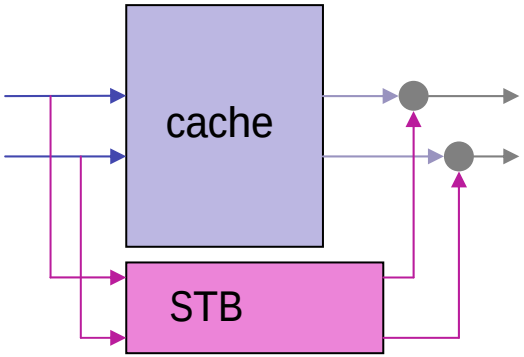
Store Buffer Design in First-Level Multibanked Data Caches

[1]E. Torres, P. Ibánez, V. Viñals, and [2]J.M. Llabería

Multibanked L1 cache

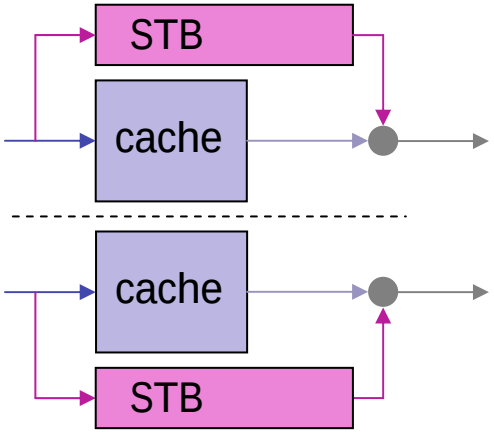
[Sohi & Franklin, ASPLOS91]

Multi-Ported



- time
- area
- power

Multi-Banked

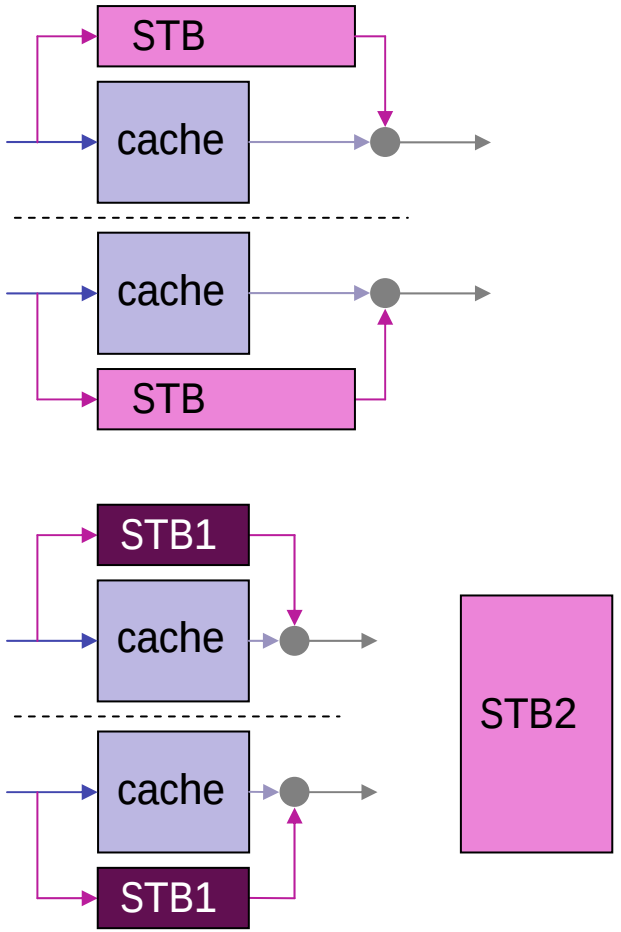


- 1 port/bank
- STB bank size remains unchanged

[Zyuban & Kogge, IEEE TonC 01]

Our Proposal: Distributed Store Buffer Design

- Basic 2-level STB
  - STB1: speculative data forwarding
  - STB2: enforce program order
- Complexity Reduction
  - STB1: does not check age
  - STB2: does not forward
- Performance Improvement
  - choose the right recovery policy
  - some stores may skip STB1

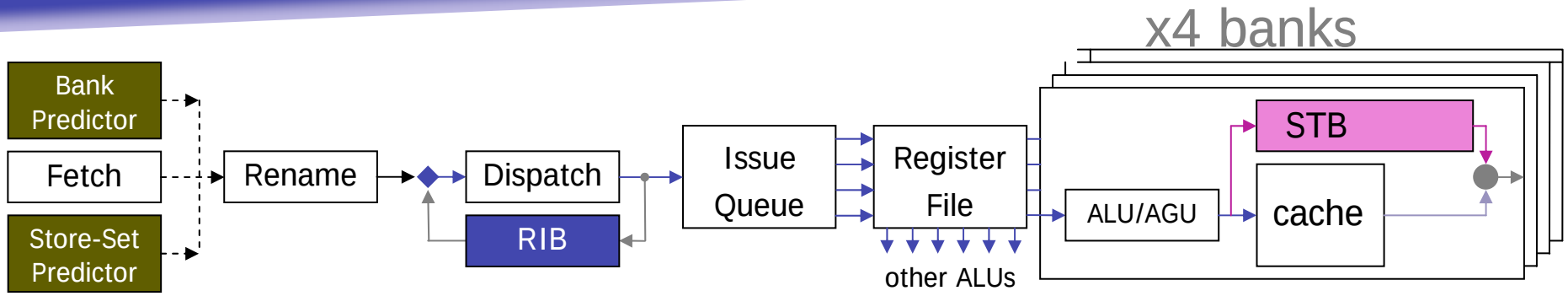


IPC: 1-level STB (128-entry) ≈ 2-level STB (8-entry STB1)

