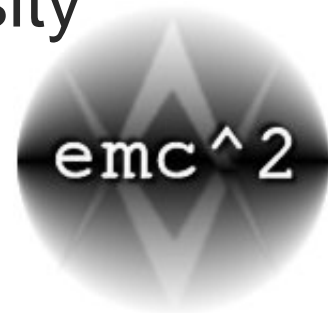




Exploring Wakeup-Free Instruction Scheduling

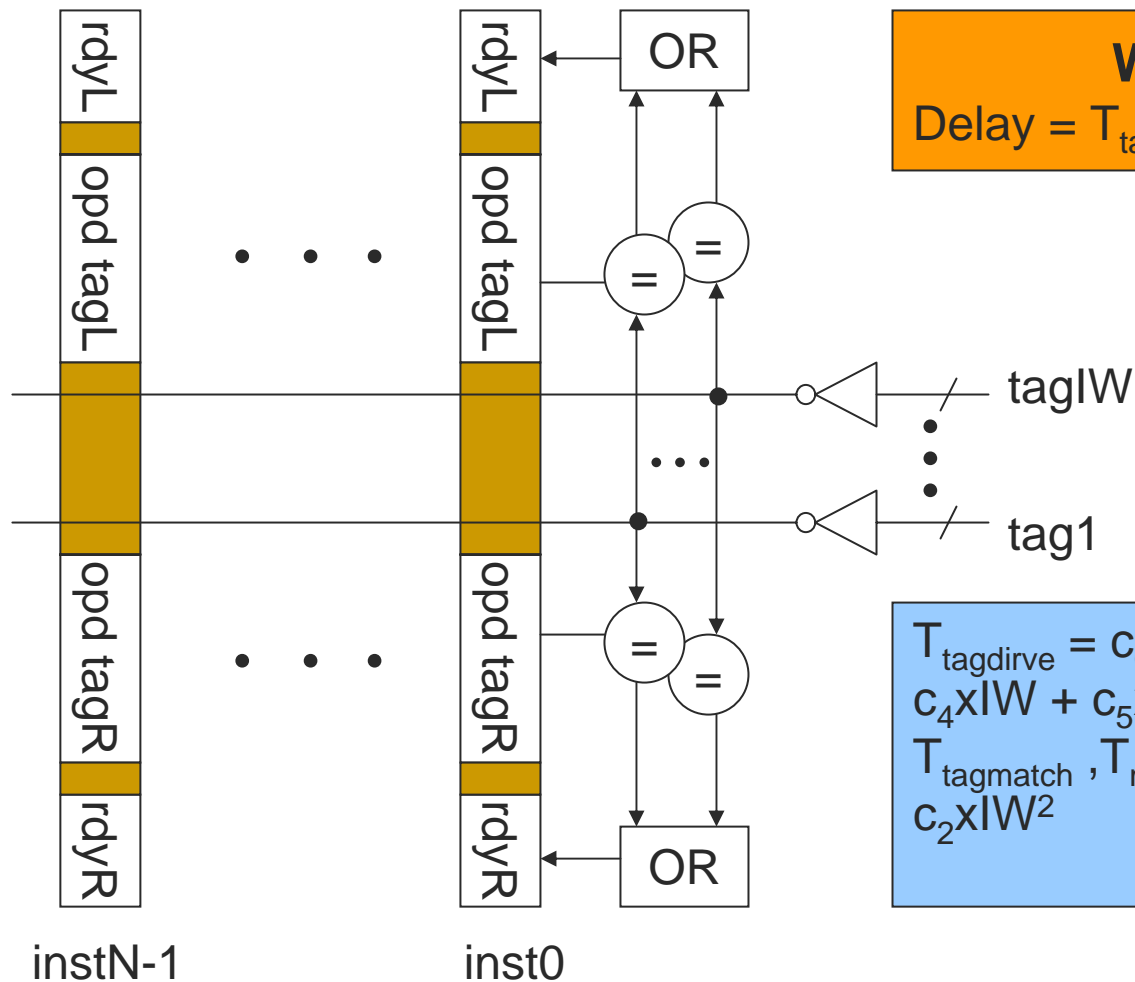
Jie S. Hu, N. Vijaykrishnan, and
Mary Jane Irwin
Microsystems Design Lab
The Pennsylvania State University



[Outline]

- Motivation
- Case study: Cyclone
- Towards high-performance wakeup-free scheduler
 - A general model
 - Employing pre-check scheme
 - A segmented issue queue
- Conclusions and future work

Superscalar Issue Queue



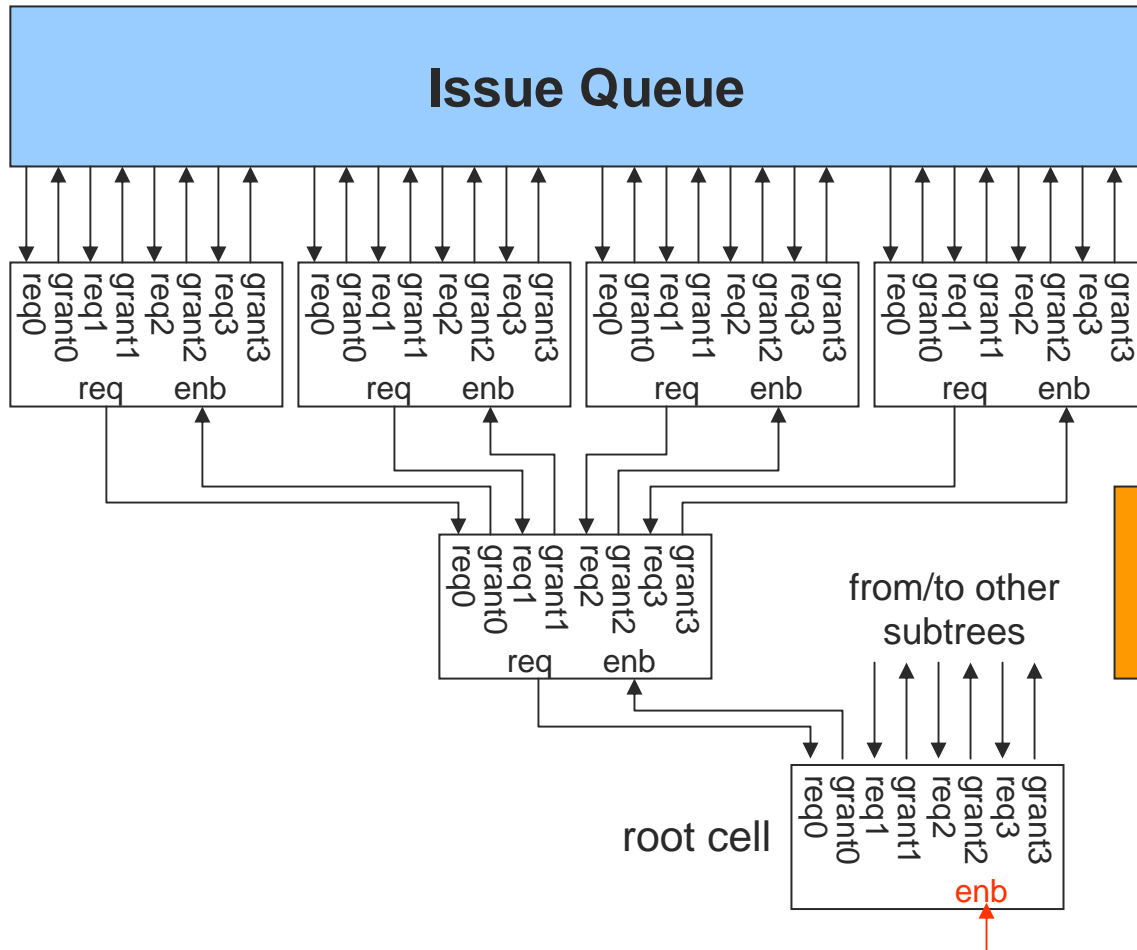
Wakeup Logic
 $Delay = T_{tagdirve} + T_{tagmatch} + T_{matchOR}$

$$T_{tagdirve} = c_0 + (c_1 + c_2 \times IW) \times N + (c_3 + c_4 \times IW + c_5 \times IW^2) \times N^2$$

