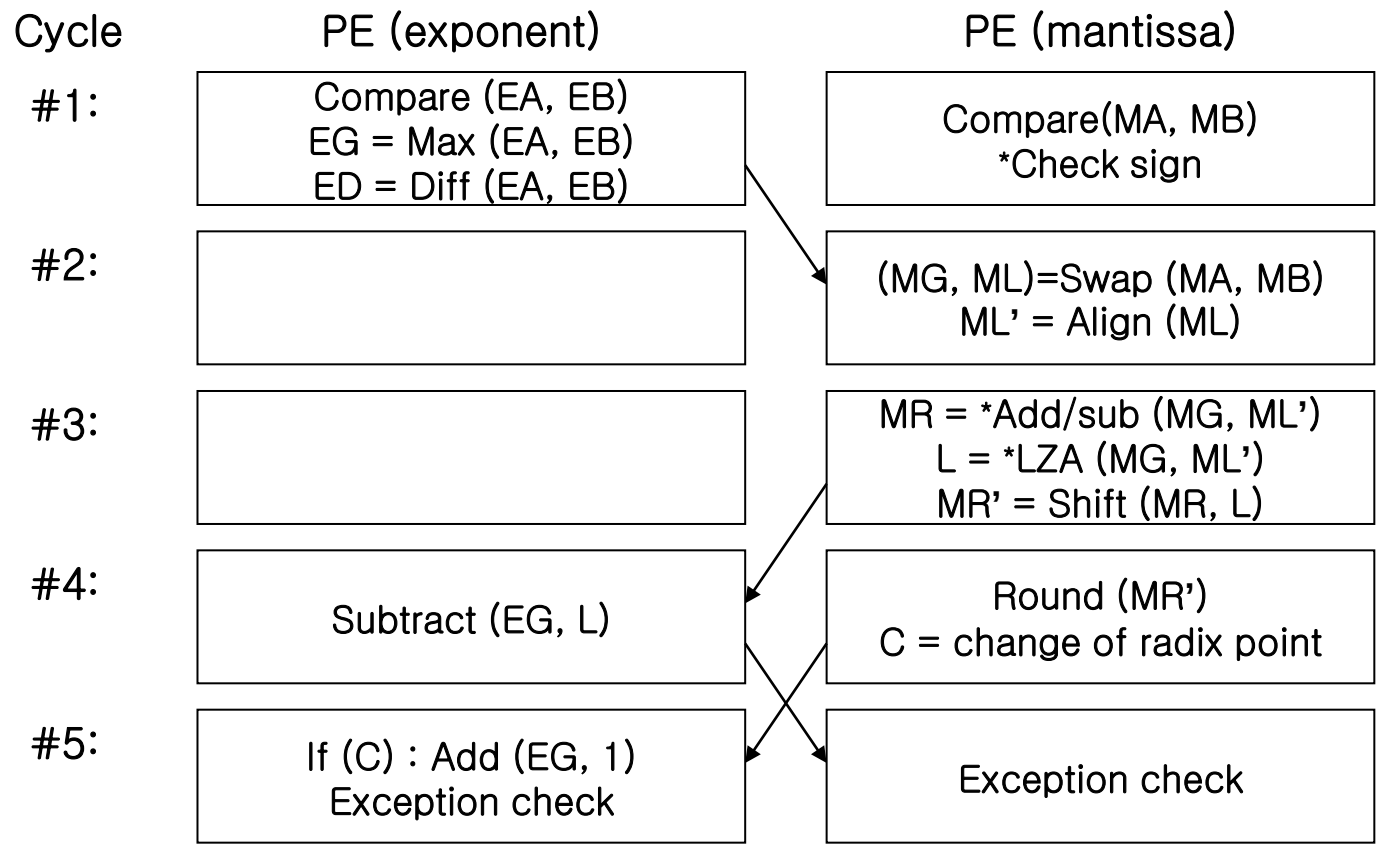
Reconfigurable Computing Module

- Finite state machine control



Reconfigurable Computing Module

- Multi-cycle operations



→ : data transfer between the PE's
EA, MA: exponent and mantissa of operand A, respectively
EB, MB: exponent and mantissa of operand B, respectively

Reconfigurable Computing Module

- Properties of floating-point functions (100MHz @0.18u)

func.	input	output	latency (no of cycles)	method
add/sub	24-bit floating	24-bit floating	5	int. unit
mult	24-bit floating	24-bit floating	3	int. unit
div	24-bit floating	24-bit floating	8	int. unit

Reconfigurable Computing Module

- Comparison of basic 3D graphics functions

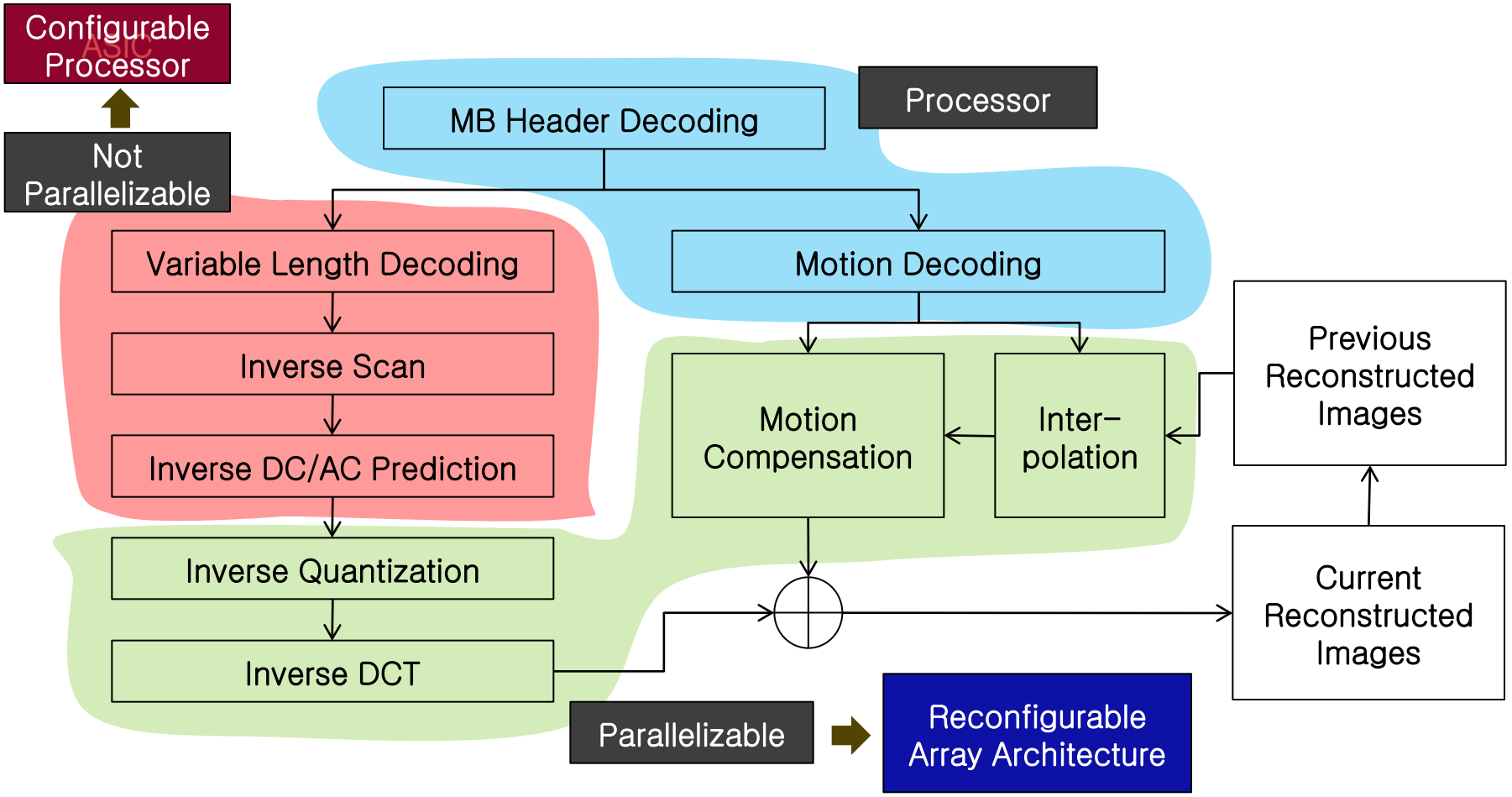
3D graphics func.	latency (#cycles)	1/throughput (average #cycles per operation)		
	our CGRA	our CGRA	scalar proc[9]	ARM VFP11[9]
4-term dot product	13	3.8	16	4.25
3-term cross product	22	5.8	25	11.67
x/w, y/w, z/w	9	3	29	17
3-term normalization	34	8	72	10