Talk Outline

- Introduction
- Processor Model
- Store Lifetime
- 2-Level STB Design
- Design Enhancements
- Conclusions

Processor Model



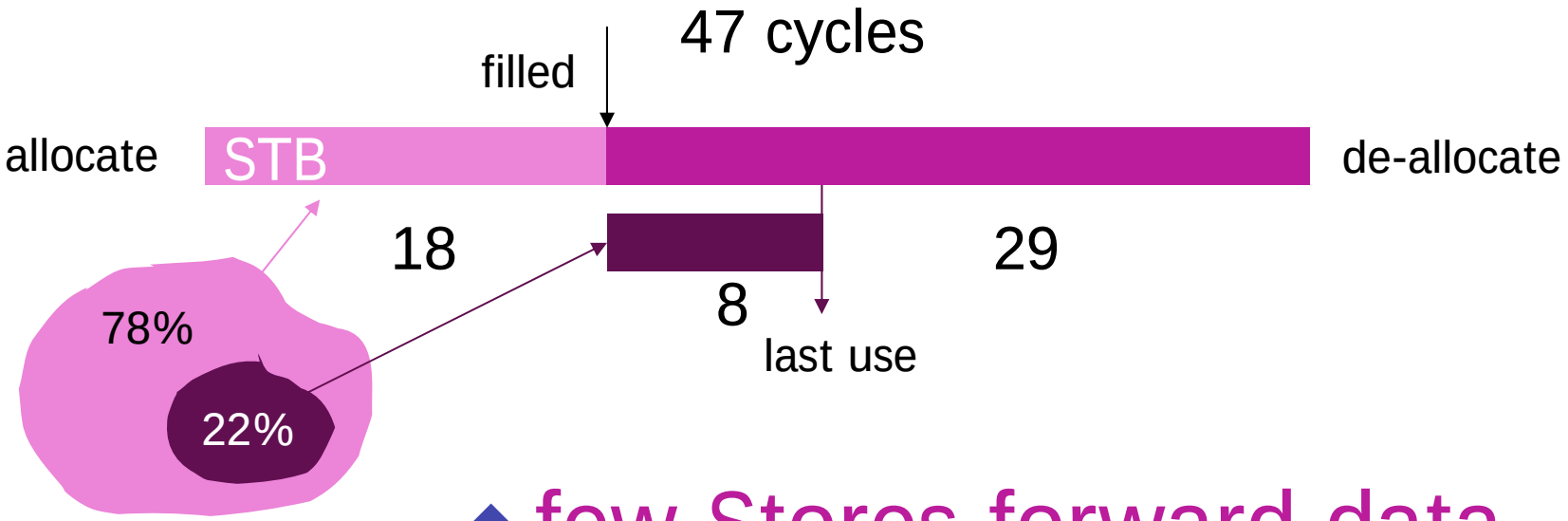
- 8-issue
- *sliced memory pipeline* [Yoaz et al., ISCA99]
  - 4 banks
- latency prediction (cache hit)
  - recovery from IQ
- memory disambiguation
  - store-sets [Chryros & Emer, ISCA 1998]
  - recovery from RIB (*Renamed Instruction Buffer*)

Talk Outline

- Introduction
- Processor Model
- Store Lifetime
- 2-Level STB Design
- Design Enhancements
- Conclusions