$$T_{tagmatch}, T_{matchOR} = c_0 + c_1 \times IW + c_2 \times IW^2$$

S. Palacharla et al., ISCA24

Superscalar Issue Queue



Selection Logic

$$T_{\text{selection}} = c_0 + c_1 x \log_4 N$$

S. Palacharla et al., ISCA24

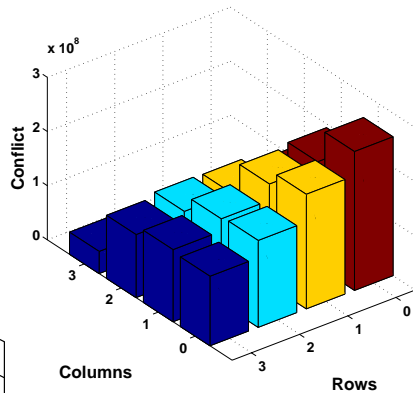
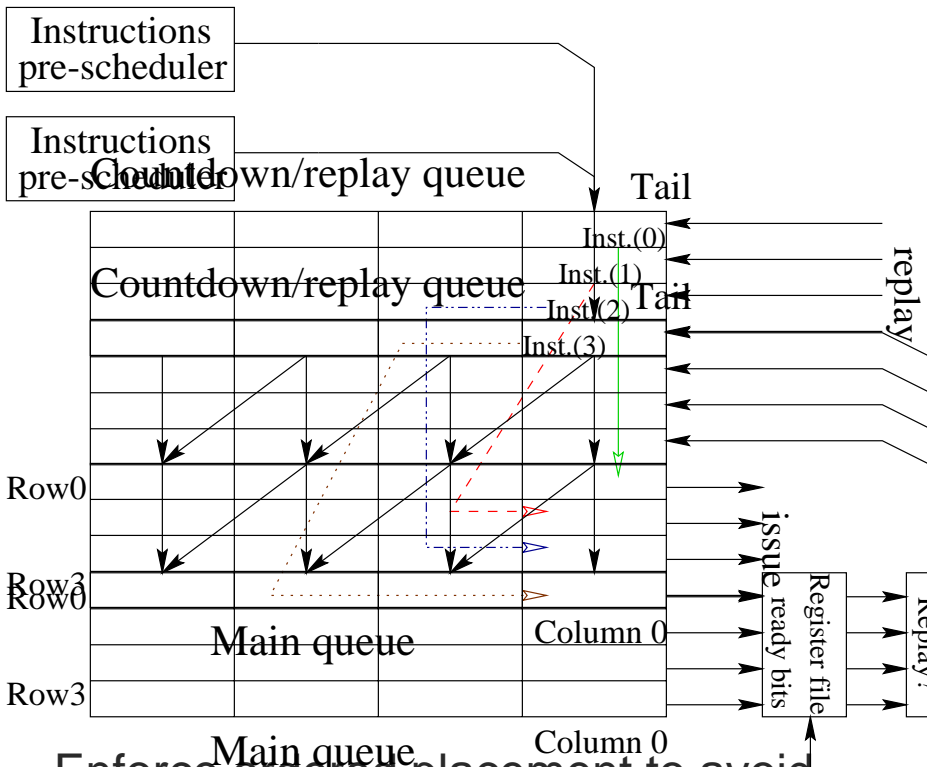
Challenges in Dynamic Instruction Scheduling

- Broadcast-based dynamic scheduler
 - Higher complexity
 - Power hungry
 - A major limiter to clock frequency: increasing issue queue size, issue width, wire delay, and shorten logic levels per pipeline stage
- Wakeup-free dynamic scheduler (e.g., Cyclone)
 - Lower complexity
 - Lower power consumption
 - Better scalability
 - 👉 Have to trade performance loss

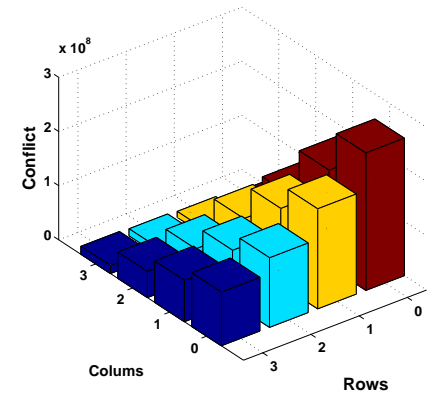
[Our Goals]

- Explore the predictability of instruction issue latency
- Identify the performance impediments in wakeup-free architectures
- Design high-performance wakeup-free schedulers

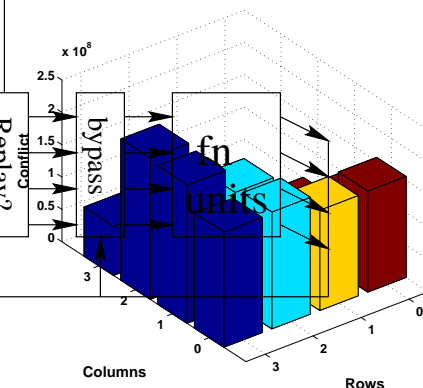
Cyclone: Conflict in the Main Queue



FP benchmarks



Int benchmarks



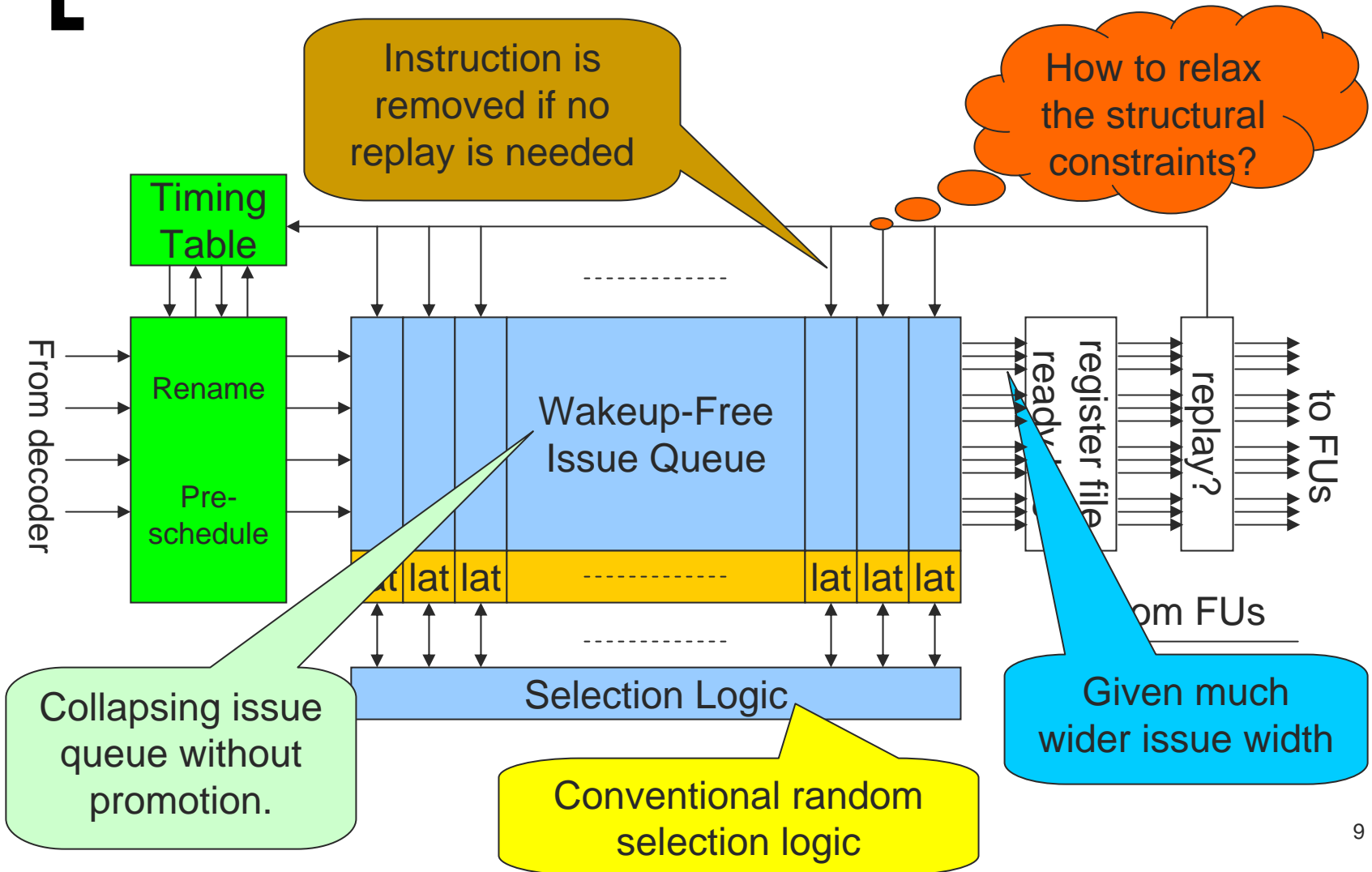
Order Enforced

Enforce ordered placement to avoid conflict between instructions with different latencies

