Cornell University
Computer Systems Laboratory

Accommodating Software Diversity in Chip Multiprocessors

José F. Martínez

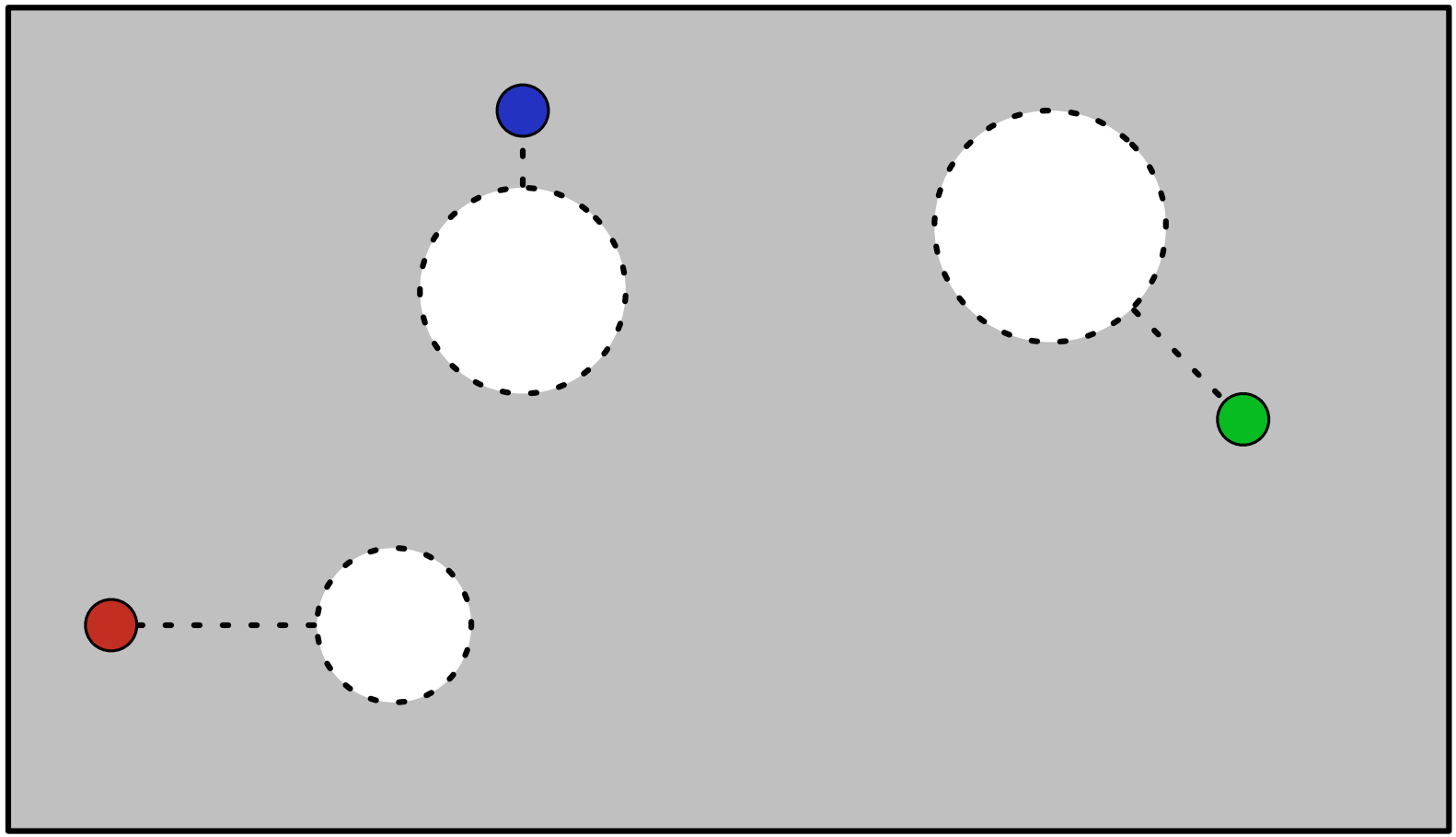
M³ Architecture Research Group
http://m3.csl.cornell.edu/

Introduction

Multicore

- In multicore, CPU is Moore's Law's "new transistor"
- Some important advantages over monolithic designs
  - Circuit locality (short CPU wires)
  - Manageable complexity (modularity)
  - Potential for power/area efficiency
  - Fault resilience (redundancy)
- But no free meal
  - Programmability, efficiency, versatility, ...

A Simplified View of Versatility

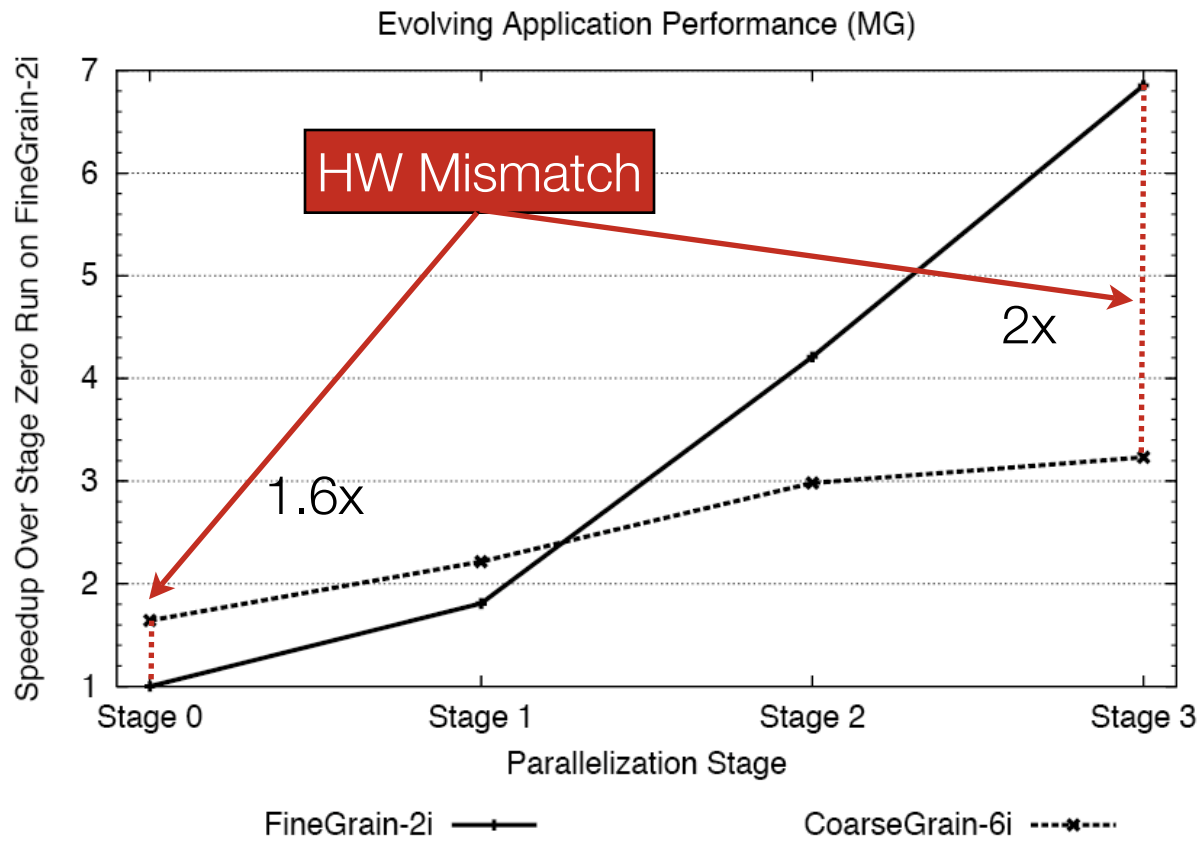


1st-order Versatility in CMPs

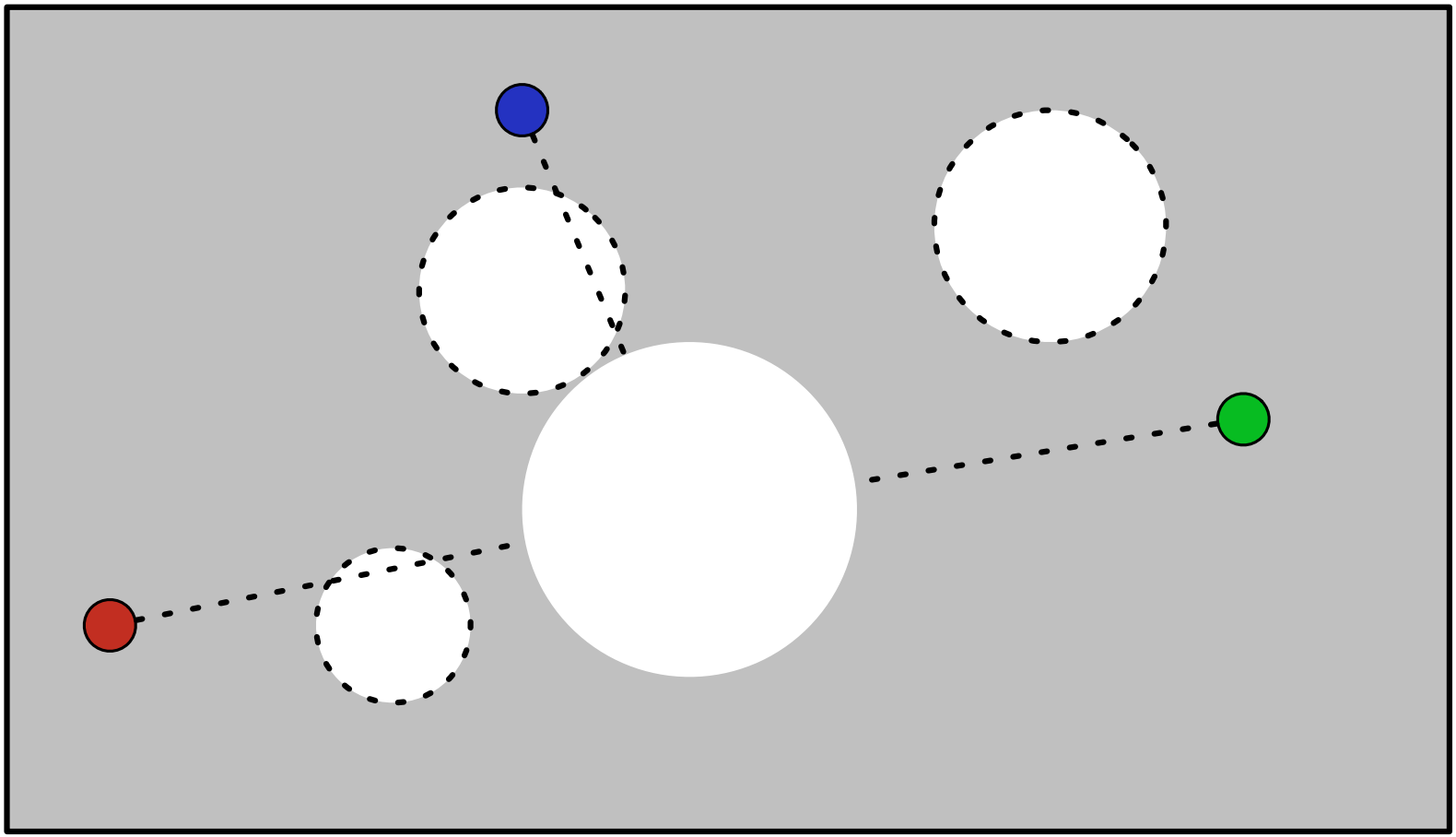
- CMPs must support diverse apps
  - Sequential
  - Multiprogrammed
  - Parallel (high or low)
  - *Evolving*
- Conflicting requirements
  - No. of cores
  - Per-core performance
- Should SW bridge gap alone?

1st-order Versatility in CMPs

- Must support diverse apps
  - Sequential
  - Multiprogrammed
  - Parallel (high or low)
  - *Evolving*
- Conflicting requirements
  - No. of cores
  - Per-core performance
- Should SW bridge gap alone?



