

A Study of Control Independence

Eric Rotenberg

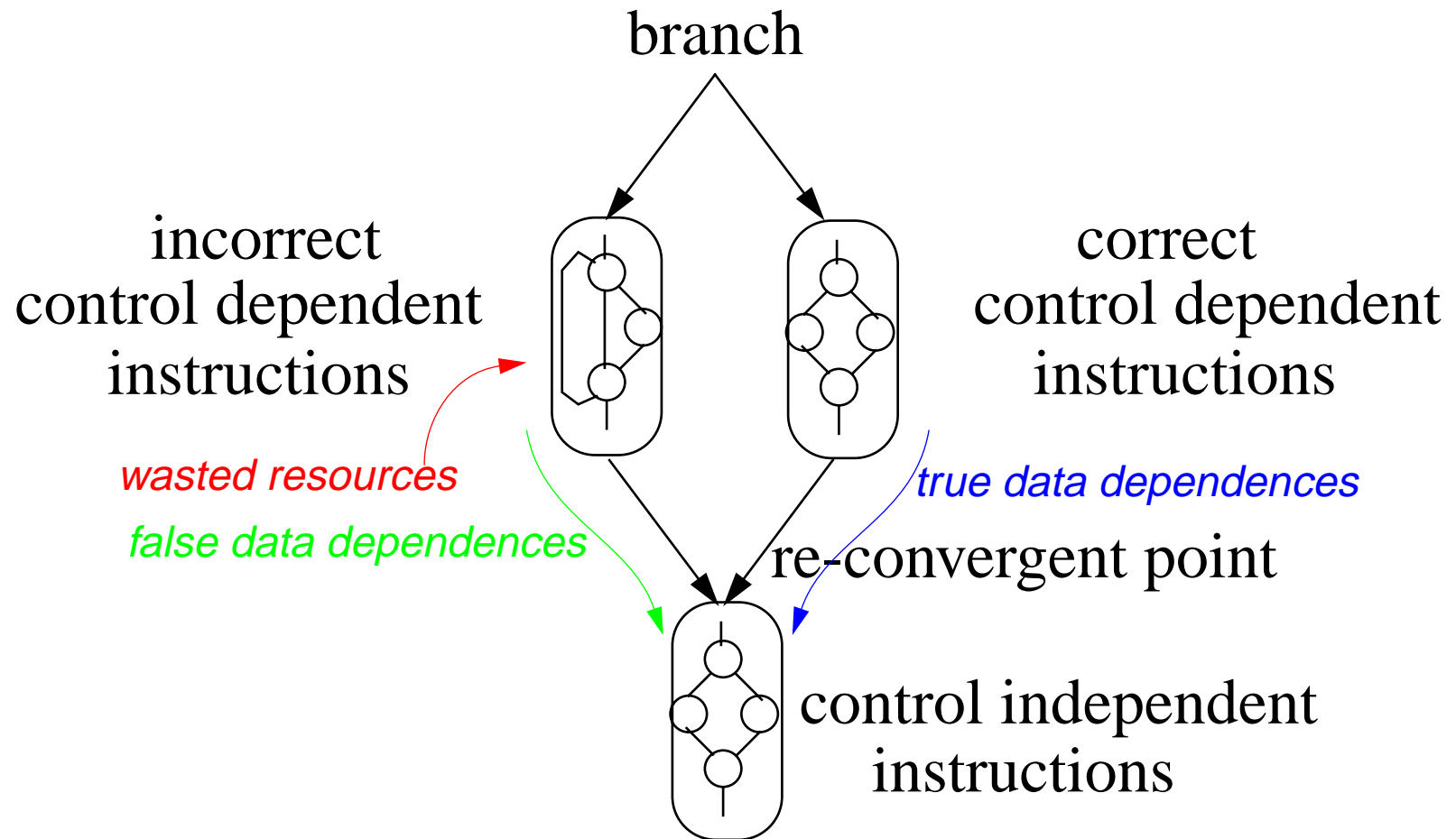
Introduction

- High instruction-level parallelism (ILP)
 - requires large instruction scheduling window
 - branches make this difficult
- Branches introduce control dependences
 - must execute branch to fetch next instructions
 - correct branch prediction/speculation: essentially eliminates control dependences
 - mispredictions are still a major bottleneck: complete squashing

Introduction

- Solutions to branch misprediction bottleneck
 1. better branch prediction
 2. forms of multi-path execution
 3. predication
 4. control independence

