Microarchitectural Support for Speculative Register Renaming

J. Alastruey[+], T. Monreal[+], V. Viñals[+], M. Valero[*]

[+]gaZ – I3A - Universidad de Zaragoza

[*] Universidad Politécnica de Cataluña-Barcelona Supercomputing Center

HiPEAC - High-Performance Embedded Architecture and Compilation

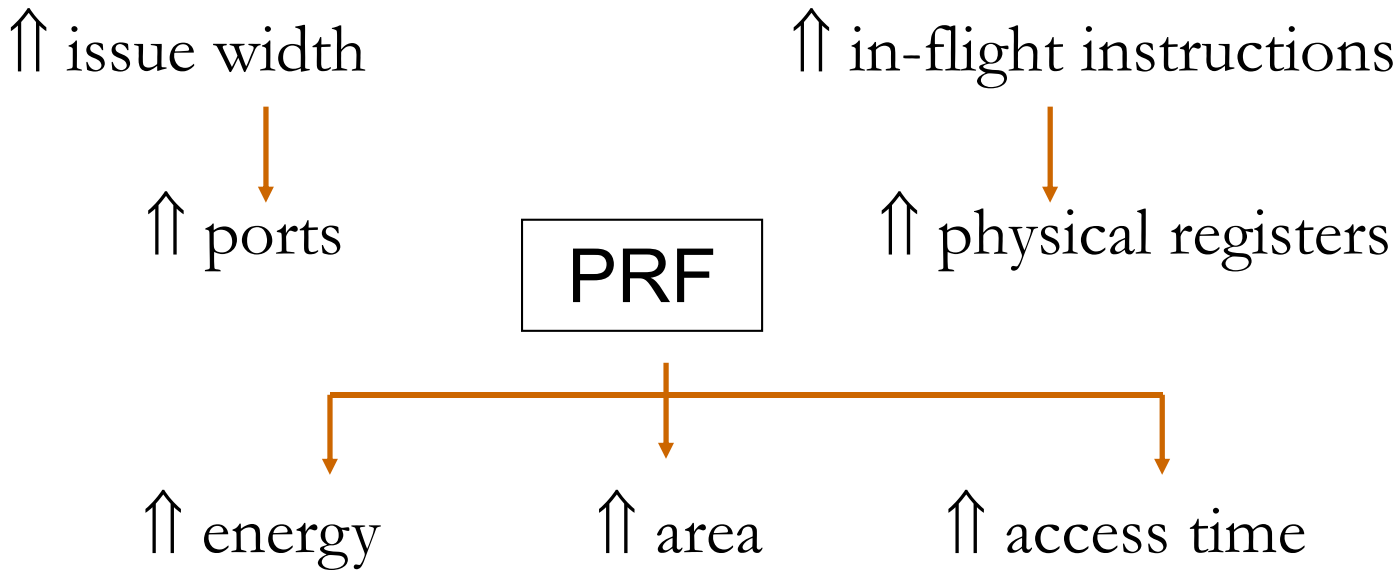
IEEE International Parallel & Distributed Processing Symposium

Long Beach, USA, 2007

gaZ

Overview

- Context
  - Monolithic register file of out-of-order superscalar processors
- Trends



Physical Register File is becoming more critical

Motivation

- Conventional renaming is inefficient
  - Many physical registers will NOT be read in the future
    - ⇑ register file pressure ⇒ ⇑ rename stalls

- Modified renaming mechanisms
  - Register reuse ⇒ ⇑ **IPC**
  - Reduce PRF size
    - ⇓ energy, area (linear)
    - ⇓ Taccess

      if PRF in critical path, ⇓ Tcycle ⇒ ⇑ **IPS**

gaZ

Motivation

- Register allocation and release
  - Late Allocation[1]
  - Omission of Physical Register Allocation (OPRA)[2,3]
  - Early Release of Physical Register (ERPR)[4,5,6]
- Speculative Renaming (SR)
  - Recovery actions
- Microarchitectures supporting SR
  - Very tuned to specific allocating or releasing policies
  - Limitations
    - Backup structure, recovery mechanism, ERPR and OPRA identification

gaZ

[1]T. Monreal et al., "Delaying Physical Register allocation …", MICRO 99.
[2]J.A. Butts and Sohi, "Dynamic dead-instruction detection and elimination", ASPLOS 02.
[3] D. Balkan et al., "SPARTAN: Speculative Avoidance of Register Allocation …", PACT 06.
[4]M. Moudgill et al., "Register Renaming and Dynamic Speculation: an Alternative Approach", MICRO 93.
[5]T. Monreal et al., "Hardware Schemes for Early Register Release", ICPP 02.
[6]O. Ergin et al., "Increasing Processor Performance Through Early Release", ICCD 04.

Contributions

- Microarchitecture supporting arbitrary Speculative Renaming policies
  - OPRA, ERPR
  - Software , hardware
- Implementation
  - Au**X**iliary **R**egister **F**ile (XRF) for recovery
  - Virtual Registers
- Evaluation of microarchitecure
  - Last-Use Predictor: OPRA and ERPR
  - Complexity-effective XRF and Last-Use Predictor

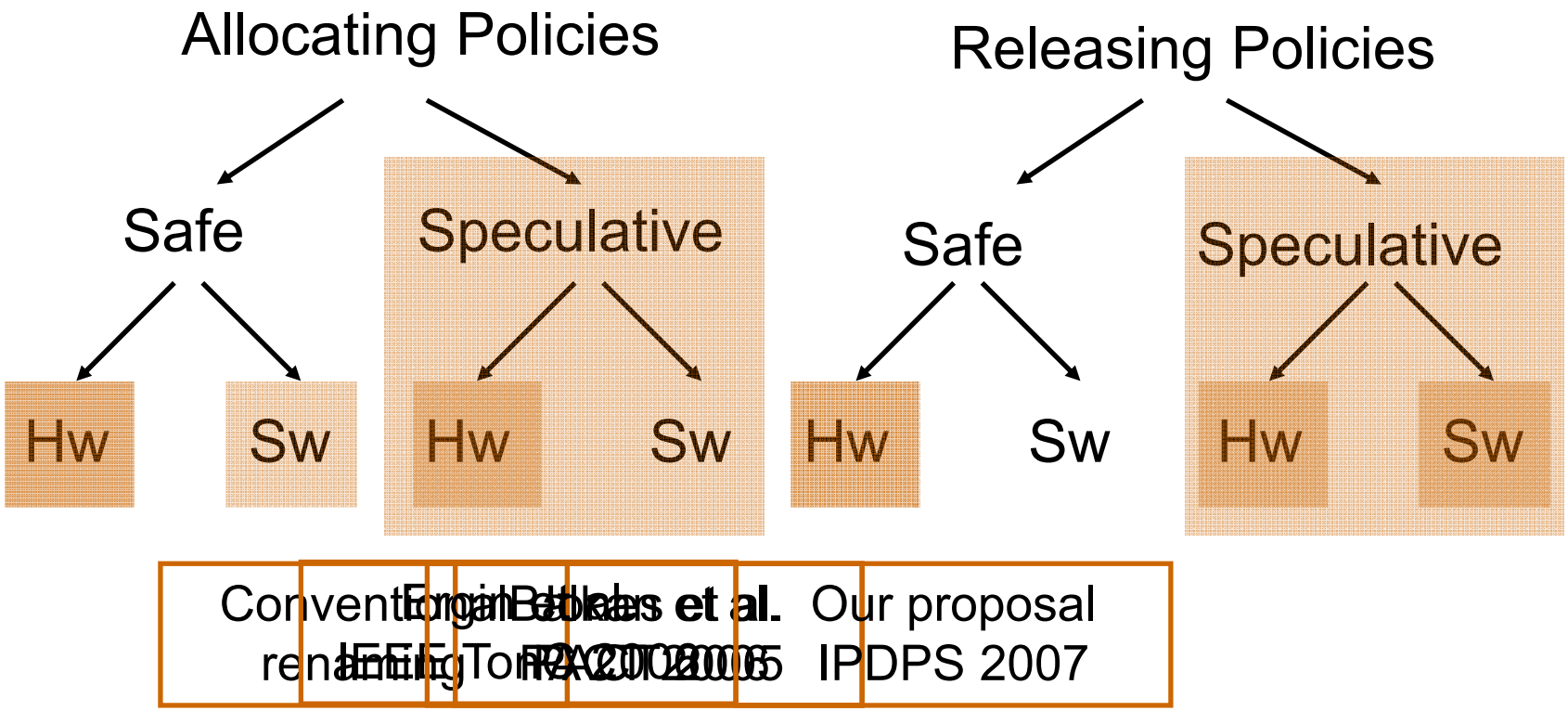
gaZ

Outline

- **Renaming mechanisms**
- SR microarchitecture
- SR-LUP
- Results
- Conclussions

gaZ

Taxonomy of Renaming Policies



Conventional renaming

V: r2 = ..

...

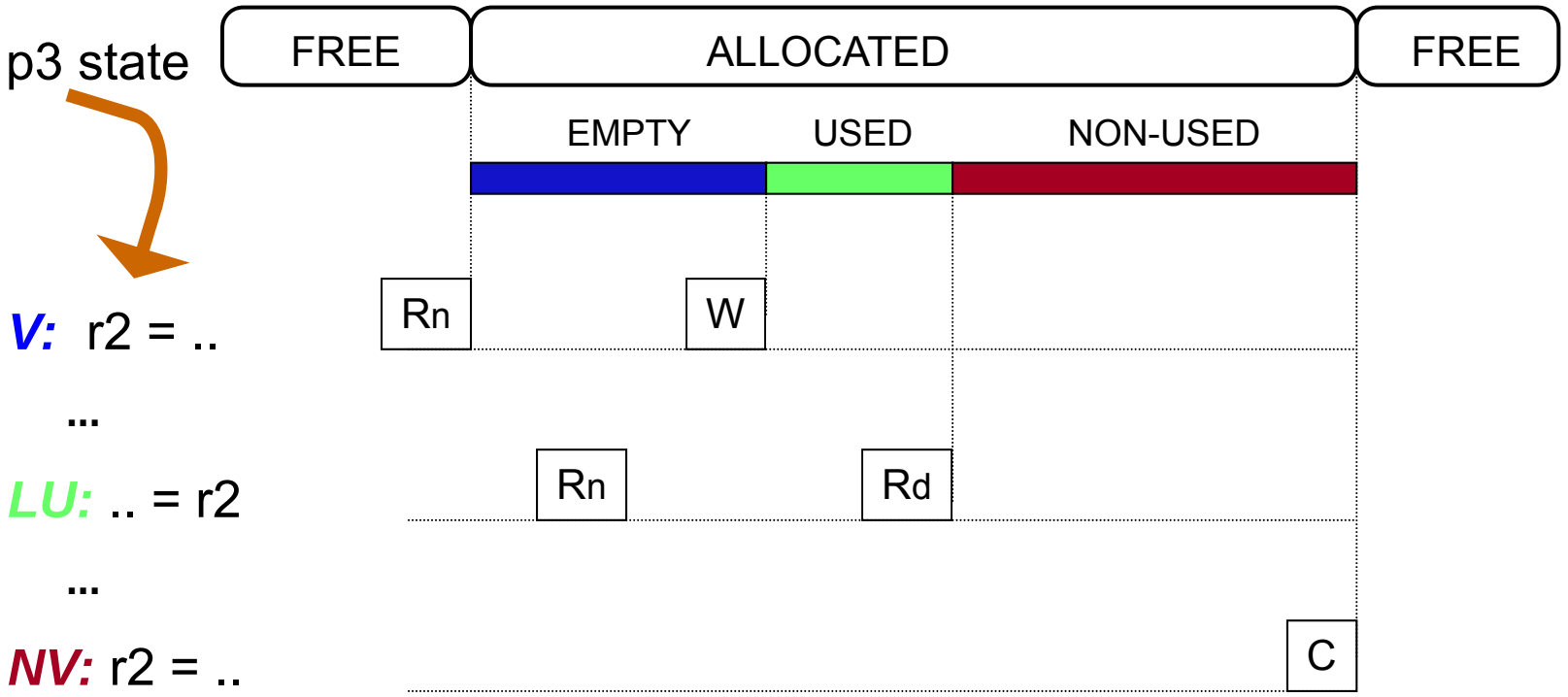
LU: .. = r2

...