A Design Trade-off

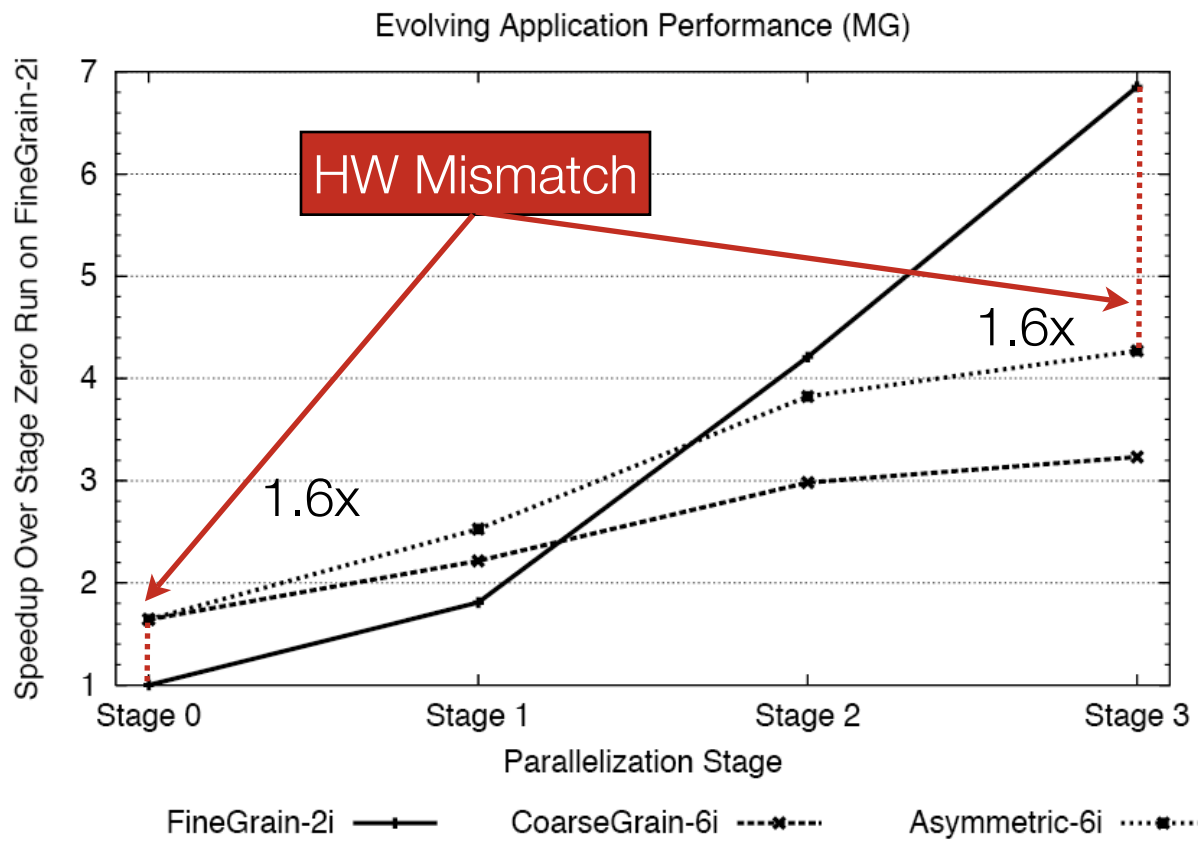


Asymmetric CMPs

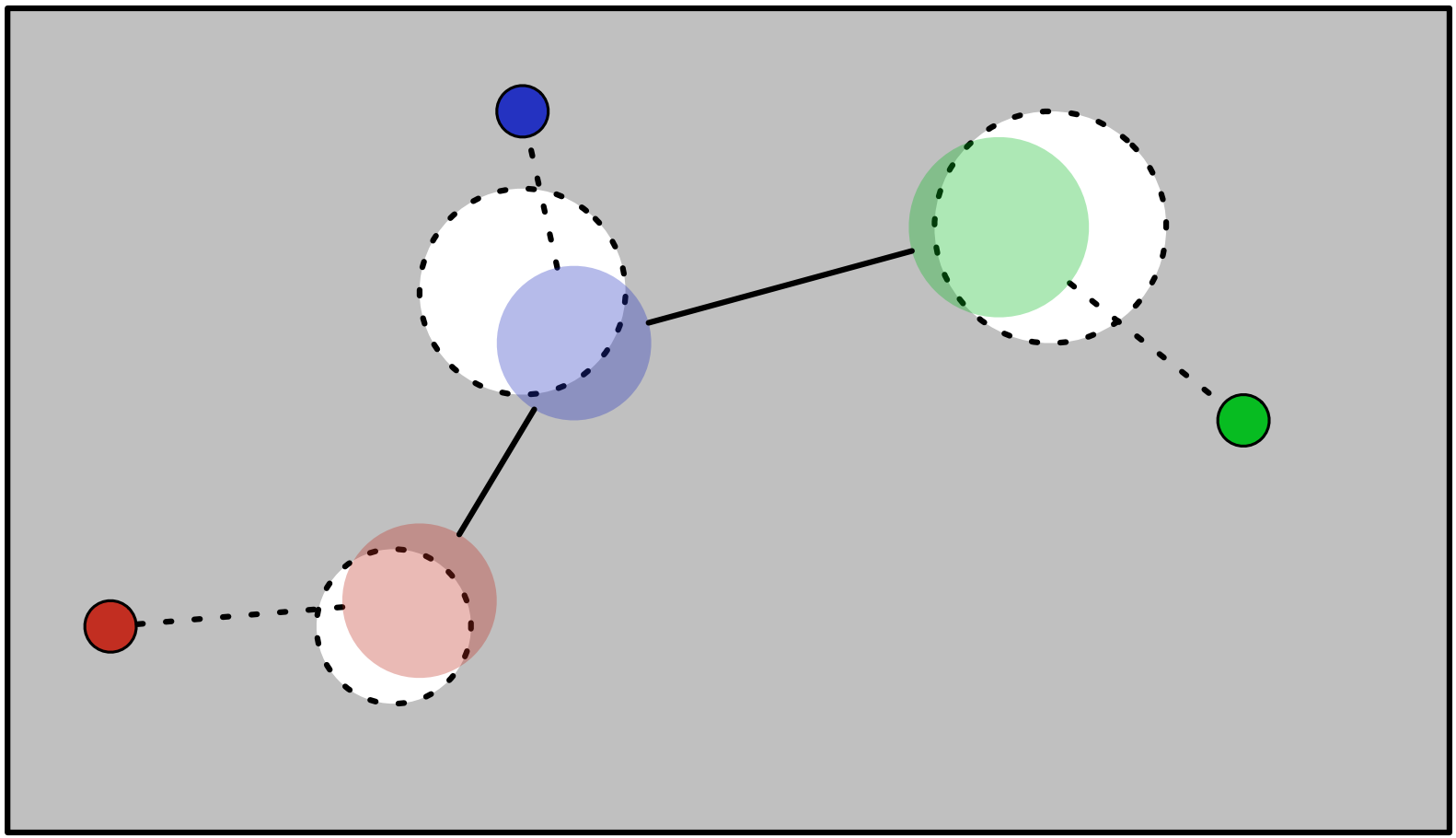
- Multiple core sizes on chip
- Trade-off set at design time
  - No “one size fits all”
- Multiple core designs
- Sophisticated SW
  [Balakrishnan et al., ISCA ’05]

Asymmetric CMPs

- Multiple core sizes on chip
- Trade-off set at design time
  - No “one size fits all”
- Multiple core designs
- Sophisticated SW
  [Balakrishnan et al., ISCA ’05]

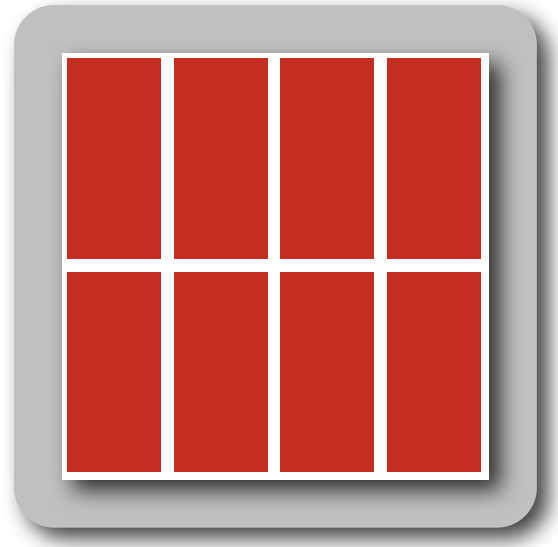


A Reconfigurable Design



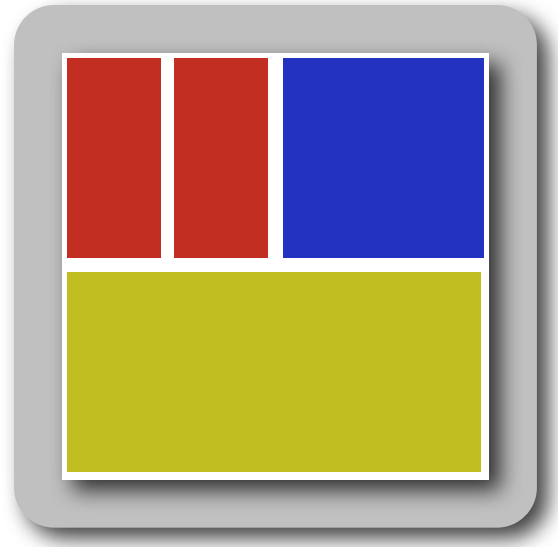
Proposal: Core Fusion

- Run-time CMP "synthesis"
- High compatibility
  - Single execution model
  - Backward-compatible ISA
  - No sophisticated SW support
  - No significant programming effort
- Bottom-up design
  - Optimized for *parallel* codes
  - Better isolation (power, faults, ...)



Proposal: Core Fusion

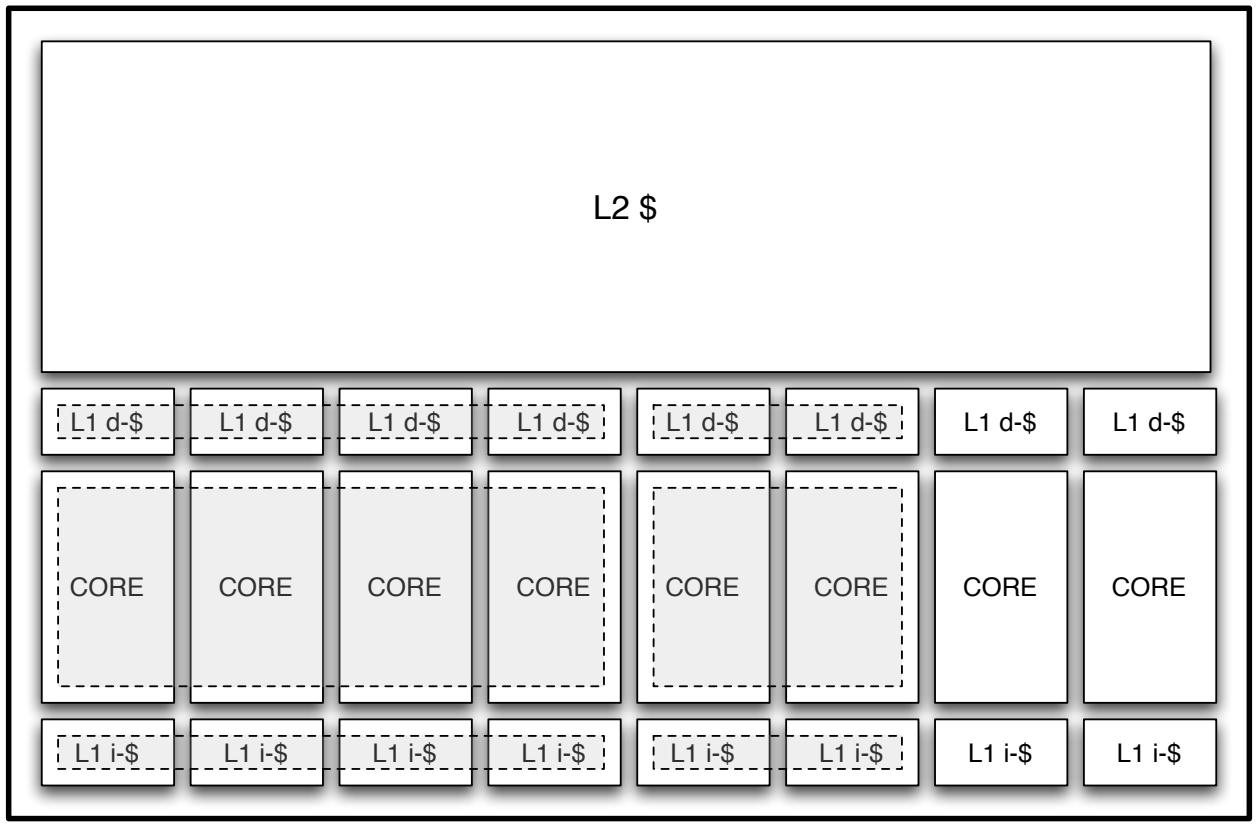
- Run-time CMP “synthesis”
- High compatibility
  - Single execution model
  - Backward-compatible ISA
  - No sophisticated SW support
  - No significant programming effort
- Bottom-up design
  - Optimized for *parallel* codes
  - Better isolation (power, faults, ...)