Store Lifetime

- Spec int2K (*simpoints)*

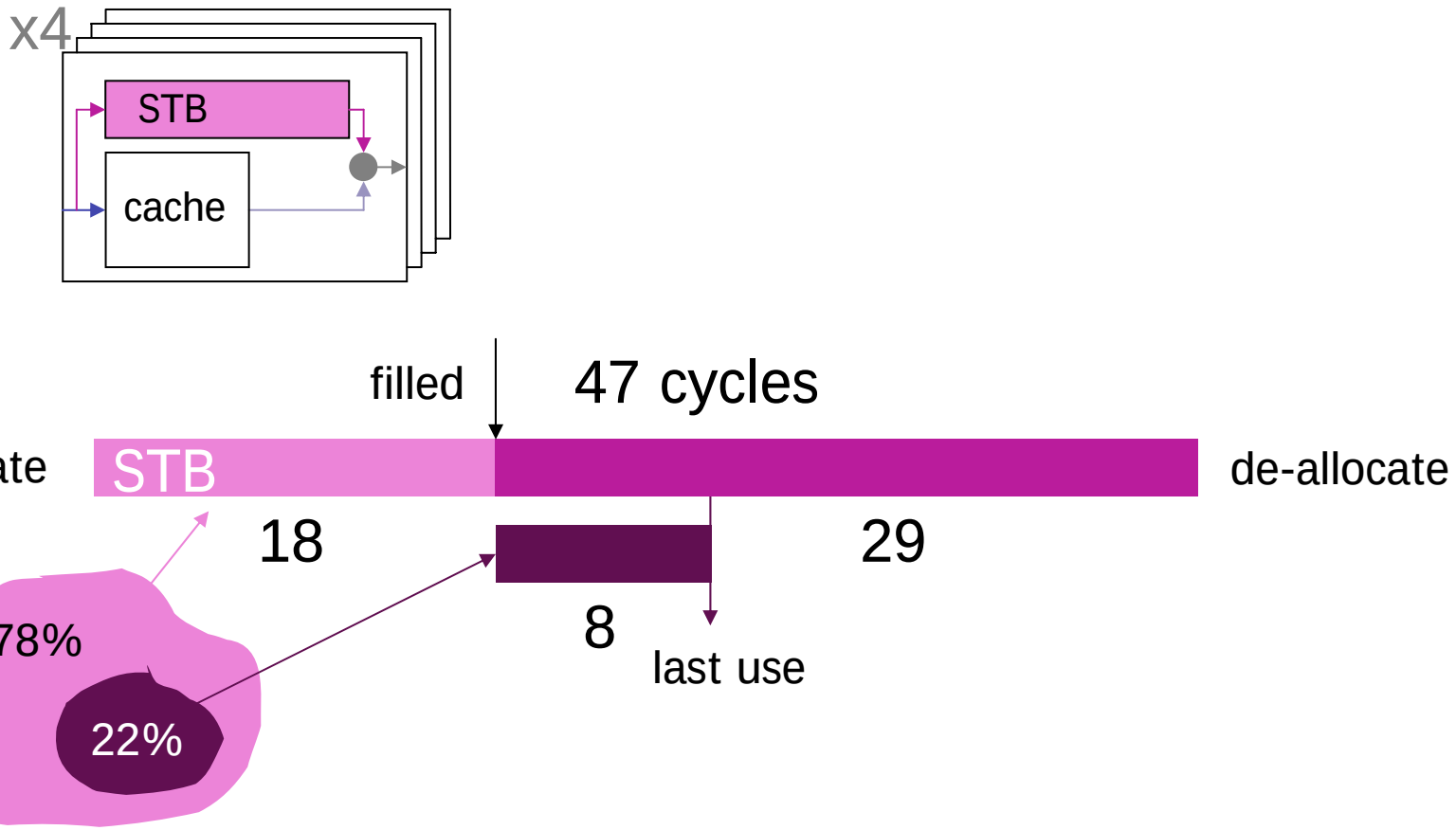


- few Stores forward data
- early entry allocation
- late entry de-allocation

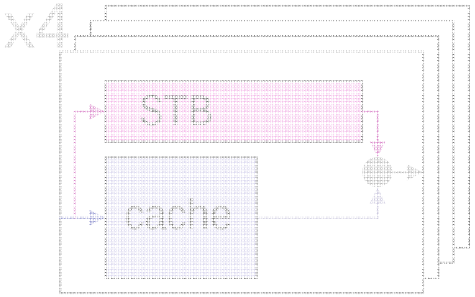
Talk Outline

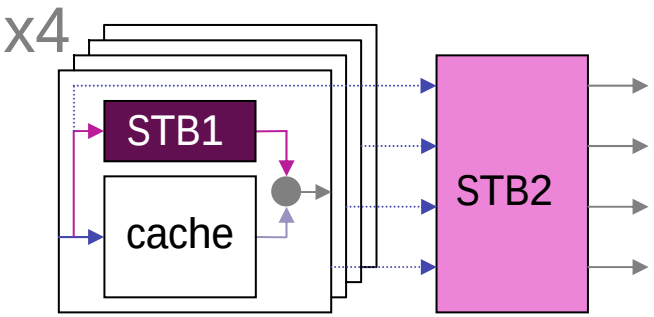
- Introduction
- Processor Model
- Store Lifetime
- 2-Level STB Design
- Design Enhancements
- Conclusions