Control Independence



Goals

1. Establish new performance bounds of control independence under implementation constraints
 - window size
 - fetch/issue width
 2. Identify and quantify primary limiting factors
 3. Assess basic implementation requirements and complexity
- Goals 1 and 2 are the focus of this talk
 - Inspired by Lam & Wilson limit study
 - Want to refine understanding of control independence
 - Can aid control independence design

Notation

- **WR**

- Model **W**asted **R**esources due to incorrect control dependent instructions
- Resources => window space and pipeline bandwidth
- Conversely: **nWR** => **n**o **w**asted **r**esources

- **FD**

- Model **F**alse **D**ependences created by incorrect control dependent instructions
- Conversely: **nFD** => **n**o **f**alse **d**ependences

Models

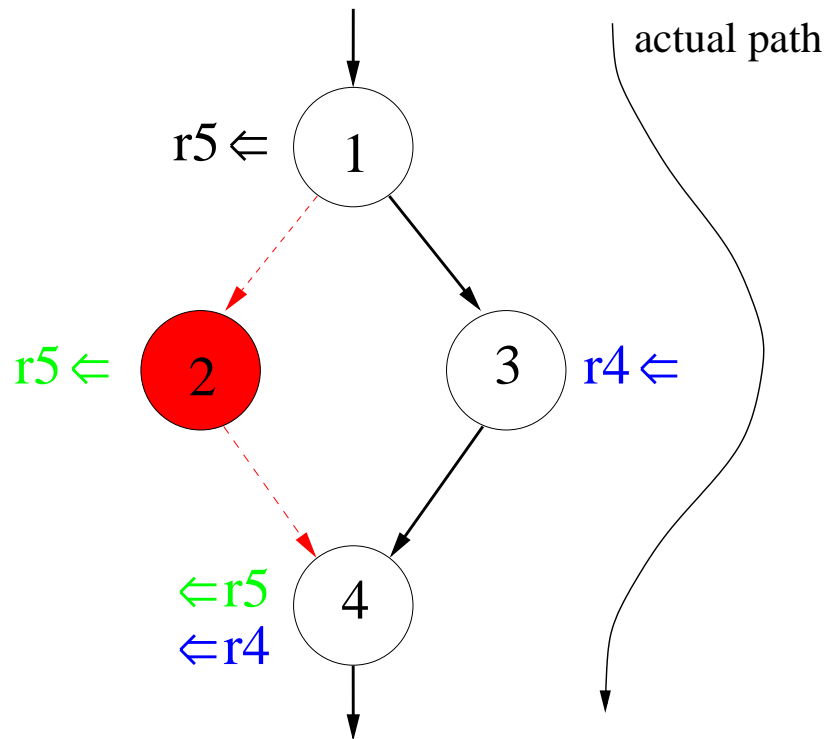
Endpoints

1. **oracle**: oracle branch prediction
2. **base**: conventional prediction/speculation

Control independence models

3. **nWR-nFD**: no wasted resources, no false dependences
4. **nWR-FD** : no wasted resources, false dependences
5. **WR-nFD** : wasted resources, no false dependences
6. **WR-FD** : wasted resources, false dependences