Reconfigurable Computing Module

- Chip test (JPEG and Fractal)

Application Mapping

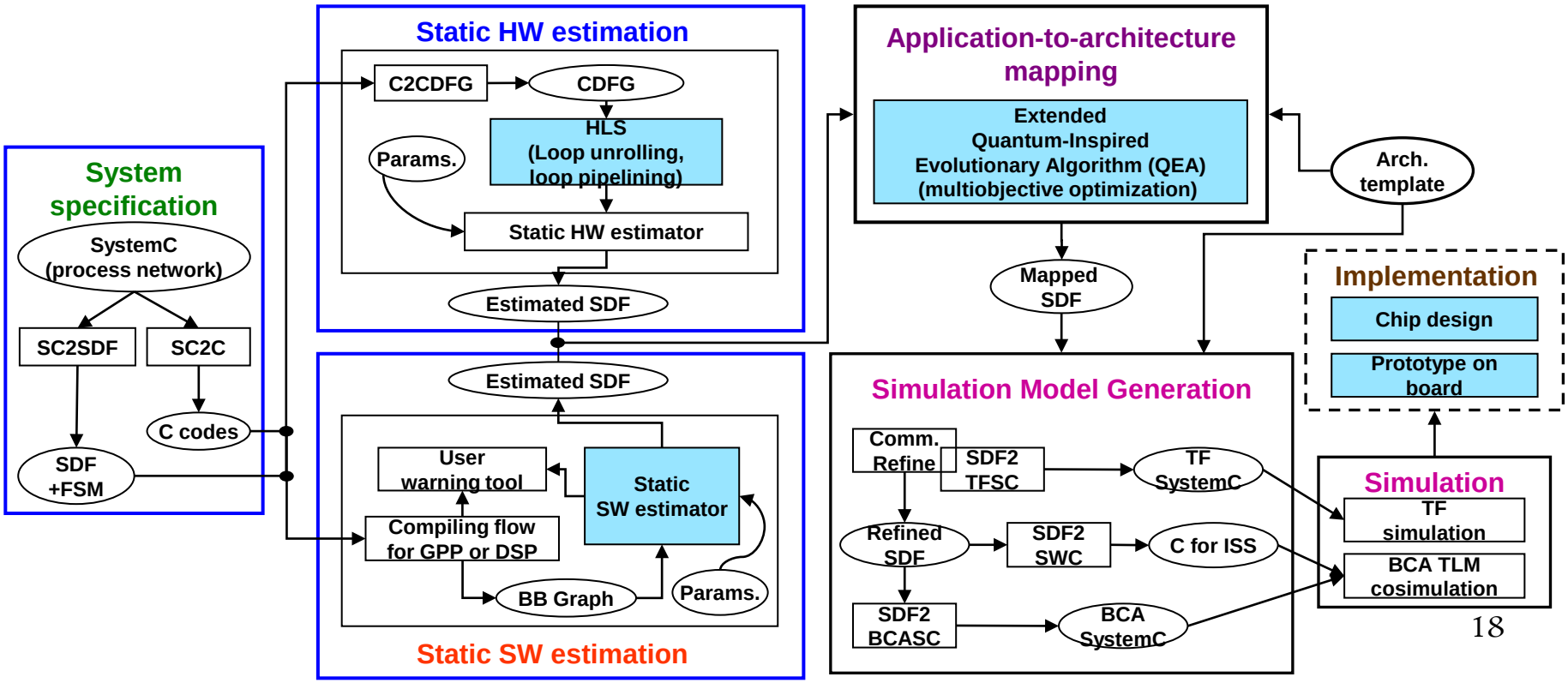
- MPEG-4 example



➡ The whole system is flexible now!

Application Mapping

- SoCDAL: an SoC design environment
  - Input in SystemC
  - Integrates task decomposition, estimation, mapping, communication synthesis, and simulation

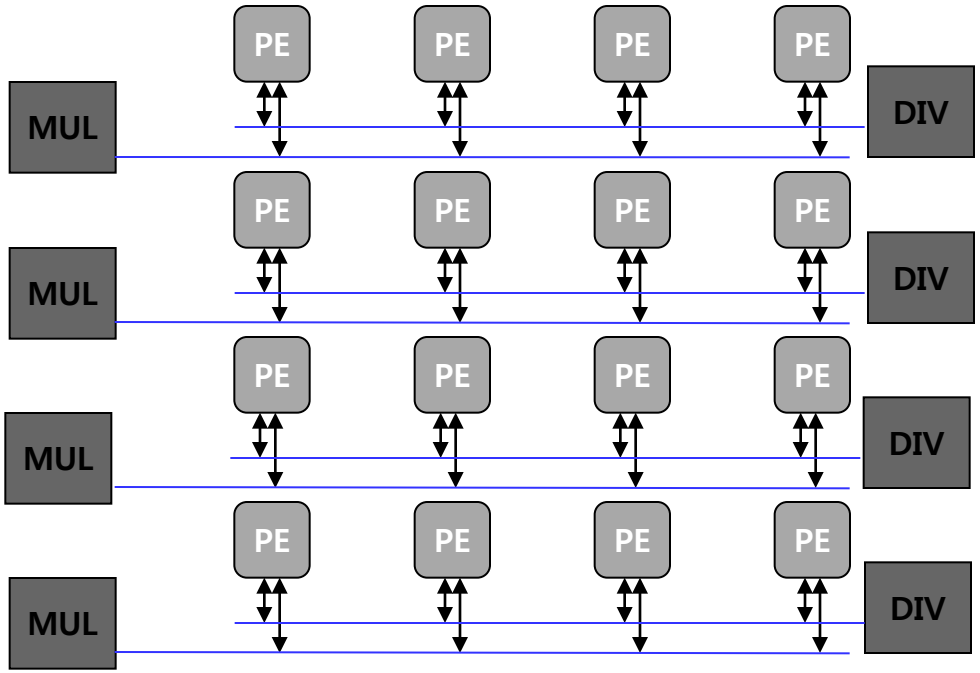


Application Mapping

- Related work
  - DRESC for ADRES (IMEC)
    - Modulo scheduling
    - Shared registers (multi-cycle access)
    - Simulated annealing (slow)
    - No pipelined functional units
  - Edge-centric modulo scheduling for ADRES
    - Fast heuristic algorithm for routing
    - Performance is degraded
  - Cyber work bench for DRP (NEC)
    - Based on high-level synthesis
    - Shared registers (long critical path delay)
    - Simple heuristic (low utilization)
    - Node centric (poor data routing)