Core Fusion Architecture

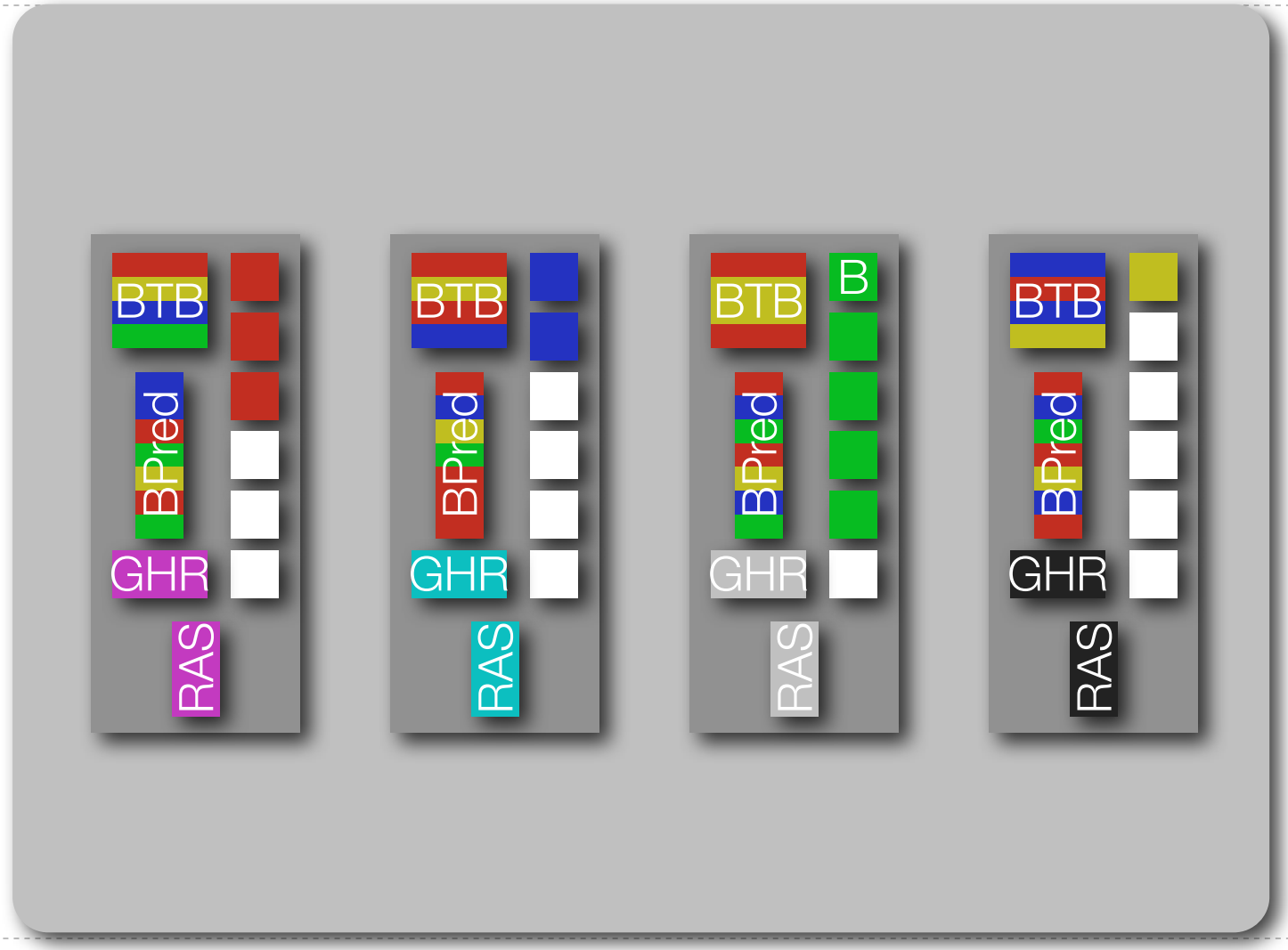
Conceptual Organization

- Concept: Add enveloping hardware to make cores cooperate

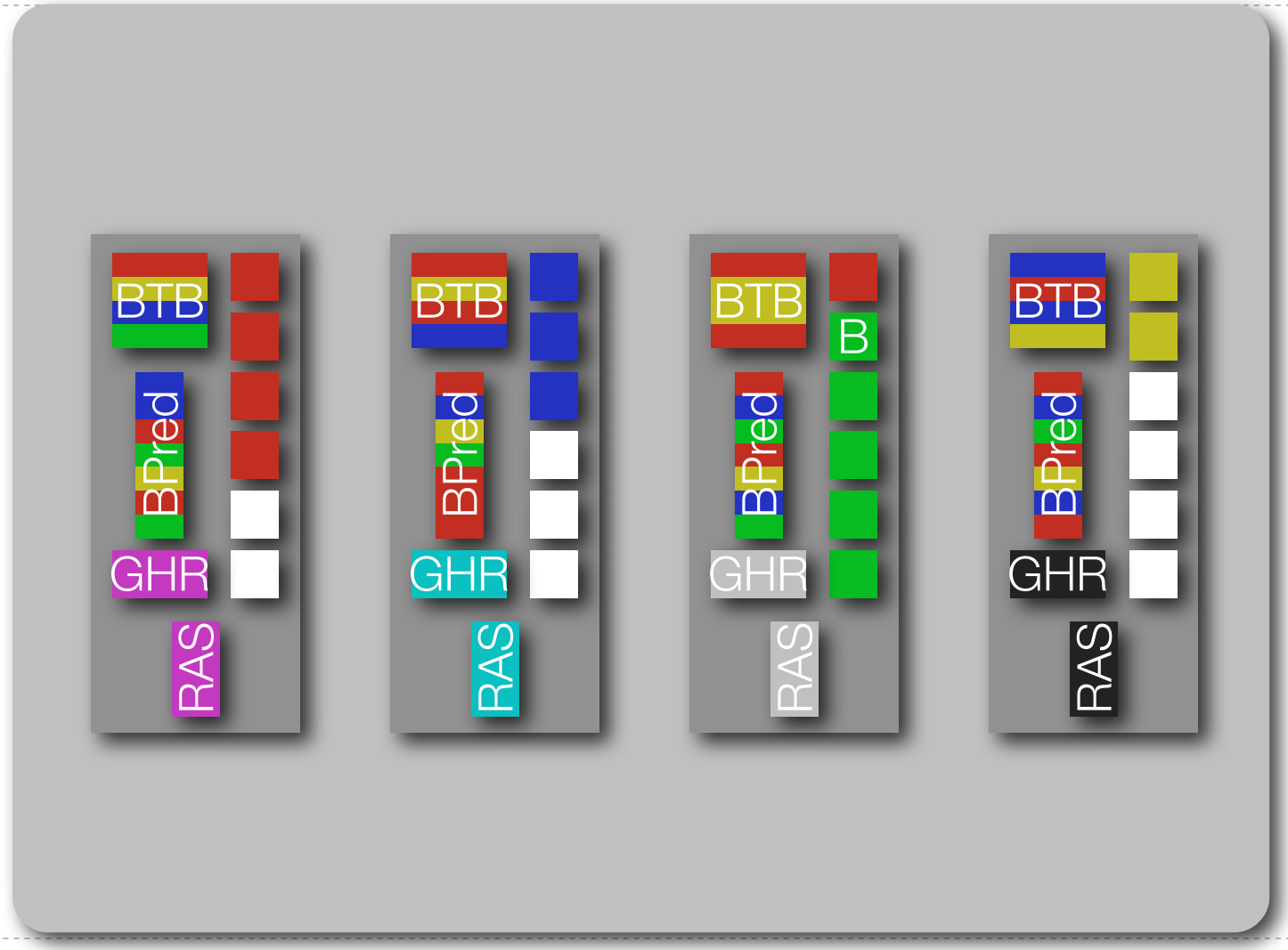


Not meant to represent actual floorplan

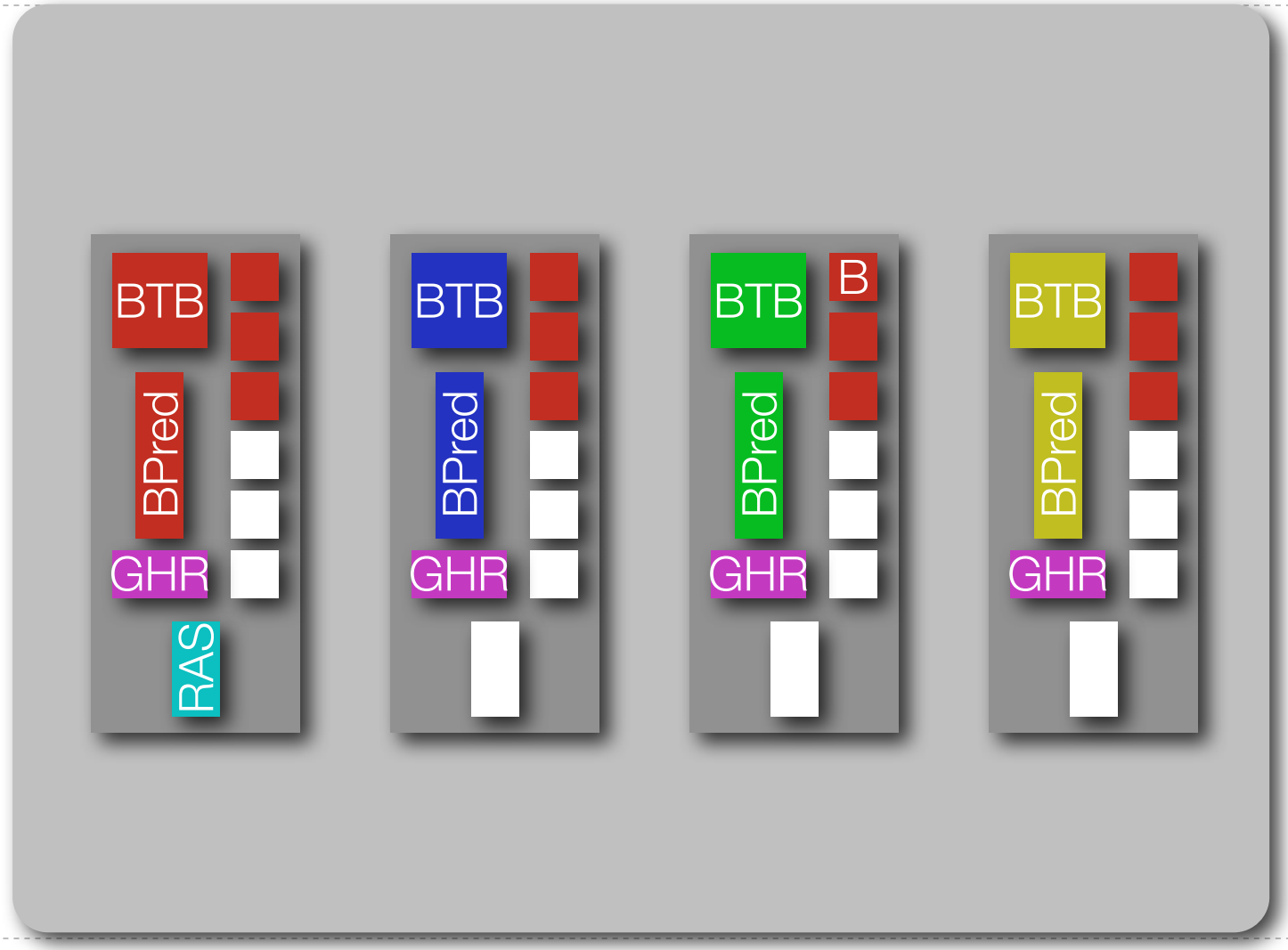
Fetch Mechanism (Split)



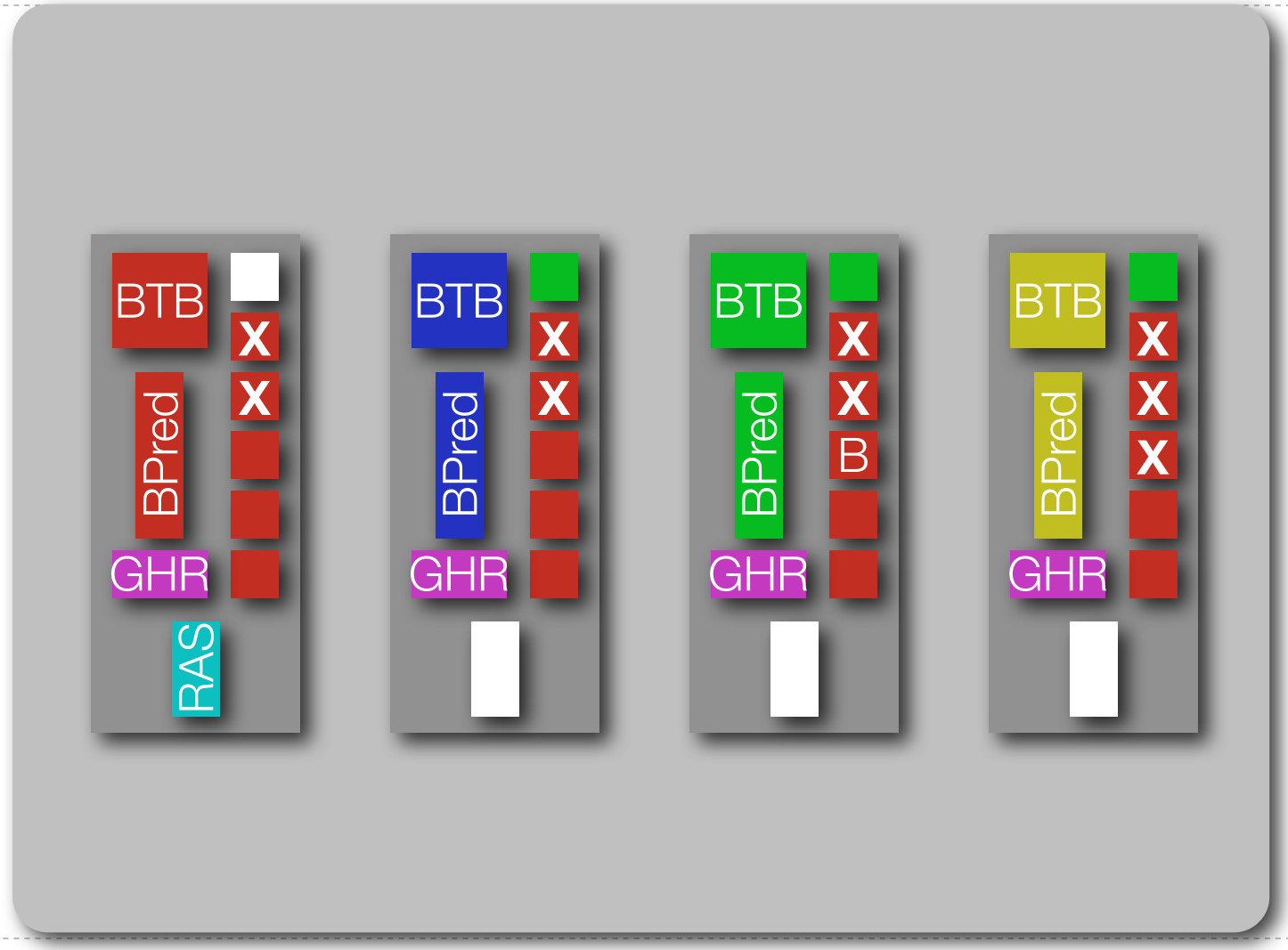
Fetch Mechanism (Split)