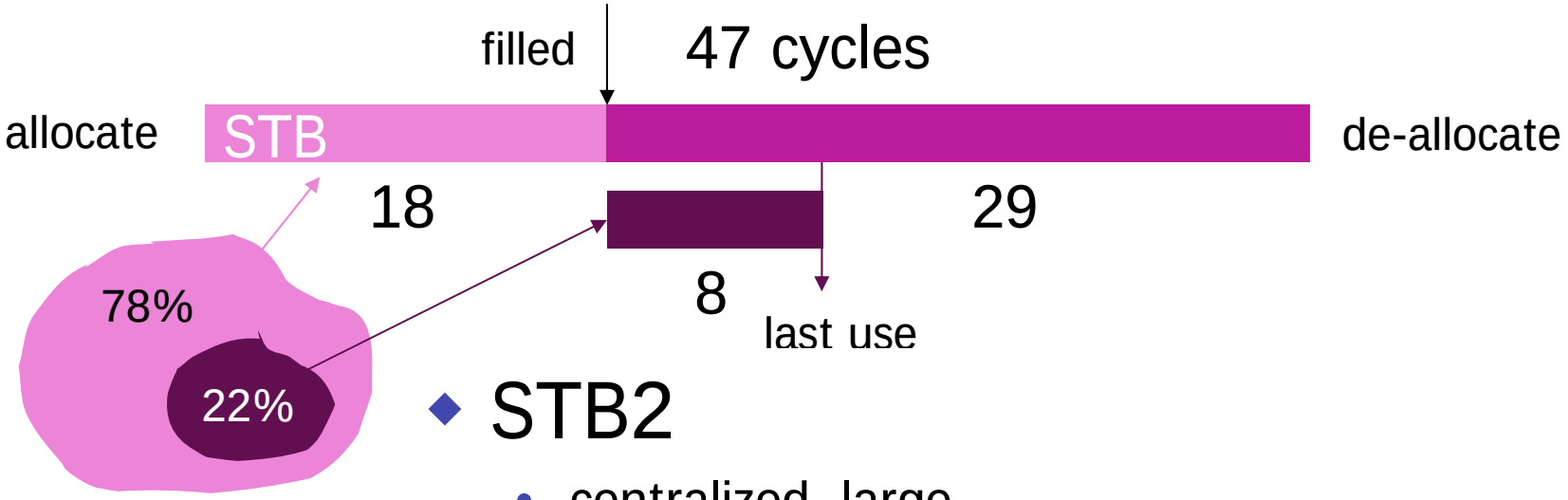
Base 2-Level Store Buffer Design



Base 2-Level Store Buffer Design

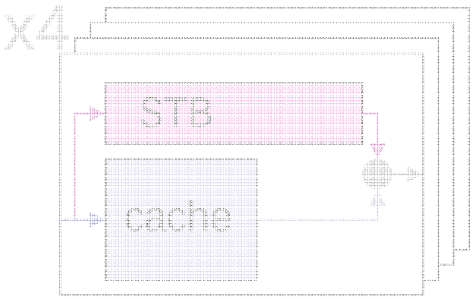


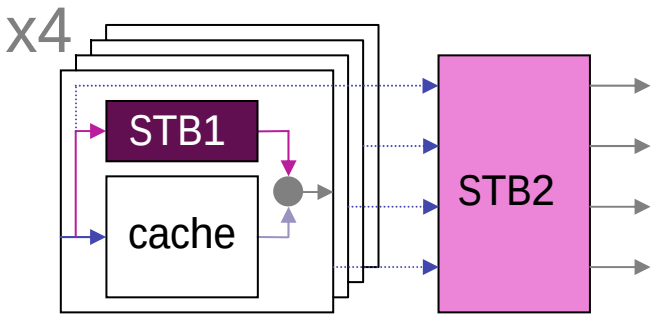


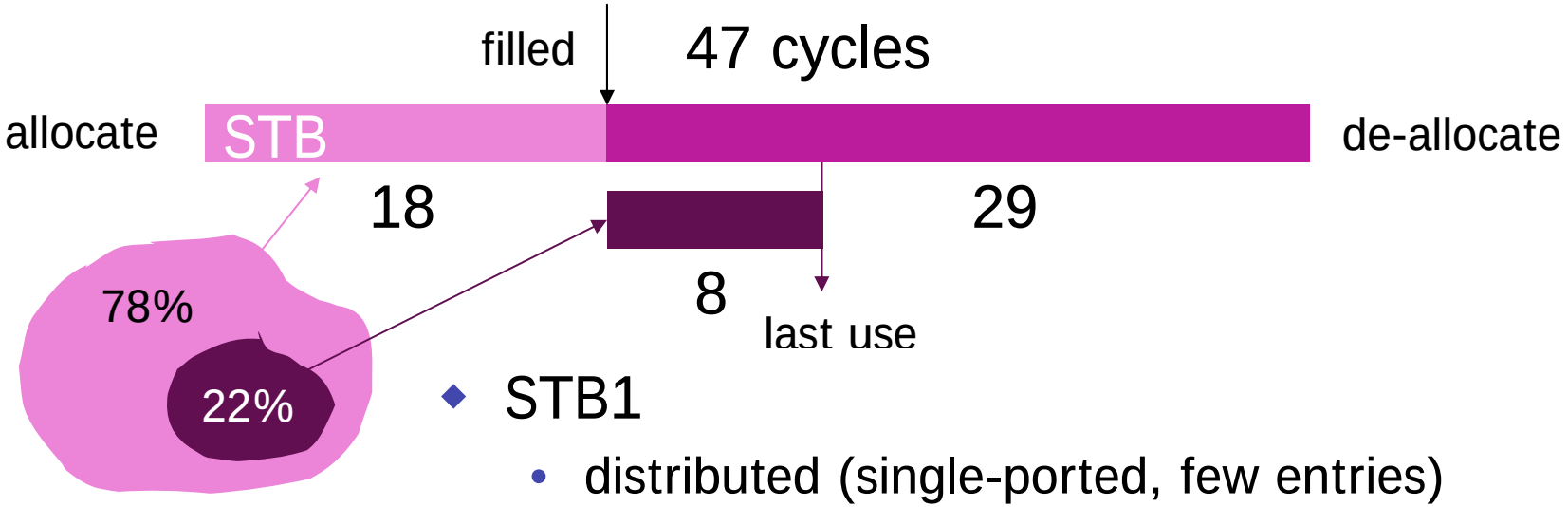


- STB2
  - centralized, large
  - keeps all in-flight stores
  - checks program order and updates cache
  - entries are allocated from dispatch to commit