oracle

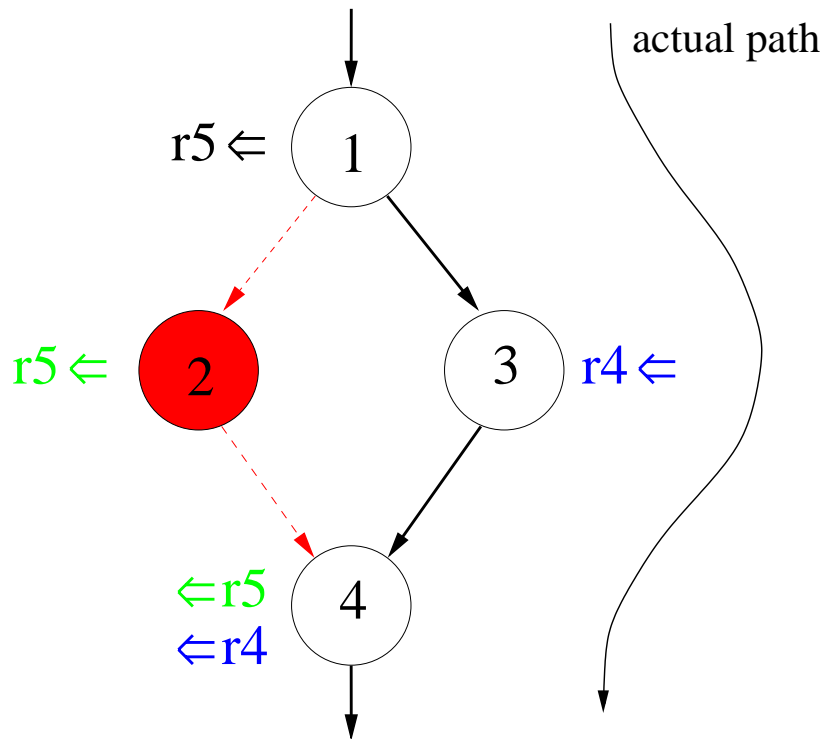


TIME

FETCH ISSUE

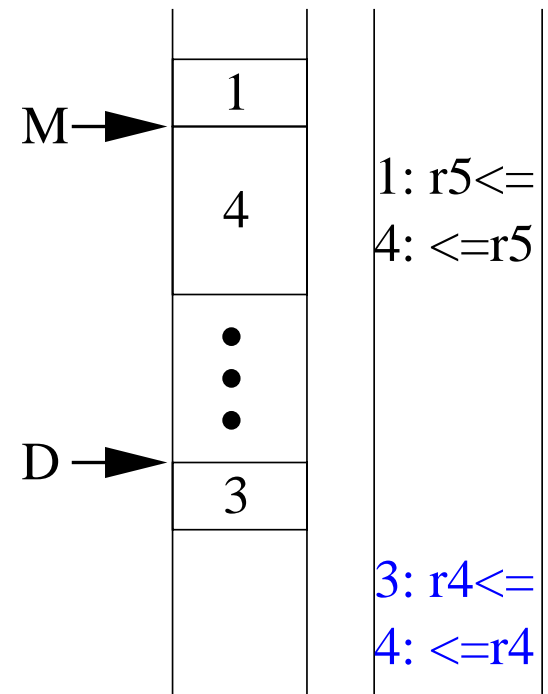
1	
3	1: $r5 \leftarrow$
4	3: $r4 \leftarrow$
	4: $\leftarrow r5$
	4: $\leftarrow r4$
•	
•	
•	