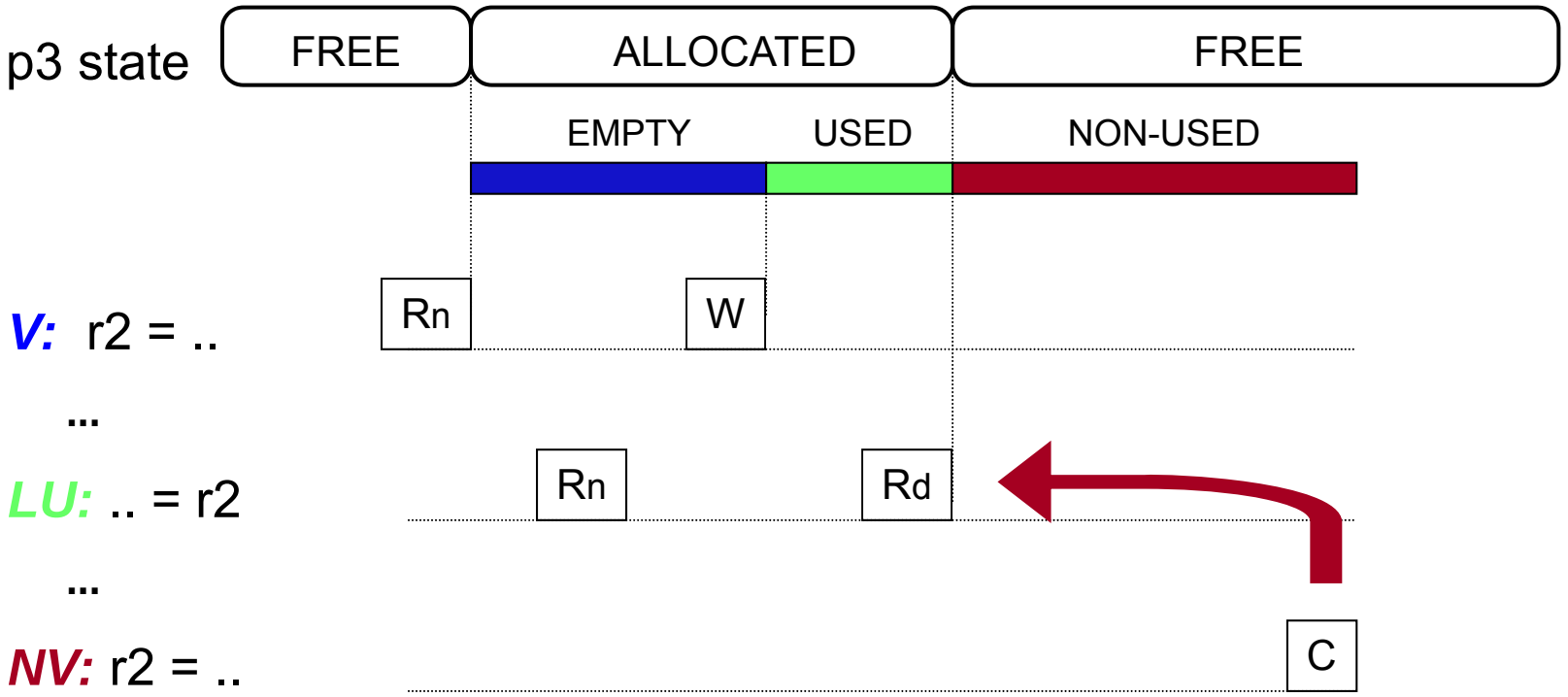
NV: r2 = ..