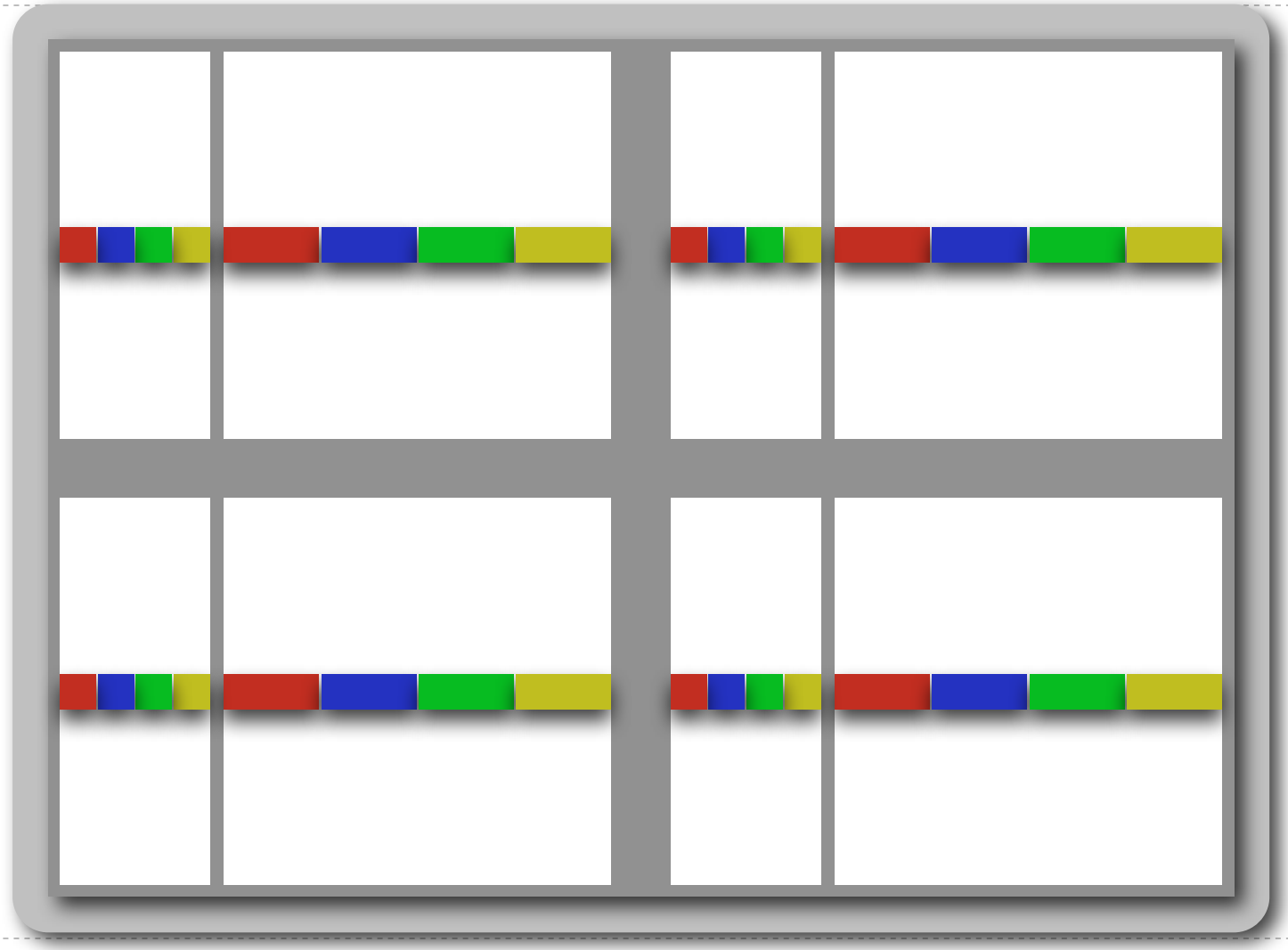
Fetch Mechanism (Fused)



Fetch Mechanism (Fused)

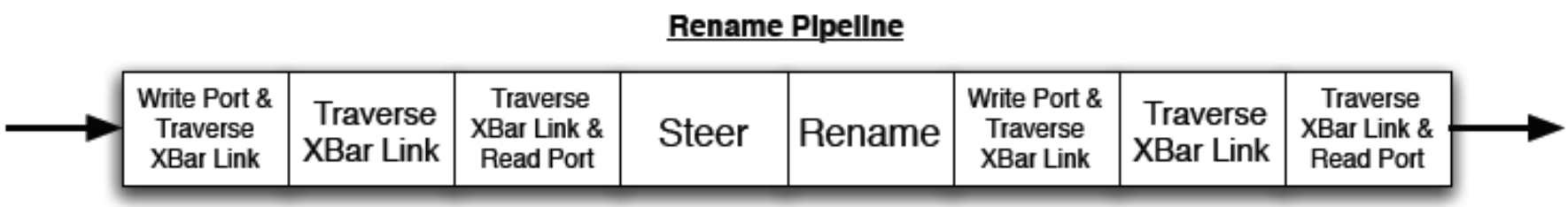


i-Cache (Fused)

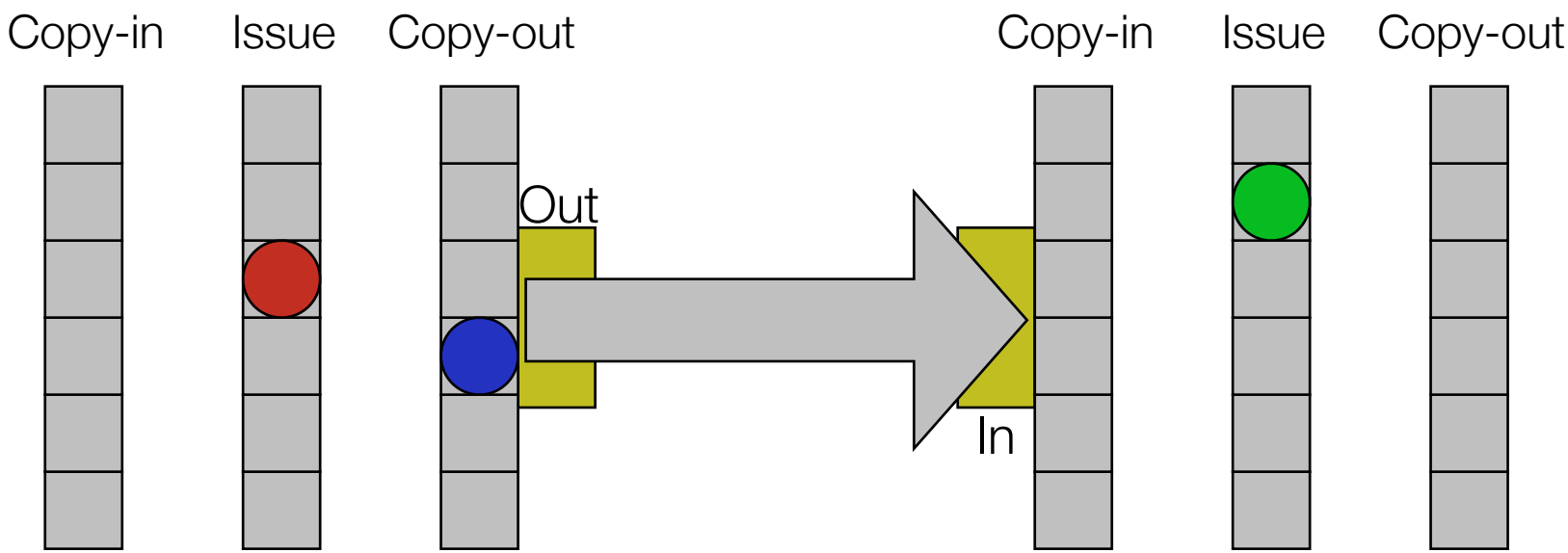


Collective Steer+Rename

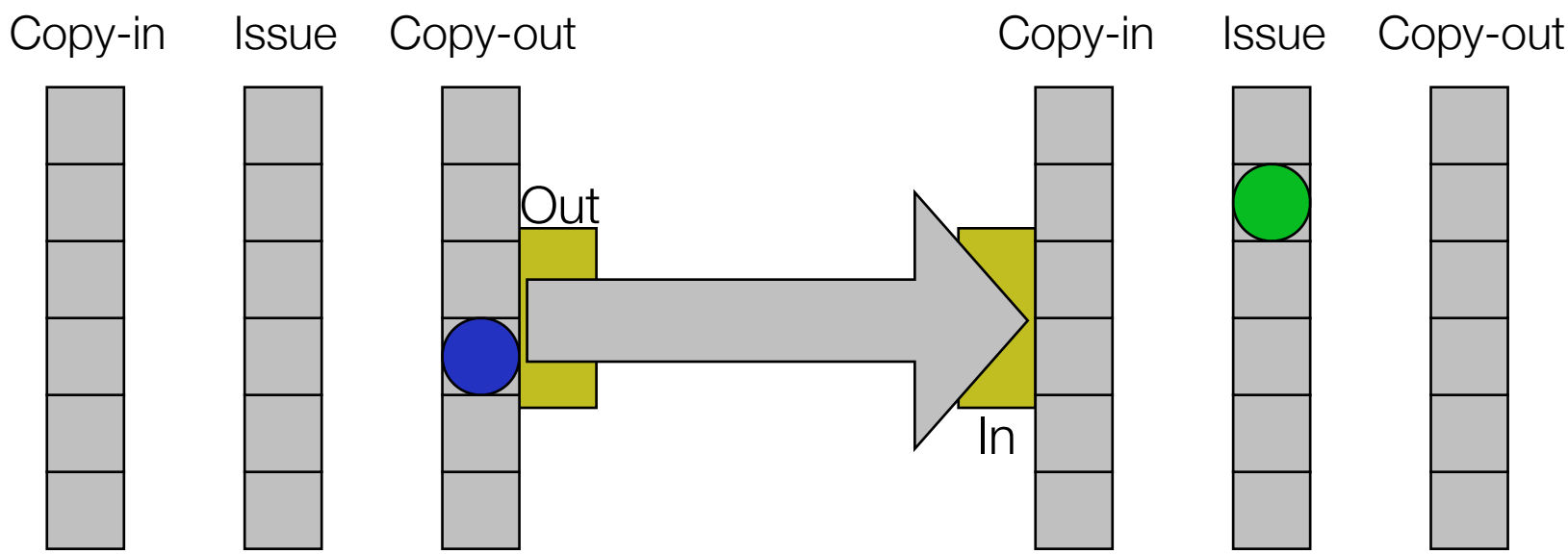
- Cores send predecoded info to Steering Management Unit (SMU)
- SMU steers and dispatches regular and copy instructions
  - Max. two regular + two copy instructions per core, cycle
- Eight extra pipeline stages (only fused mode)
- More wires than FMU → three-cycle interconnect