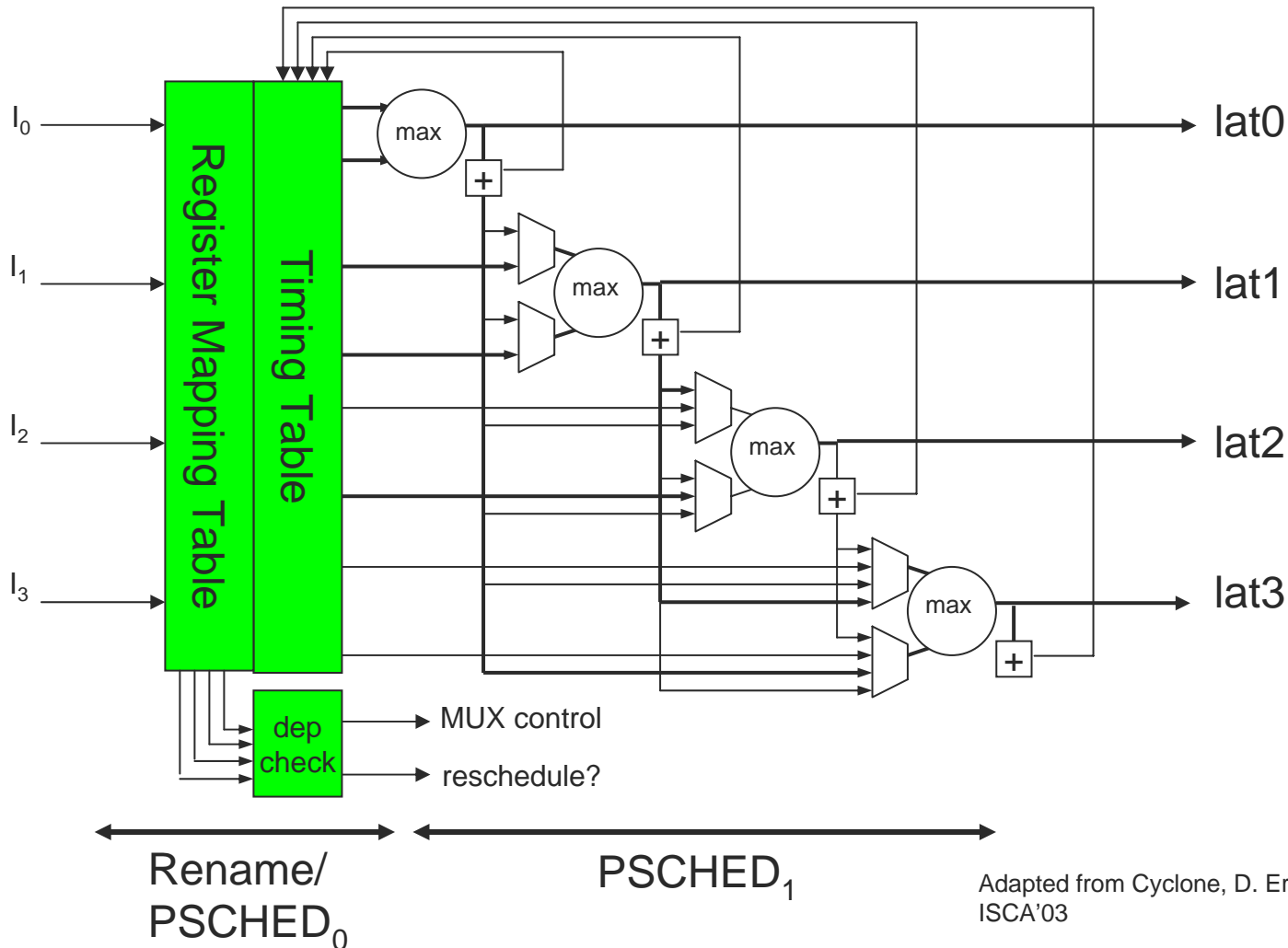
Possible Structural Problems

- Instruction promotion/forwarding incurs conflict along the path
- Very limited instruction pool for selection
 - Only entries in column 0 in the main queue can be issued
 - Ready instructions (not in column 0) are delayed due to conflict
- Limited number of issue ports has less tolerance to mispredicted ready instructions
 - Waste issue port
 - Prevent ready instruction from issue
 - Complete with newly decoded instructions due to replay