gaZ

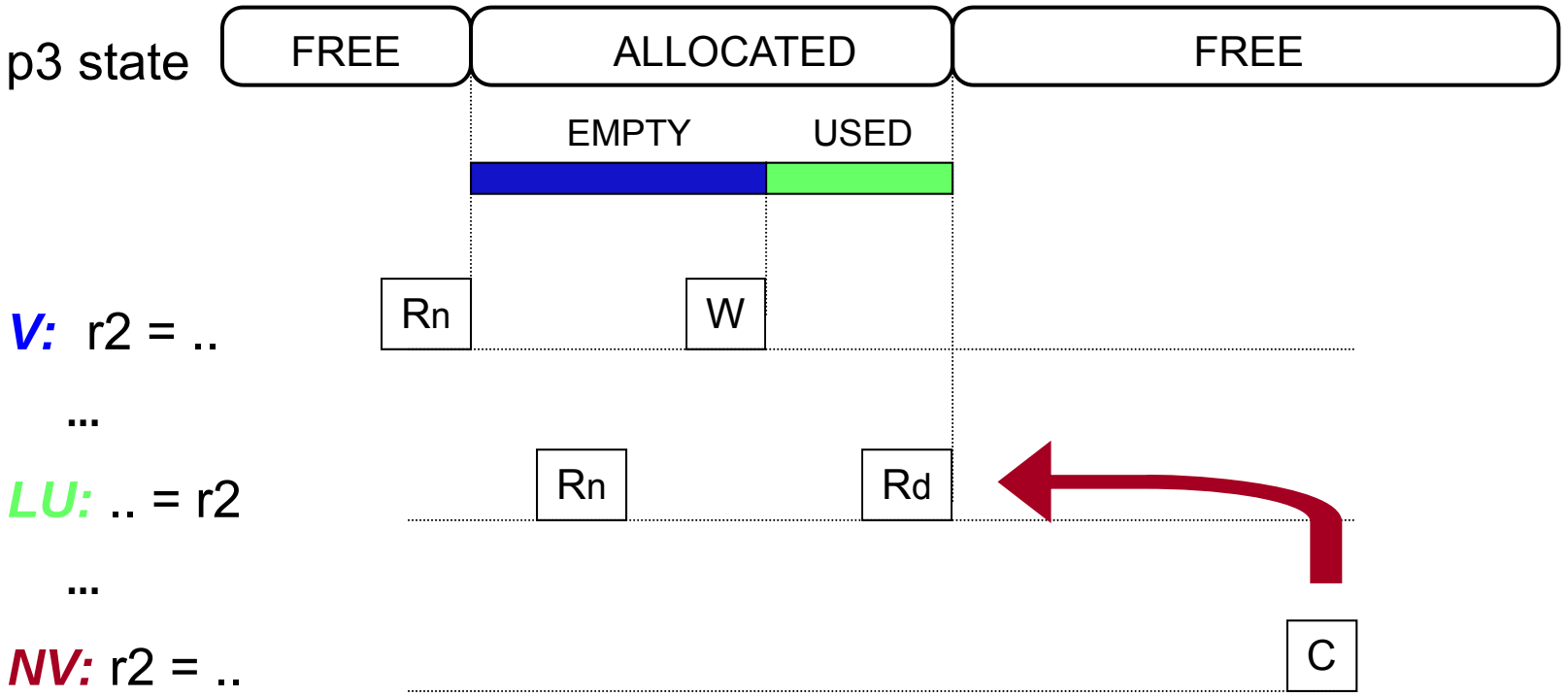
Conventional renaming



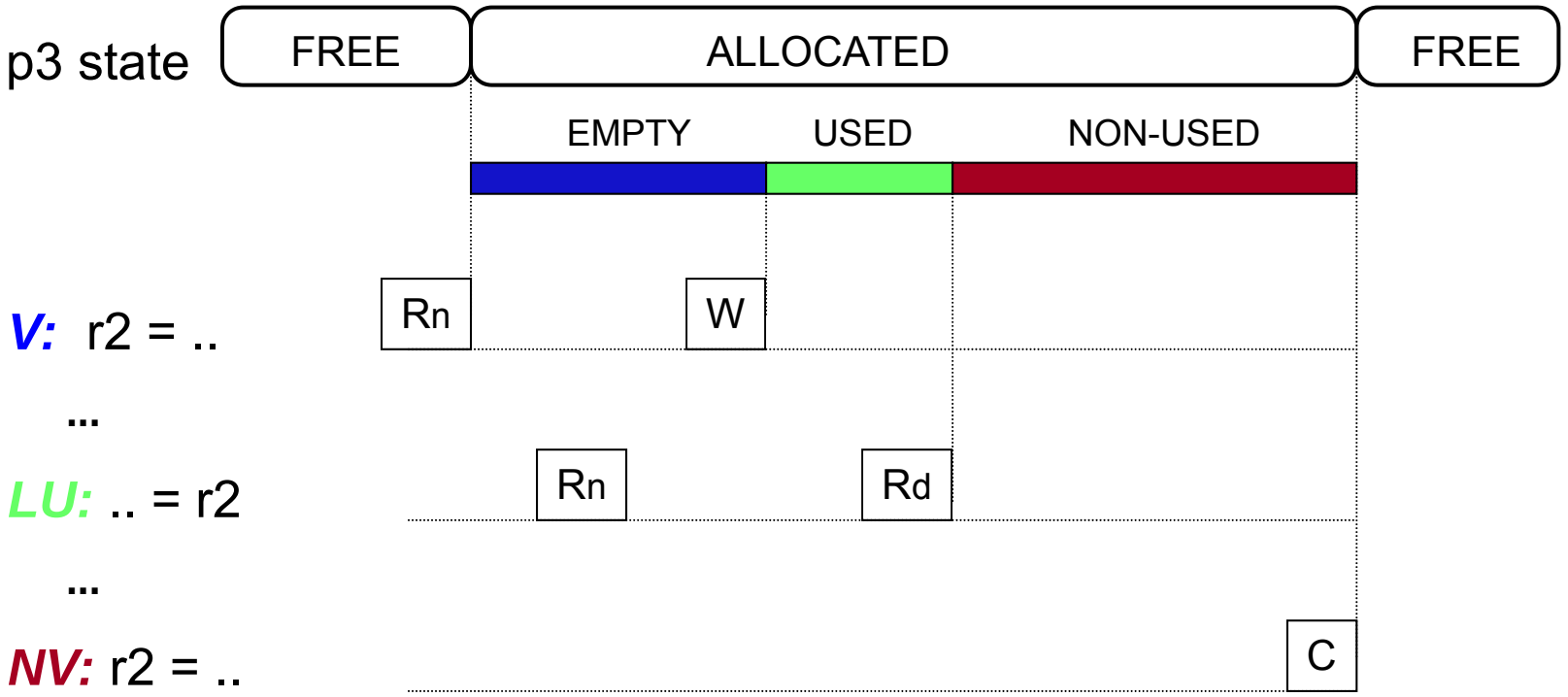
Speculative Early Release: ERPR



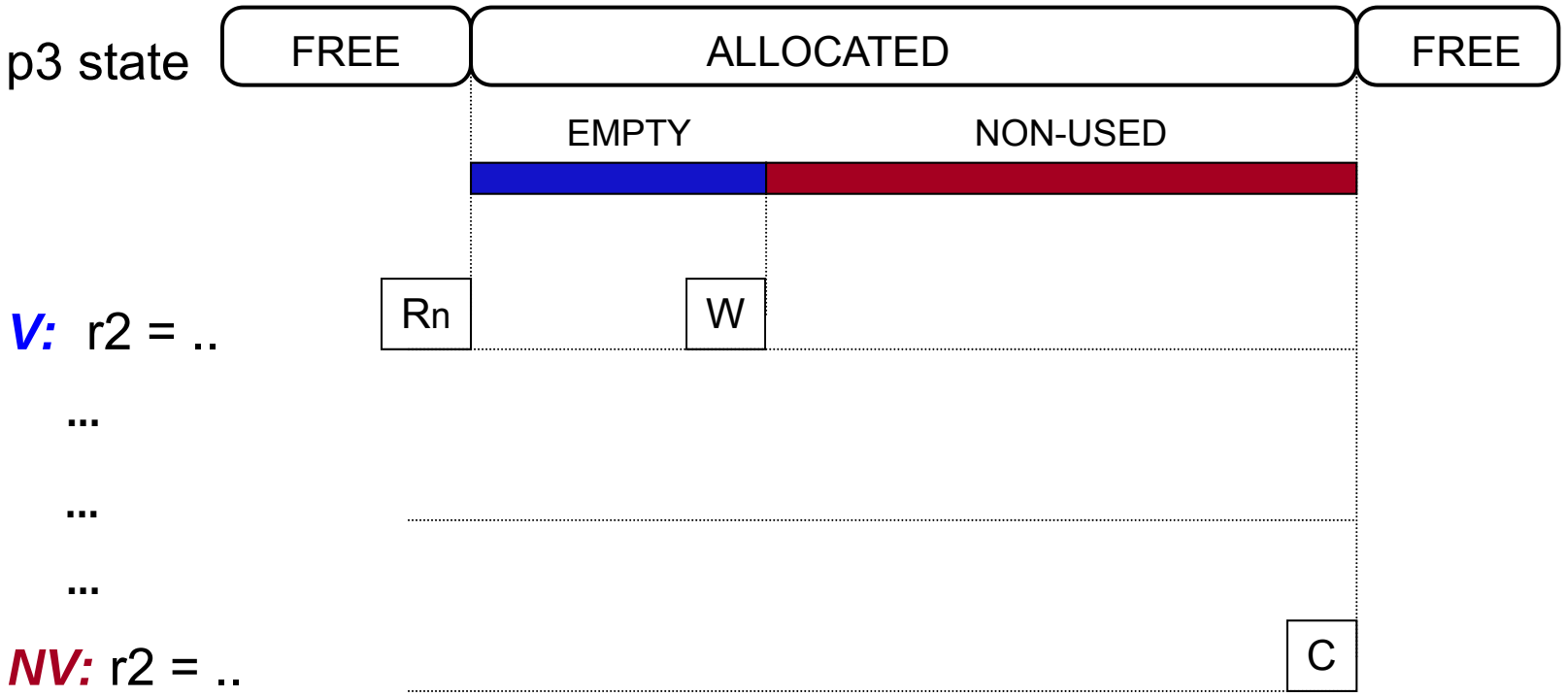
Speculative Early Release: ERPR



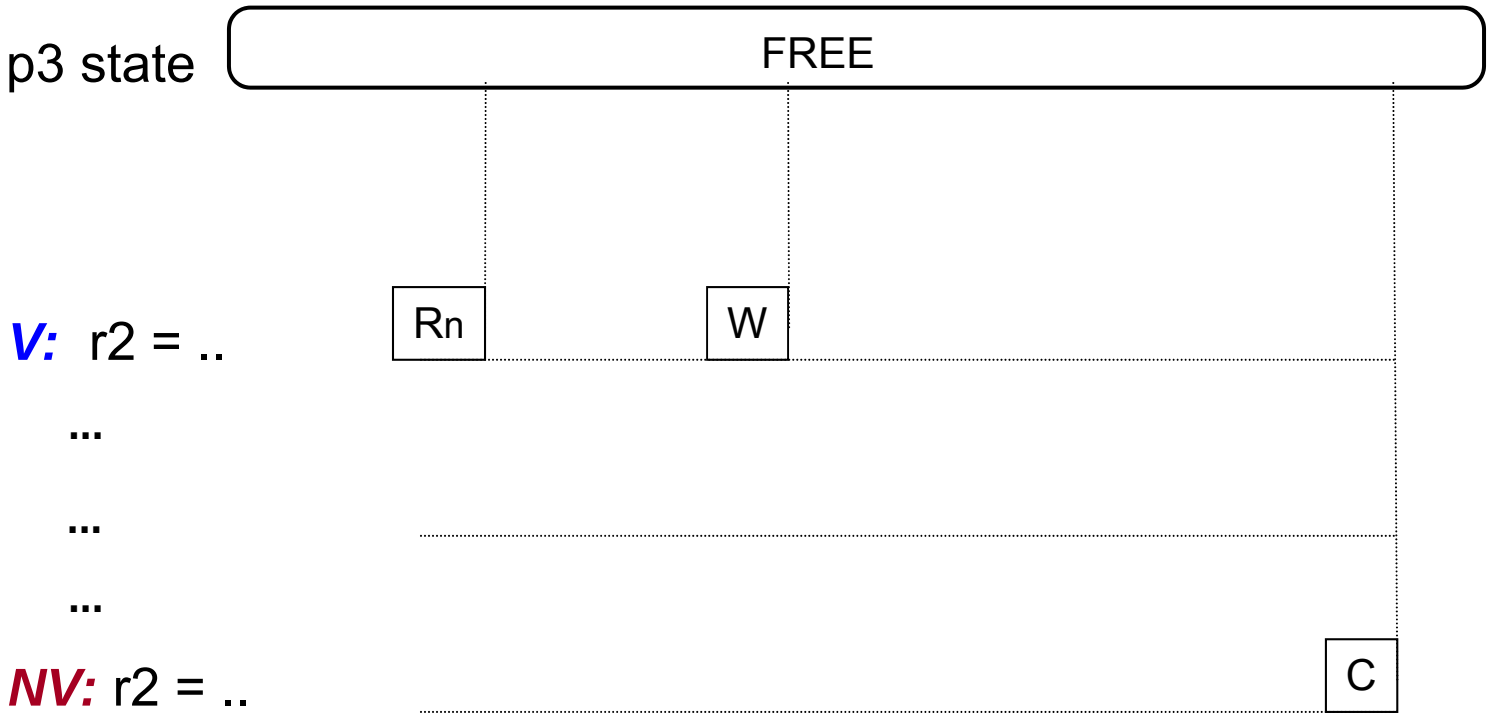
Conventional renaming



Speculative Omission of Allocation: OPRA



Speculative Omission of Allocation: OPRA



Outline

- Renaming mechanisms
- **SR microarchitecture**
  - **ERPR**
  - **OPRA**
  - **Details**
- SR-LUP
- Results
- Conclussions

gaZ

SR: ERPR

- Releasing policy
  - Early released values
    sent to auXiliary Register File

V: r2= ..

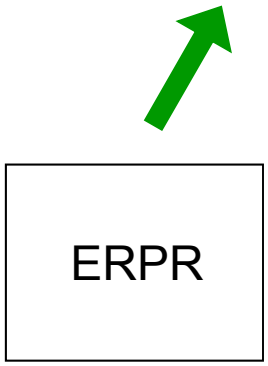
...

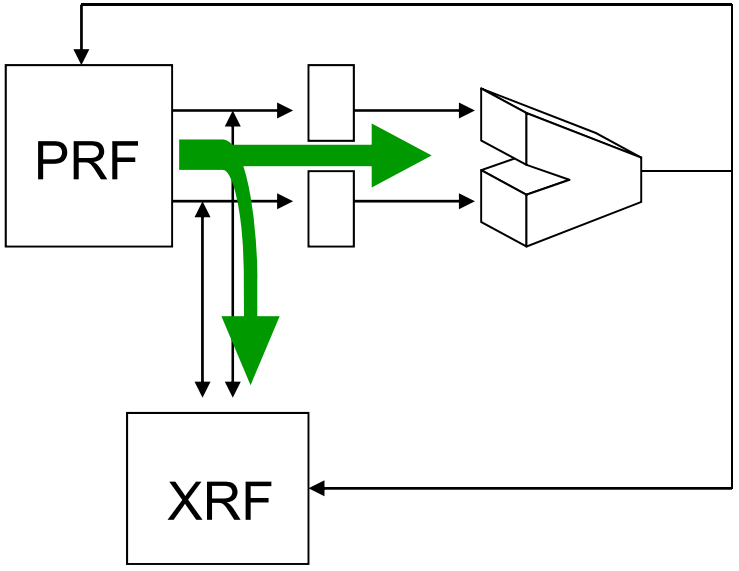
U: .. = r2

...

NV: r2 = ..

U: add r1 <- **r2**, r3





SR: OPRA

- Allocating policy
  - Not allocated values
    sent to auXiliary Register File

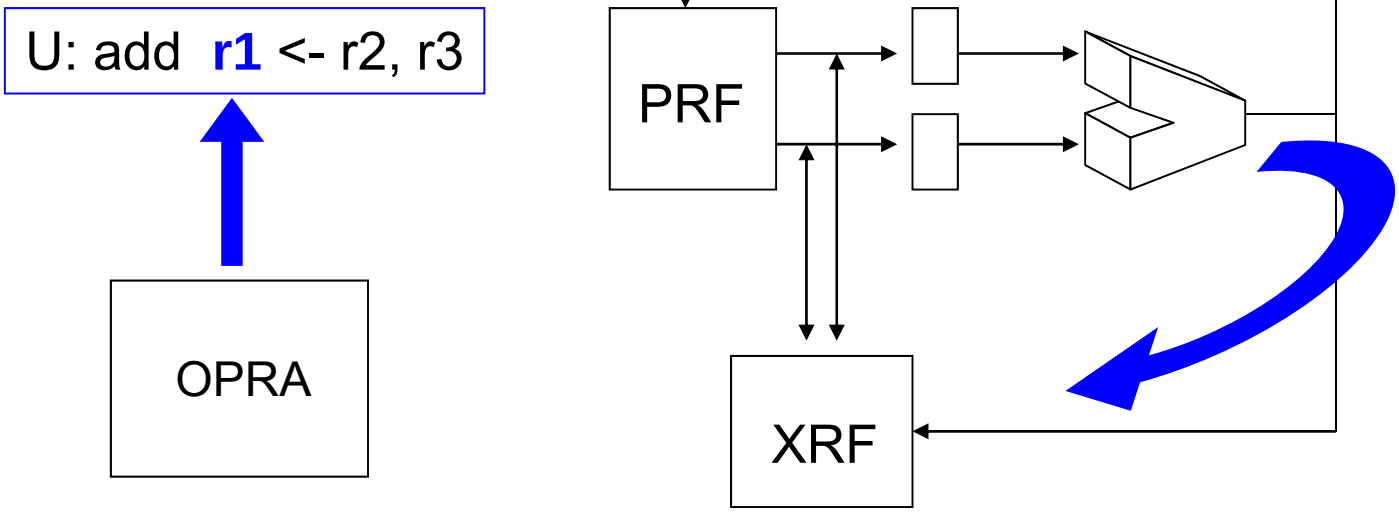
V: r1= ..