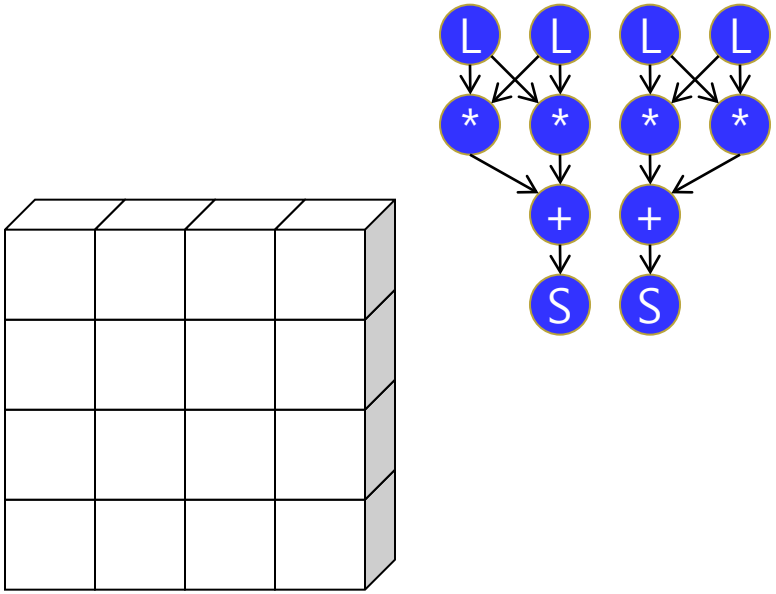
Kernel Mapping onto RCM

- Target architecture
  - Area critical resources are located outside the PEs
  - PEs in the same row share the resources thru buses



Kernel Mapping onto RCM

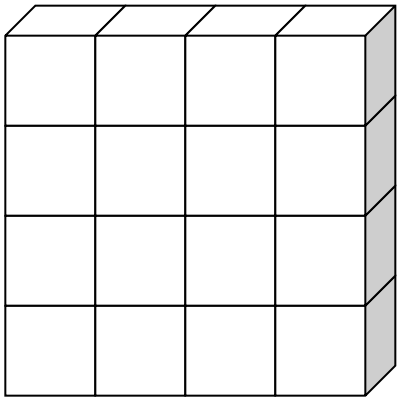
- Spatial mapping

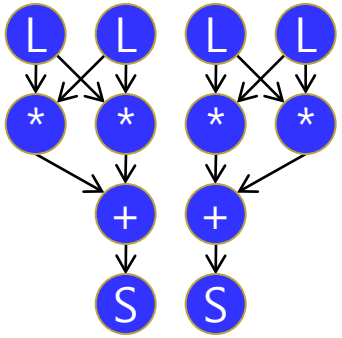


  - Each operation in a loop body is spatially mapped to a dedicated PE
  - Each PE executes a fixed operation with static configuration

Kernel Mapping onto RCM

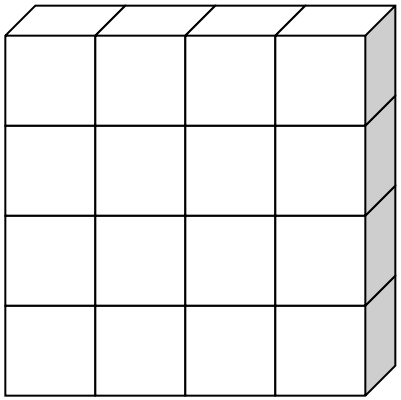
- Temporal mapping
  - A PE executes multiple operations in a loop by changing the configuration dynamically
  - Each column executes an iteration of the loop

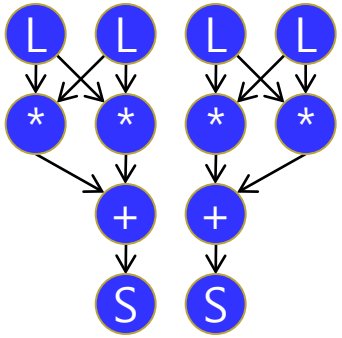




Kernel Mapping onto RCM

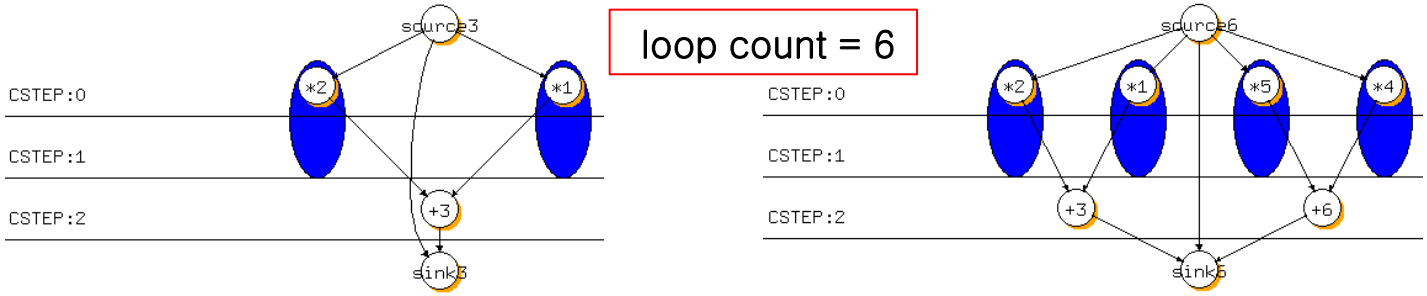
- Temporal mapping
  - Loop pipelining
  - Configuration is also pipelined





