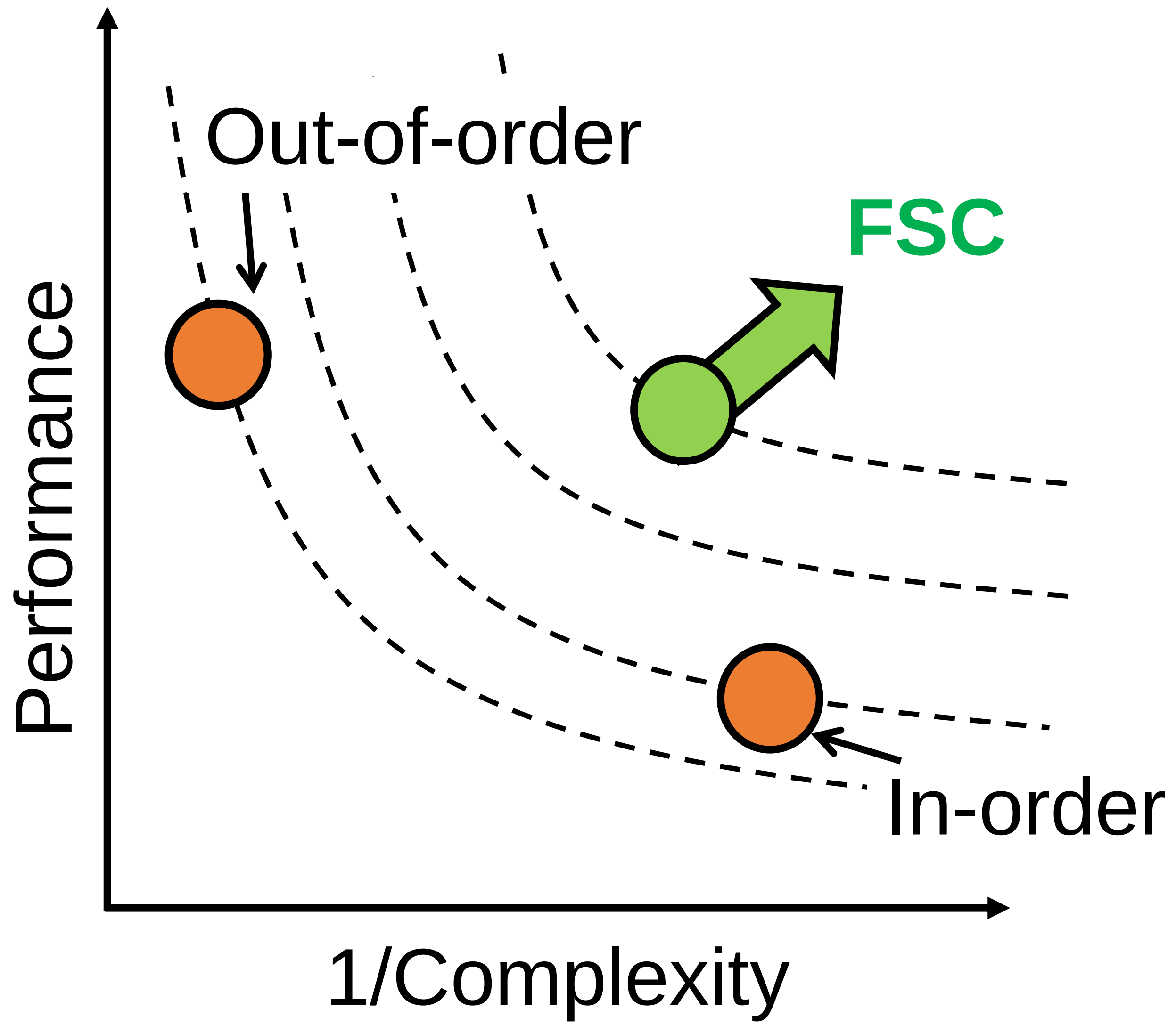
The Forward Slice Core Microarchitecture

Kartik Lakshminarasimhan *(Ghent University)*
Ajeya Naithani *(Ghent University)*
Josué Feliu Pérez *(University of Murcia)*
Lieven Eeckhout *(Ghent University)*

UNIVERSITEIT GENT

October 7, 2020 -- PACT 2020

Forward Slice Core (FSC) Delivers High Performance at Low Complexity



Modern computing systems are energy-constrained

OoO cores deliver high performance but are complex

Goal: OoO performance at in-order complexity

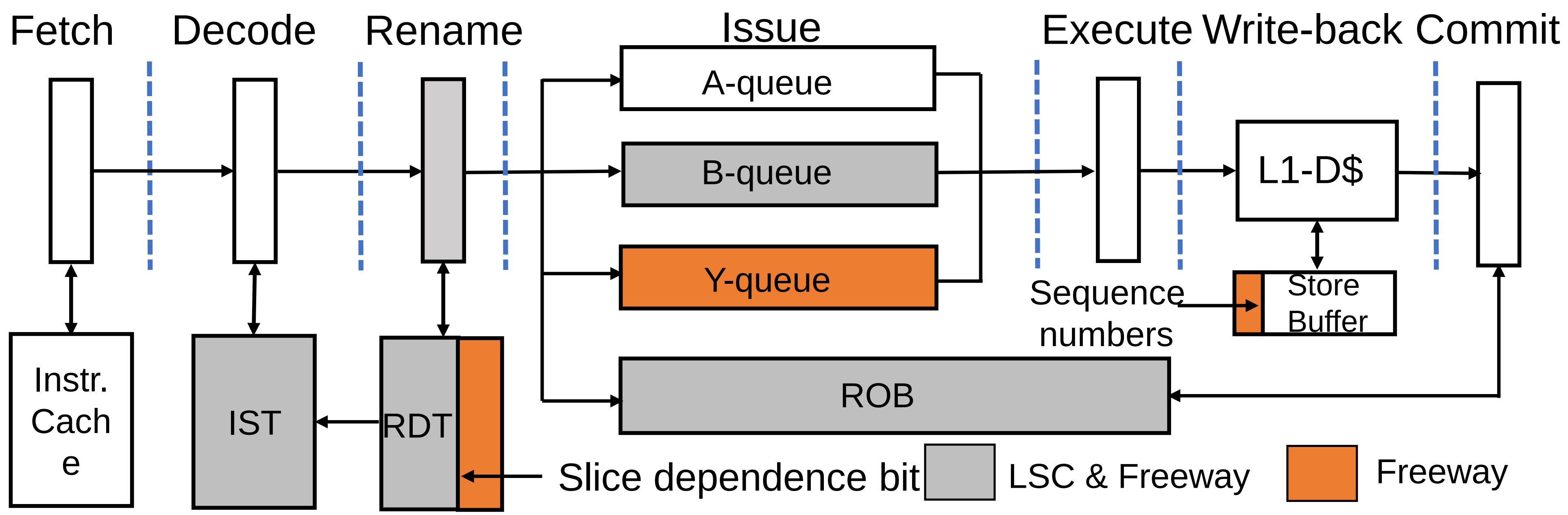
Comparison Against Prior Work

Processor	ILP	MHP	Complexity
InO	-	-	+
LSC *[ISCA '15]*	+	+	+
Freeway *[HPCA '19]*	+	++	+++
FSC	+++	+++	++
OoO	++++	++++	+++++++