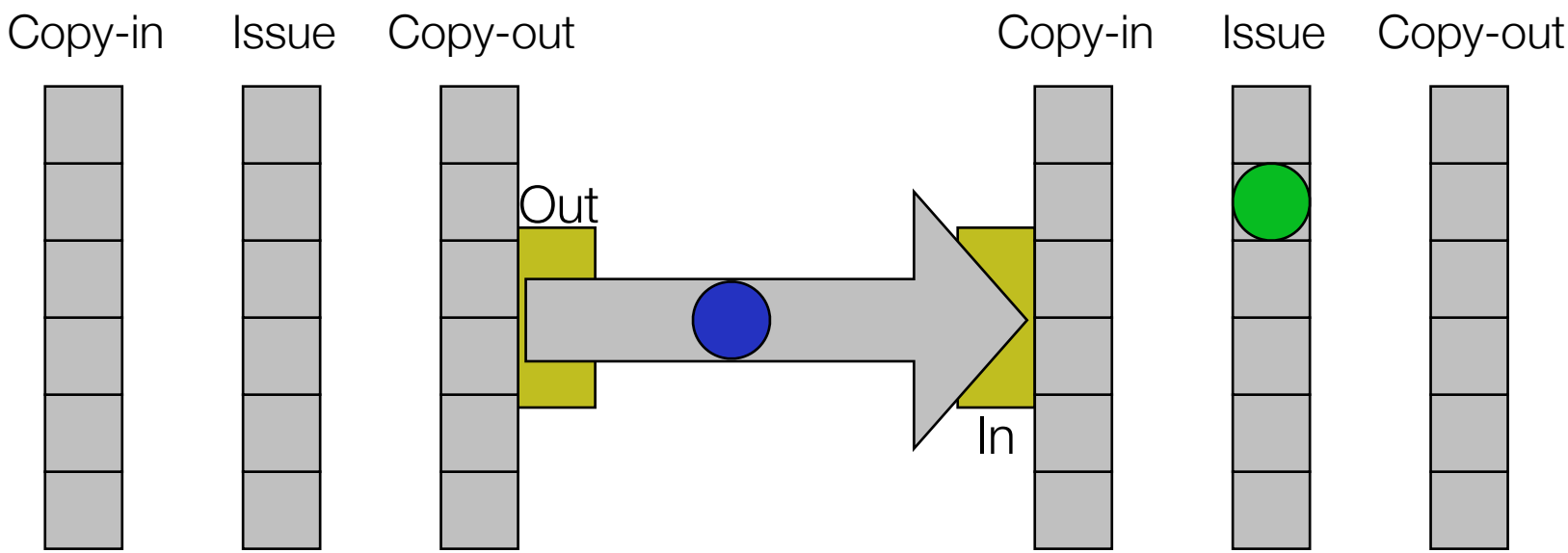
Collective Execution



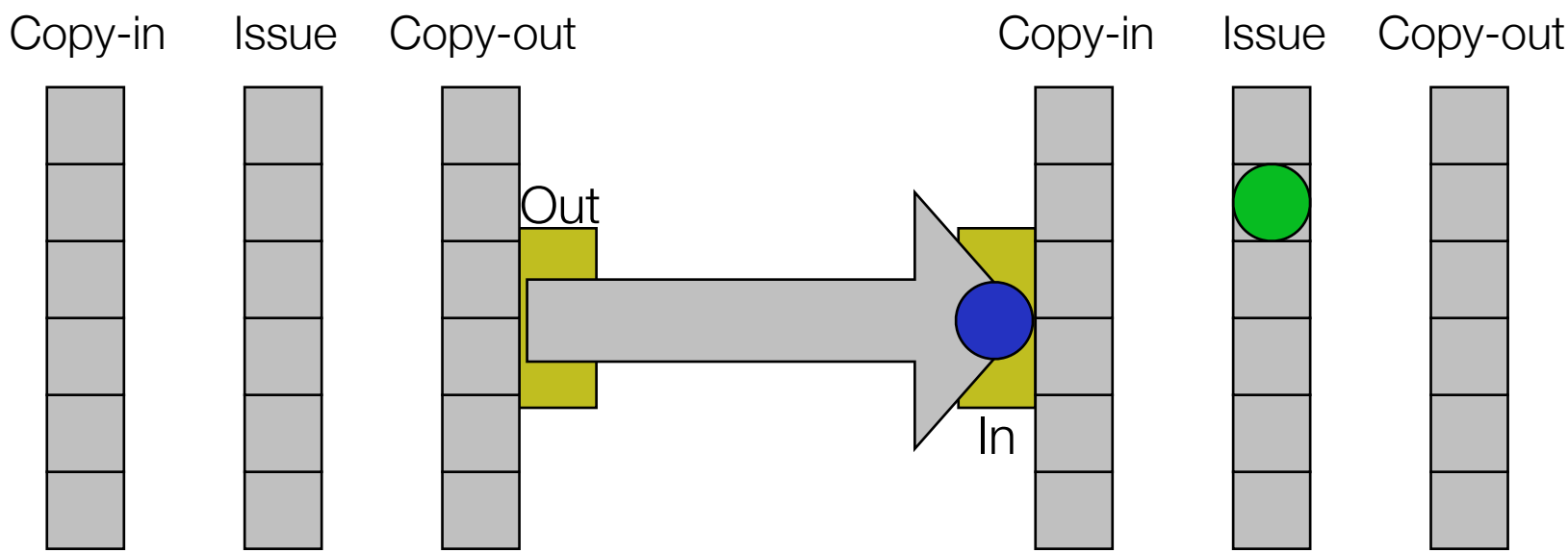
Collective Execution



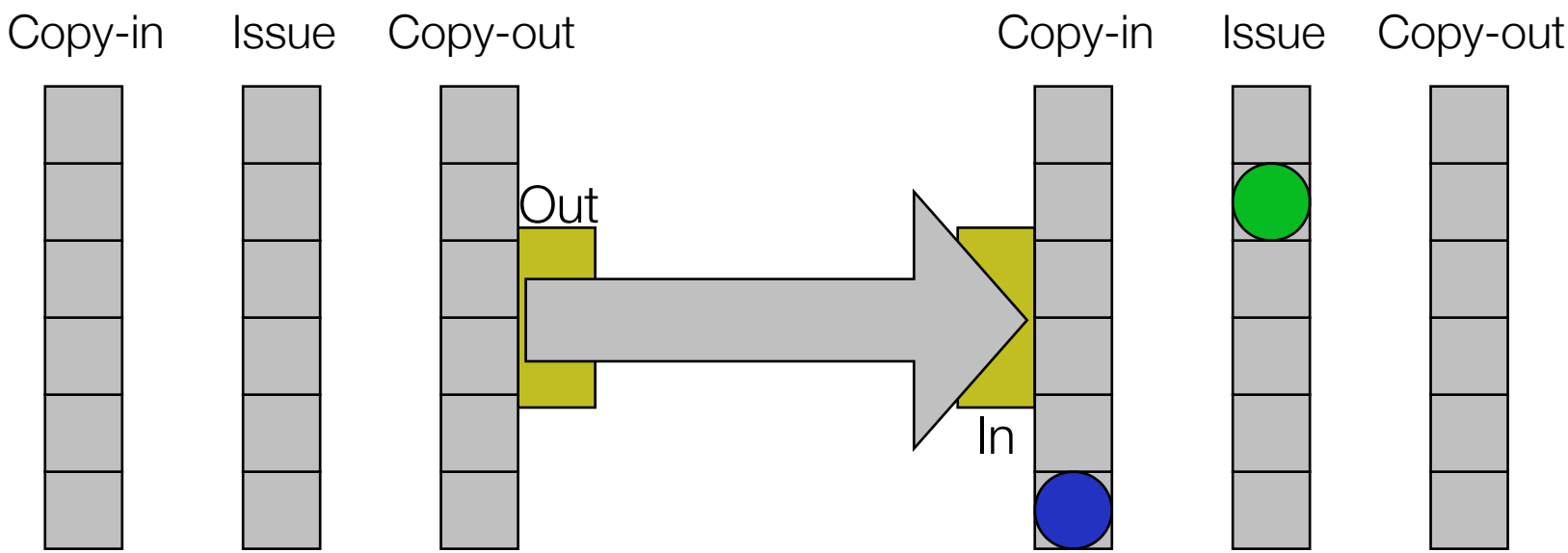
Collective Execution



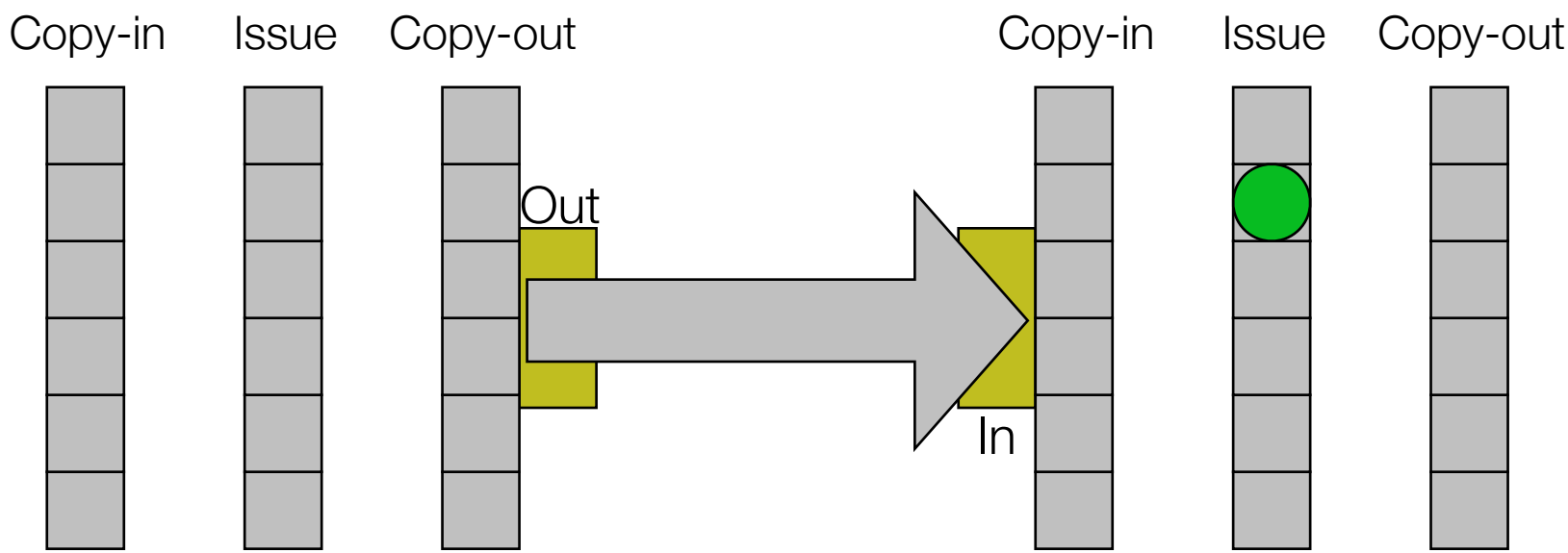
Collective Execution



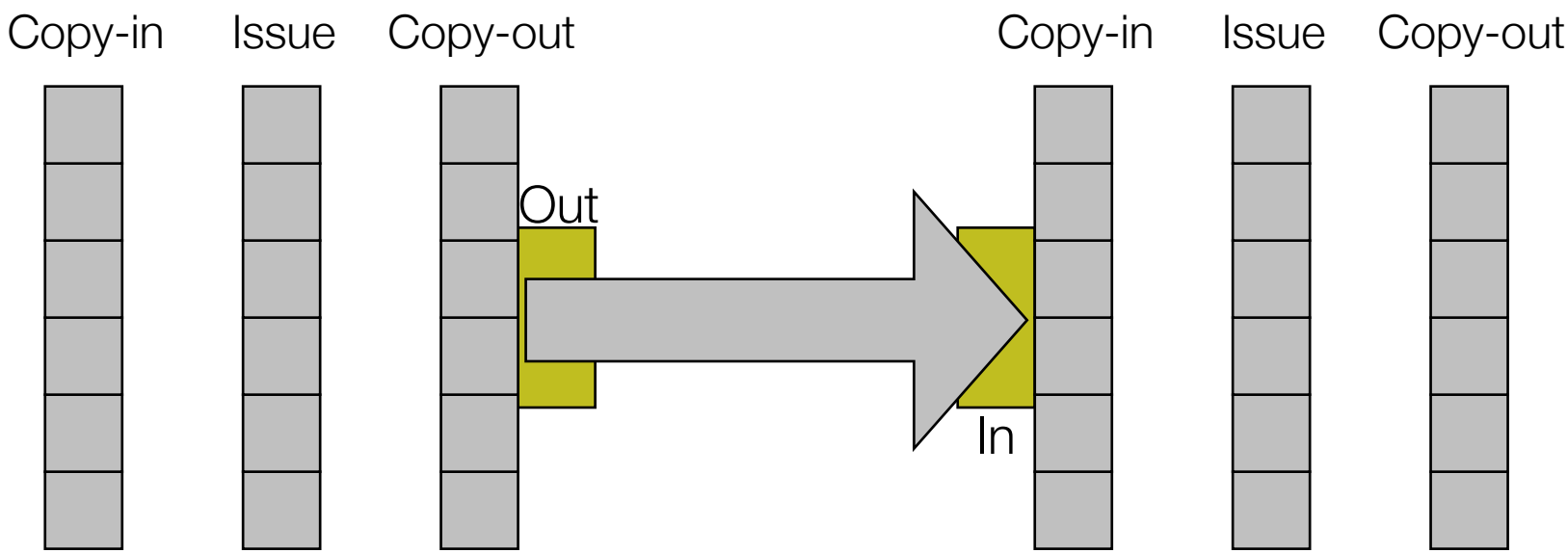
Collective Execution



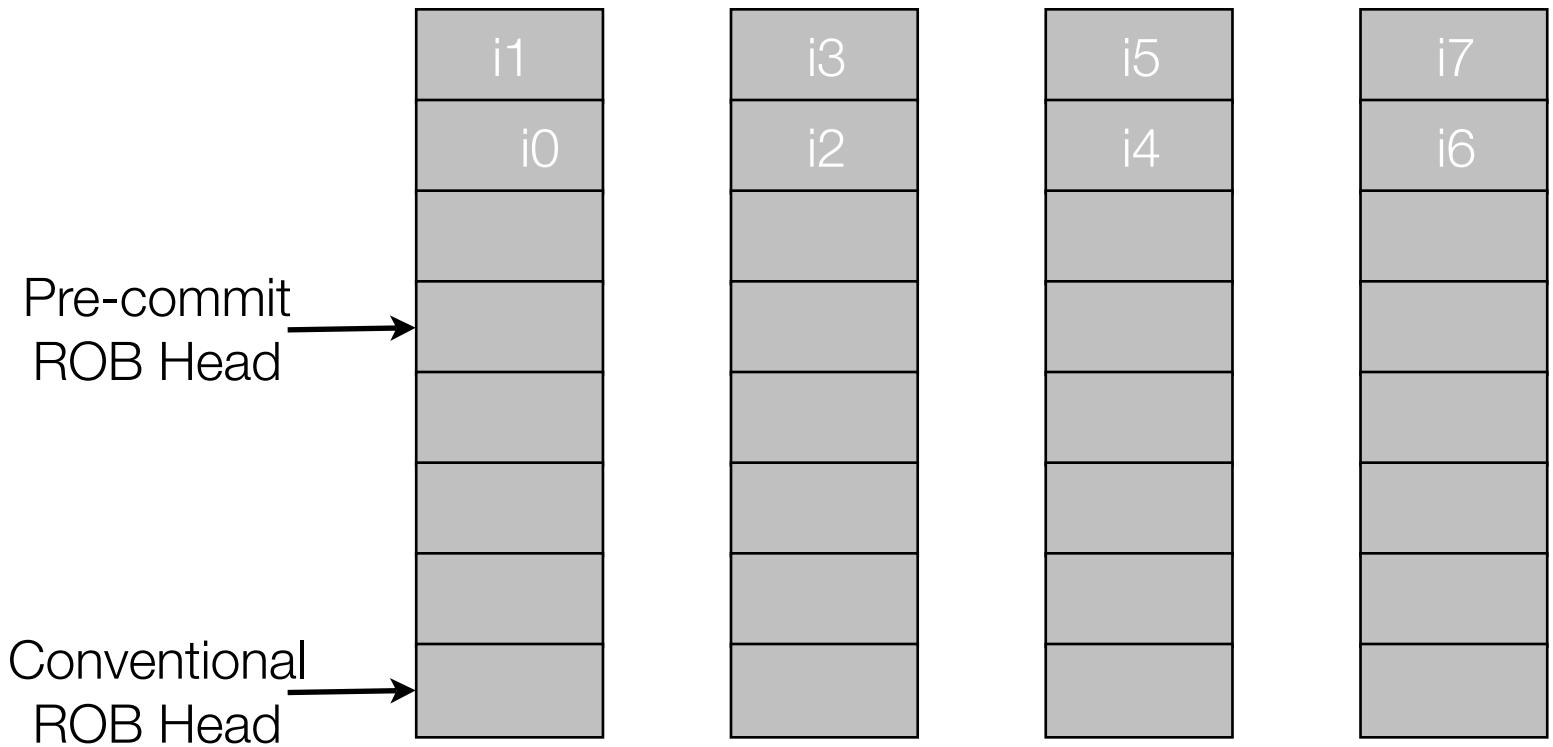