nWR-nFD

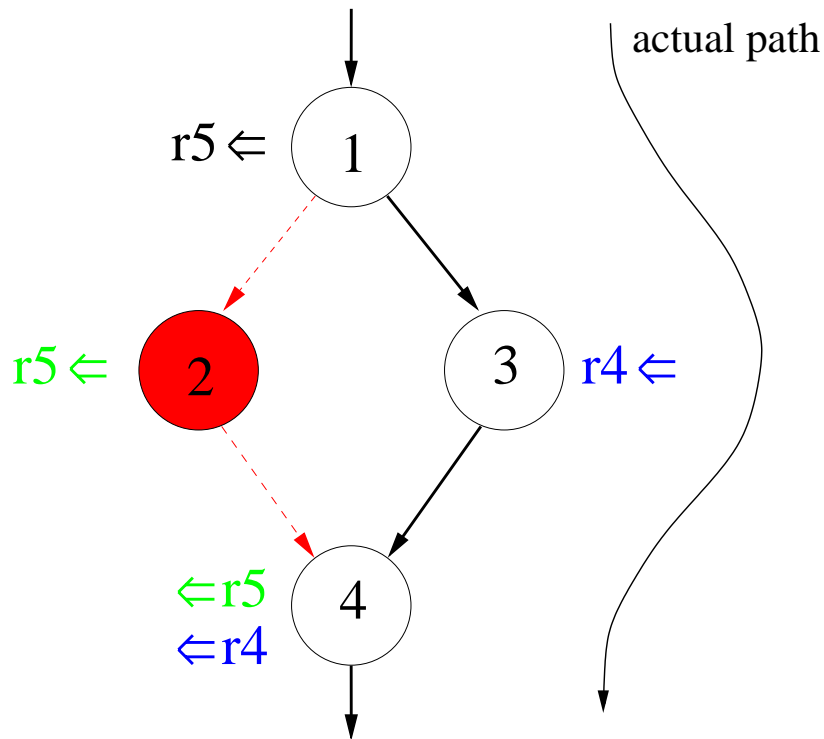


TIME

FETCH ISSUE

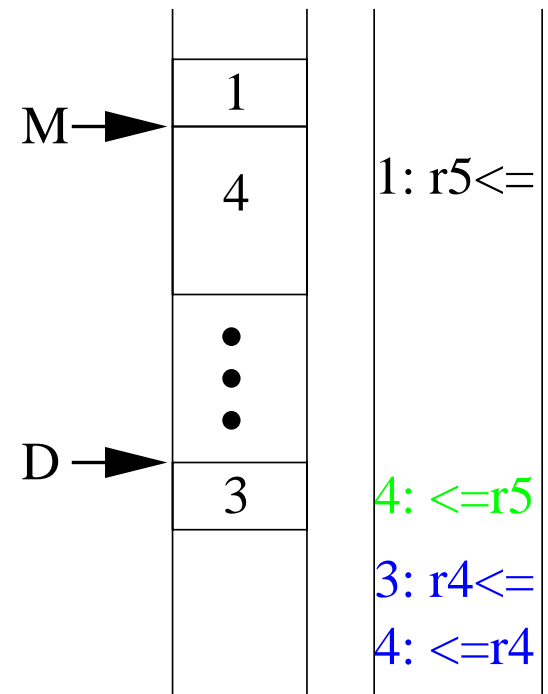


nWR-FD

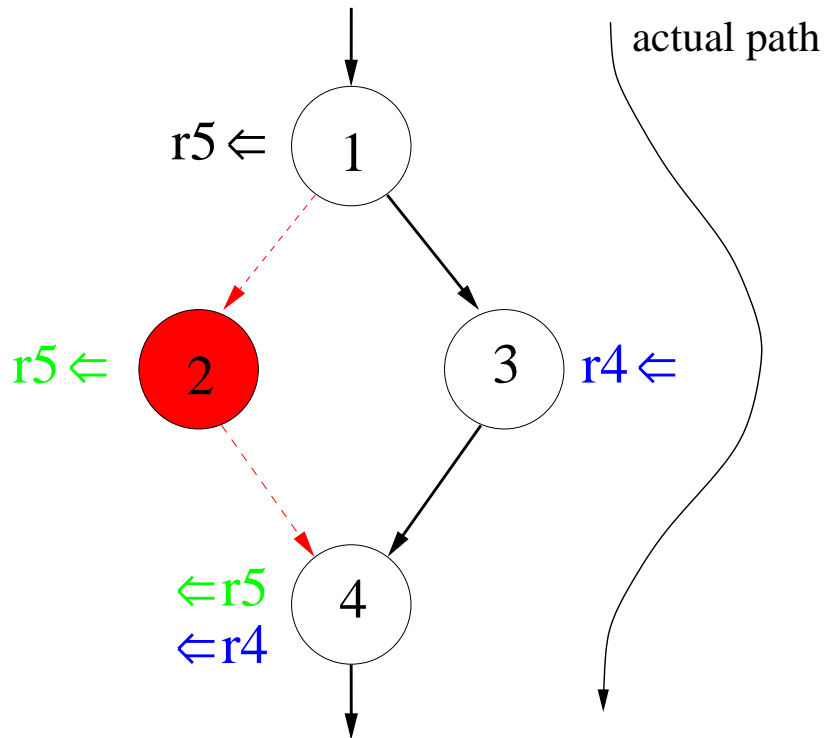


TIME

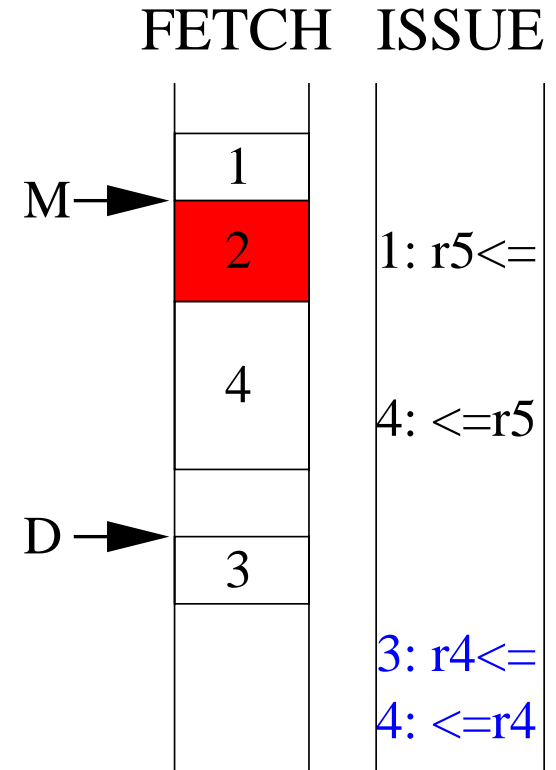