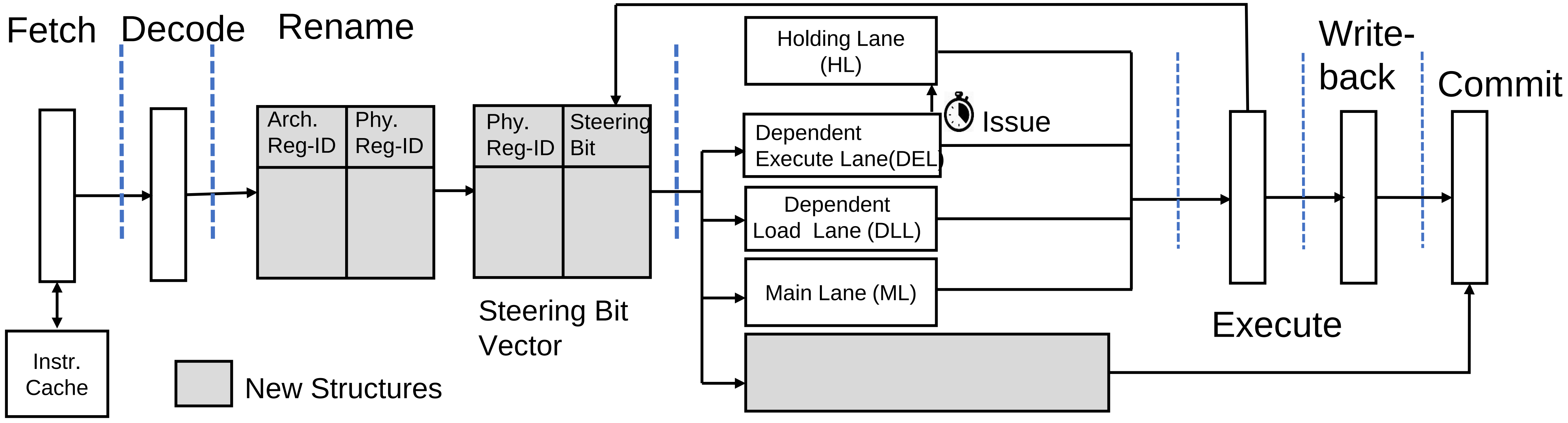
MHP = Memory Hierarchy Parallelism = number of overlapping memory accesses that hit anywhere in the memory hierarchy

Prior Work: Slice-Out-of-Order Processors

- Slice-out-of-order (sOoO) processors
    - Load Slice Core (LSC) *[ISCA'15]* and Freeway *[HPCA'19]*
- Limitations for sOoO processors:
    - Imperfect backward slice identification
    - Lack of ILP



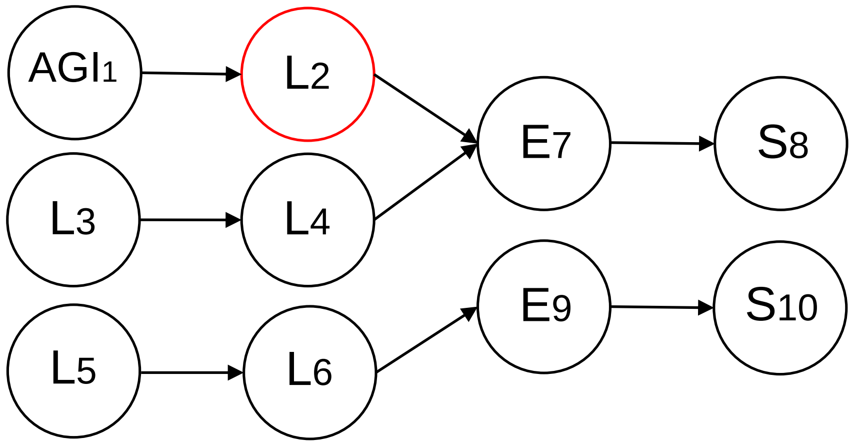
The Forward Slice Core



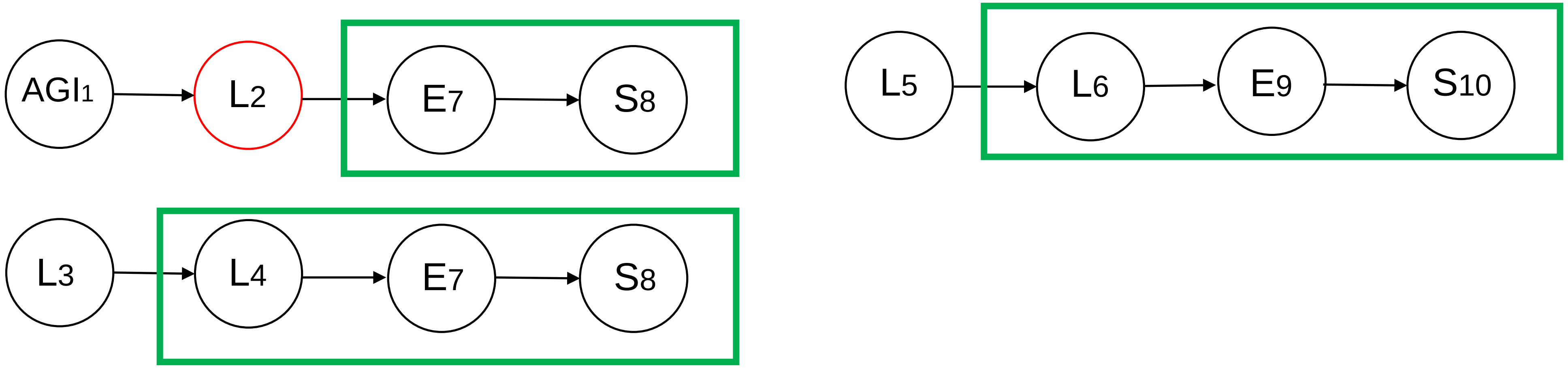
FSC's key features:

- Restricted OoO execution through in-order lanes
- Identify and steer forward slice instructions
- Move stalling forward slice instructions to Holding Lane (HL)
- Replicate store address micro-ops across issue lanes for in-order memory address resolution