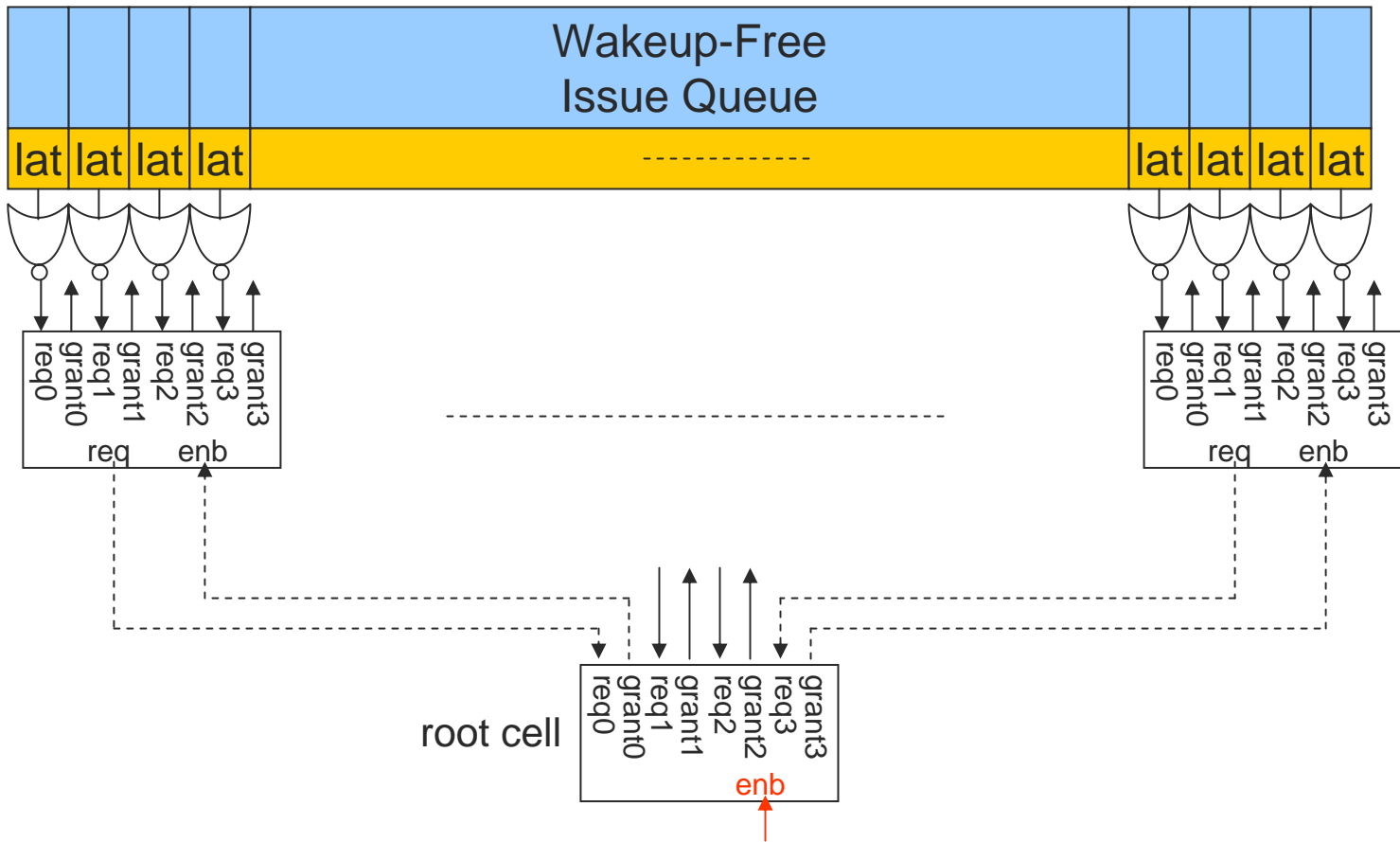
A General Model: WF-Replay



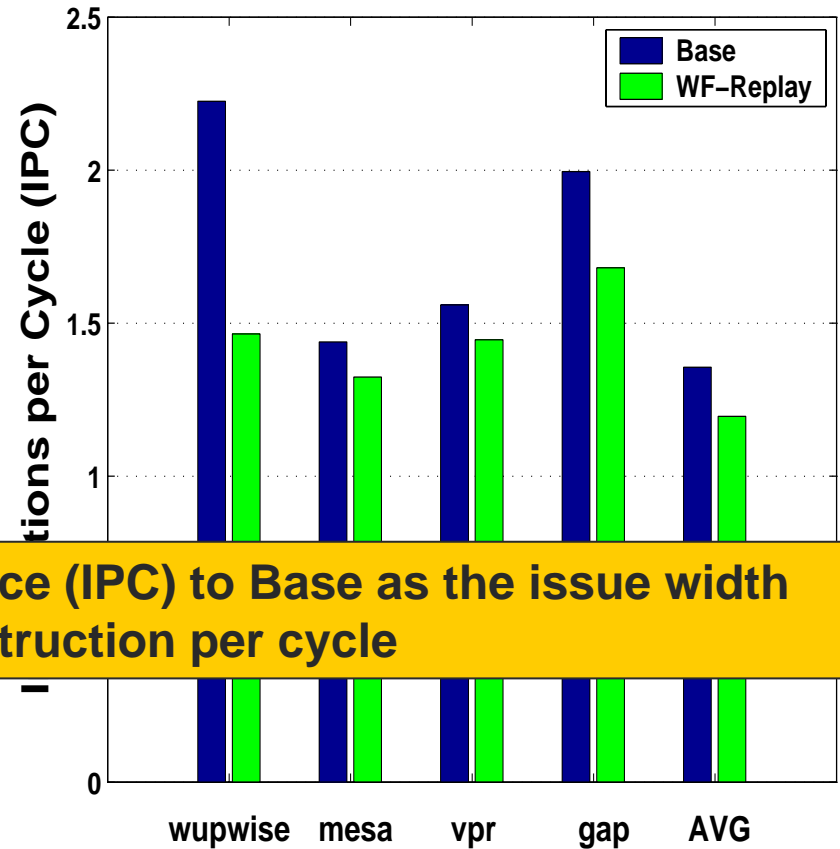
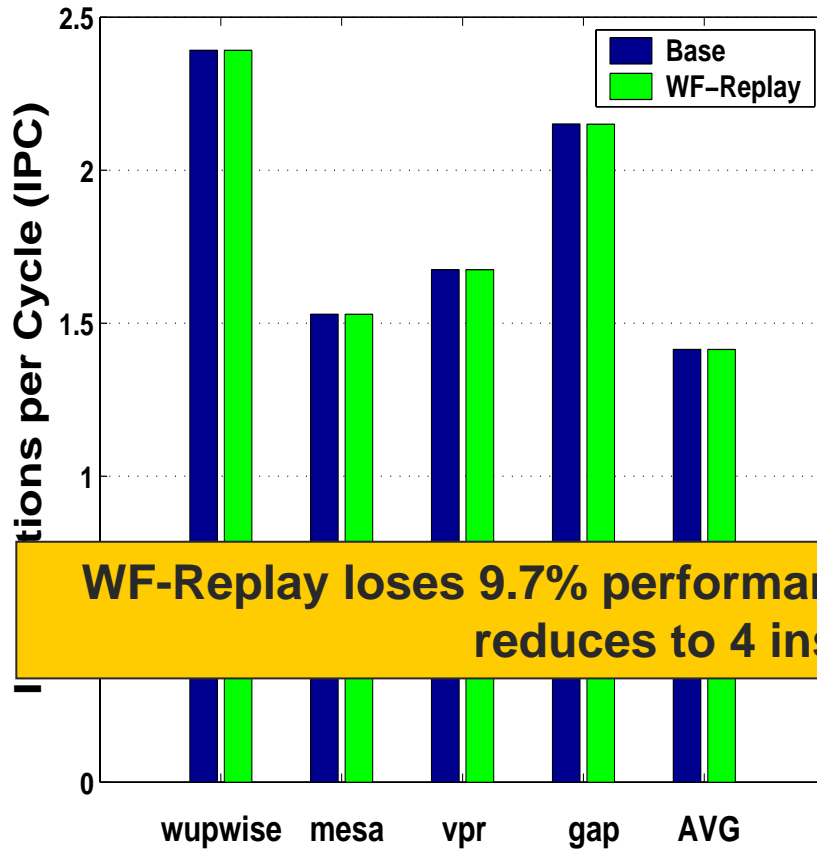
Instruction Pre-scheduling



Latency Triggered Selection



WF-Replay IPC (F4-I8 vs F4-I4)

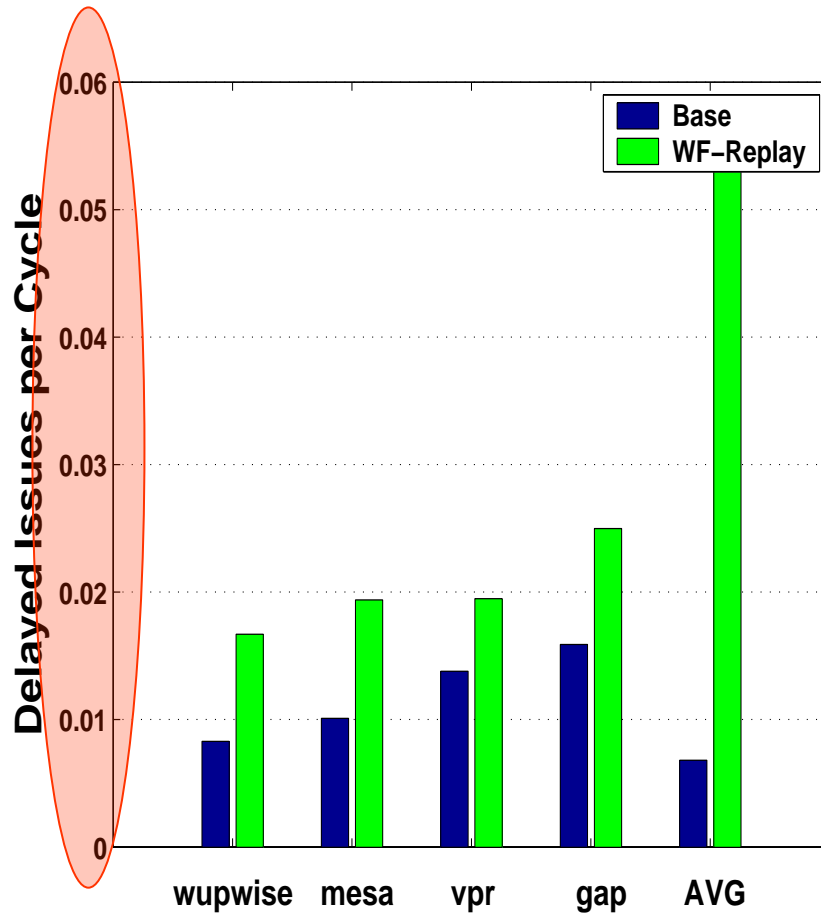


WF-Replay loses 9.7% performance (IPC) to Base as the issue width reduces to 4 instruction per cycle

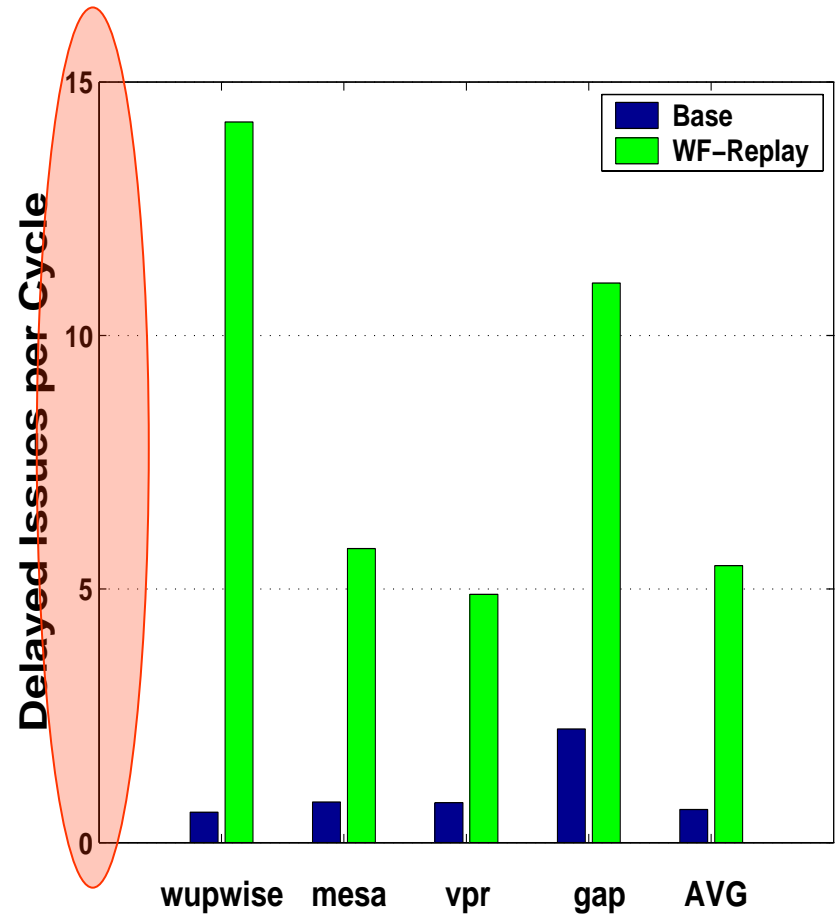
Issue Width: 8

Issue Width: 4

Competition at Issue Ports?