Collective Execution



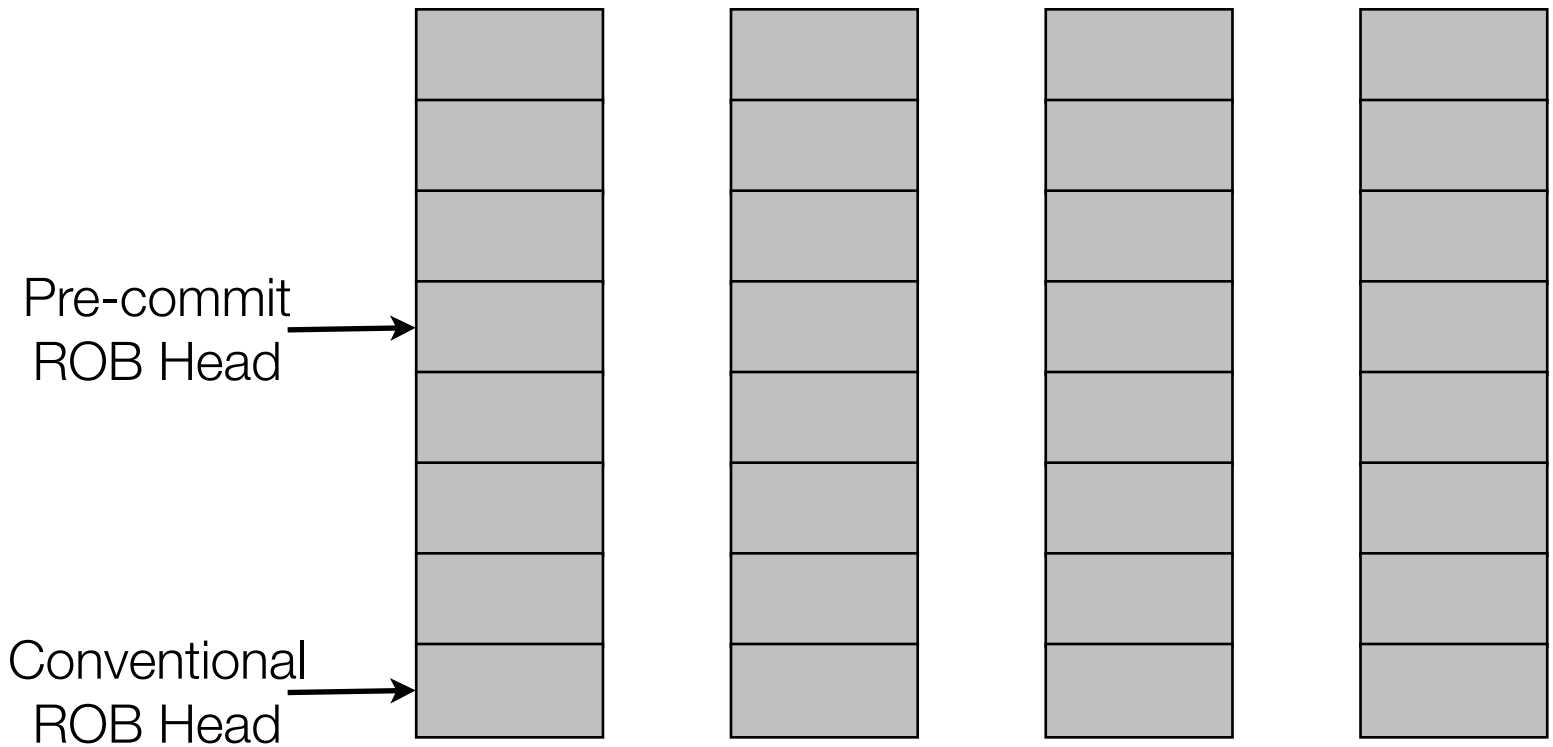
Collective Execution



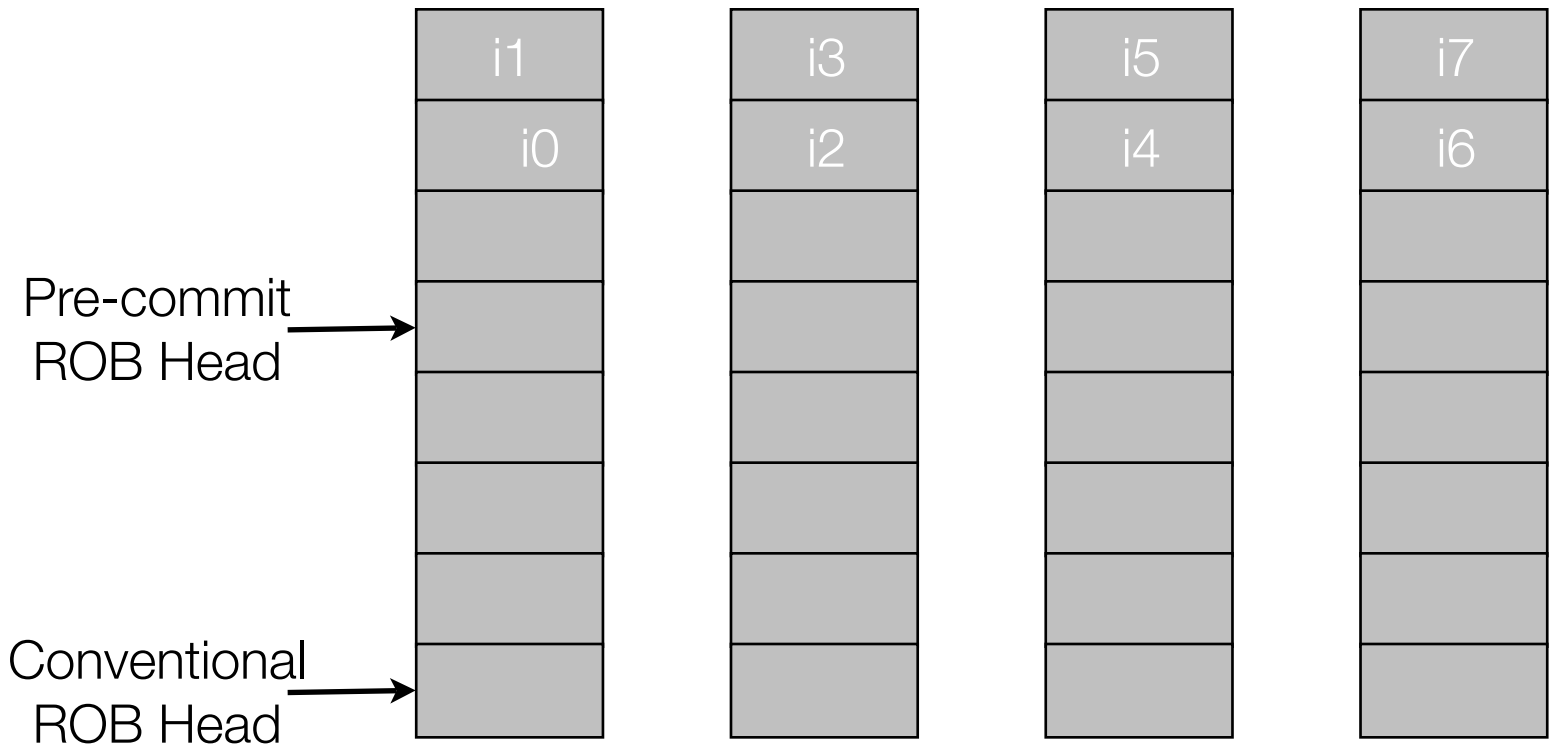
Collective Commit I



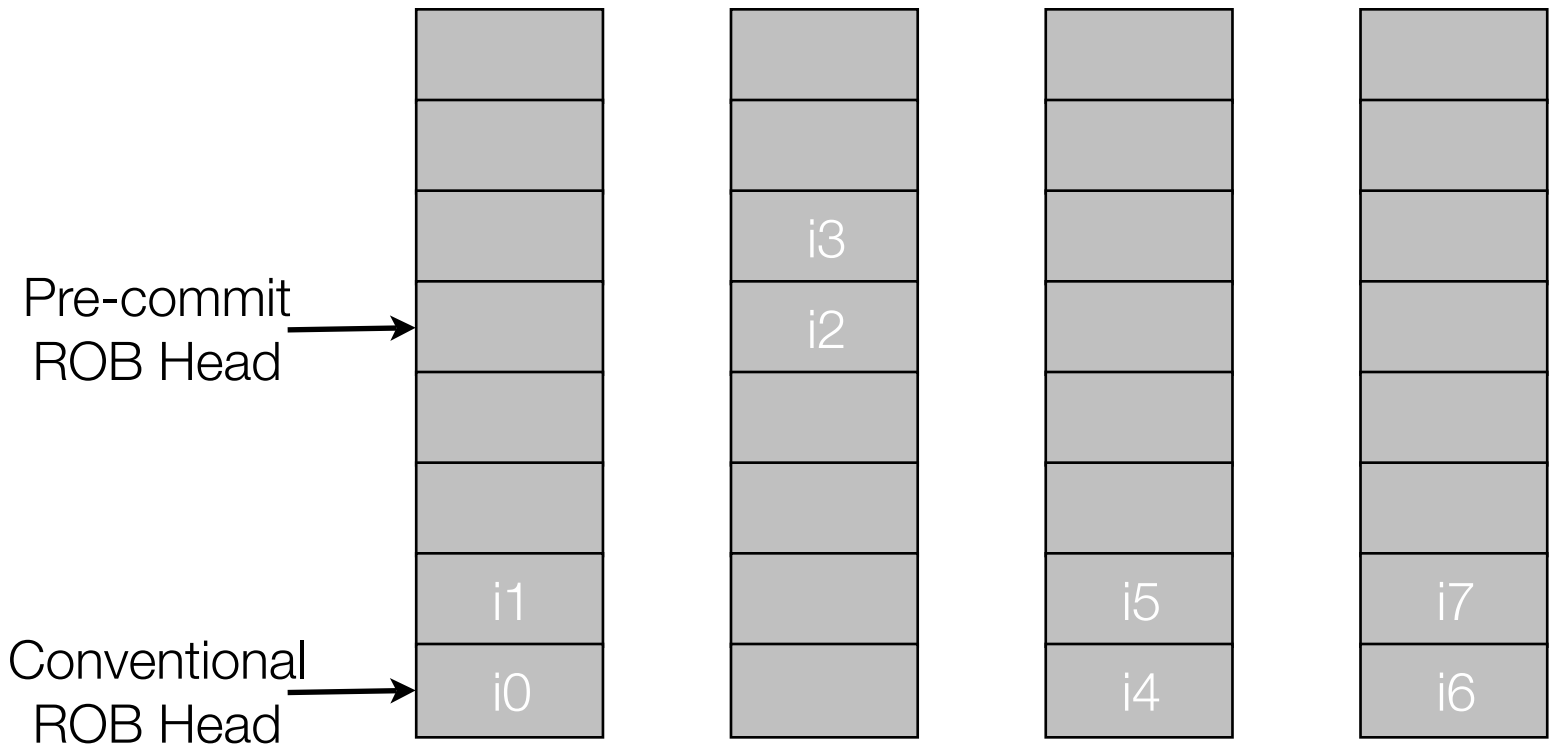
Collective Commit I



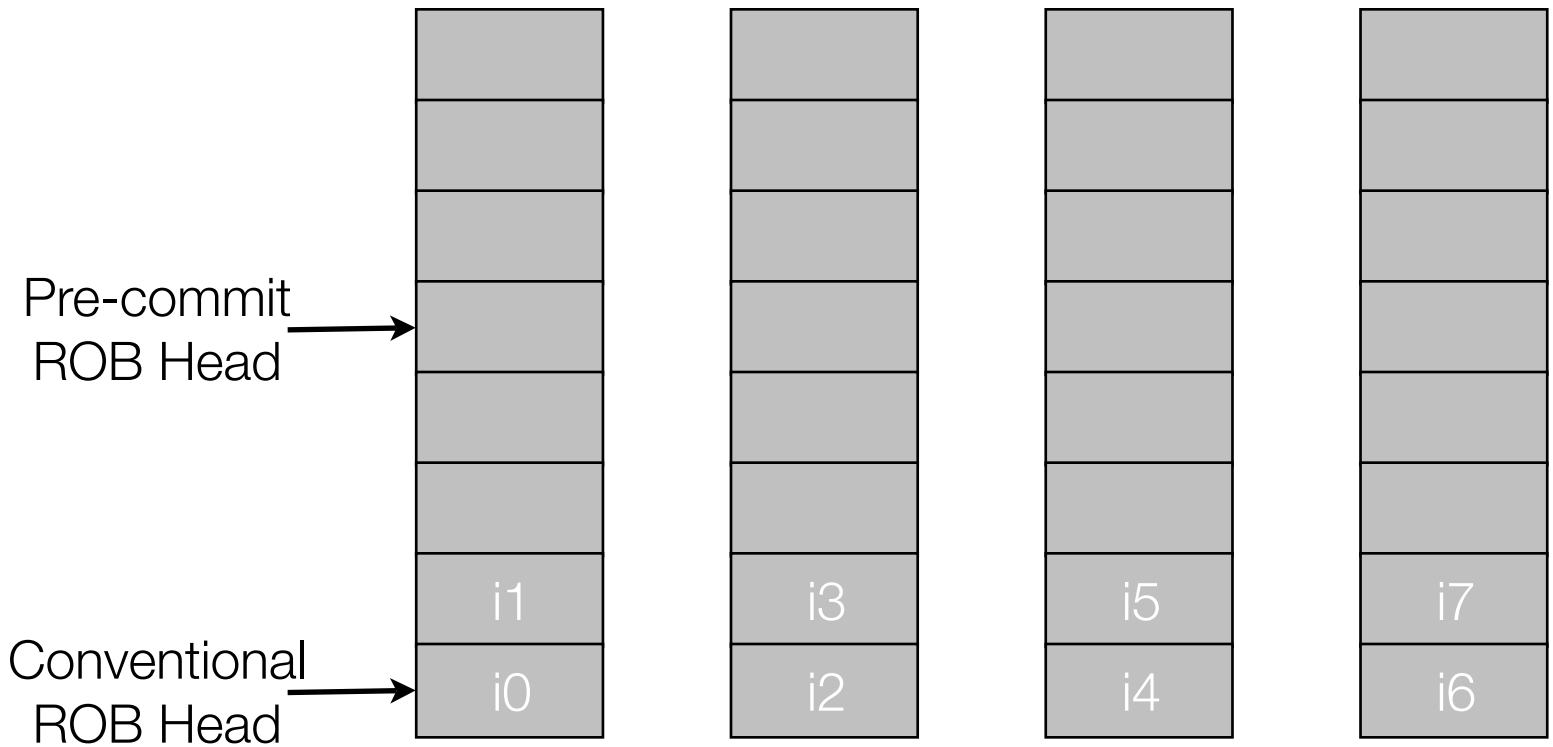
Collective Commit II



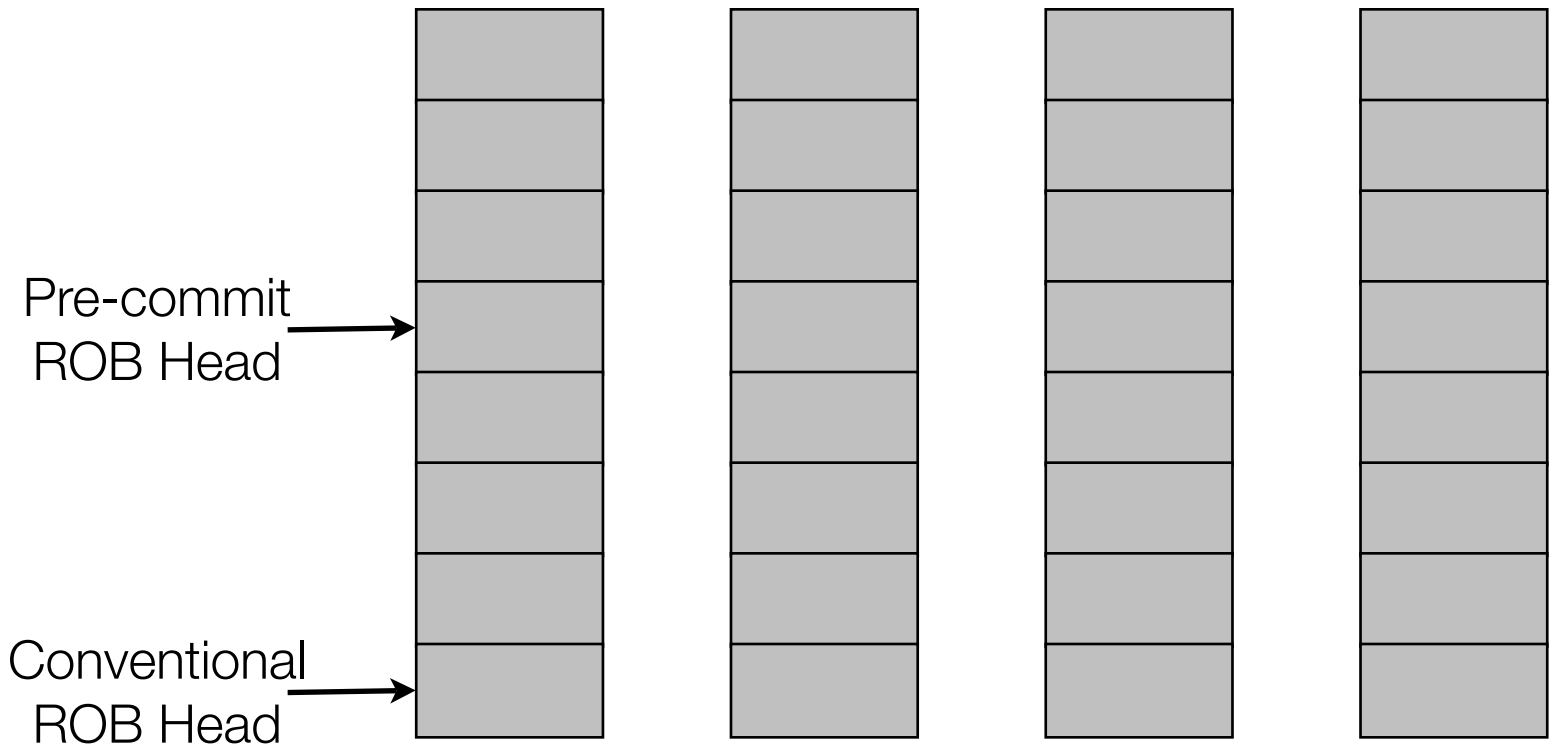