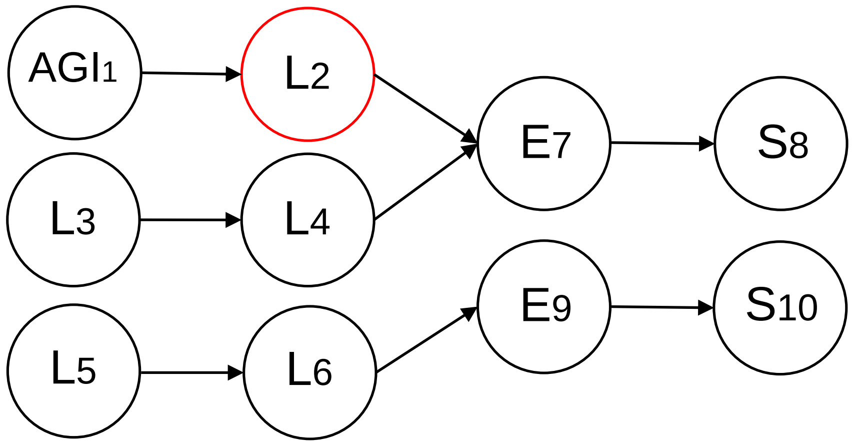
Forward Slice Example

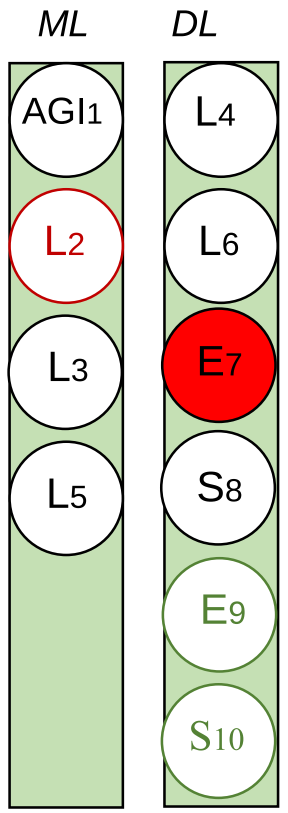


a code example from mcf

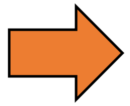


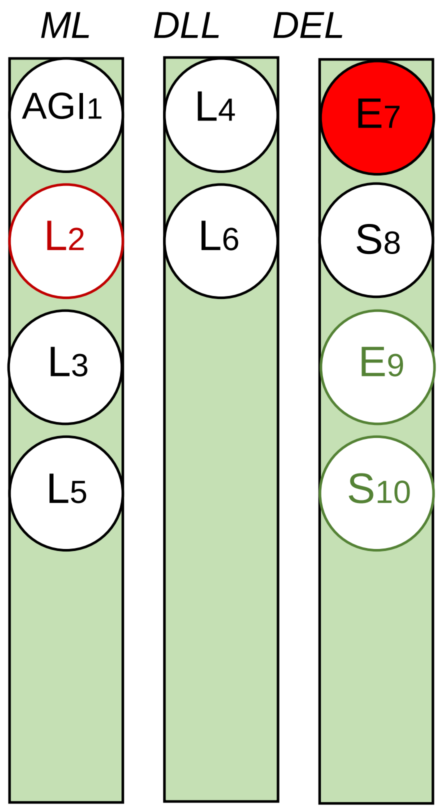
Steering In-Flight Load Consumers to DEL and DLL Lanes



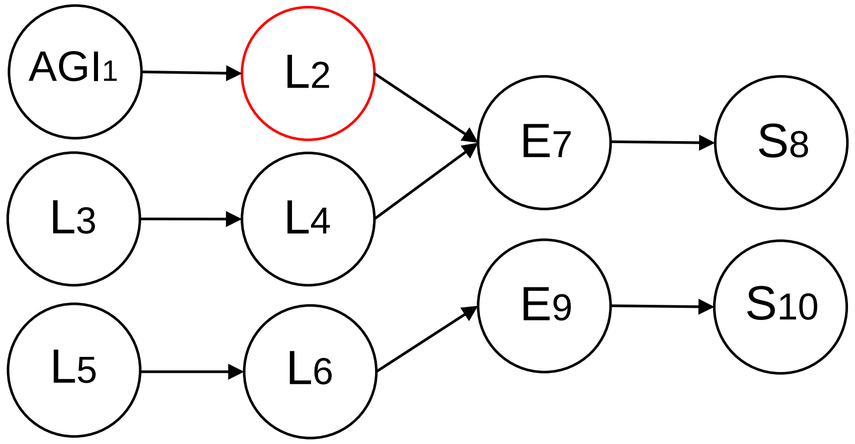


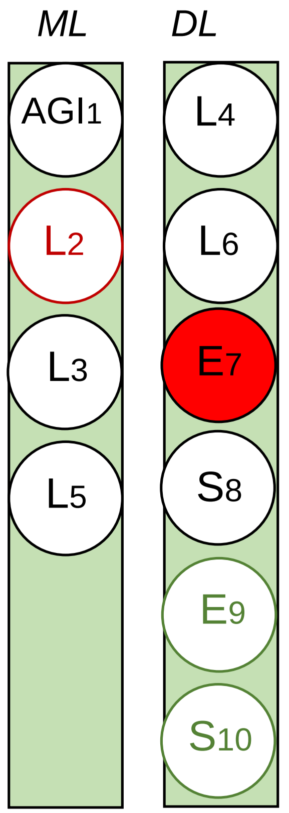
load consumers are steered to separate lanes