Kernel Mapping onto RCM

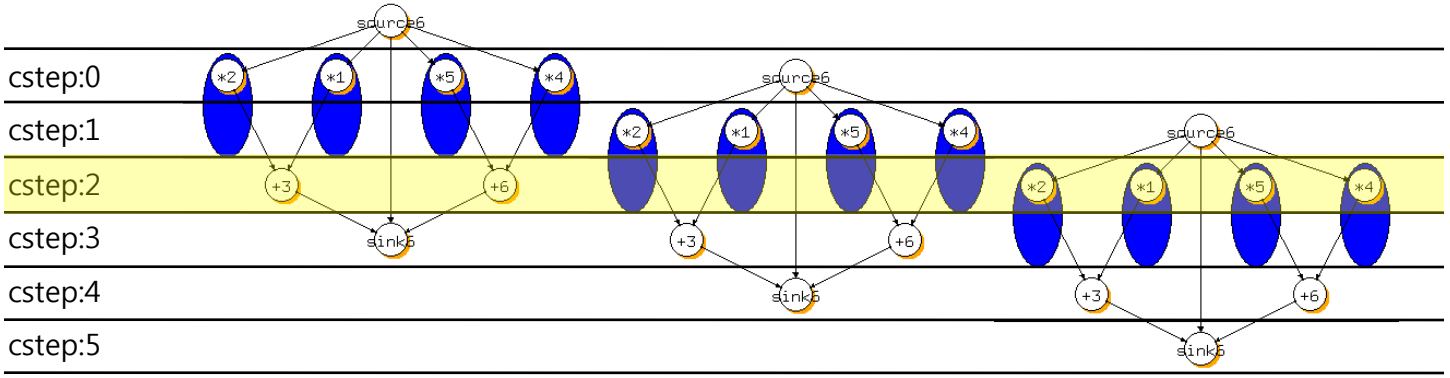
- Loop transformation
  - Loop unrolling



Total latency = 3*6 = 18 cycles

Total latency = 3*3 = 9 cycles

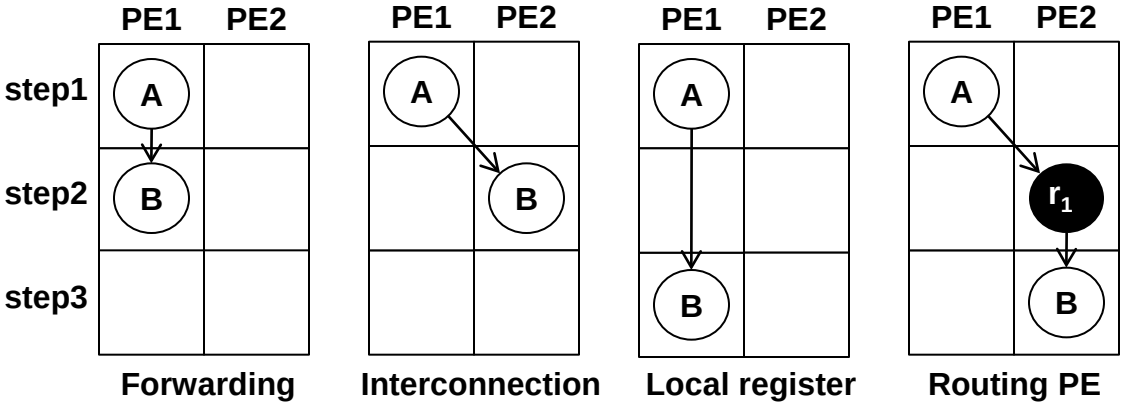
  - Loop pipelining



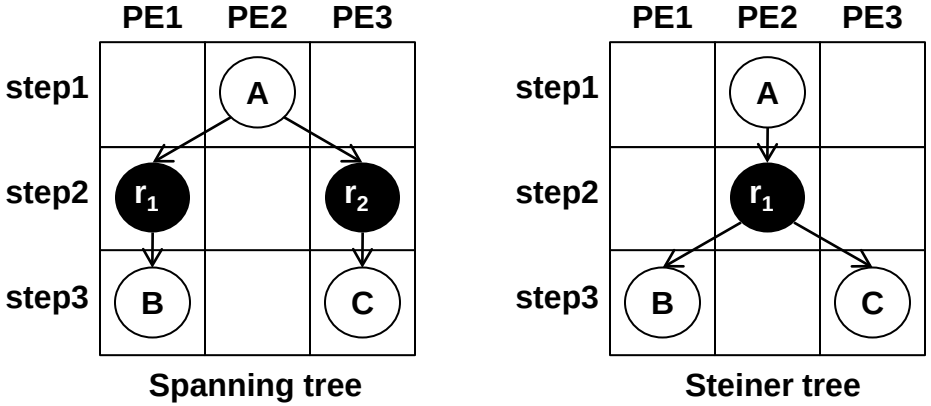
Total latency = 3 + 2*1 = 5 cycles

Kernel Mapping onto RCM

- Finding routing paths
  - Single fanout



  - Multiple fanout

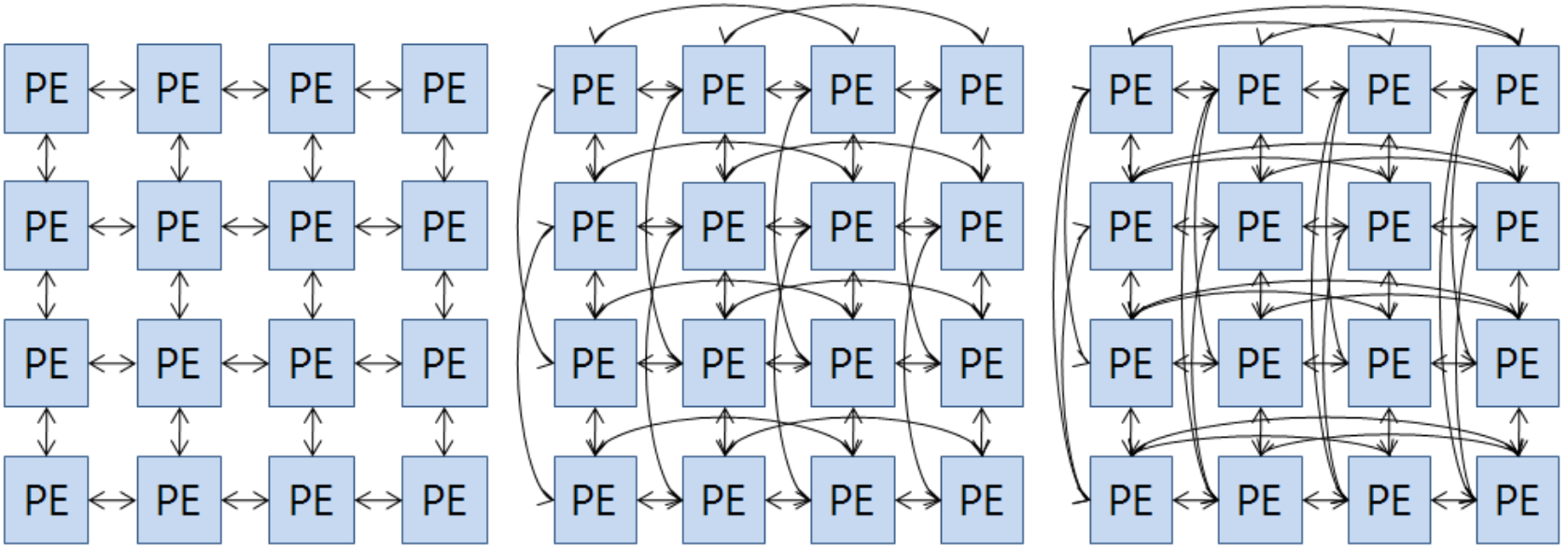


Kernel Mapping onto RCM

- Approaches
  - Integer linear programming (ILP)
  - List scheduling
  - Evolutionary algorithm
    - Quantum-inspired Evolutionary Algorithm (QEA)
  - Mixed
    - List scheduling + iterative improvement (QEA)