...

...

...

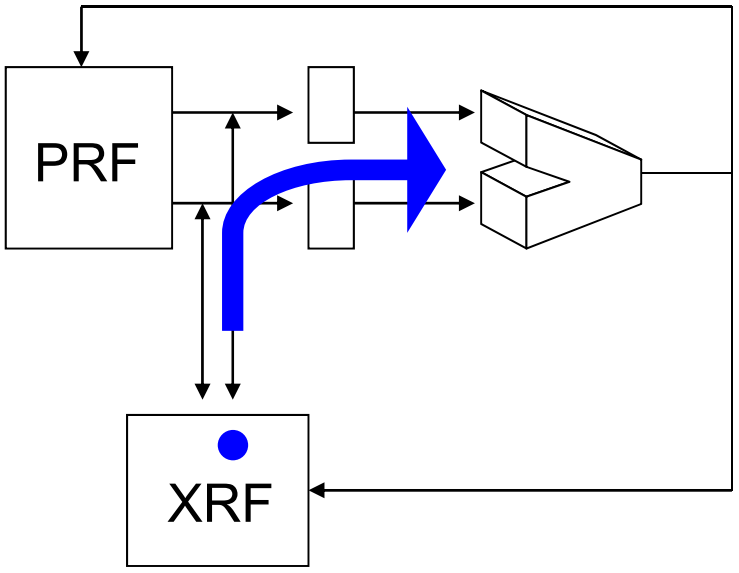
NV: r1 = ..



Microarchitecture details (1/3)

- Unexpected Uses (UU)
  - OPRA: read from XRF

```
V:   r1= .. ; OPRA
     ...
UU:  .. = r1
```



Microarchitecture details (1/3)

- Unexpected Uses (UU)
  - OPRA: read from XRF
  - ERPR: read from PRF or XRF

V: r1= .. ; OPRA

...

UU: .. = r1

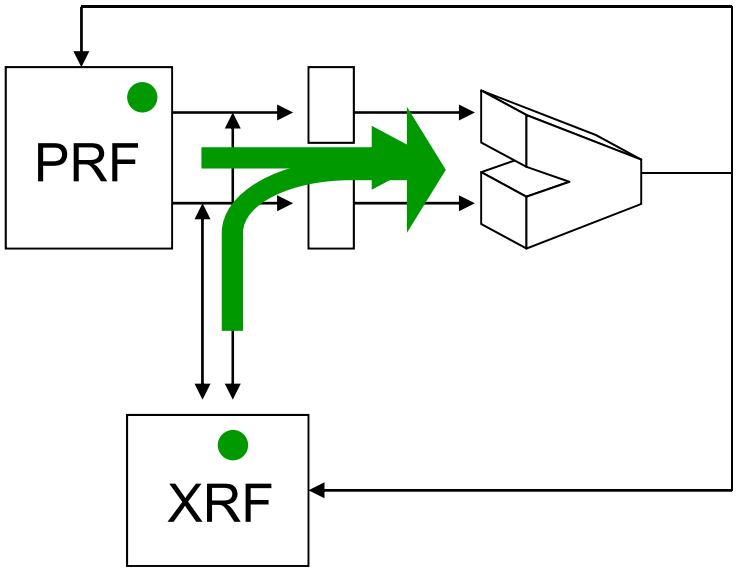
V: r1 = ..

...

U: = r1 ; ERPR

...

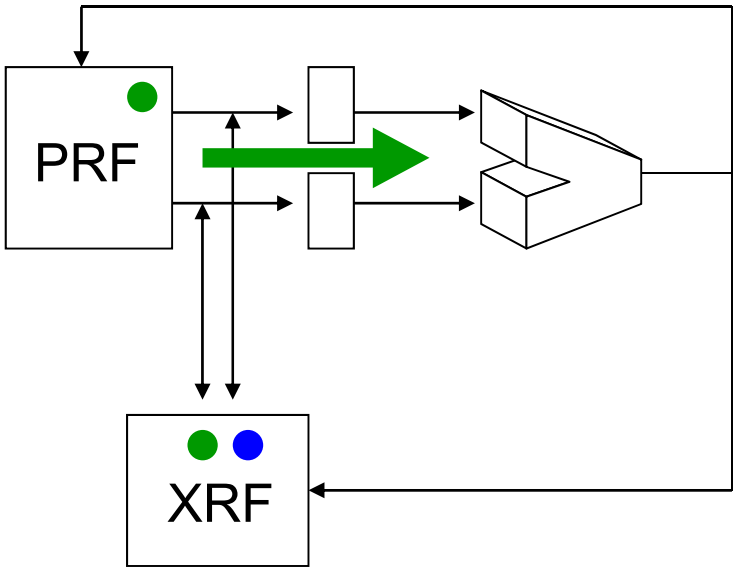
UU: .. = r1



Microarchitecture details (2/3)

- Speculative Issue
  - Operands predicted to be in PRF
- Misspeculations
  - Simple Chained Recovery Mechanism

[1]Torres et al. “Counteracting bank misprediction in sliced first-leves caches”, EuroPAR 2003.



Microarchitecture details (3/3)

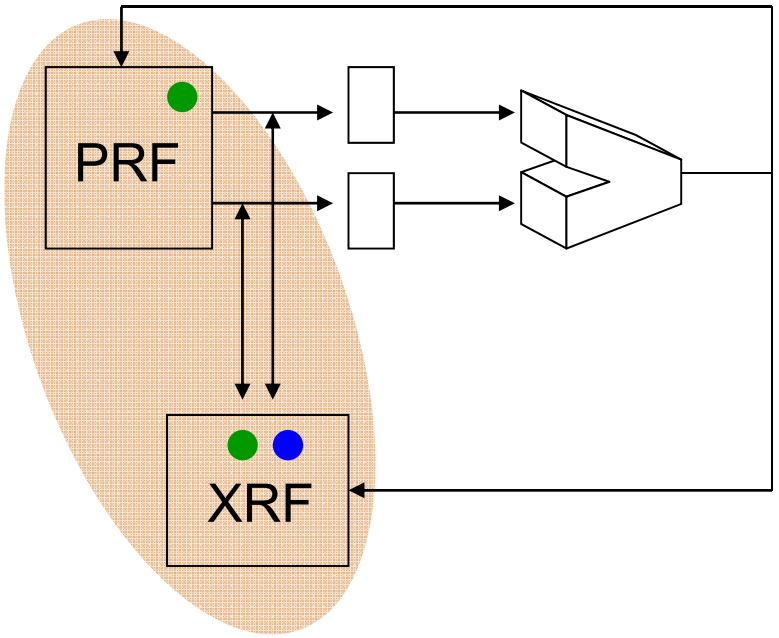
- Dependence Tracking
  - From Physical to Virtual Registers[1]

[1]Monreal et al. “Delaying physical register allocation …”, MICRO 99

V: r1= .. ; OPRA

...

U: .. = r1



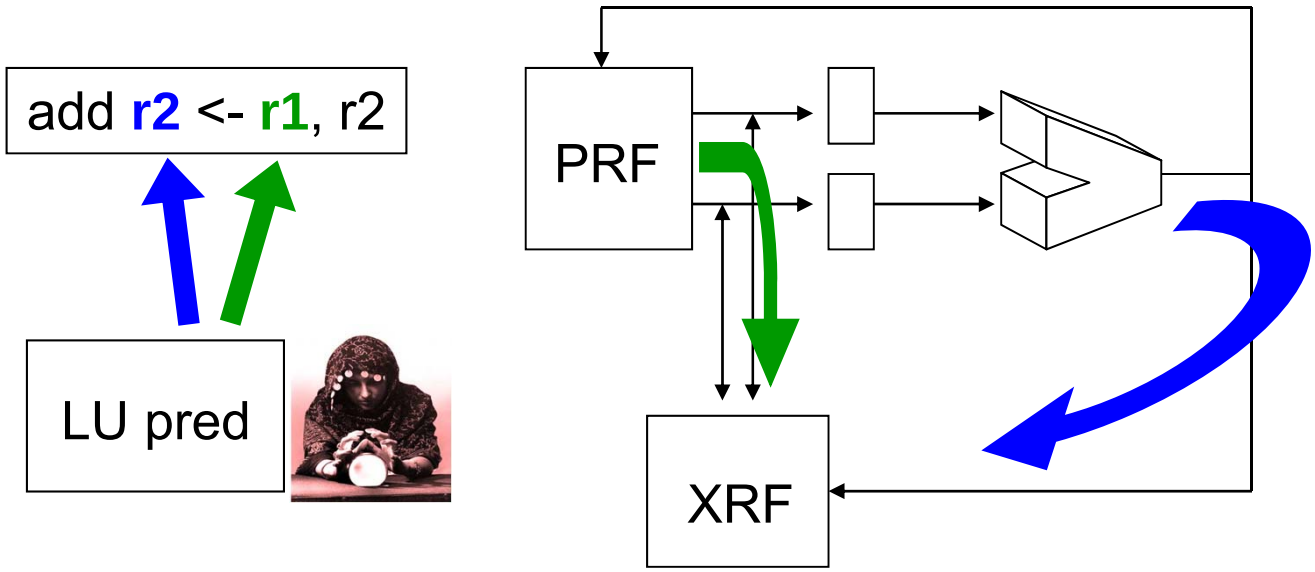
Outline

- Renaming mechanisms
- SR microarchitecture
- **SR-LUP**
- Results
- Conclussions