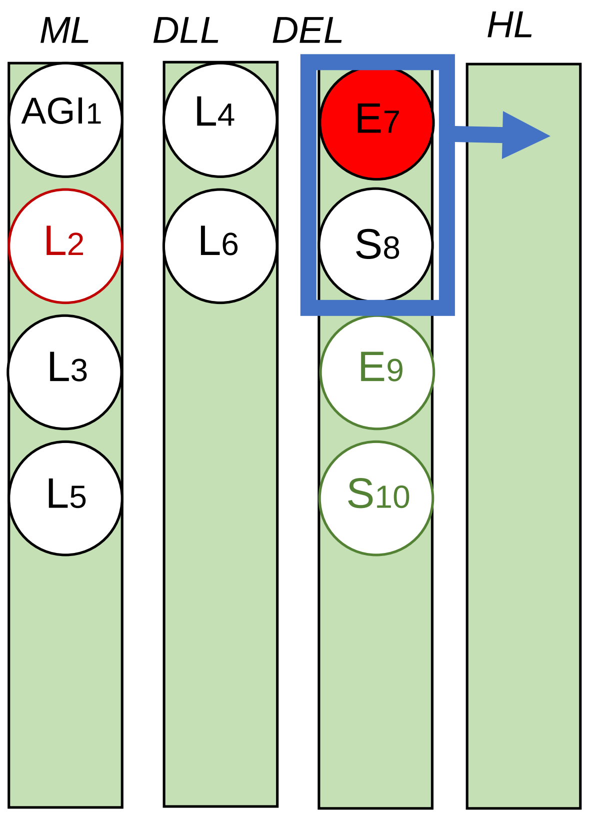


Redirecting Long-Latency Load Consumers to Holding Lane

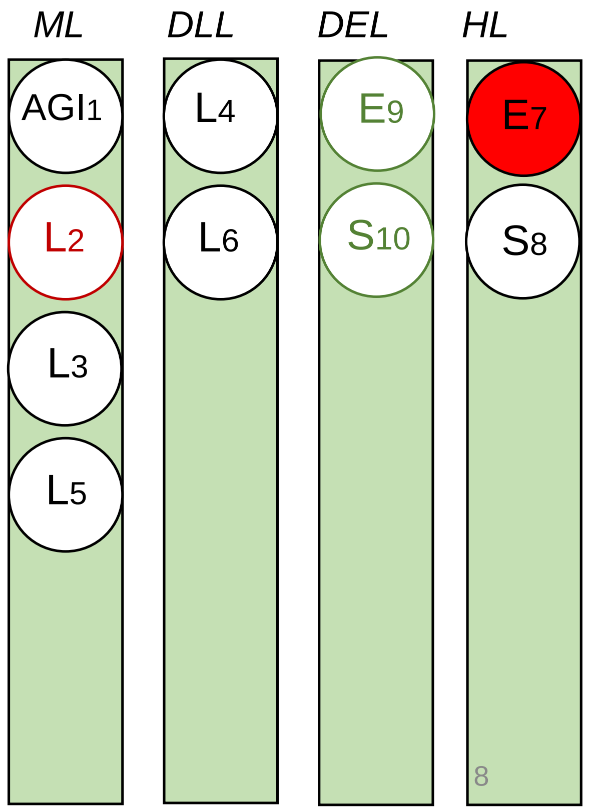




load consumers are steered to separate lanes

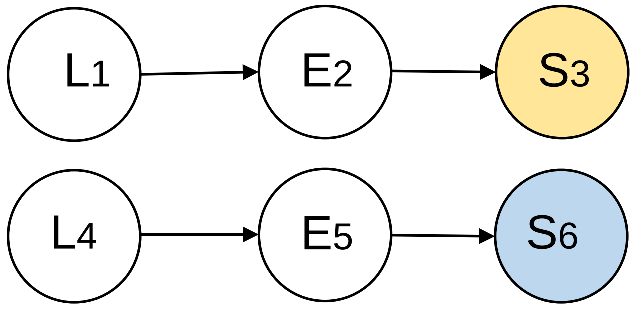


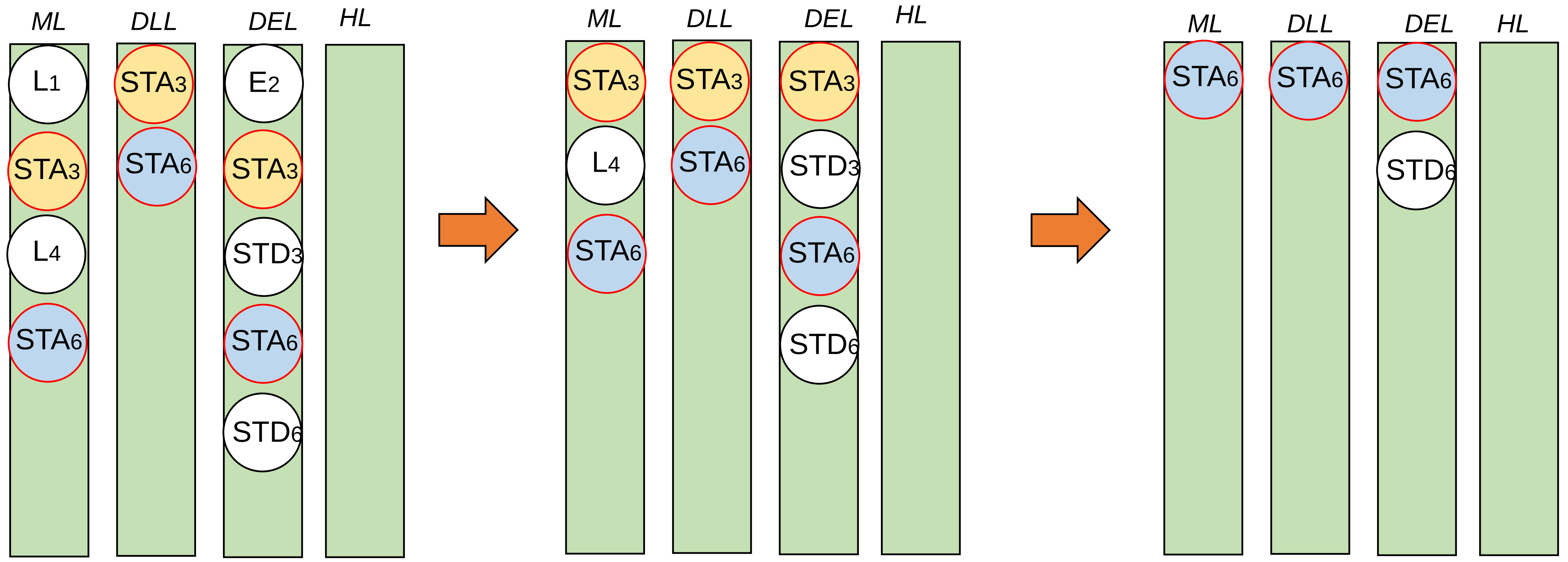
re-directing L1D-miss load consumers from DEL lane to HL



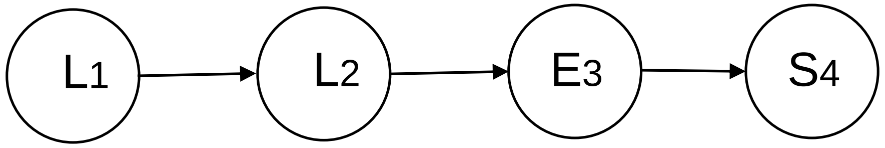
Store Address Replication (SAR)

No memory dependence violations by replicating store-address instructions
→ in-order address computation





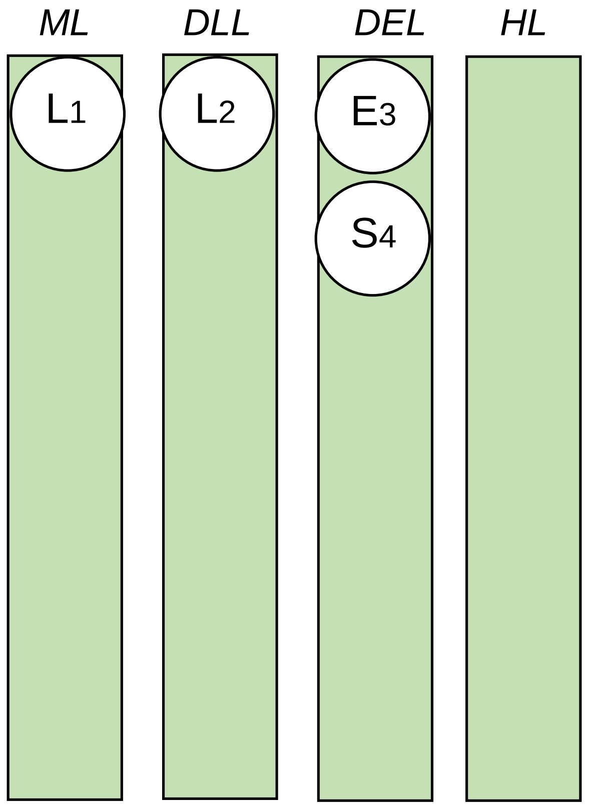
Steering Bit Vector (SBV): Tracking In-Flight Load Consumers