Base 2-Level Store Buffer Design

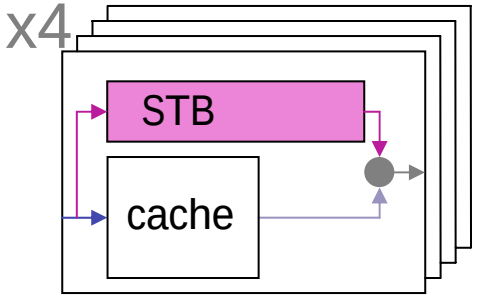


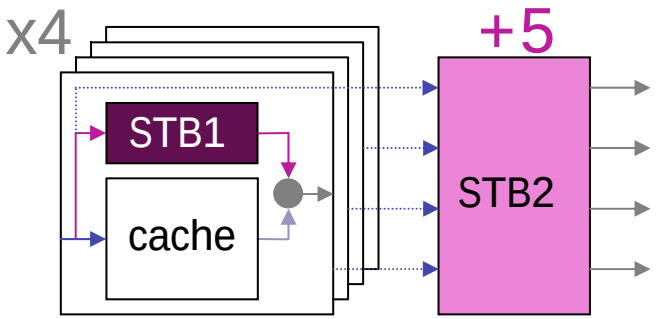


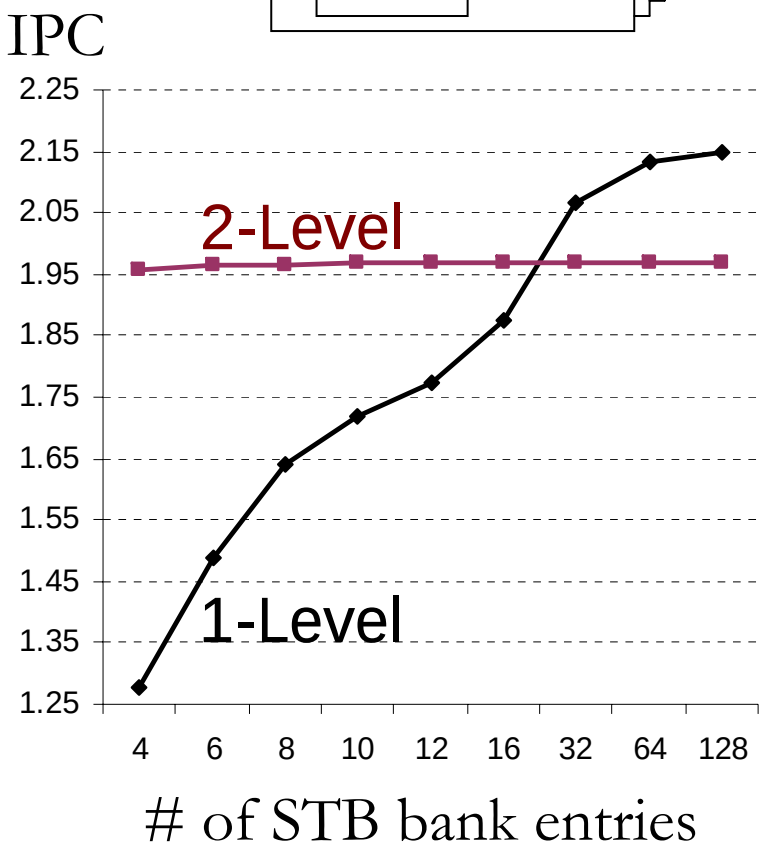


- STB1
  - distributed (single-ported, few entries)
  - cache latency
  - circular buffer, allocated at execution
  - speculative forwarding (STB2, IQ)

Base 2-level STB *vs* 1-level STB







- 2-Level STB
  - flat performance
  - high LD fwd coverage
    - 8 entry STB1: 99%
  - high IQ pressure

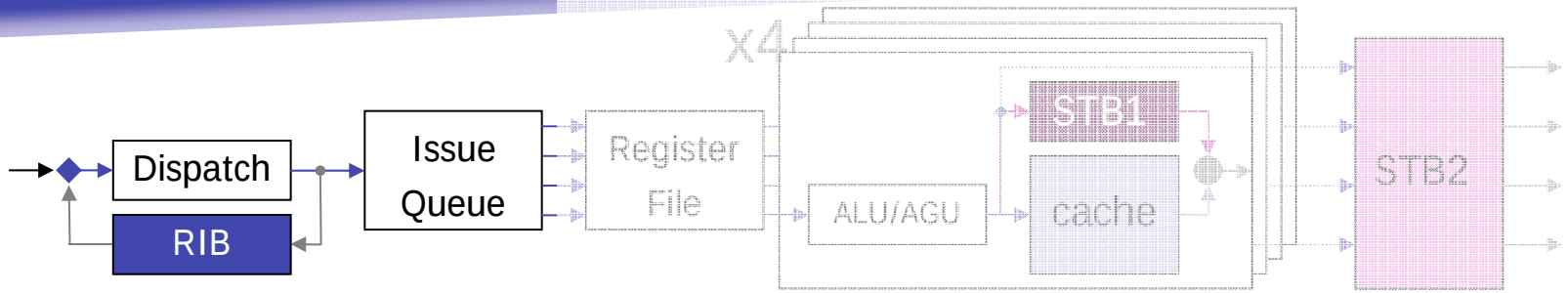
Talk Outline

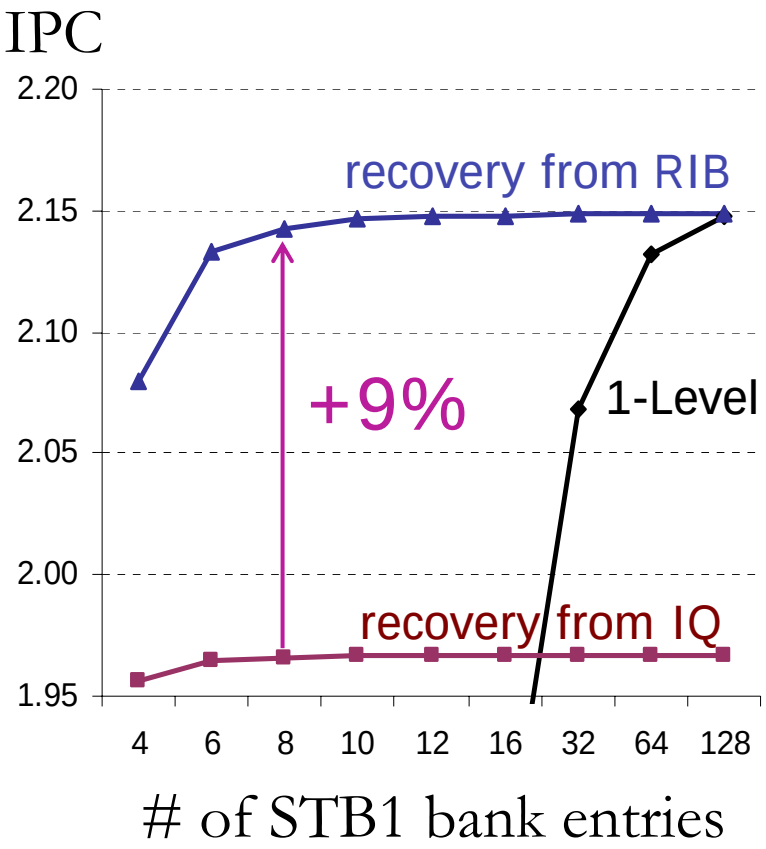
- Introduction
- Processor Model
- Store Lifetime
- 2-Level STB Design
- Design Enhancements
- Conclusions