gaZ

SR-LUP

- Speculative Renaming based on Last Use Prediction



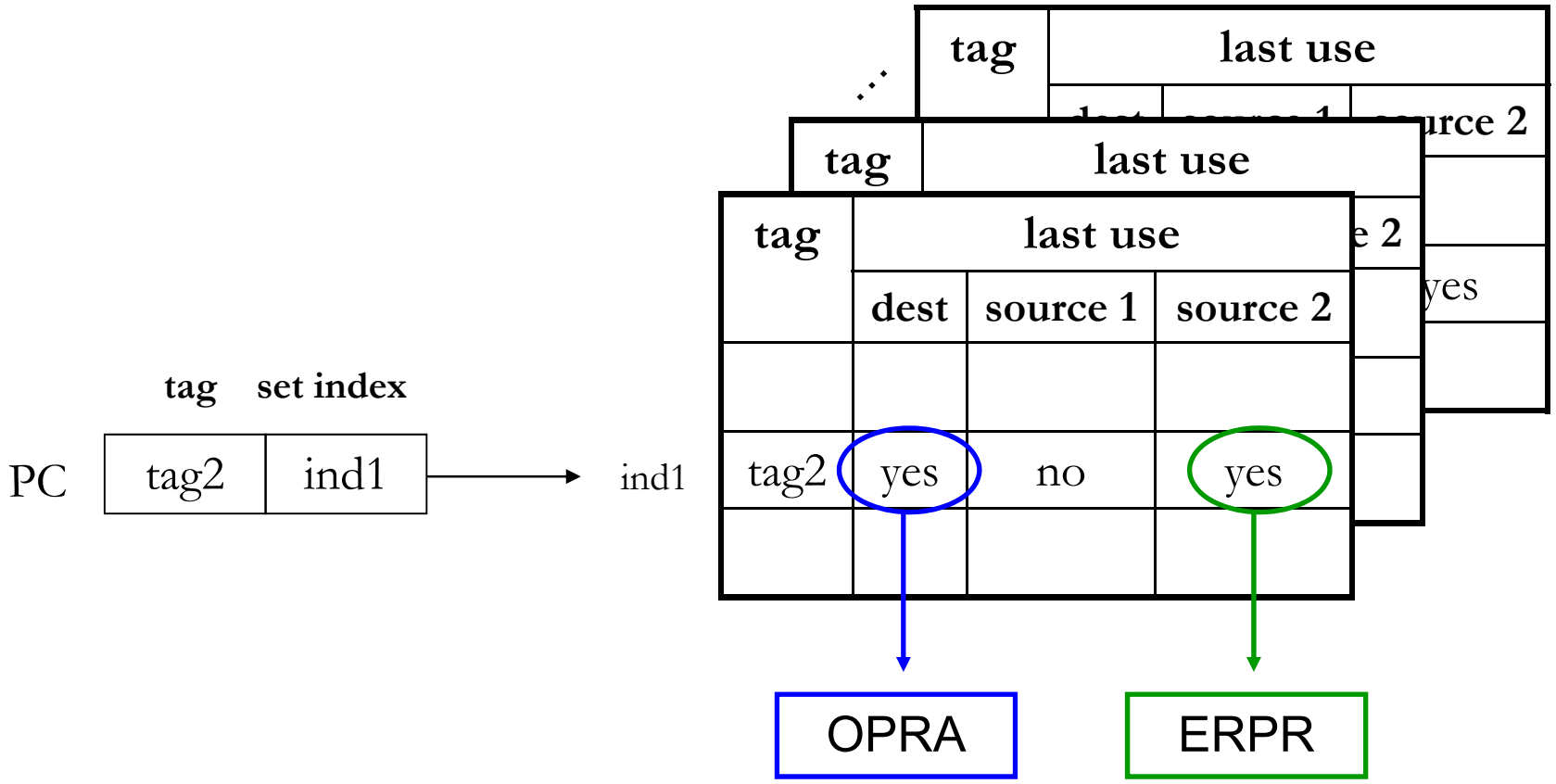
2 designs
- Sticky[1] Last-Use Predictor (SLUP)
- Degree of Use[2] Last-Use Predictor (DULUP)

[1]J. Alastruey et al., "Speculative Early Register Release", ICCF 06.

[2]J.A. Butts and Sohi, "Characterizing and Predicting Value Degree of Use", MICRO 02.

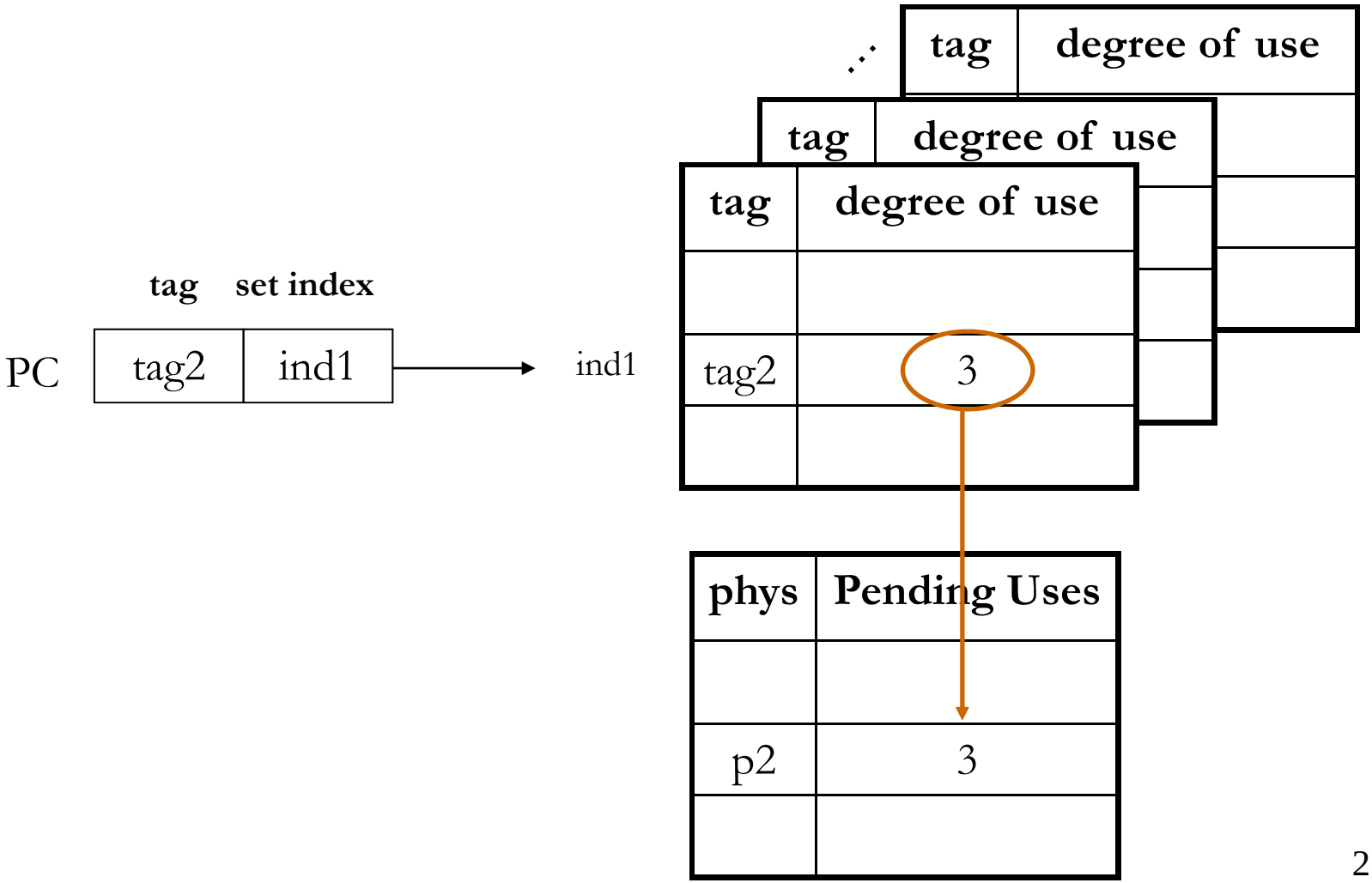
Sticky Last-Use Predictor (SLUP)

- Records "is a LU register" information of commited instructions



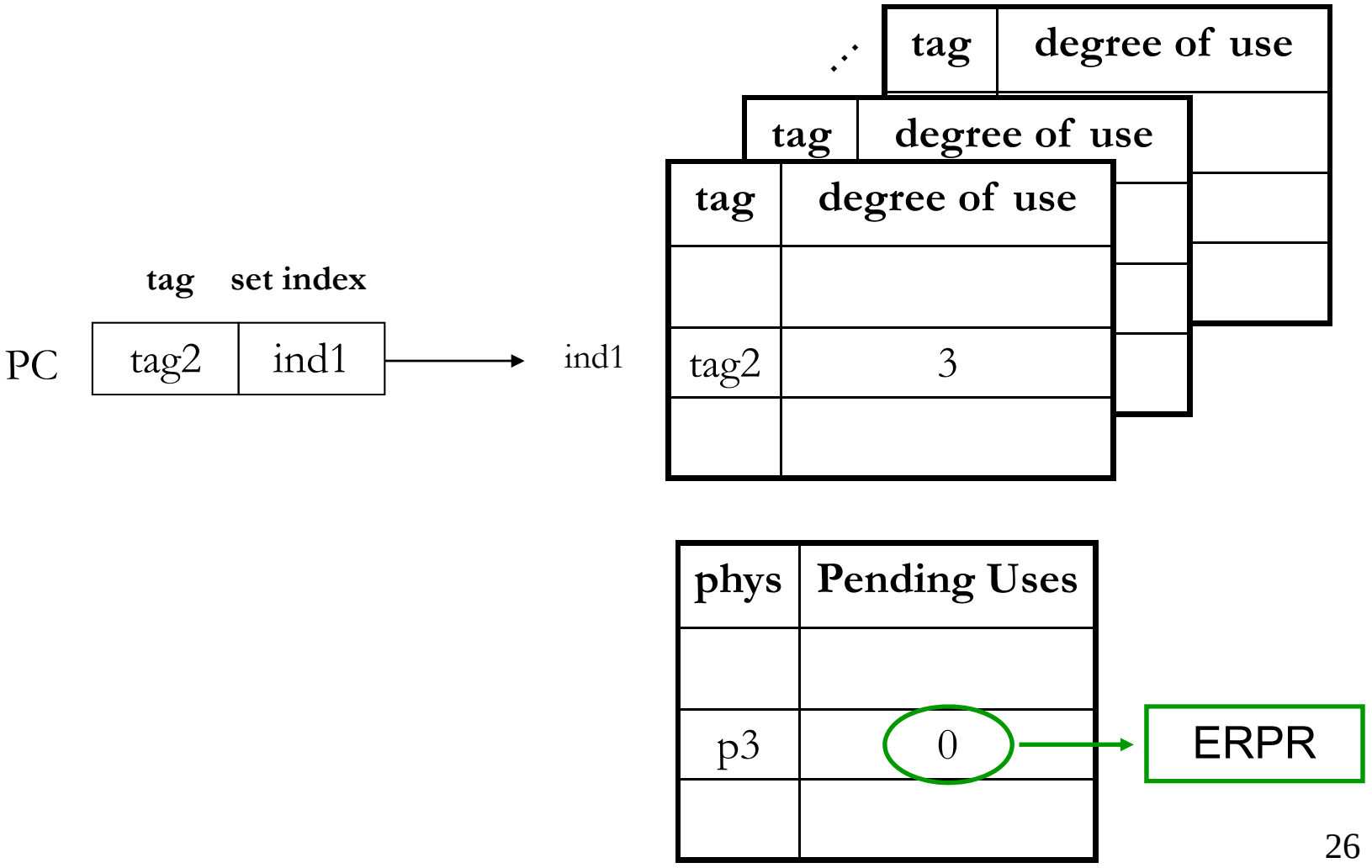
Degree-of-Use Last-Use Predictor (DULUP)