Kernel Mapping onto RCM

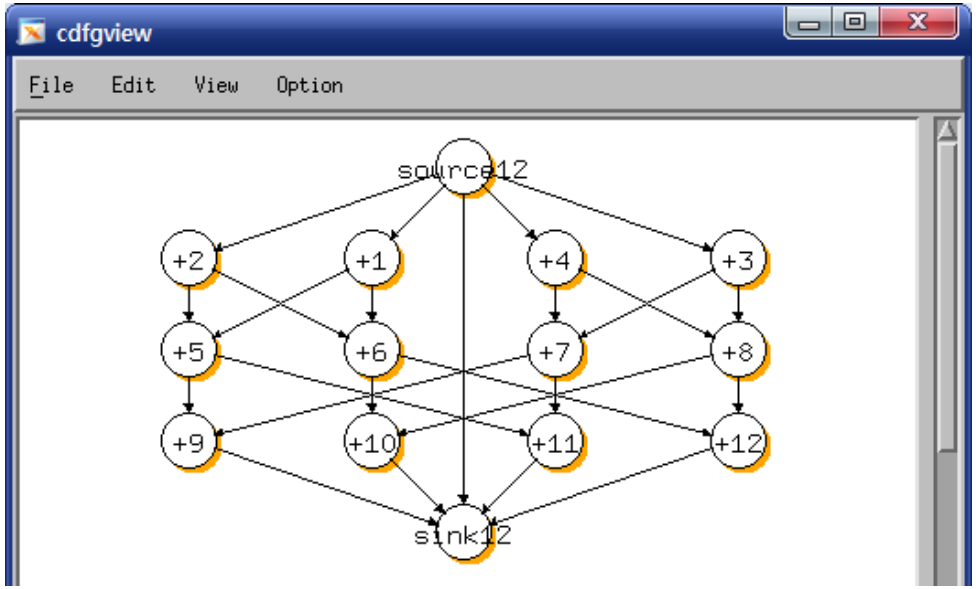
- Design space exploration
  - Interconnection topology



(a) Mesh (b) Mesh-plus (c) Full

Kernel Mapping onto RCM

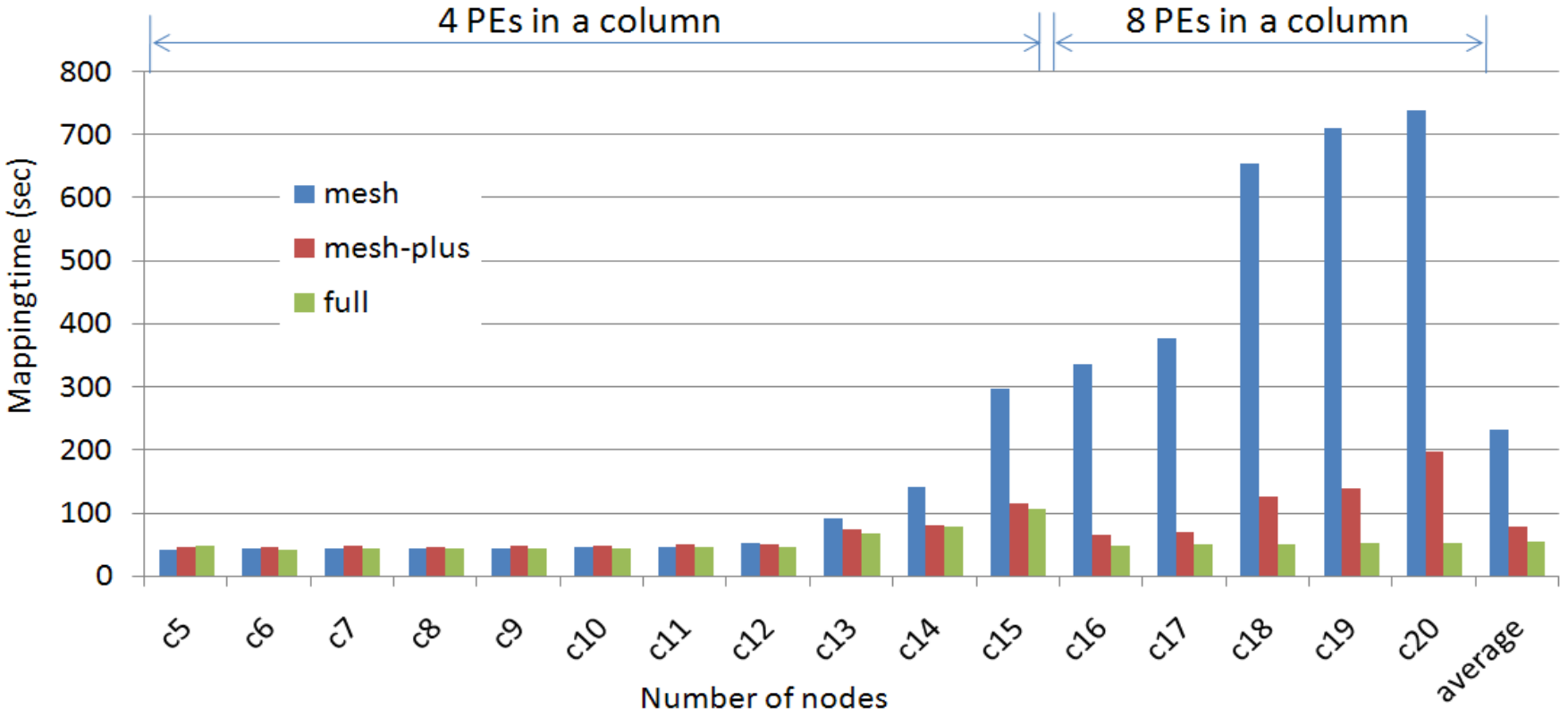
- Performance by interconnection topology



	MESH	M-PLUS	FULL
Latency (cycle)	4	3	3

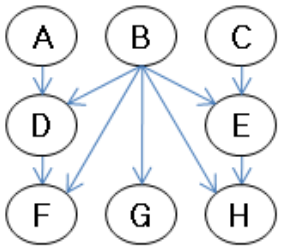
Kernel Mapping onto RCM

- Mapping time by interconnection topology

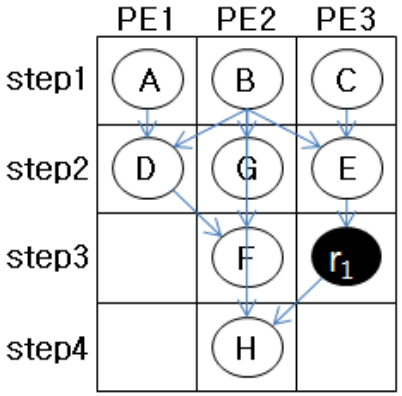


Kernel Mapping onto RCM

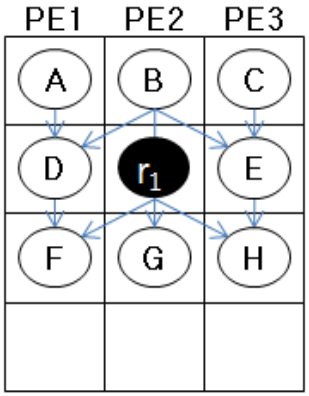
- ILP vs. Mixed, Spanning tree vs. Steiner tree