Issue Width: 8

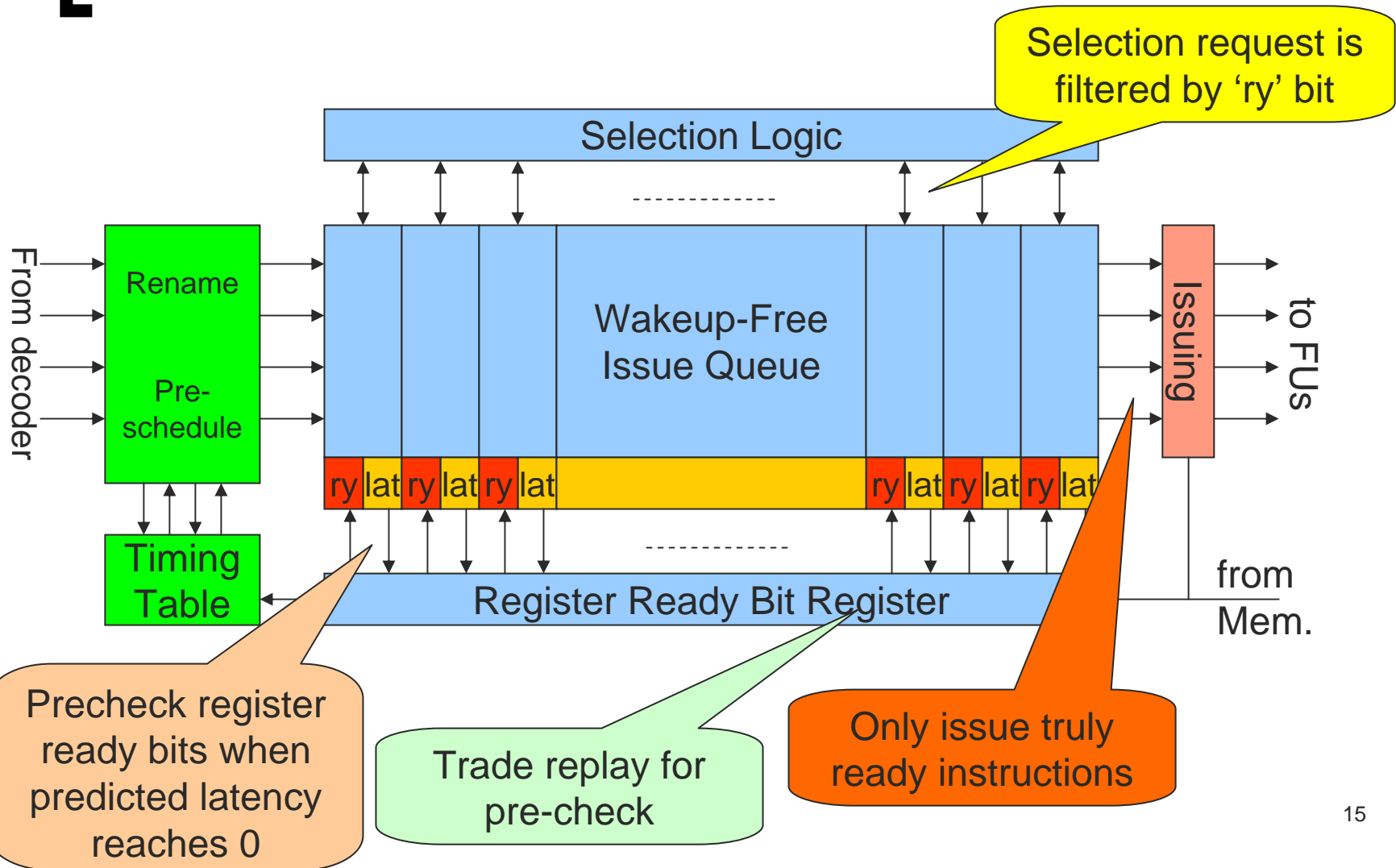


Issue Width: 4

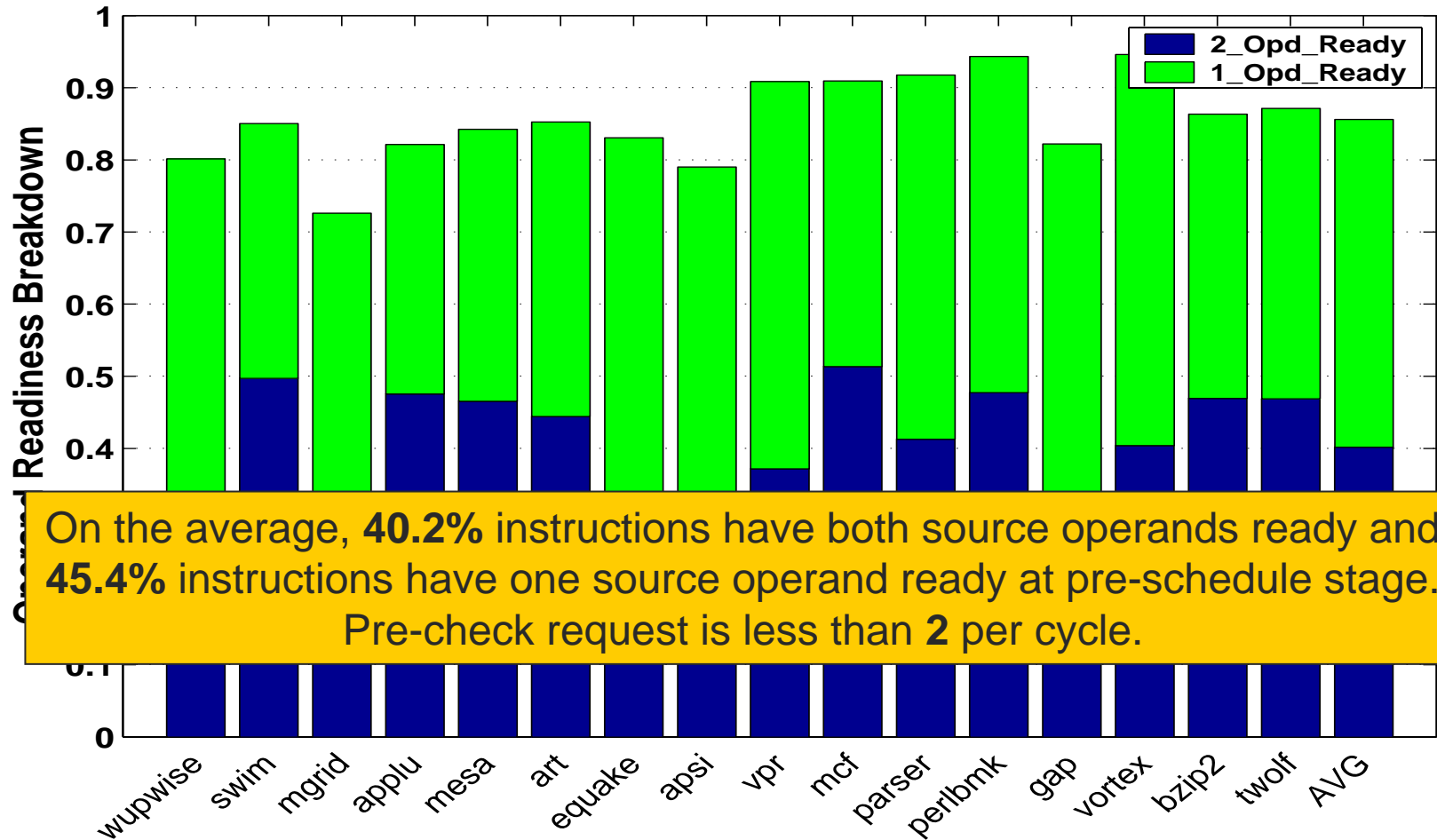
[Precheck to Avoid Competition]

- Competition at issue port may delay ready (predictive) instructions
- Delayed instructions may again compete with instructions dependent on them
- Causing more instructions falsely ready or to be delayed
- Wider issue port can avoid unnecessary competition at cost of **higher complexity**
- **Solution: preventing falsely ready instructions from selection by pre-checking register ready bits**