Design Enhancements

- Reducing IQ Occupancy
  - Recovery from RIB
- STB2 simplification:
  - do not no forward
- Reducing Contention
  - Non Forwarding Store Predictor
- STB1 simplification
  - do not check ages

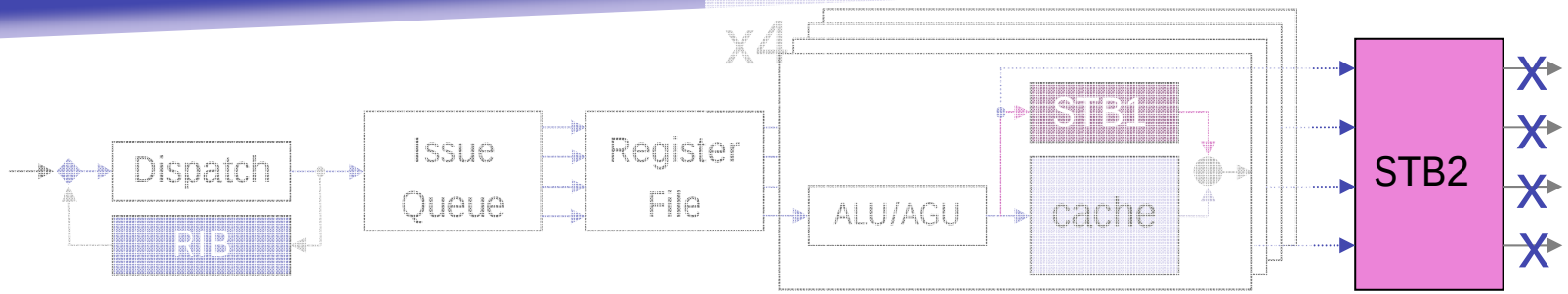
IQ Occupancy: Recovery from RIB

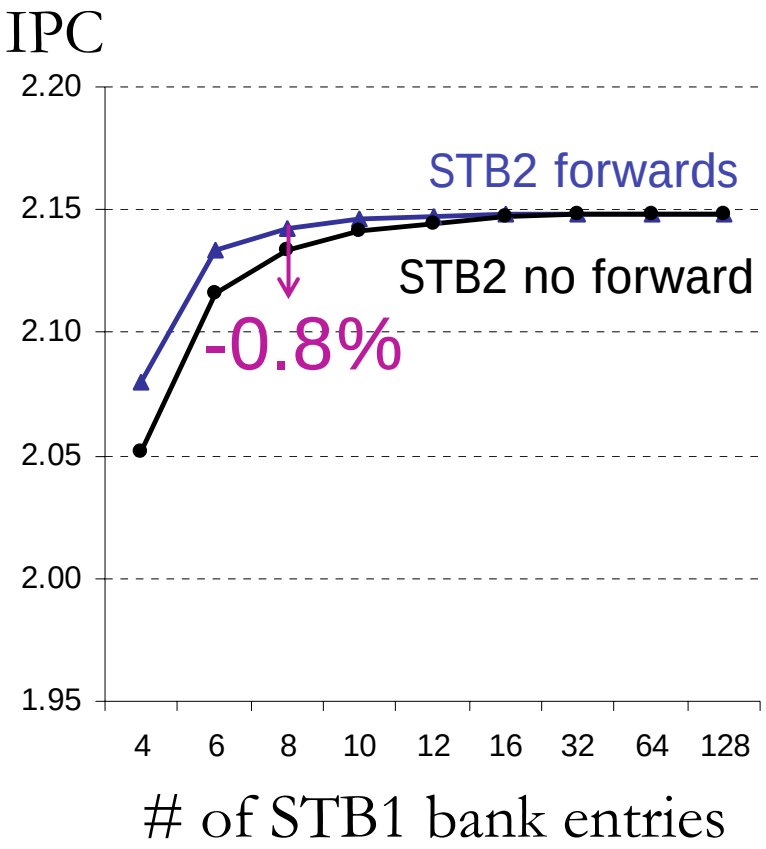




- high FWD coverage
  - 0.1% LD misspeculations
- recovery from RIB instead of recovery from IQ