Data flow graph



Spanning tree

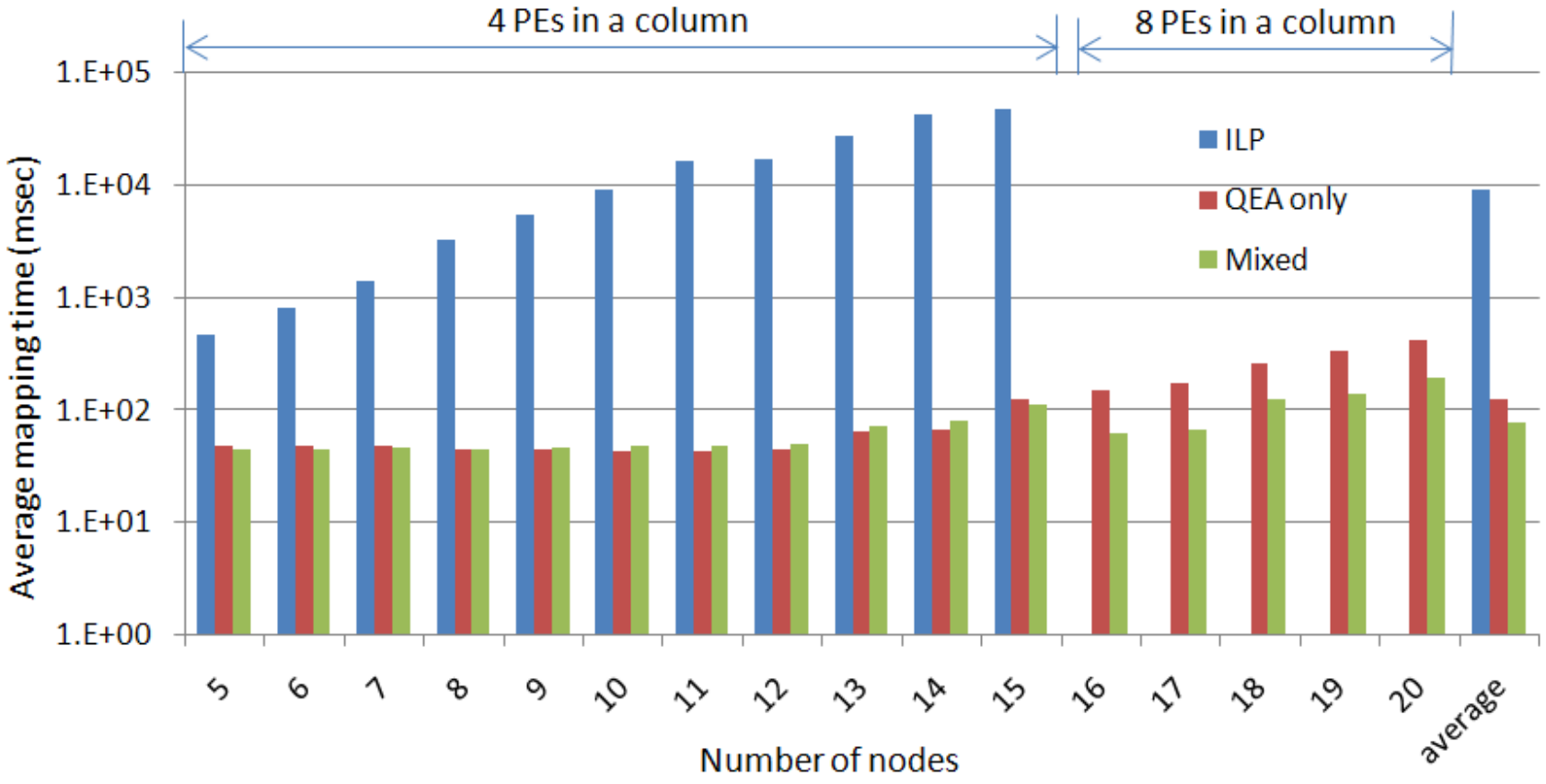


Steiner tree

		Latency (cycle)	Mapping time (second)
ILP	Spanning tree	4	1022
	Steiner tree	3	965
Mixed	Spanning tree	4	≤ 1
	Steiner tree	3	≤ 1

Kernel Mapping onto RCM

- Mapping time (randomly generated examples)