Collective Commit II



Collective Commit II



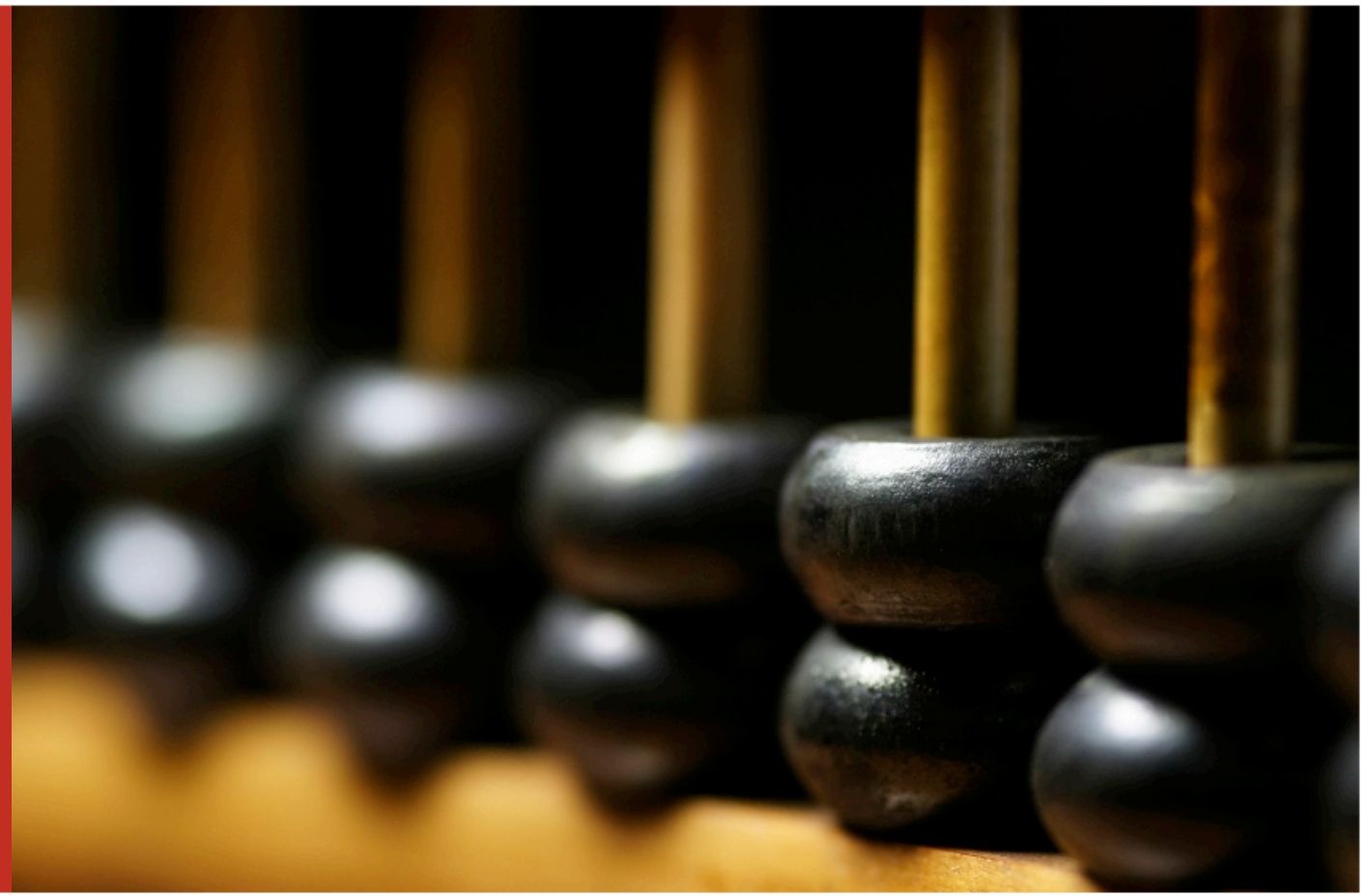
Collective Commit II



Other Issues (see ISCA'07 paper)

- Distribution of memory operations
  - Correctness (e.g., disambiguation)
  - Performance
- Run-time control of granularity
  - Serial vs. parallel sections
  - Variable granularity in parallel sections