- Records degree-of-use information of commited instructions



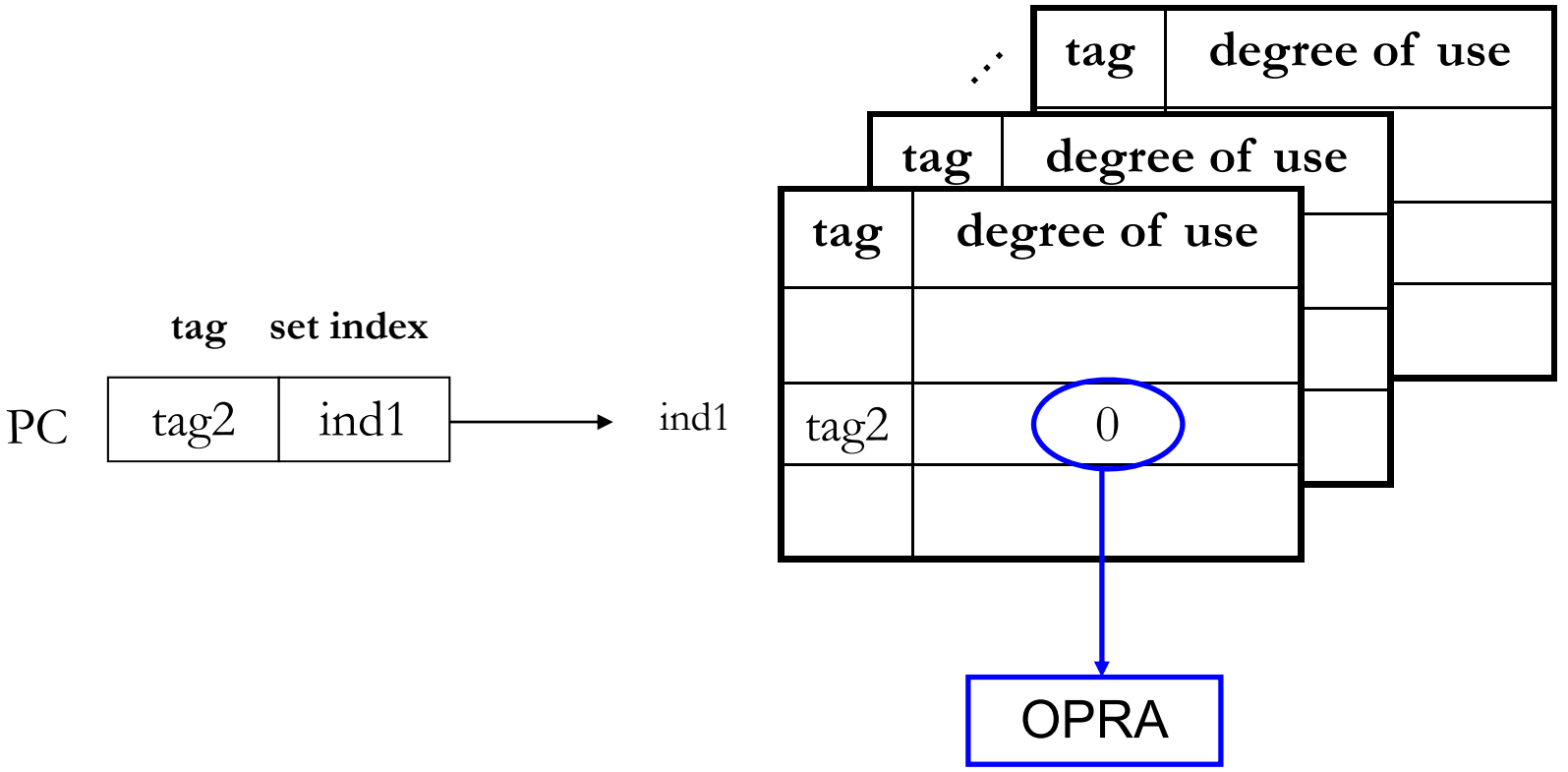
Degree-of-Use Last-Use Predictor (DULUP)

- Records degree-of-use information of commited instructions



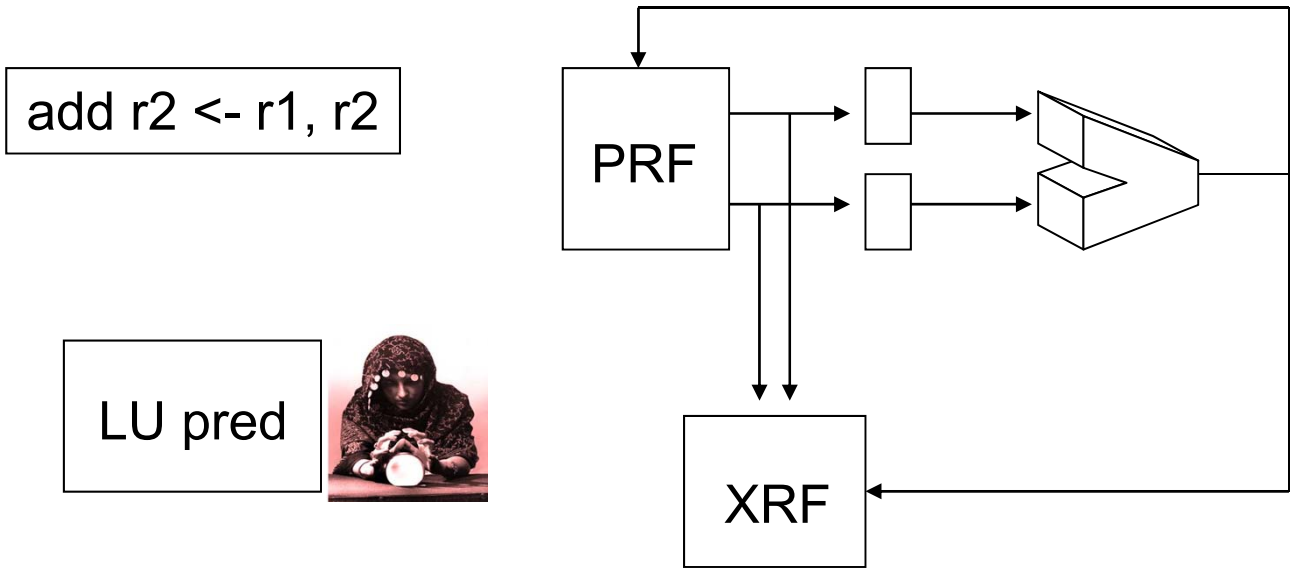
Degree-of-Use Last-Use Predictor (DULUP)

- Records degree-of-use information of commited instructions



Last-Use Predictor

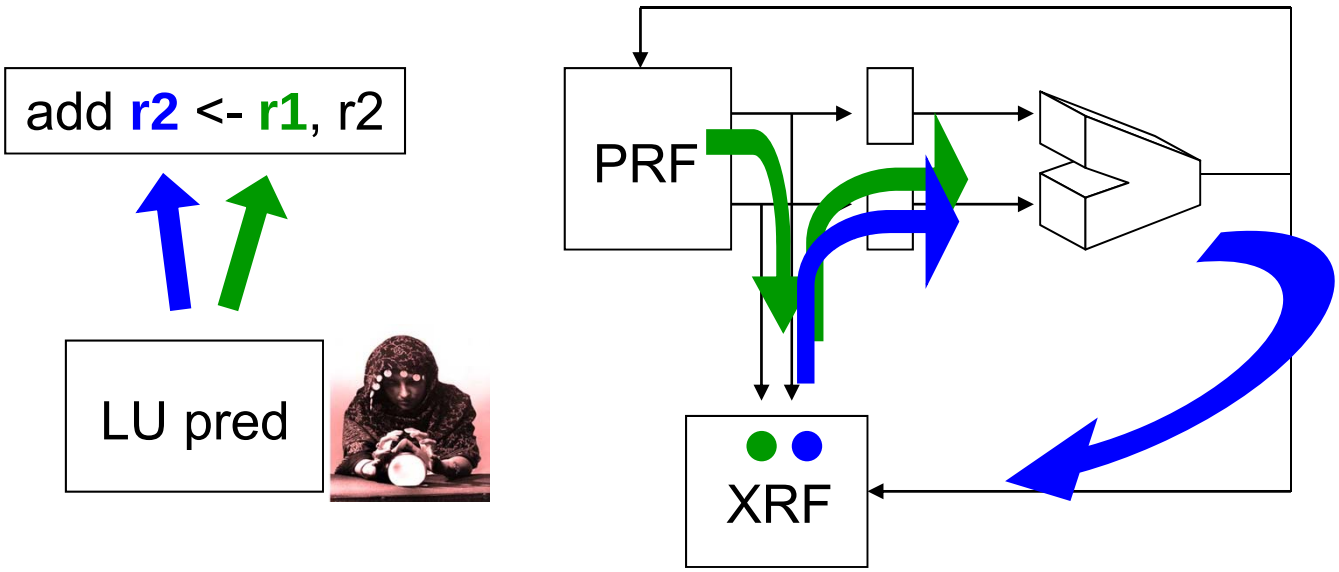
- **Last-use predictor** mispredictions
  - Wrong "LU register" predictions
  - Wrong "not LU register" predictions



gaZ

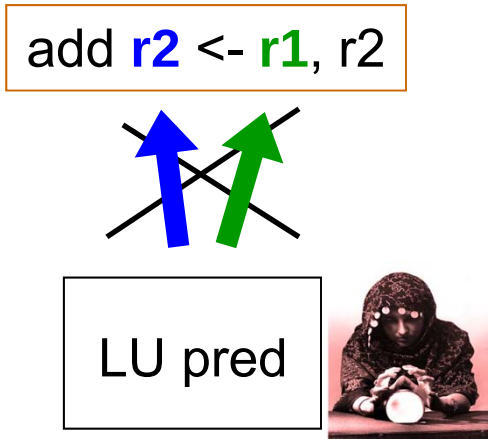
Last-Use Predictor

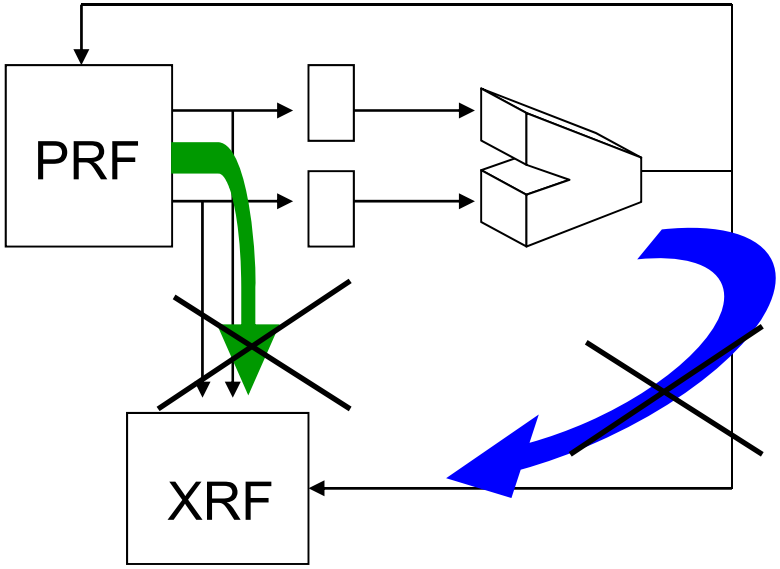
- **Last-use predictor** mispredictions
  - Wrong "LU register" predictions
  - Wrong "not LU register" predictions