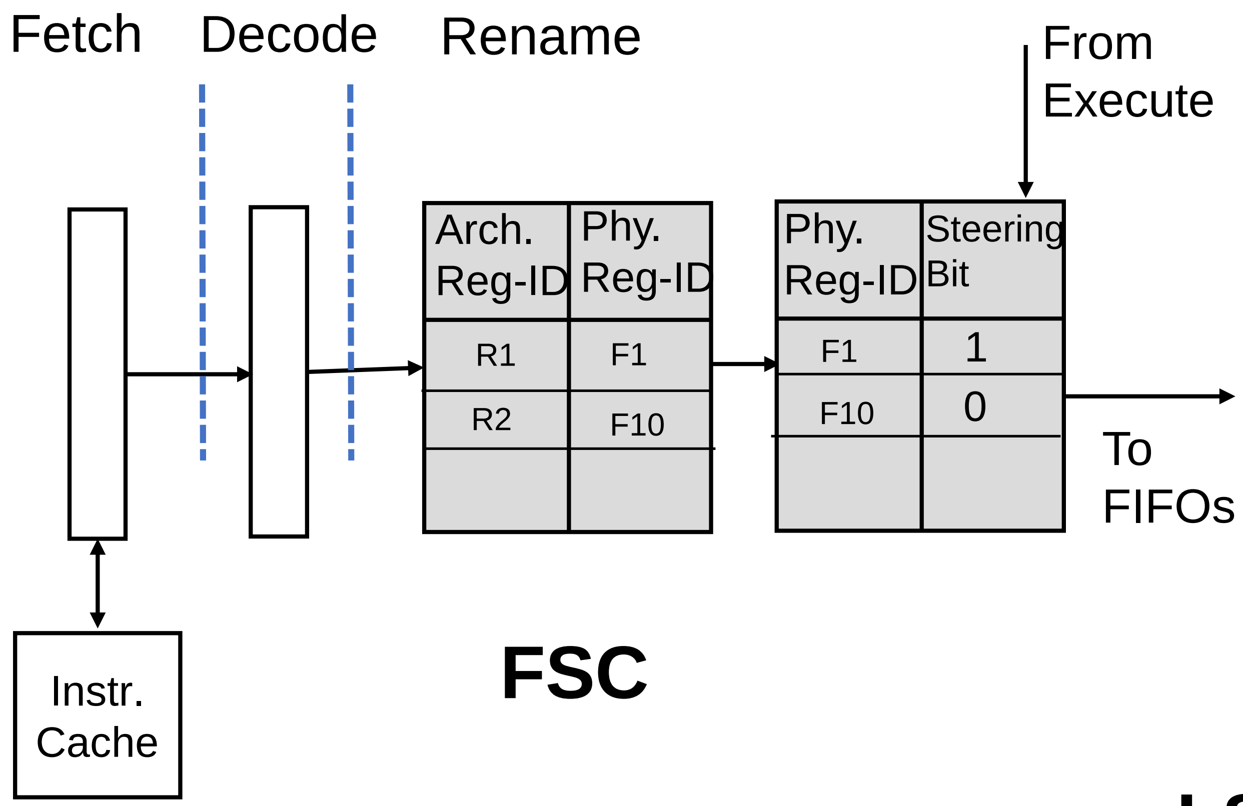
```
L1: ld ( r8 + r7*8 ), r5
L2: ld ( r5 + r6*8 ), r1
E3: add r1, r1
S4: st r1, (r2 + r3*4)
```

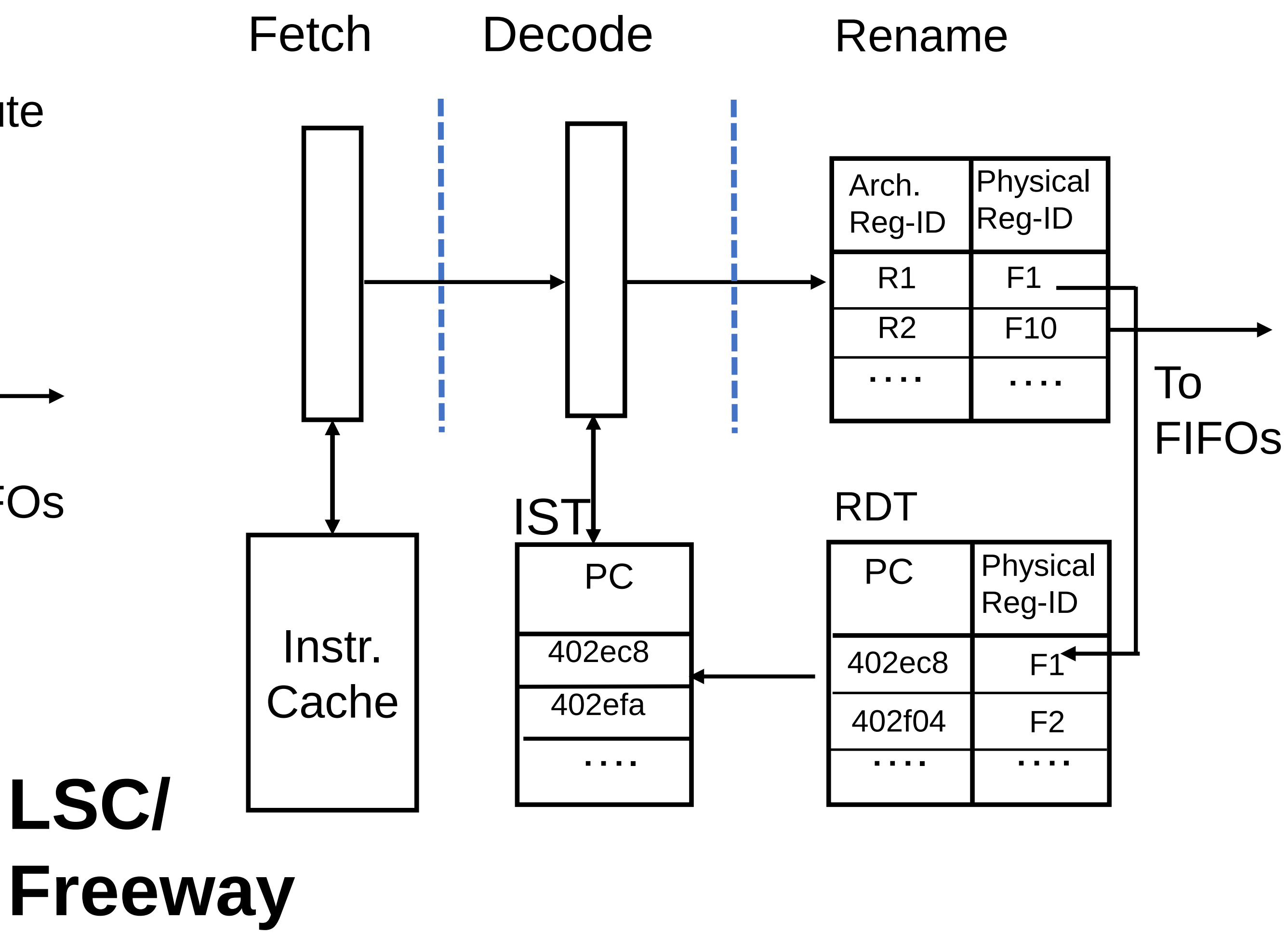
Steering Bit Vector (SBV)