FETCH ISSUE



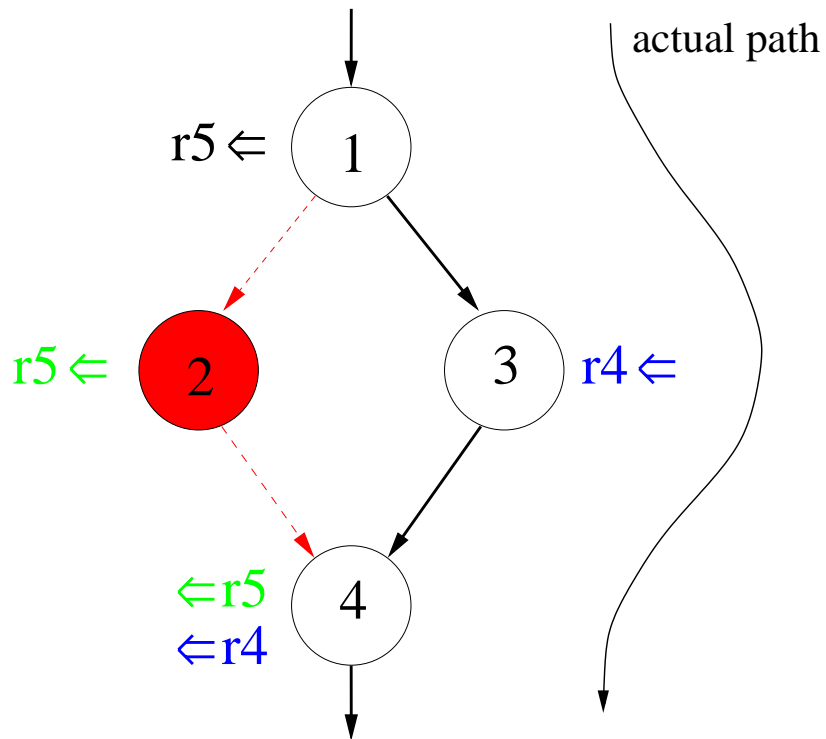
WR-nFD



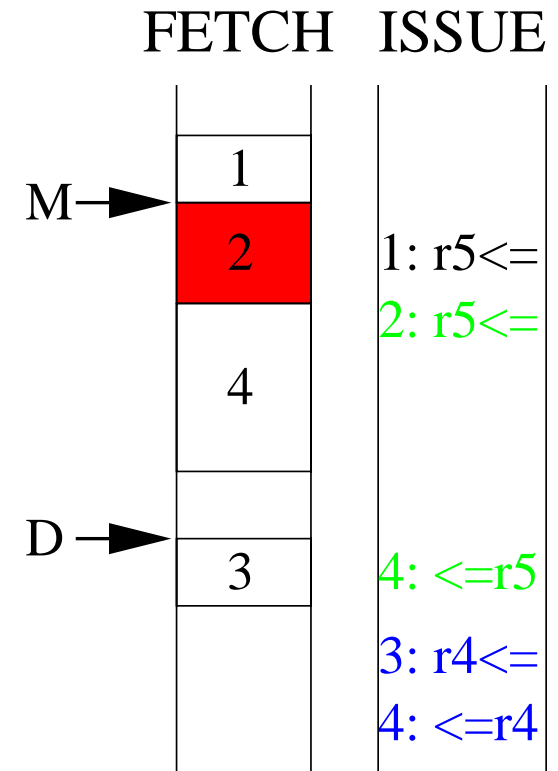
TIME



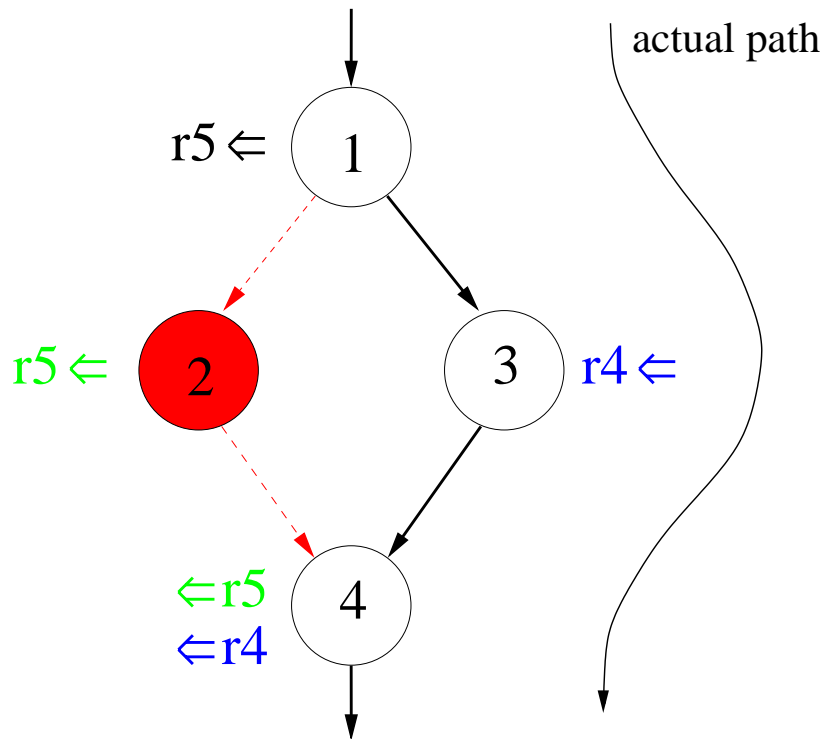
WR-FD



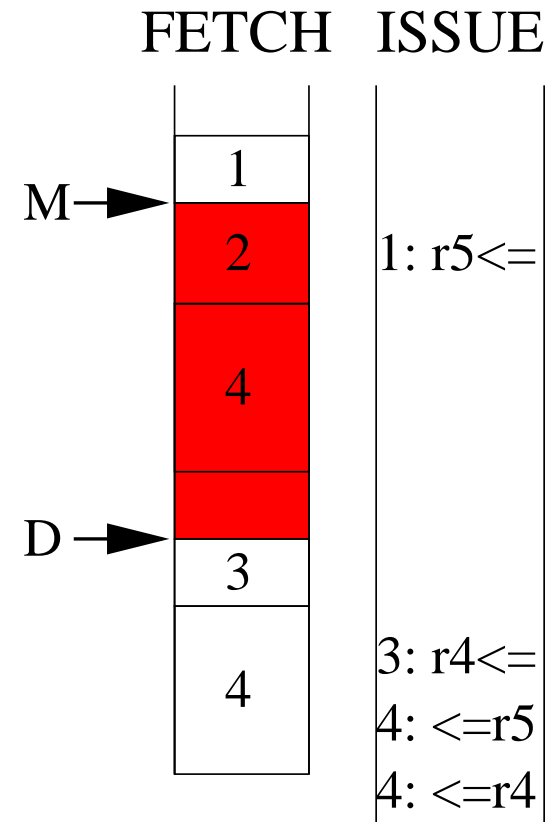