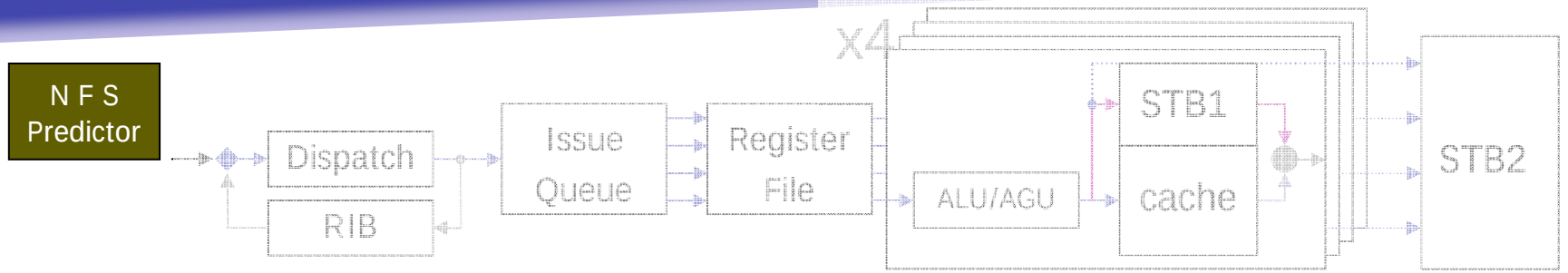
STB2 simplification: no forward

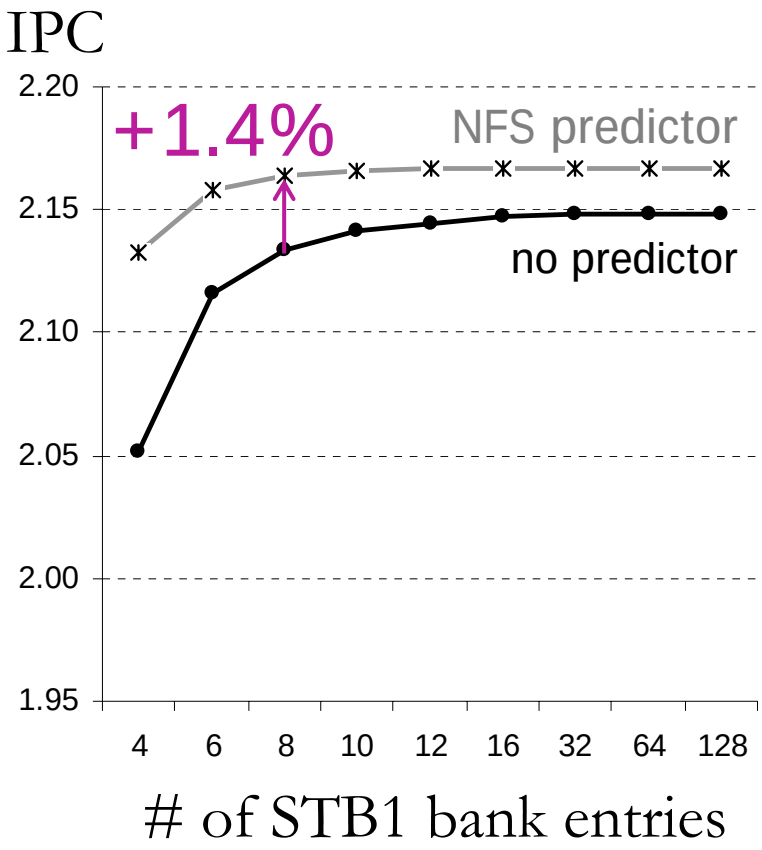




- STB2 forwards
  - STB2 data read ports
  - datapath (bypass network)
  - IQ scheduling (2 latencies)
- STB2 does not forward
  - LD wait until ST commits

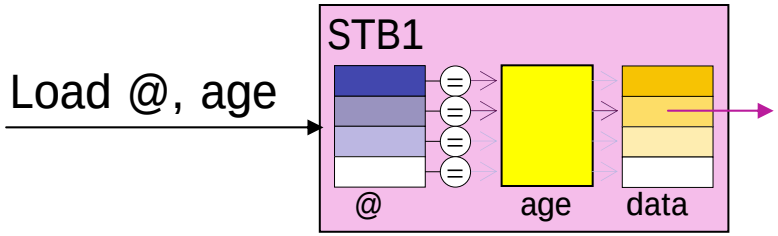
Reducing Contention: NFS predictor

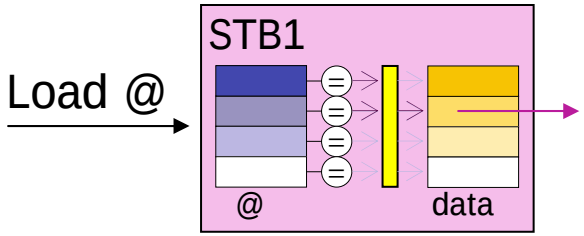


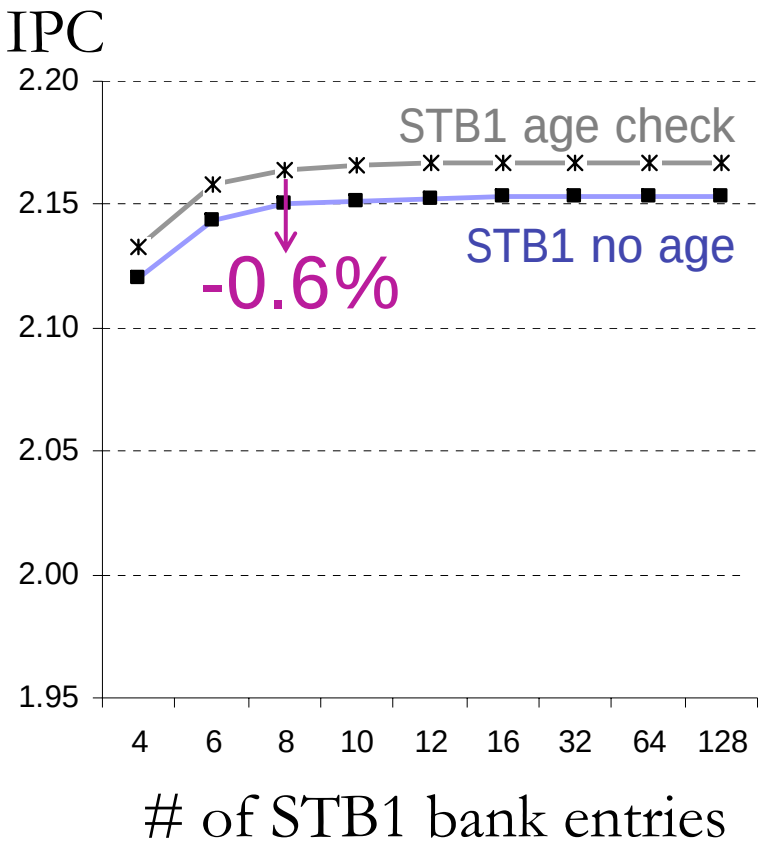


- stores: 70% do not FWD
  - use free issue memory port
- NFS predictor
  - 4K sat. counters, 3 bits
  - 64% ST classified as NFS
    - issue memory port
  - 0.47% false negative NFS
    - recover & wait