Phy. Reg-ID	Steering Bit
F5	1 → 0
F1	1



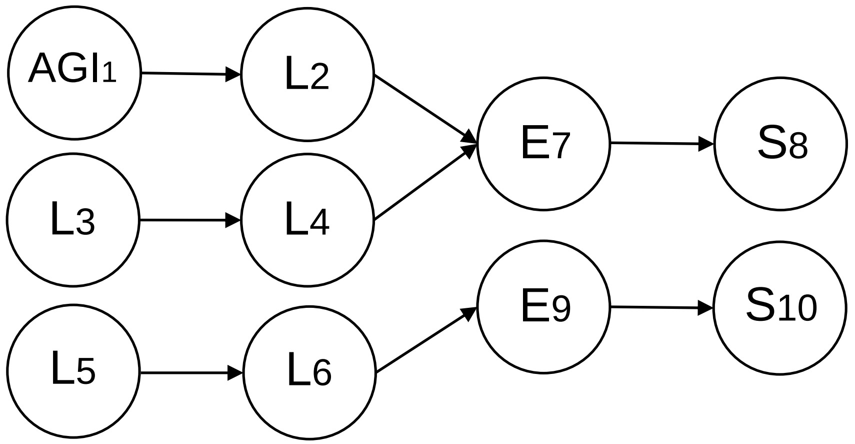
Complexity of Steering Logic: FSC vs Freeway

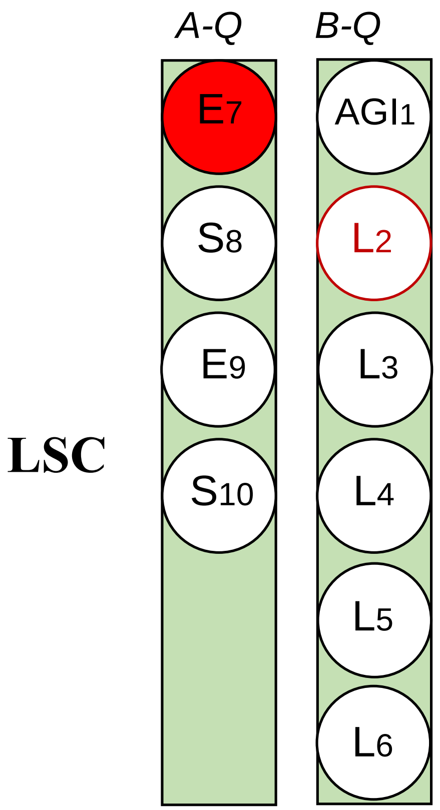


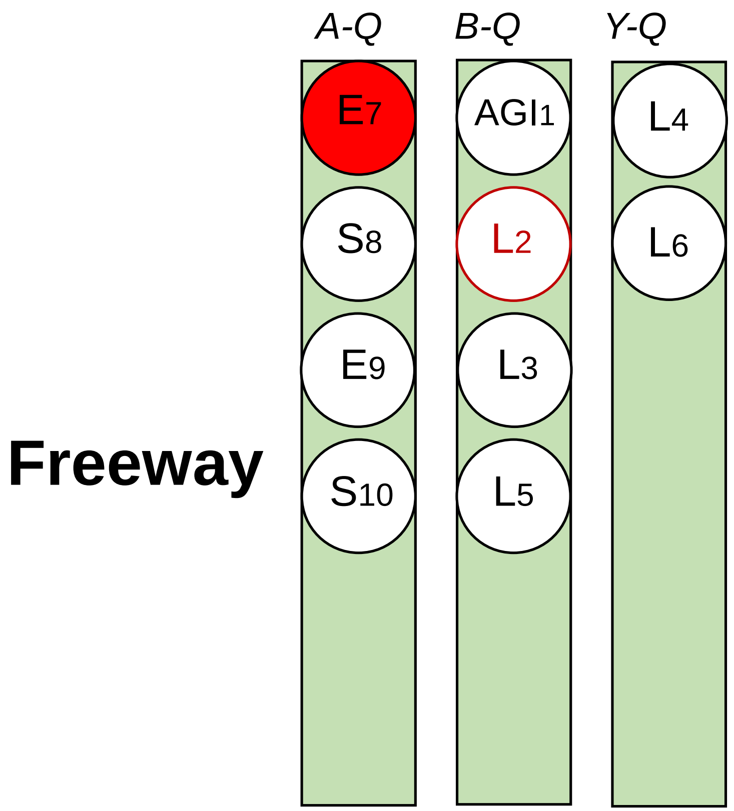


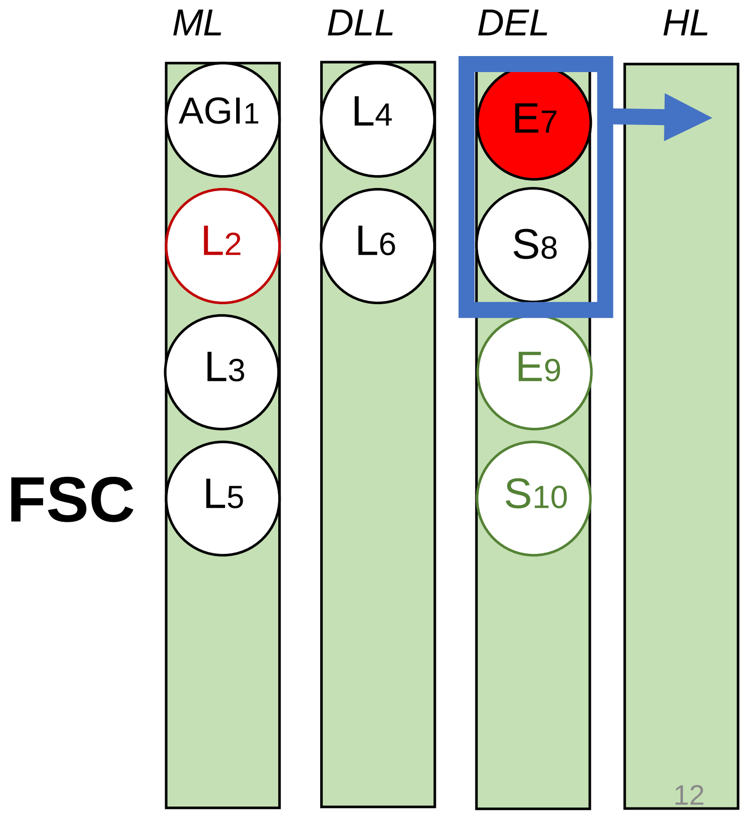
FSC requires 1048 less bytes than Freeway *and* does not require CAM logic for memory disambiguation

Why Does FSC Outperform LSC and Freeway?