WF-Precheck Scheduler

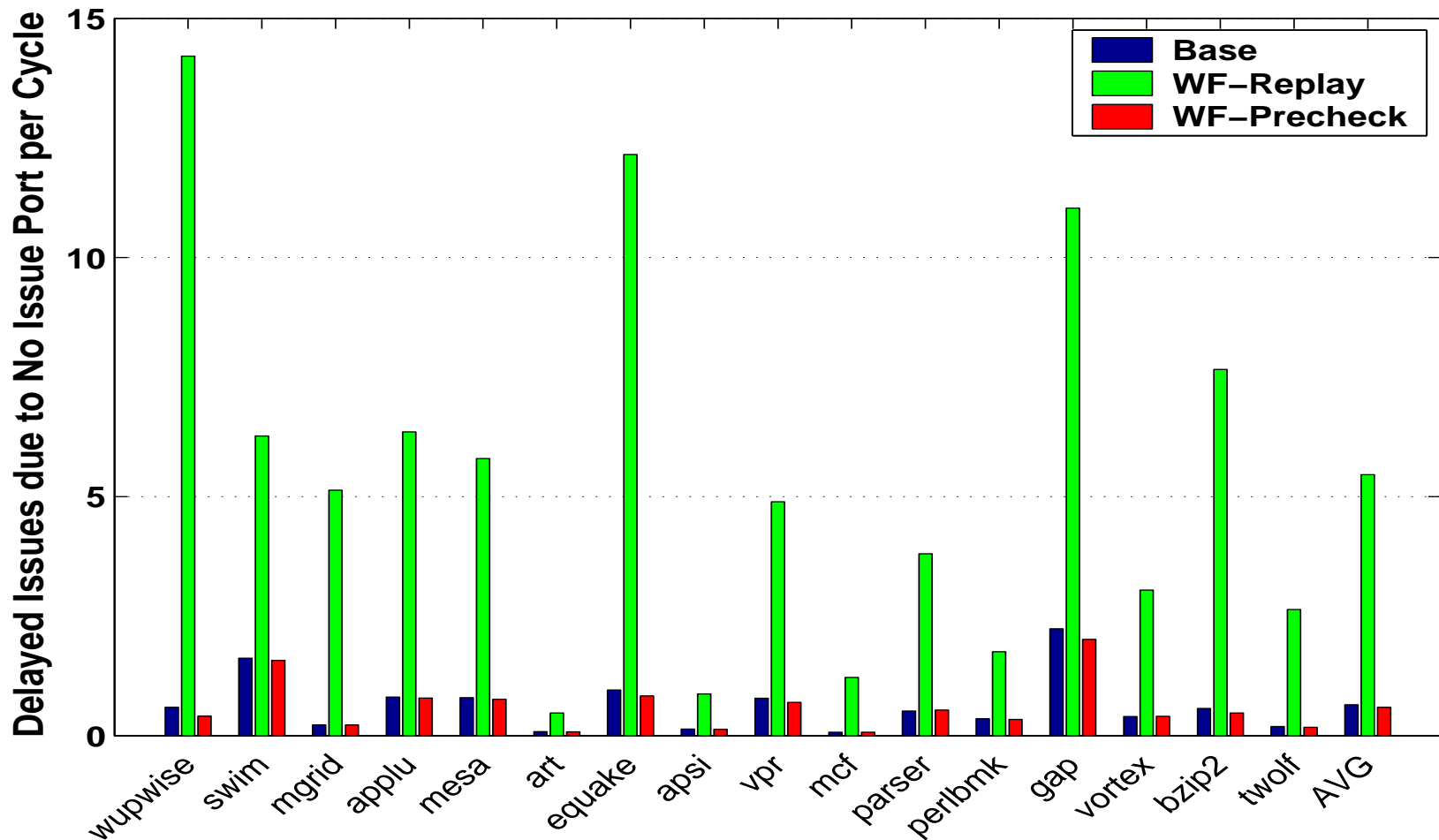


Complexity of Pre-checking

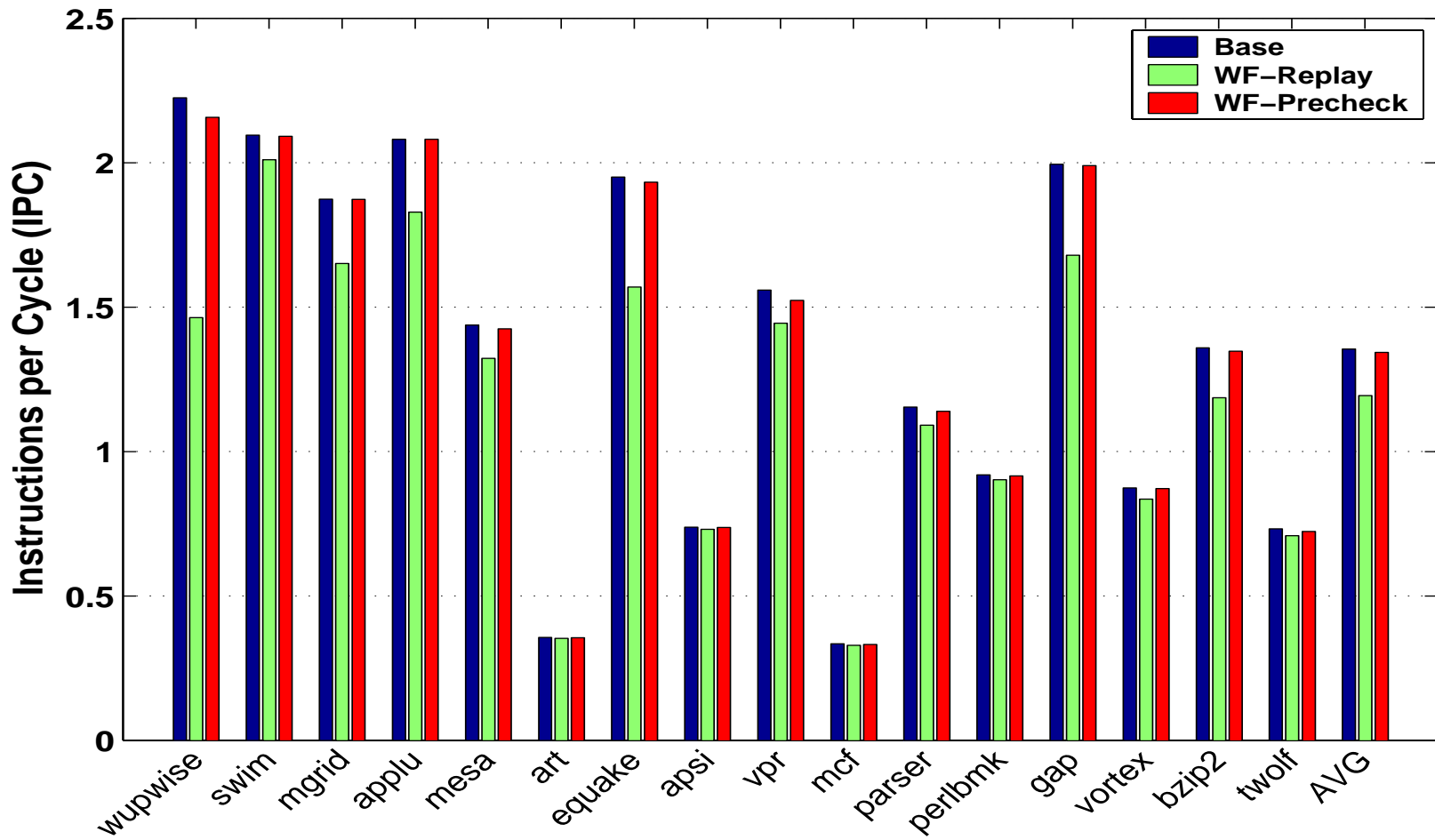


On the average, **40.2%** instructions have both source operands ready and **45.4%** instructions have one source operand ready at pre-schedule stage. Pre-check request is less than **2** per cycle.

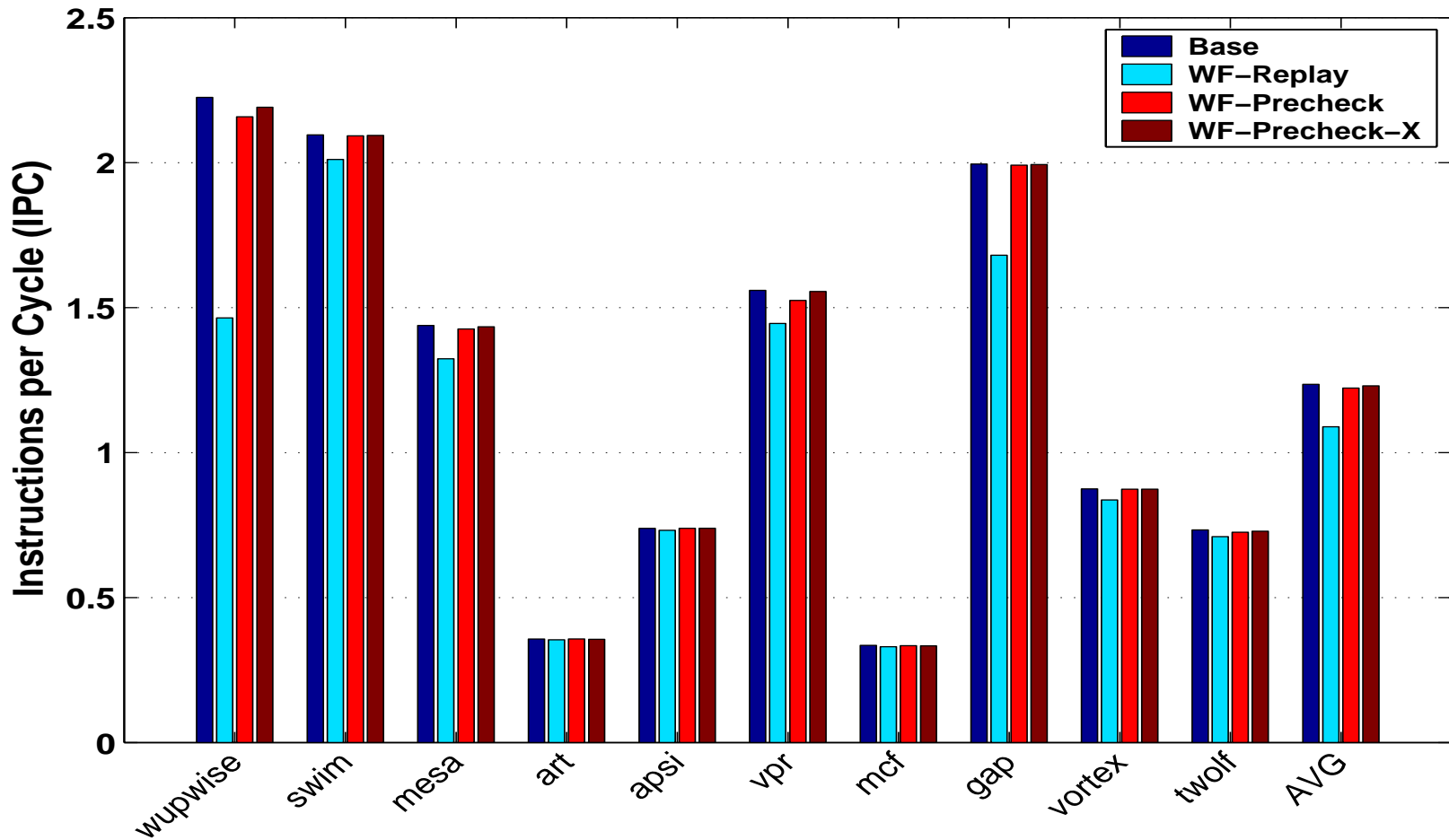
Issue Port Competition (F4-I4)