TIME



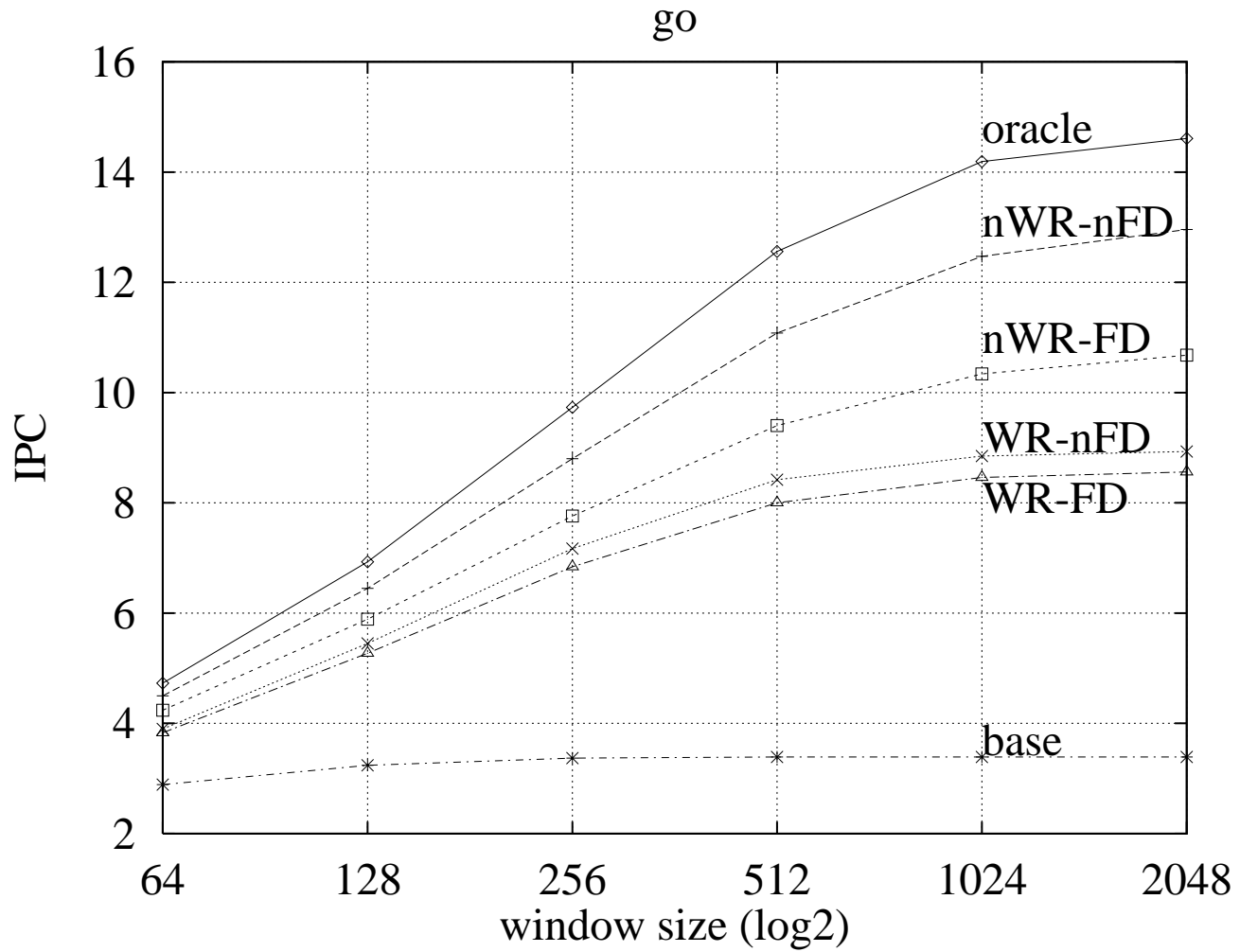
base



TIME



Ideal Study



Summary of Study

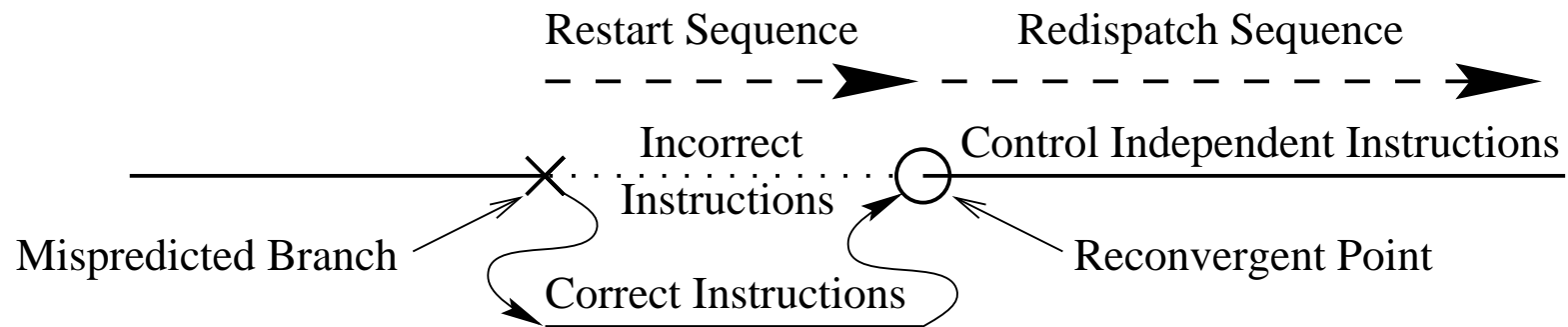
- Control independence seriously limited by **incorrect control dependent instructions**
 - wasted **window space**, fetch/execute resources
 - false data dependences
- Still has large potential
 - potentially reduce gap between oracle and real branch prediction by 1/2