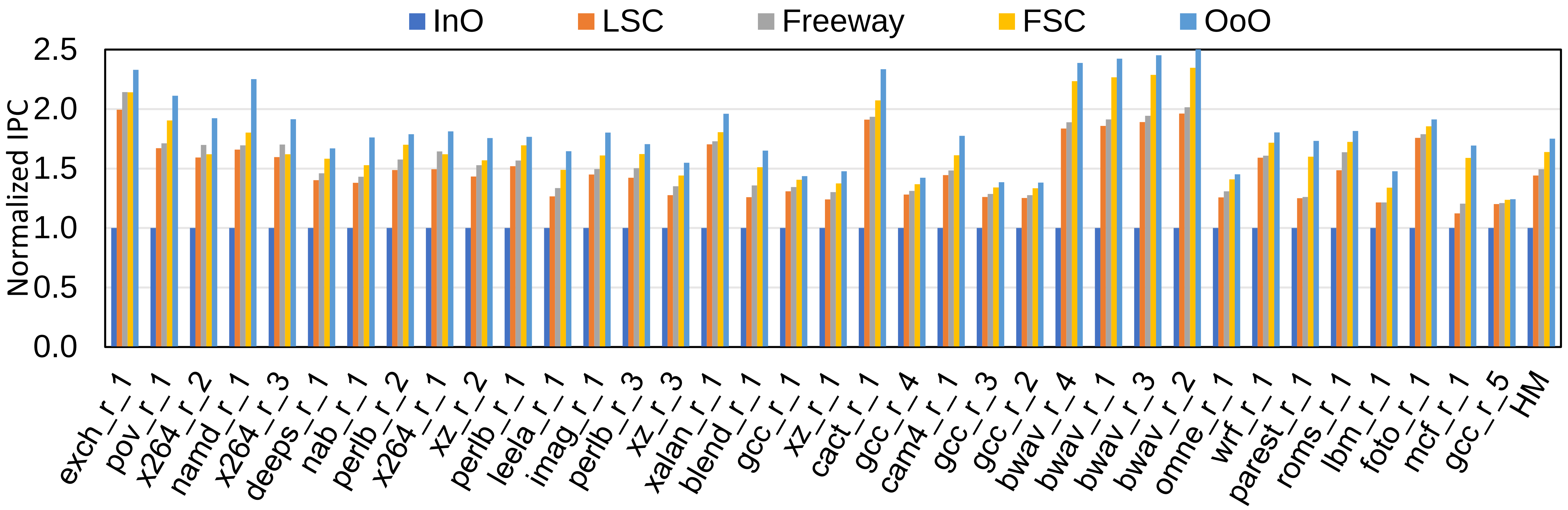
Experimental Setup

Sniper multi-core simulator
- ARM Cortex-A7-like configuration
- 32KB L1, 512 KB L2
- 22 nm (McPAT and CACTI 6.5)

SPEC CPU2017 representative 1B instruction SimPoints

InO	LSC	Freeway	FSC	OoO
Stall-on-use	Restricted OoO	Restricted OoO	Restricted OoO	OoO
32-entry queue	16-entry A and B queues	12-entry A, B and Y queues	8-entry ML, DEL, DLL and HL	32-entry ROB and issue queue
2-wide issue	2-wide issue	2-wide issue	2-wide issue	2-wide issue
-	128-entry IST 64-entry RDT	128 entry IST 64-entry RDT	64-entry SBV	-