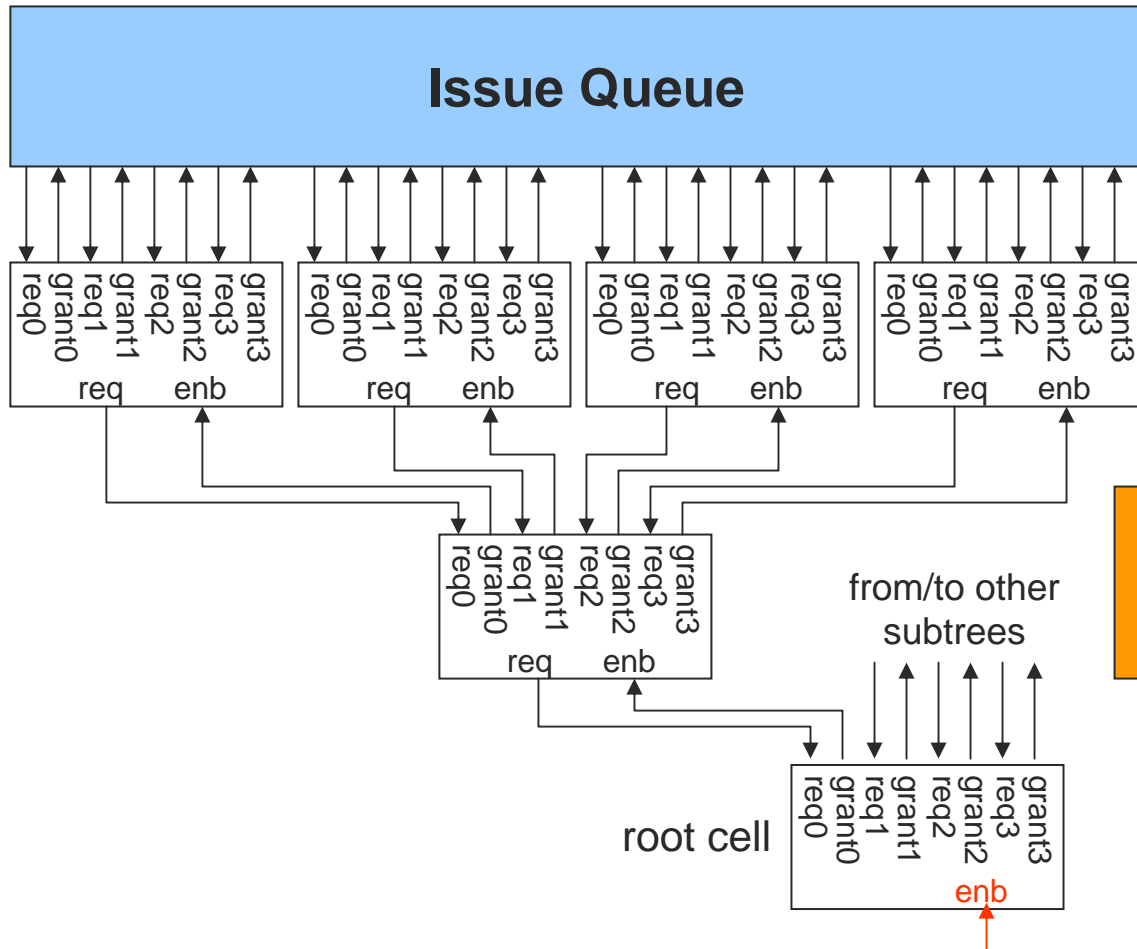
WF-Precheck IPC (F4-I4)



Precheck as A Single Stage



How about Selection Logic?