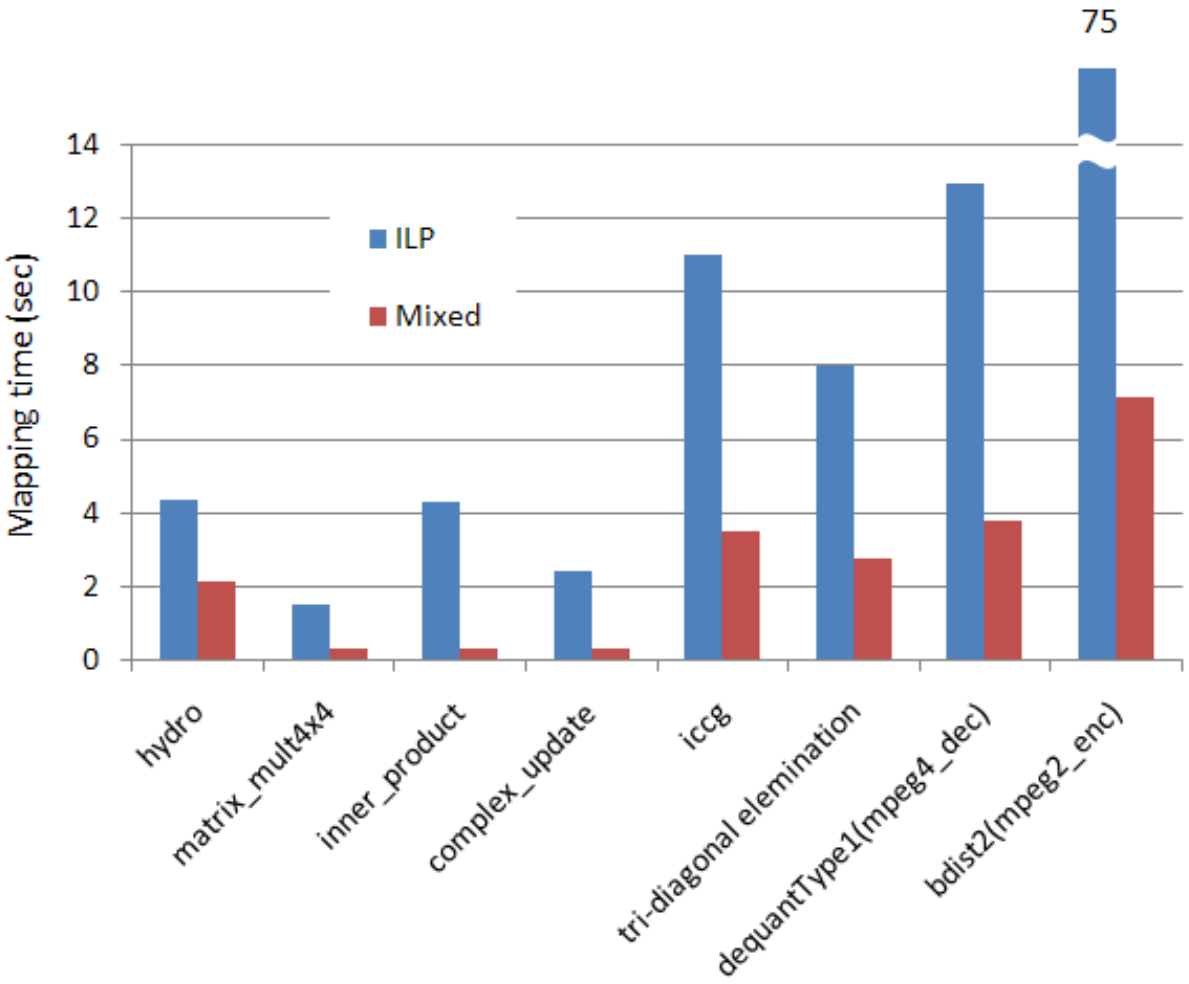
Kernel Mapping onto RCM

- Mapping time for deblocking filter

	Latency (cycle)	Mapping time (second)
QEA only	18	235.34
Mixed (List + QEA)	15	1.08