Last-Use Predictor

- **Last-use predictor** mispredictions
  - Wrong "LU register" predictions
  - Wrong "not LU register" predictions





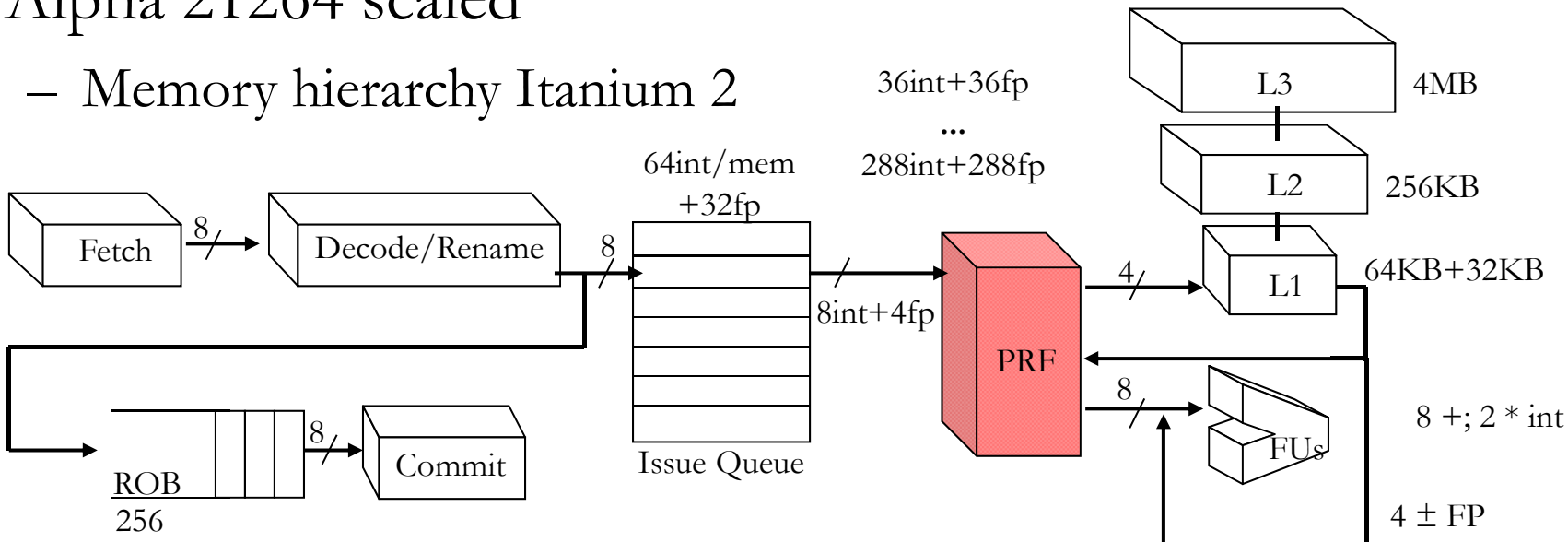
Ouline

- Renaming mechanisms
- SR microarchitecture
- SR-LUP
- **Results**
- Conclussions

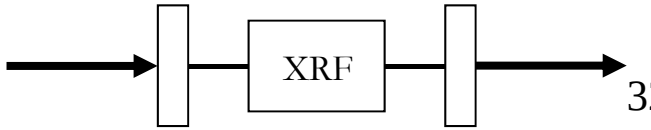
gaZ

Simulated processor

- Alpha 21264 scaled
  - Memory hierarchy Itanium 2



- SPEC2000
  - Simpoints
- simplescalar v3.0c



Sensitivity to XRF design

- Performance vs. complexity
  - Entries
  - Ports

$PRF_{size} \Rightarrow T_{XRF} <= T_{PRF}$

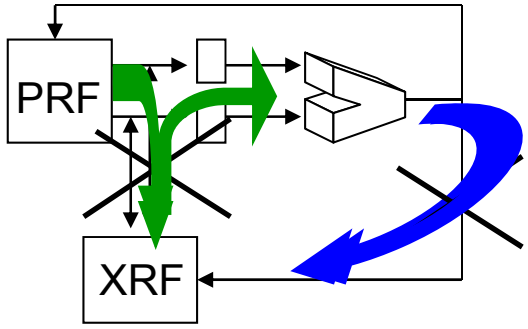
- Lack of XRF entries
  - Cancelation of LU predictions

Minimum 160

- Lack of XRF ports
  - Read: UU contention
  - Write: LUpred contention

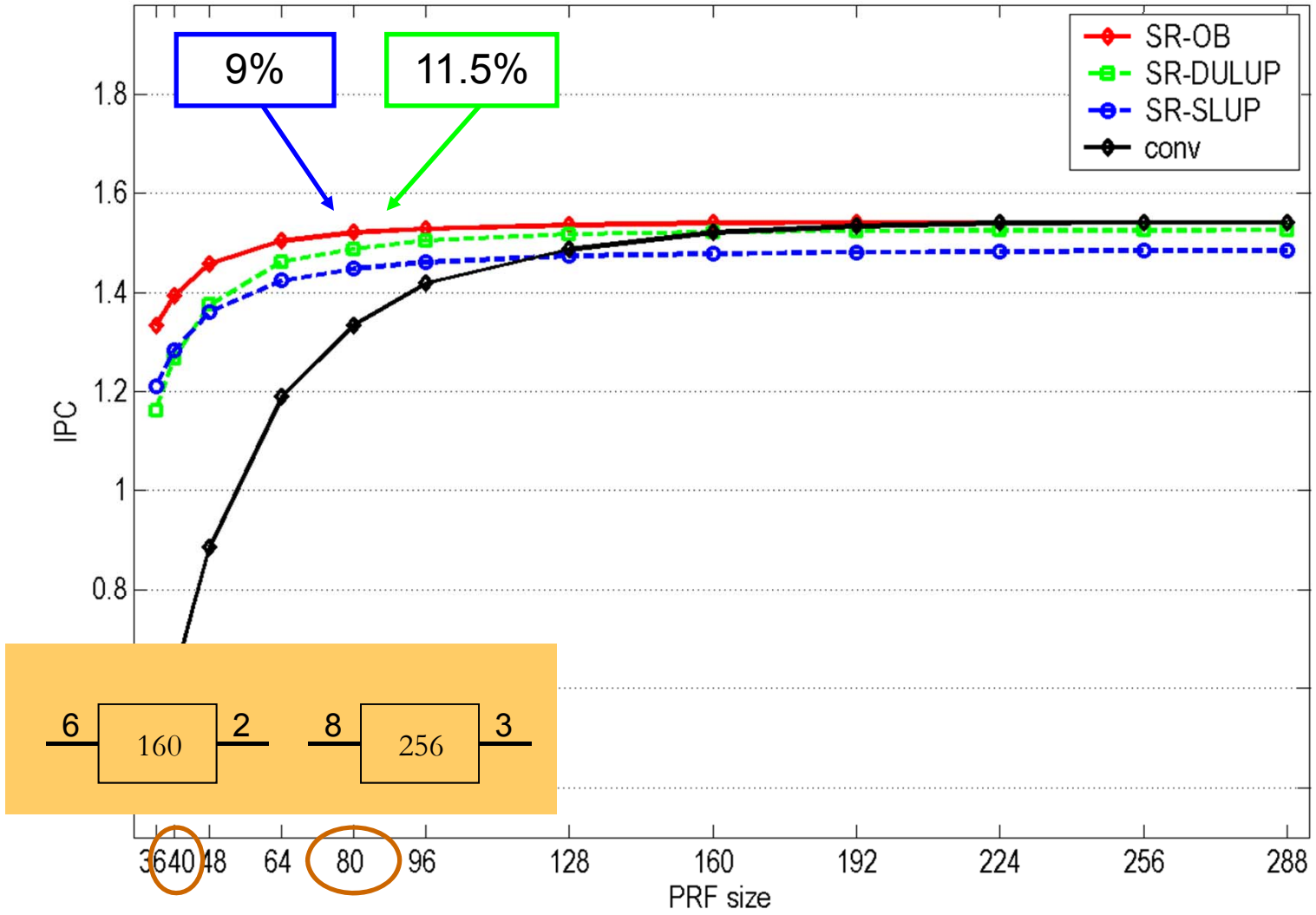
2-3 ports

5-8 ports



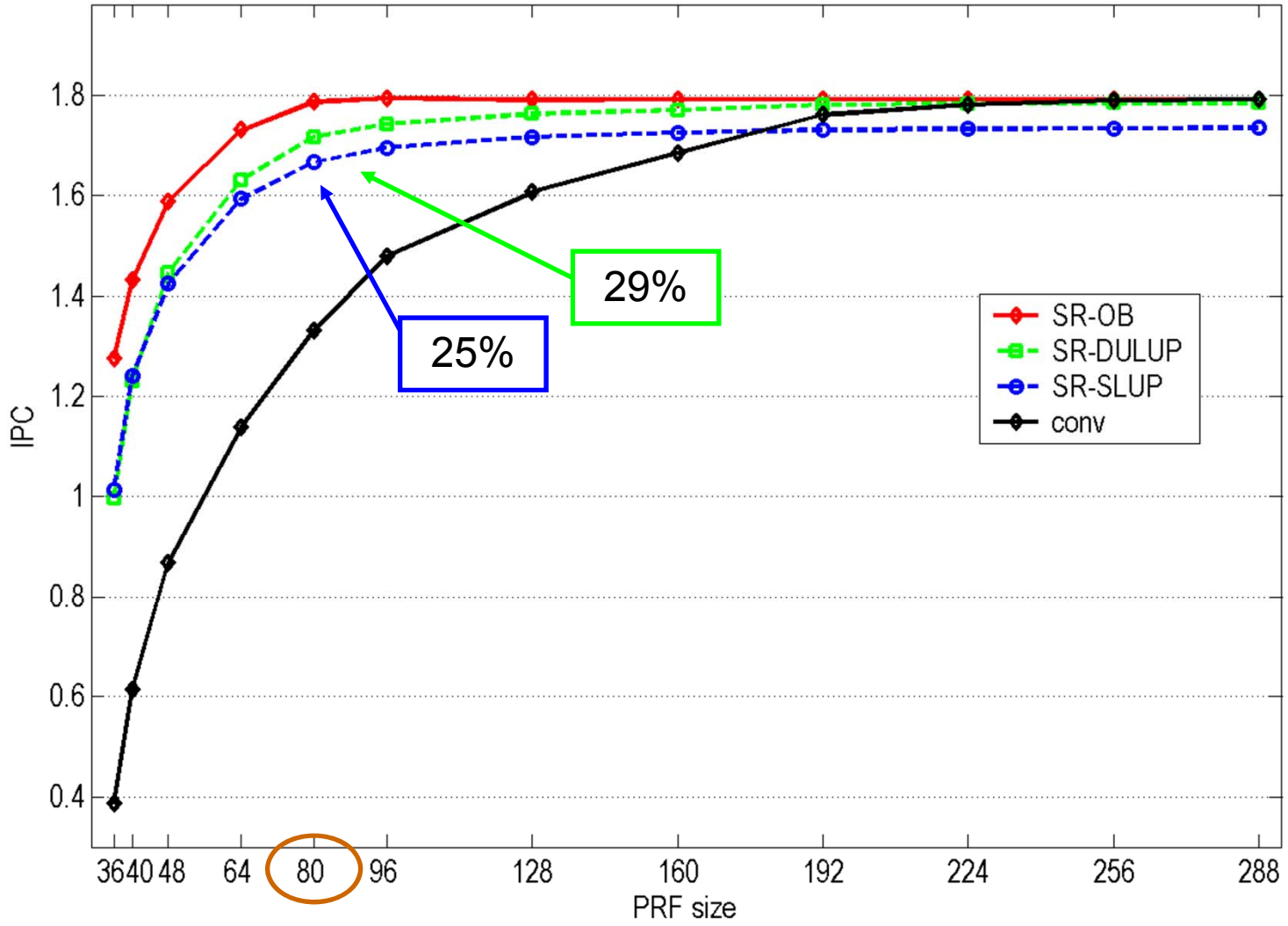
SR-LUP performance

- IPC vs. number of physical registers (specINT2000)