Evaluation

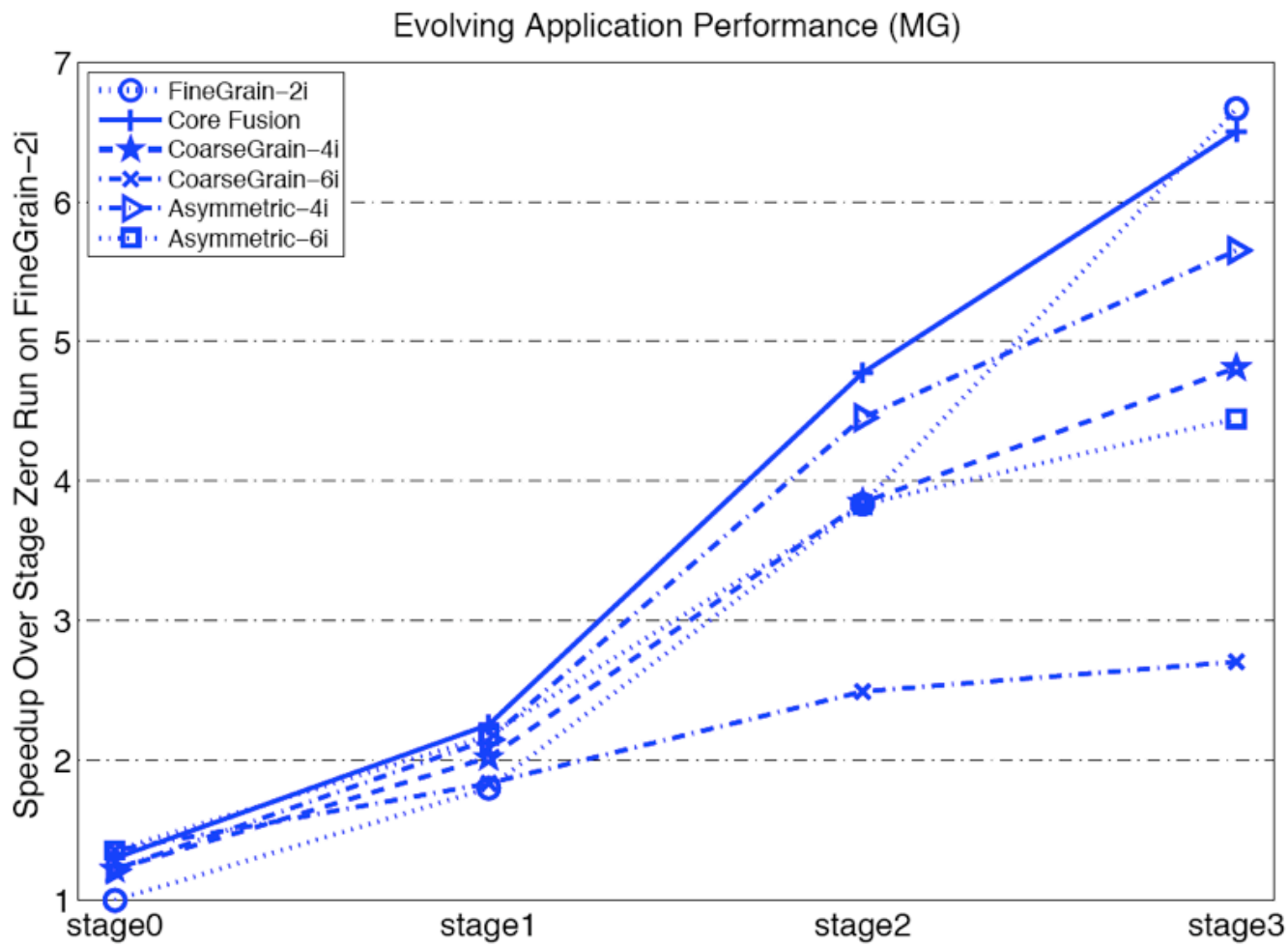


Experimental Setup

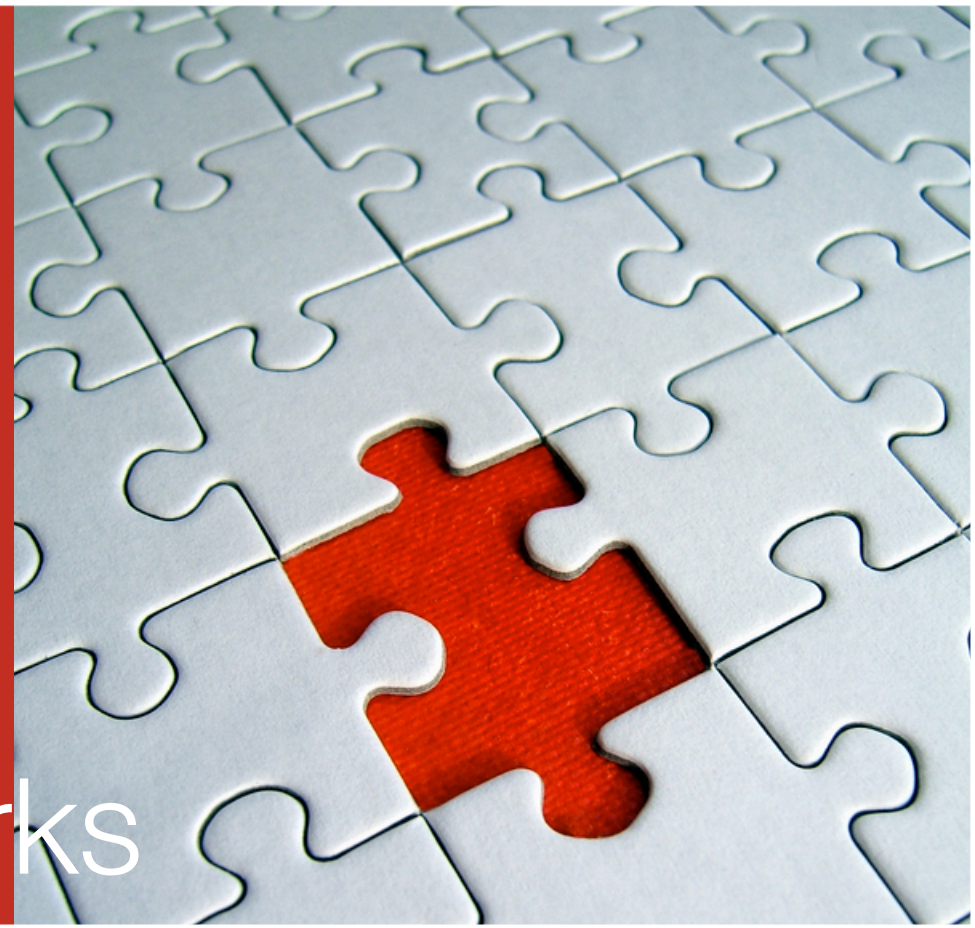
Frequency	4.0 GHz	**Fetch/issue/commit**	2/2/2	**Integer FUs**	1×ALU 1×AGU 1×Br 1×Mul 1×Div
Int/FP issue queues	16/16	**ROB entries**	48	**Int/FP registers**	32+40 / 32+40 (Architectural+Rename)
Floating-point FUs	1×ALU 1×Mul 1×Div	**Ld/St queue entries**	12/12	**Bank predictor**	2K-entries
Max. br. pred. rate	1 taken/cycle	**Max. unresolved br.**	12	**Br. penalty**	7 cycles minimum (14 cycles when fused)
Br. predictor	Alpha 21264	**RAS entries**	32	**BTB size**	512 entries, direct mapped
iL1/dL1 size	16 kB	**iL1/dL1 block size**	32B/64B	**iL1/dL1 round-trip**	2 cycles (uncontended)
iL1/dL1 ports	1 / 2	**iL1/dL1 MSHR entries**	8	**iL1/dL1 associativity**	2-way
Coherence protocol	MESI			**Consistency model**	Release consistency

Shared-memory Subsystem		**CMP Configuration**	**Composition (Cores)**
System bus transfer rate	64GB/s	CoreFusion	8x2-issue
System bus width	256 bits	FineGrain-2i	9x2-issue
Shared L2	4MB, 64B block size	CoarseGrain-4i	4x4-issue
Shared L2 associativity	8-way	CoarseGrain-6i	2x6-issue
Shared L2 banks	16	Asymmetric-4i	1x4-issue + 6x2-issue
L2 MSHR entries	32	Asymmetric-6i	1x6-issue + 4x2-issue
L2 round-trip	10 cycles (uncontended)		
Memory round-trip	320 cycles (uncontended)		

Performance: Evolving Workloads



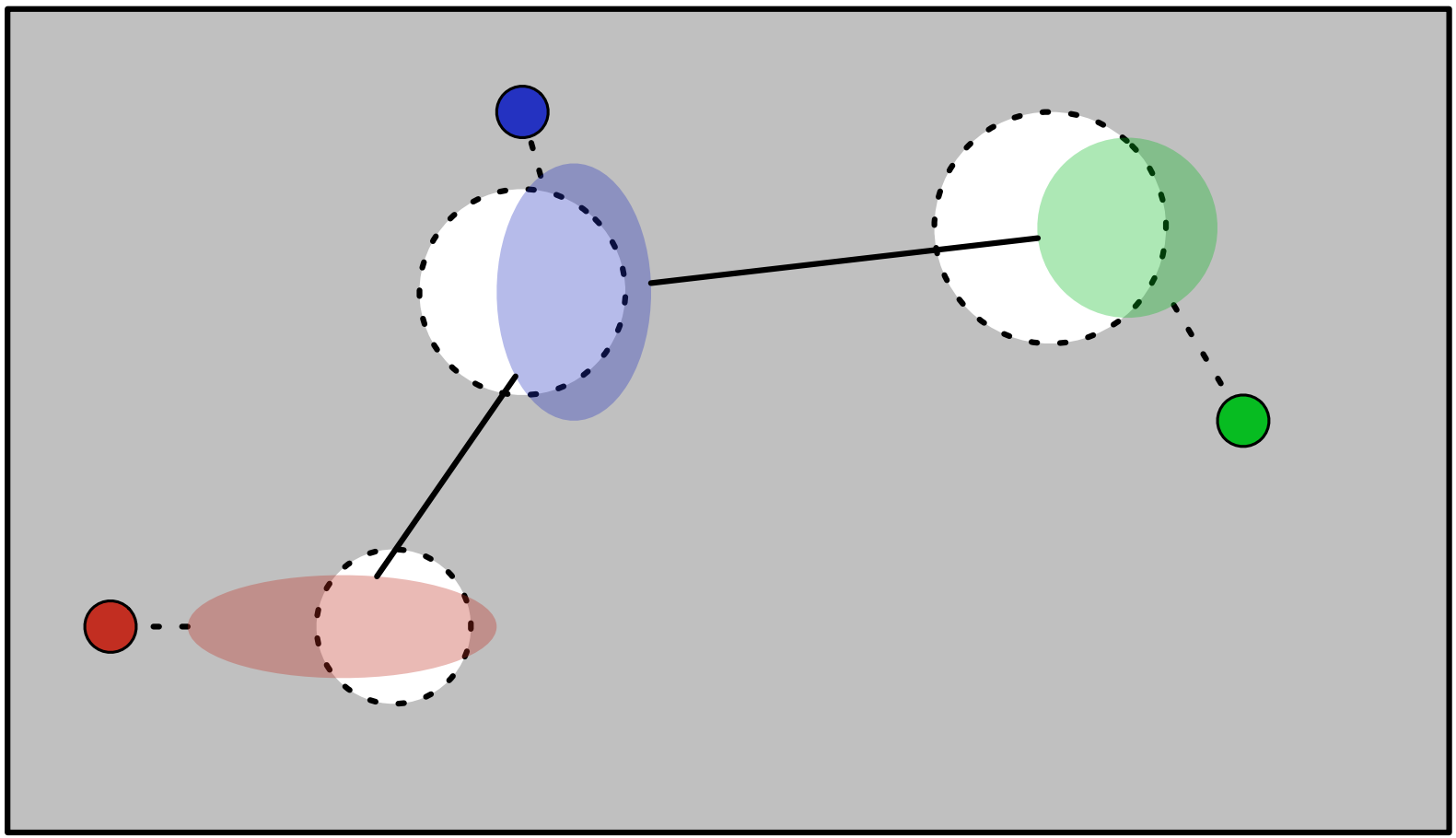
Concluding Remarks



Contributions and Findings

- Run-time fully reconfigurable and distributed
  - Front-end + i-Cache
  - LSQ + d-Cache
  - ROB
- Thorough evaluation using diverse workload classes
  - Sequential
  - Parallel
  - Multiprogrammed
  - Evolving
- Effective
  - Always best or 2nd best
  - Always best in intermediate parallelization stages
  - Others lag significantly in 1+ cases
- Highly compatible

Toward a Truly Versatile Design



Beyond Core Fusion

- Are there better “fusion” solutions?
- What is the ideal level of “transparency”?
  - Virtualization
  - Interaction w/ OS scheduling policies
  - Synergy w/ compiler technology
- What about other dimensions of versatility?
- What about adaptation within a configuration?
- Can all of this be automated?

Acknowledgments

- Outstanding students
  - Engin İpek
  - Meyrem Kırman
  - Nevin Kırman
- Generous support
  - NSF CAREER Award CCF-0545995
  - NSF Grants CNS-0720773, CNS-0509404, CCF-0429922
  - IBM Faculty Award
  - Intel graduate fellowships (M. and N. Kırman)
  - Microsoft gifts; Intel gifts and equipment donations

Cornell University
Computer Systems Laboratory

Accommodating Software Diversity in Chip Multiprocessors

José F. Martínez

M³ Architecture Research Group
http://m3.csl.cornell.edu/