STB1 simplification: no check age

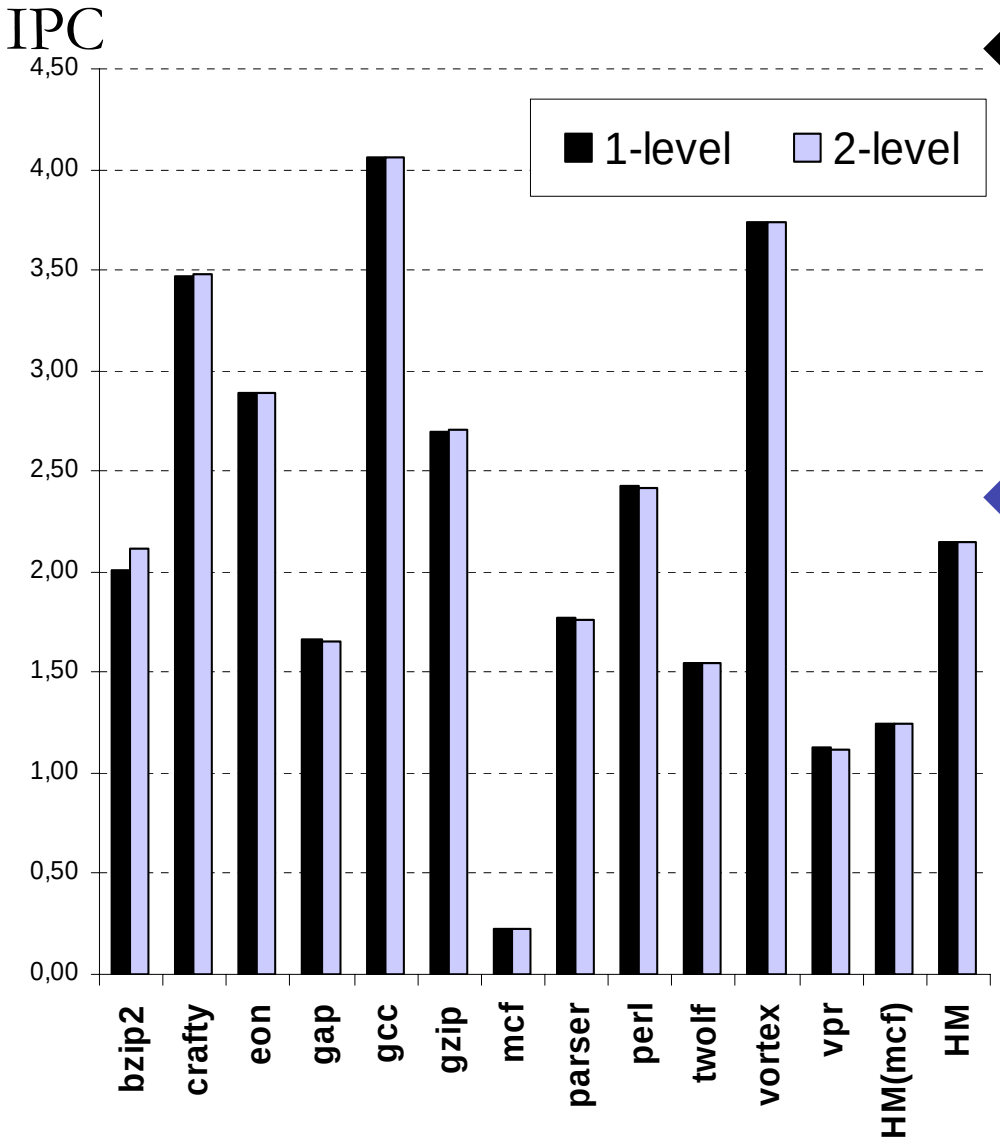






- compare address
  - on a match
    - ~~select in program order using Instruction age~~
    - select last allocated
- STB2 checks STB1

Summarizing



- 1-Level
  - 128-entry banks cache latency
- 2-Level
  - 8-entry STB1 banks cache latency
  - 128-entry STB2 cache latency +5 cycles

Talk Outline

- Introduction
- Processor Model
- Store Lifetime
- 2-Level STB Design
- Design Enhancements
- Conclusions

Related Work

- Akkary et al., MICRO 2003

  Checkpoint Processing and Recovery: Towards Scalable Large IW Processors

- Sethumadhavan et al. MICRO 2003

  Scalable Hardware Memory Disambiguation for High ILP Processors

- Park et al. MICRO 2003

  Reducing Design Complexity of the Load/Store Queue

- Cain & Lipasti, ISCA 2004

  Memory Ordering: A Value-Based Approach

- Baugh & Zilles, $P=ac^2$ 2004

  Decomposing the Load-Store Queue by Function for Power Reduction and Scalability

Conclusions

- two-level distributed STB
  - distributed STB1
    - speculative forwarding
    - 1 port, small banks (within cache bank latency)
    - circular buffer, allocated at execution
  - simplifications: hardware
    - STB1 does not use instruction age
    - STB2 does not forward
  - improvements
    - recovery from RIB
    - NFS Predictor