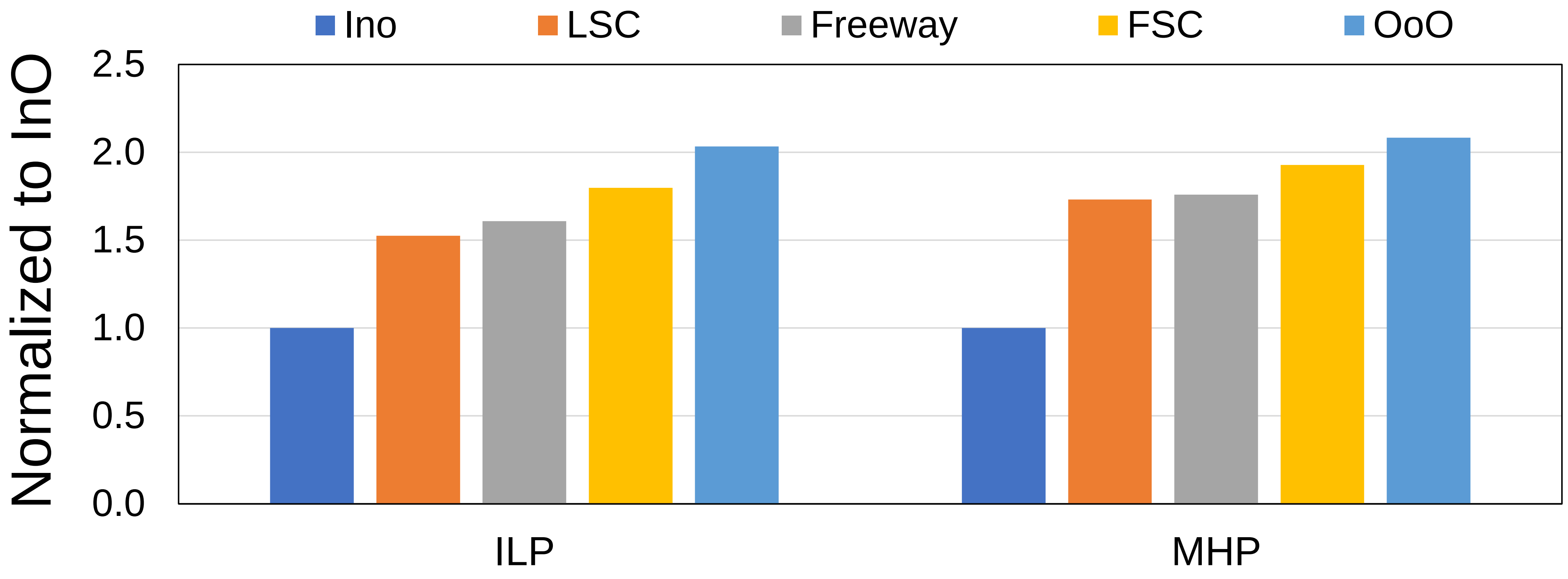
FSC Performance Benefit



9.7% avg performance improvement over Freeway with 1408 bytes less hardware

Within 6.9% of OoO, while consuming 56% less power

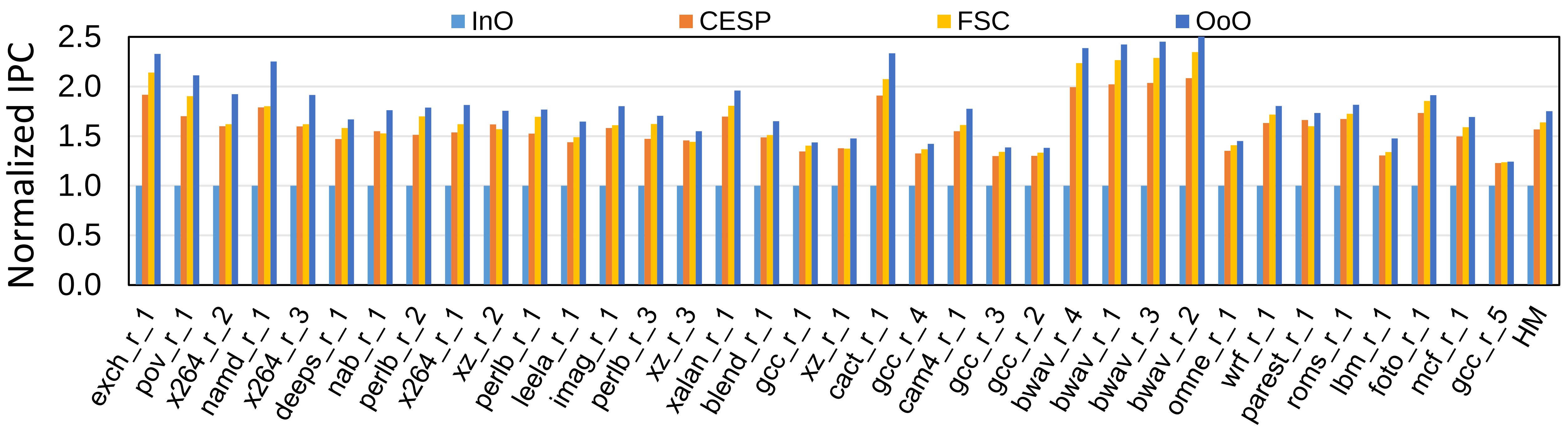
FSC Improves ILP and MHP



FSC improves ILP by 12% and MHP by 10% over Freeway

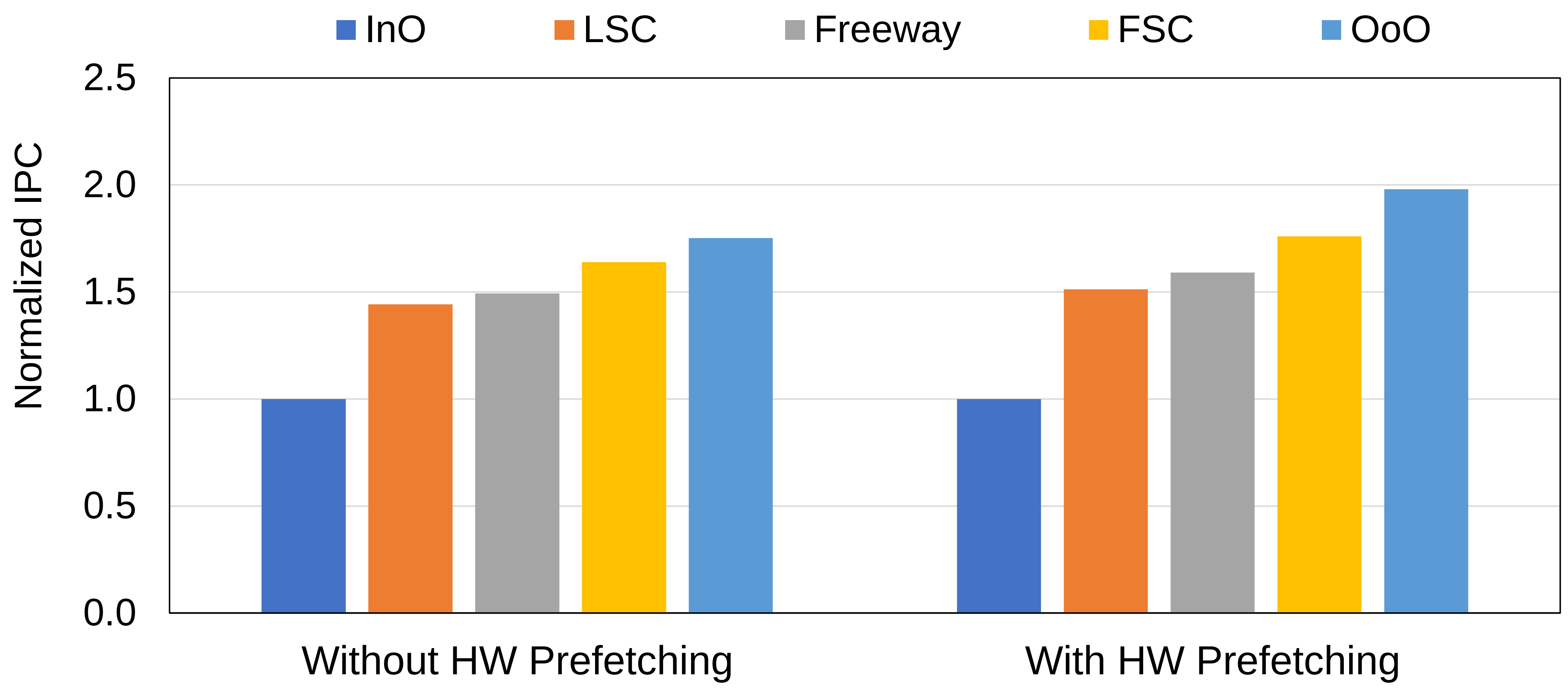
MHP = Memory Hierarchy Parallelism = number of overlapping memory accesses that hit anywhere in the memory hierarchy

FSC Performance Benefit over CESP



FSC improves performance by 4.5% on average (and up to 12.6%) over CESP *[Palacharla et al. ISCA'97]* at reduced hardware complexity

FSC with Hardware Prefetcher



FSC improves performance by 75% on average compared to an InO core

Conclusion: Forward Slice Core

Three key innovations

- Steering load-consumers to separate in-order queues
- Redirect L1D-miss load-consumers to holding lane
- Store address replication

Higher ILP and MHP than prior sOoO cores, at reduced hardware cost and power consumption

Key results

- 9.7% higher performance than Freeway with less HW
- within 6.9% of OoO core with 56% less power consumption
- 44% higher performance than InO at small HW cost