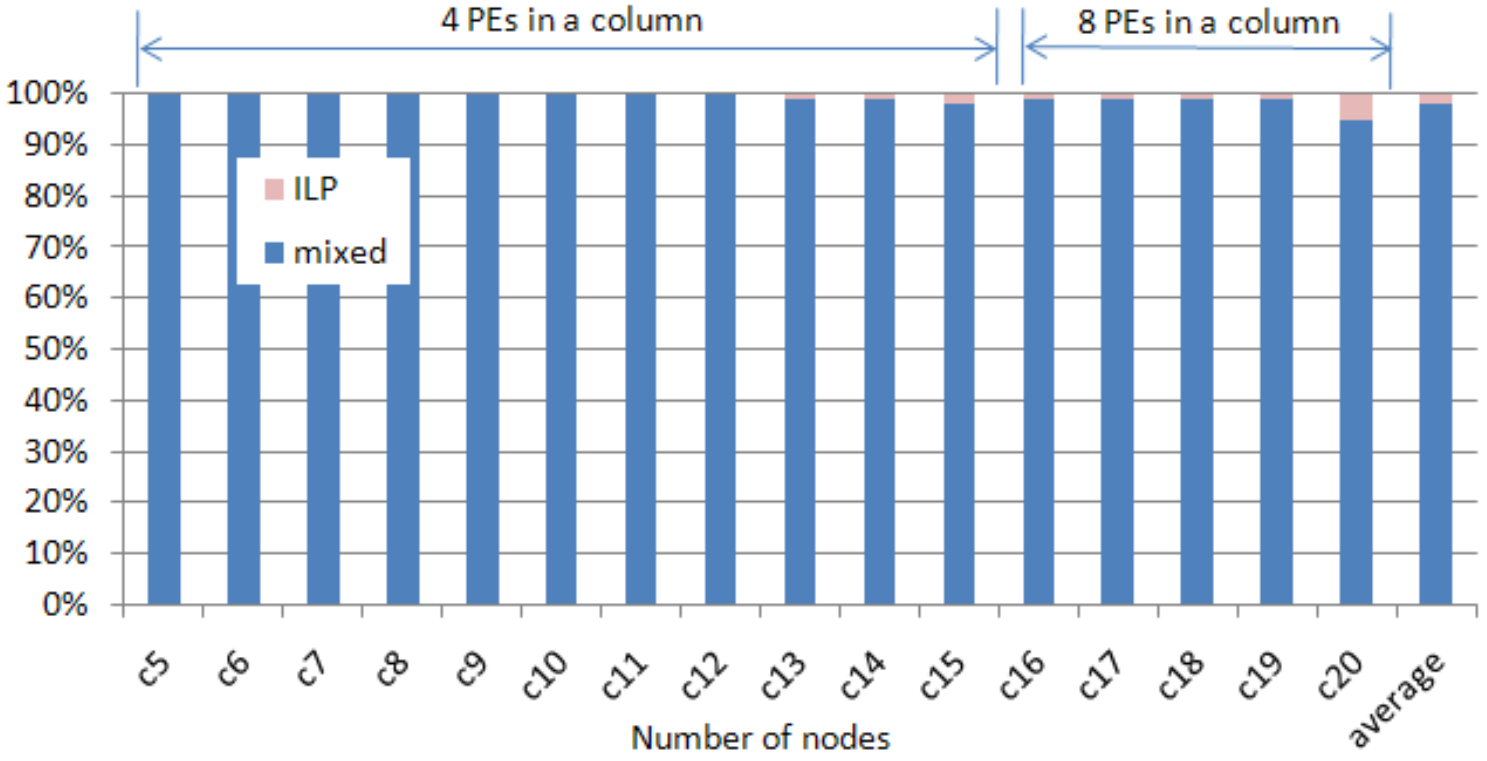
Kernel Mapping onto RCM

- Mapping time of real benchmarks



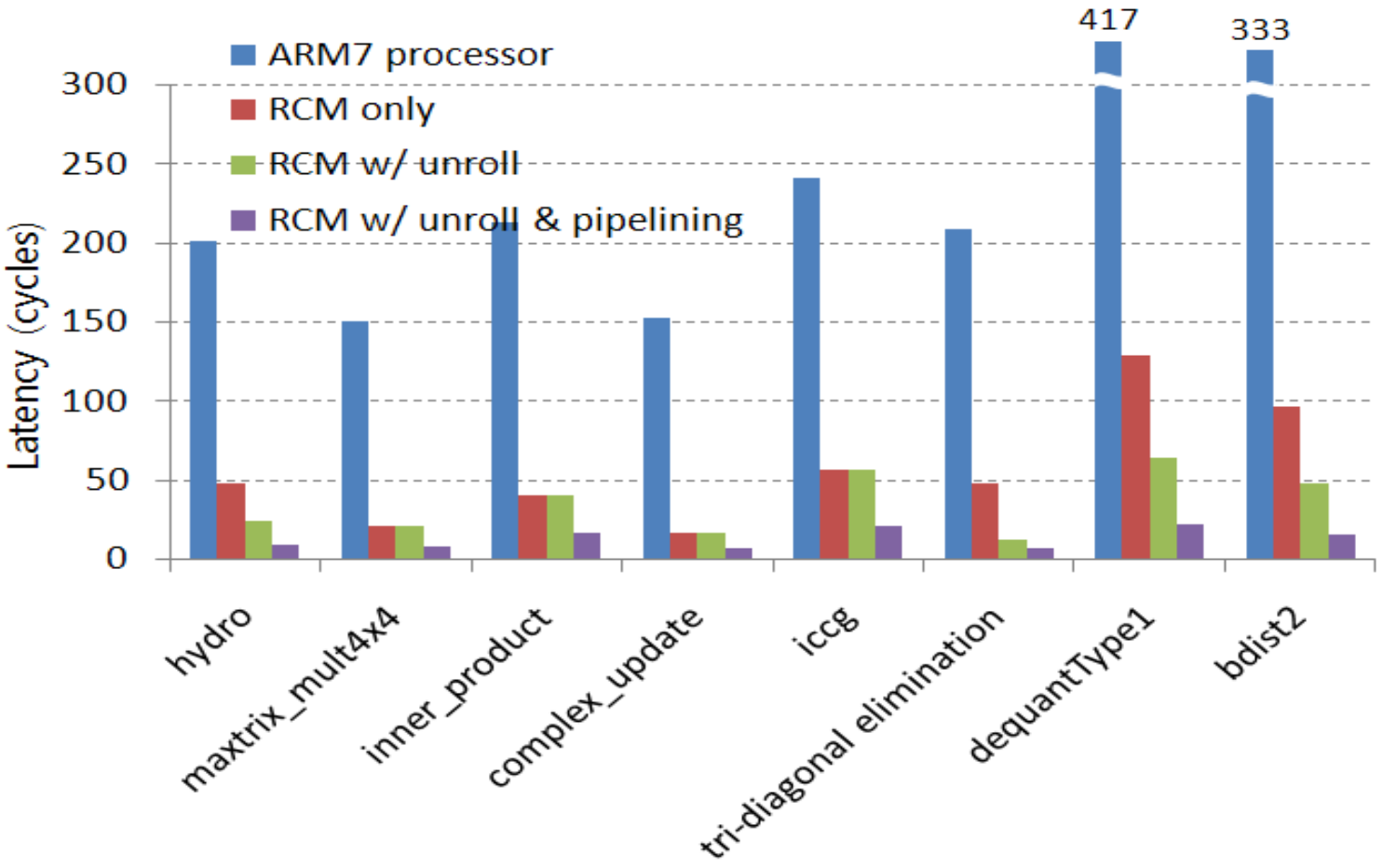
Kernel Mapping onto RCM

- Optimality of mixed approach



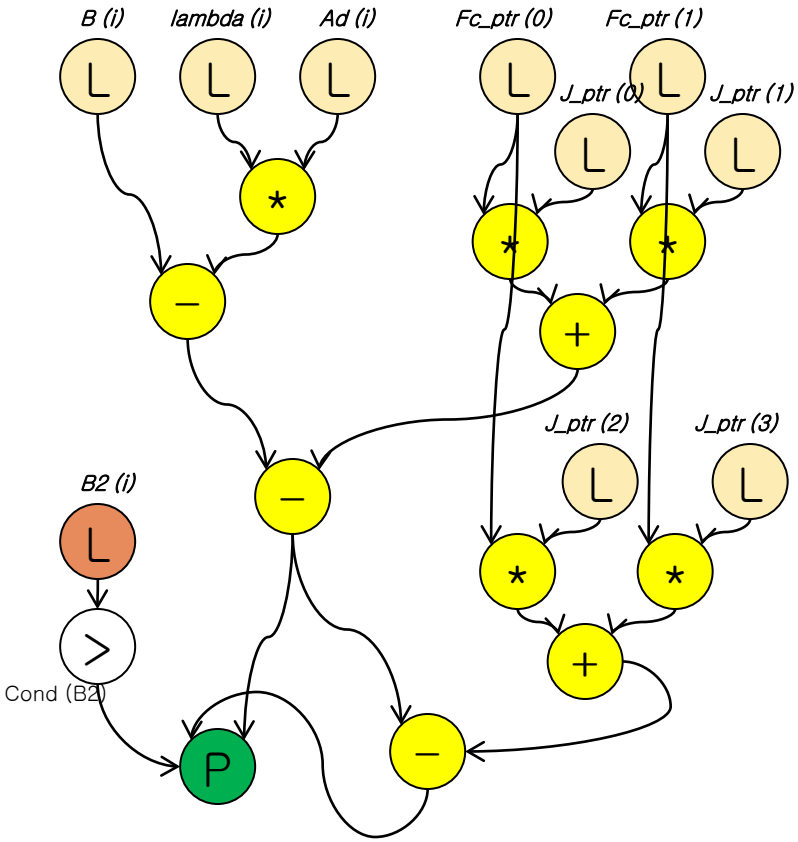
Kernel Mapping onto RCM

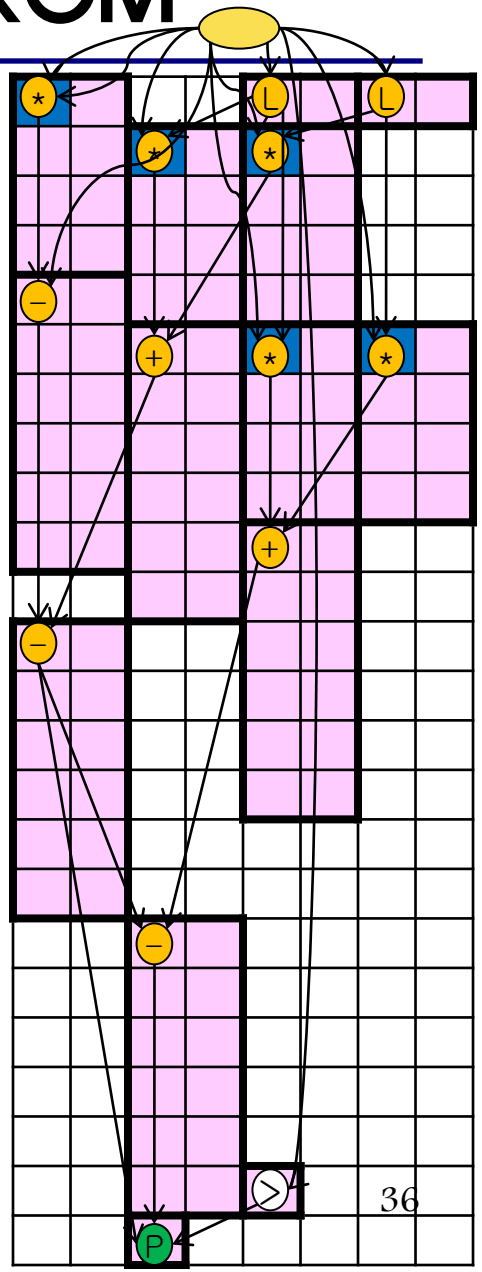
- Latency of real benchmarks



Kernel Mapping onto RCM

- Mapping floating-point operations
  - Crash-wall (3D physics engine)





Conclusion

- FloRA
  - Coarse-grained reconfigurable architecture
  - Floating-point operation
  - Memory-centric communication
- Application mapping
  - Temporal mapping
  - Loop unrolling and pipelining
  - Routing PE
    - Find a solution without any global registers
  - List scheduling followed by QEA
  - Optimal solution for 98.8 % of cases
- Future work
  - Speculation
  - Full integration into SoCDAL
  - Optimization of memory-centric communication architecture
  - Library-based mapping