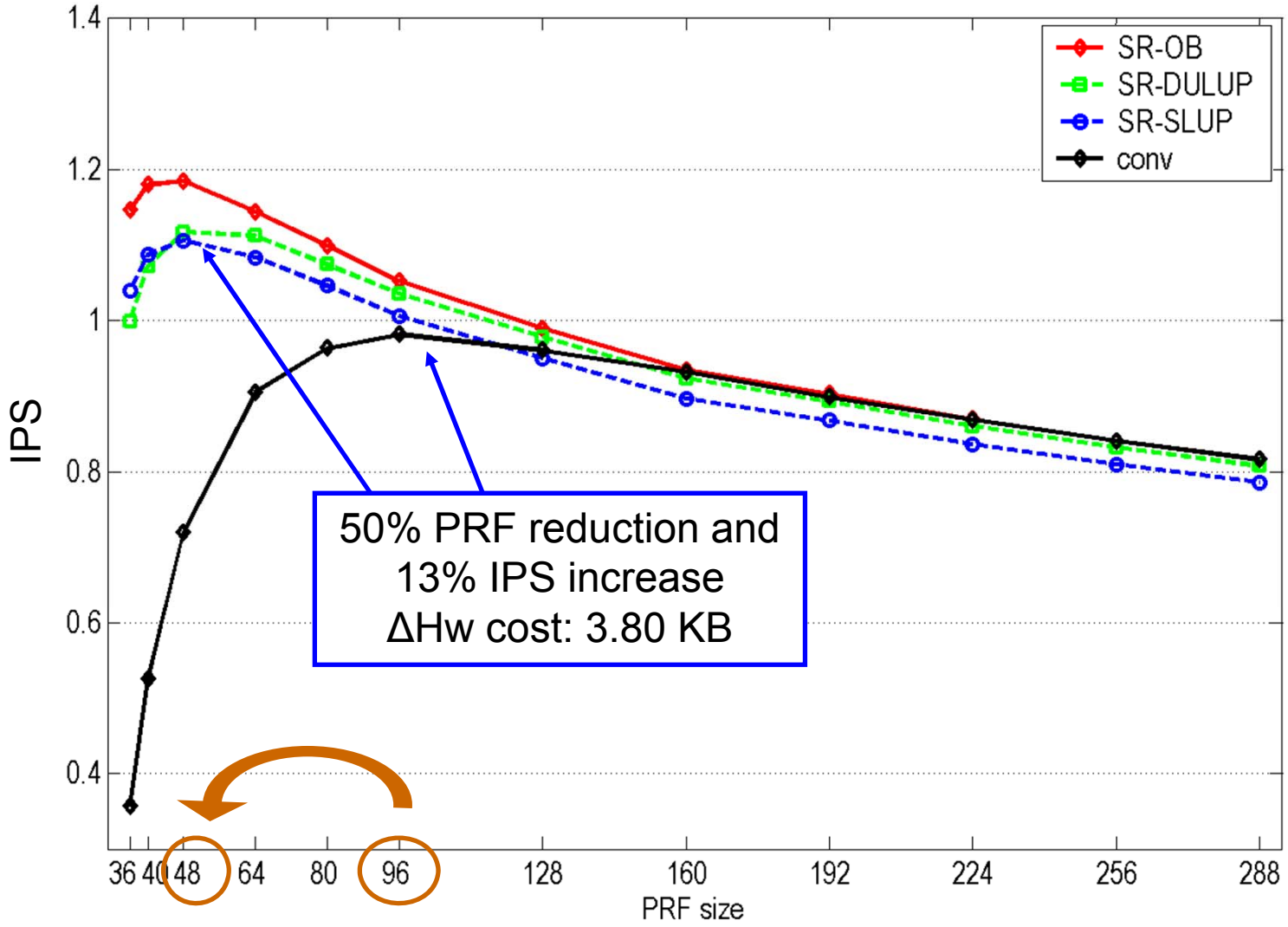
SR-LUP performance

- IPC vs. number of physical registers (specFP2000)



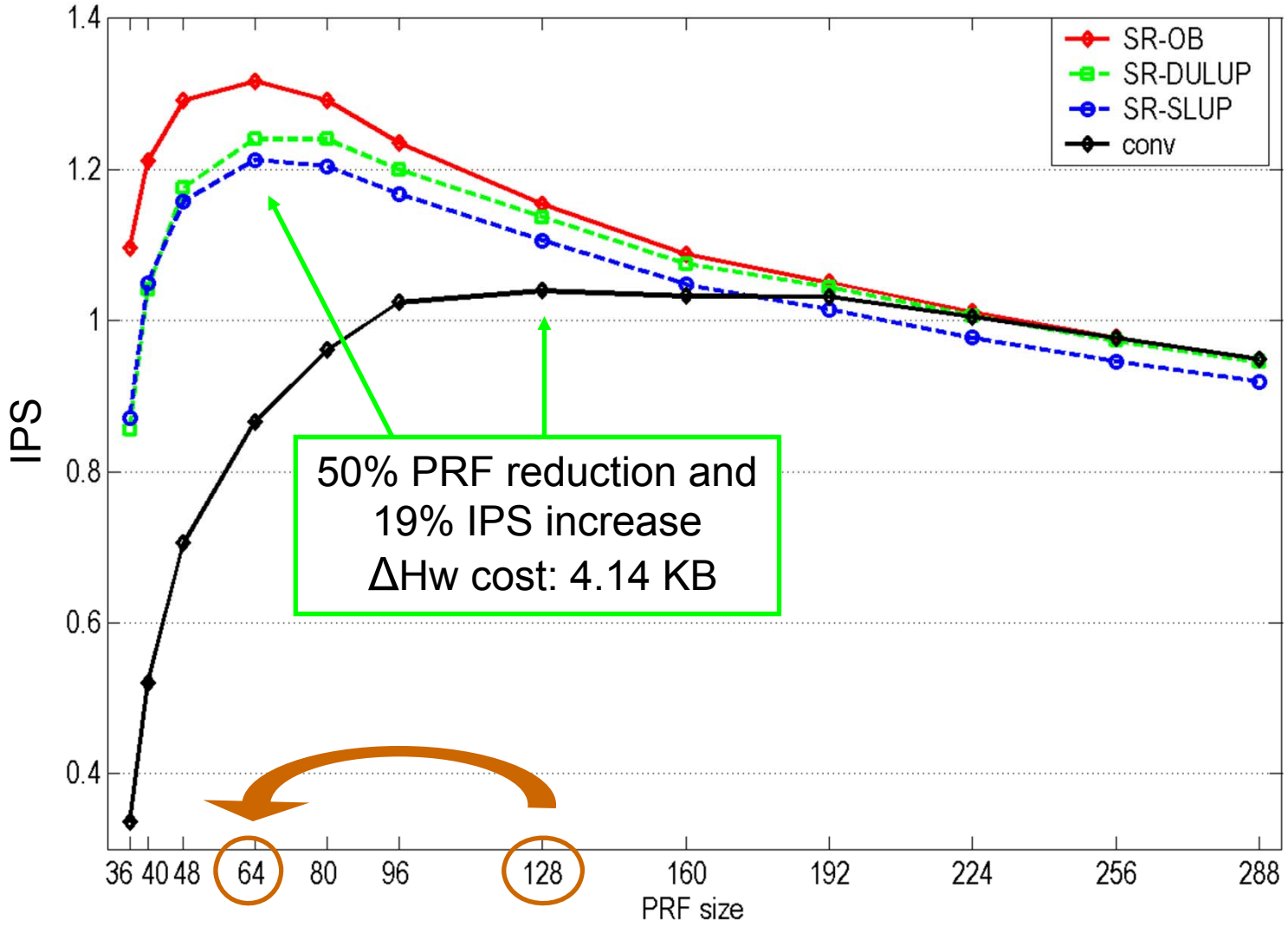
SR-LUP performance

- IPS vs. number of physical registers (specINT2000)



SR-LUP performance

- IPS vs. number of physical registers (specFP2000)



Outline

- Renaming mechanisms
- SR microarchitecture
- SR-LUP
- Results
- **Conclussions**

gaZ

Conclussions

- Microarchitecture supporting SR
  - Arbitrary speculative OPRA and ERPR policies
  - Recovery mechanism based on Auxiliary Register File
- Evaluation of microarchitecture
  - LU prediction for OPRA and ERPR
  - Simple LU predictor + Realistic XRF design

gaZ

Conclussions

- Microarchitecture supporting SR
  - Arbitrary speculative OPRA and ERPR policies
  - Recovery mechanism based on Auxiliary Register File
- Evaluation of microarchitecture
  - LU prediction for OPRA and ERPR
  - Simple LU predictor + Realistic XRF design
- Performance improvement (IPC)
  - int: 11.5%
  - fp: 29%

  80 PRF + 256 XRF
- Significant PRF size reduction
  - 50% PRF reduction with 13%/19% IPS increase (int/fp)
  - 48int reach 90% of 224int IPC (conv)

gaZ

Future work

- Evaluate SR microarchitecture with other OPRA and ERPR policies
- Improve last-use predictor
- Apply to MT architectures
  - PRF critical

gaZ

Microarchitectural Support for Speculative Register Renaming

J. Alastruey[+], T. Monreal[+], V. Viñals[+], M. Valero[*]

[+]gaZ – I3A - Universidad de Zaragoza

[*] Universidad Politécnica de Cataluña-Barcelona Supercomputing Center

HiPEAC - High-Performance Embedded Architecture and Compilation

IEEE International Parallel & Distributed Processing Symposium

Long Beach, USA, 2007

gaZ