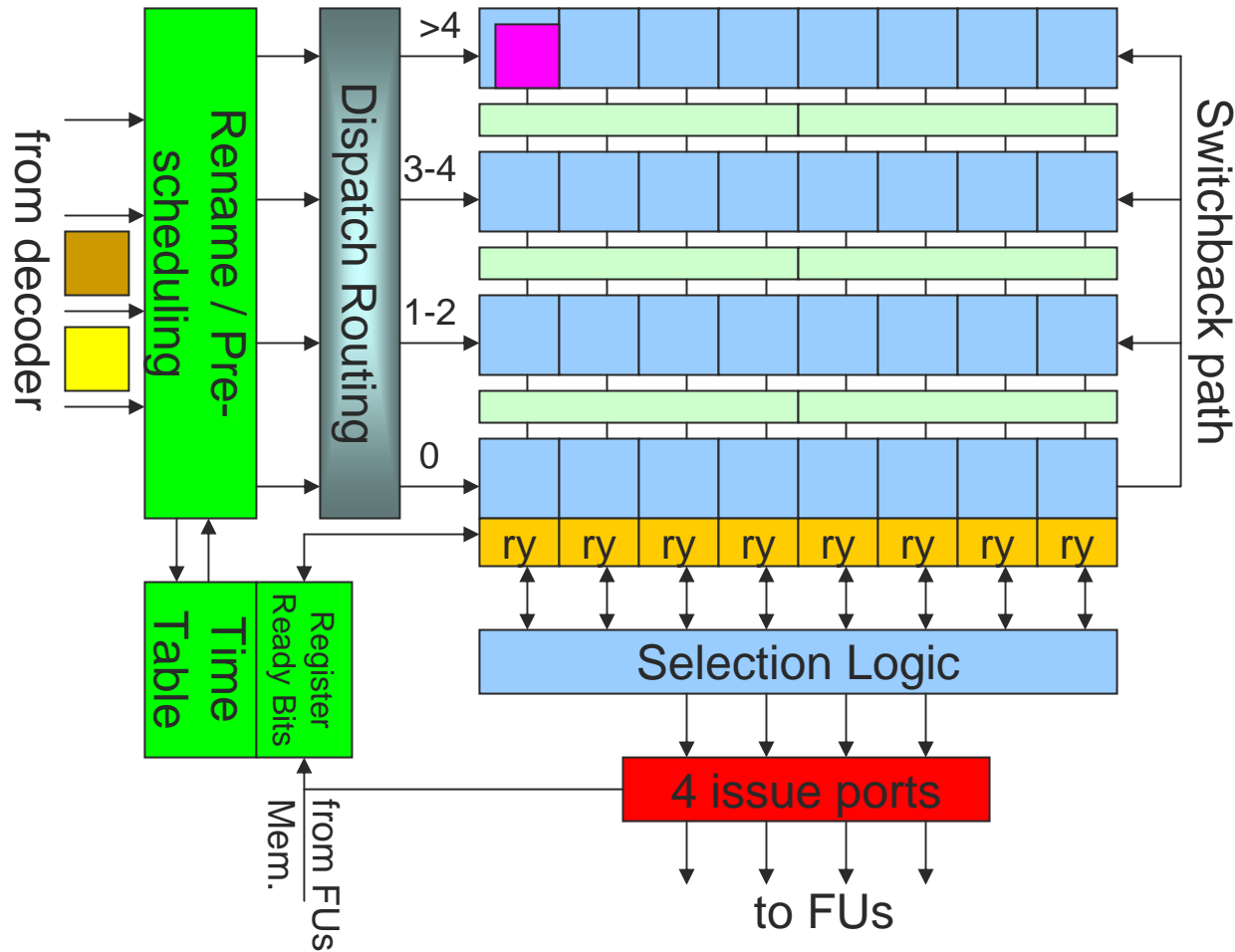


Selection Logic

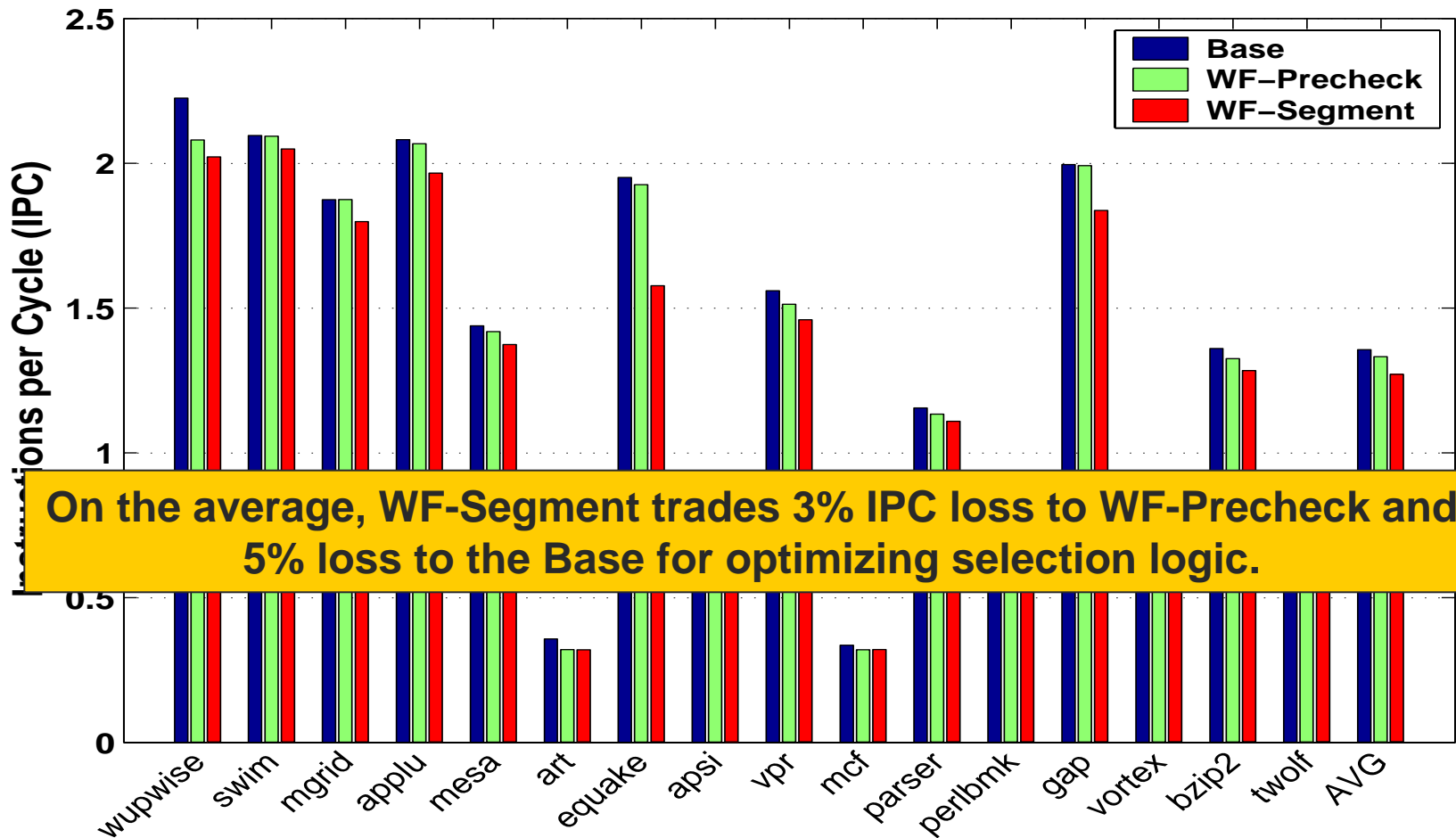
$$T_{\text{selection}} = c_0 + c_1 x \log_4 N$$

S. Palacharla et al., ISCA24

WF-Segment Issue Queue



WF-Segment Issue Queue



Conclusions

- Explore and identify the performance impediments in wakeup-free scheduling
- High-performance wakeup-free dynamic schedulers
 - **WF-Replay**: eliminates structural constraints
 - **WF-Precheck**: avoids unnecessary competition at issue ports
 - **WF-Segment**: optimizes selection logic for high clock speed

[Future Work]

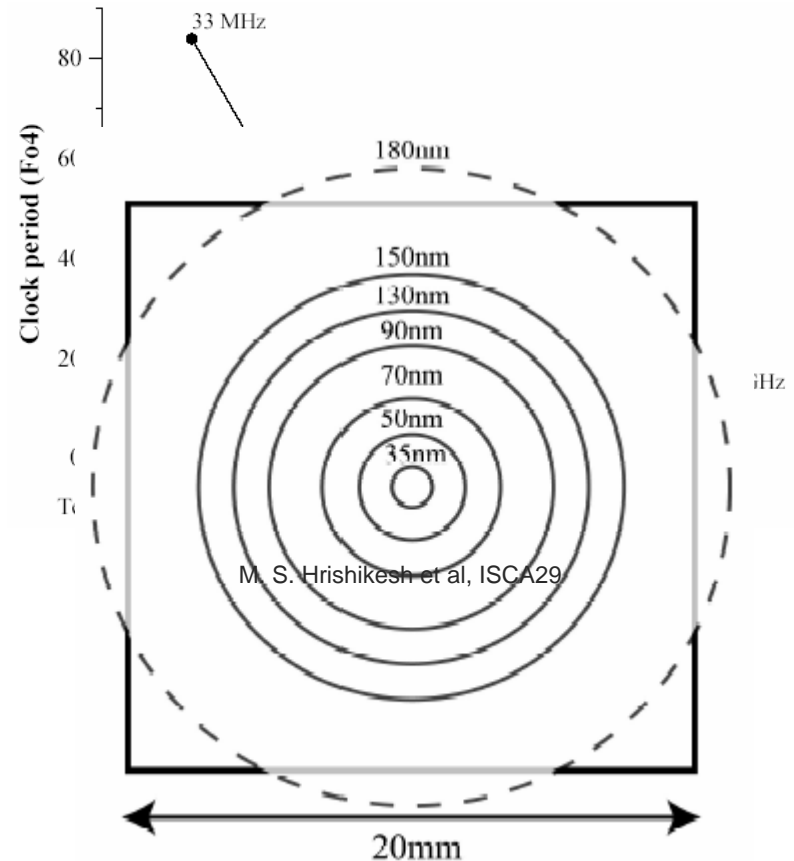
- Routing complexity analysis in **WF-Segment** scheduler
- Power analysis for wakeup-free schedulers
- Sophisticated pre-scheduler
 - Utilize FU usage and reservation information
 - Towards selection-free



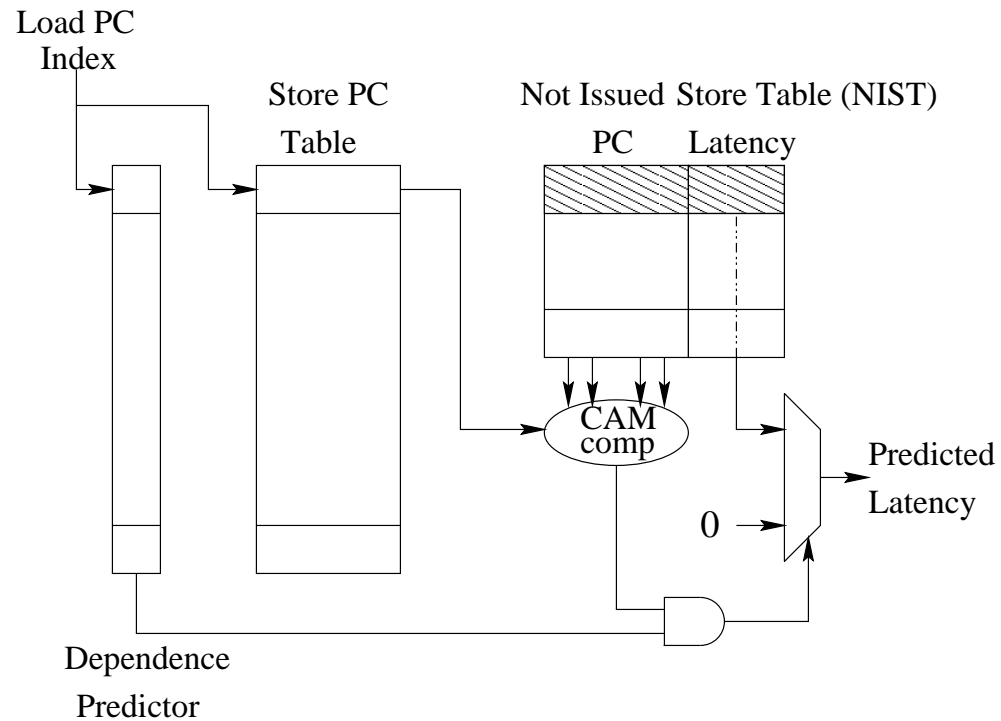
Thank You!

Wire Delay Challenges

- Increasing pipeline depth for high performance
- Clock period (FO4) decreases dramatically
- Cross-chip wire delay will be up to 10 cycles as technology shrinks



Load/Store Dependence Predictor



Impact of Load Related Predictions