Applications

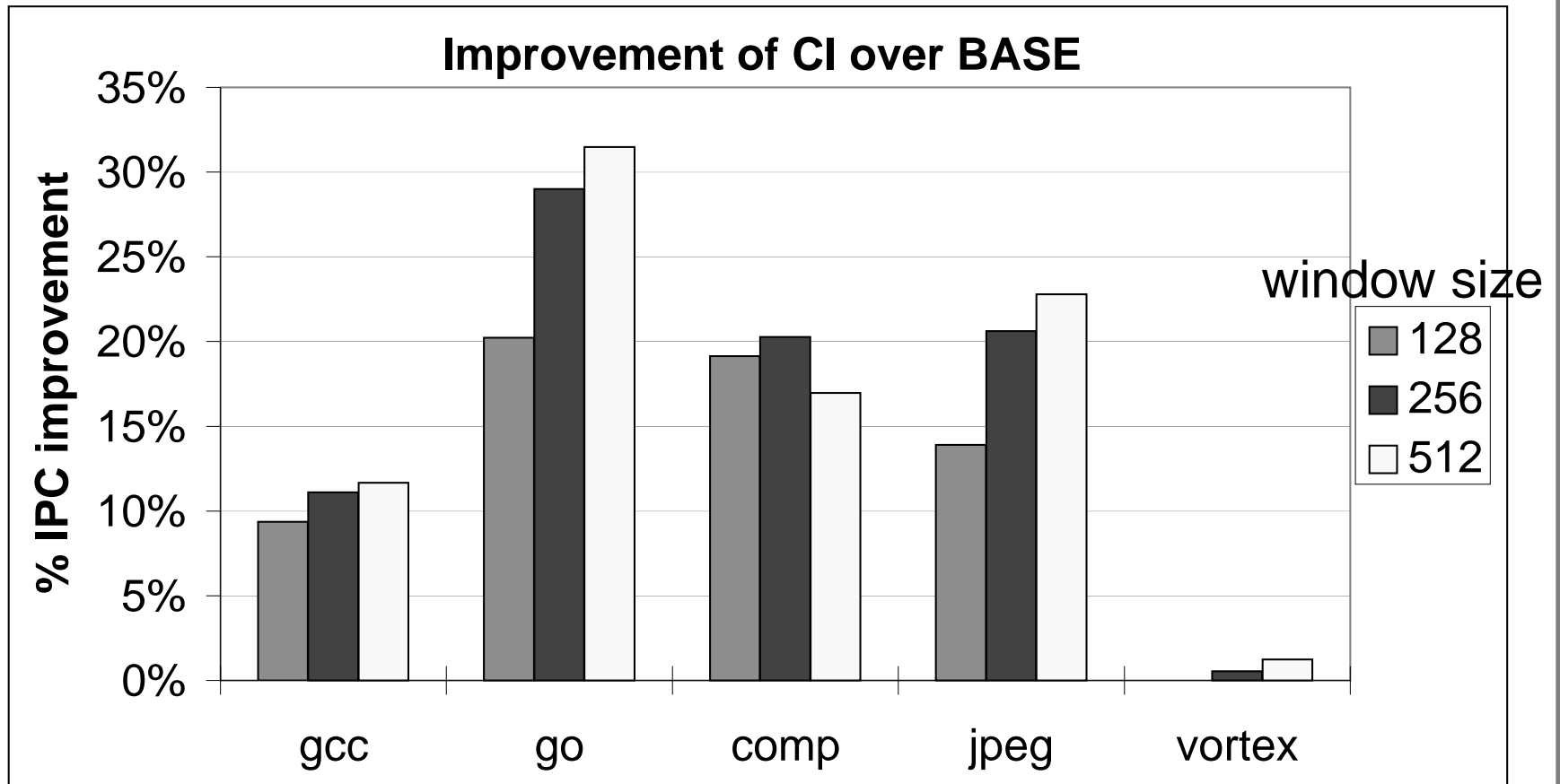
- Apply insight to guide implementations of **WR-FD model**
 - e.g. Trace Processors
- Large WR component suggests designs capable of “absorbing” incorrect paths
 - e.g. “expandable window” and multiple fetch units of Multiscalar Processors
 - **nWR-FD model** => performance potential for Multiscalar with aggressive data dependence resolution/recovery
- The study suggests several new directions...
 - **nWR-nFD model**
 - Interesting ways of handling data dependent/control independent instructions

Implementation Study

- Basic requirements for misprediction recovery
 - Detect the re-convergent point
 - Insert/remove instructions from middle of window
 - Form correct data dependences
 - Selectively reissue instructions



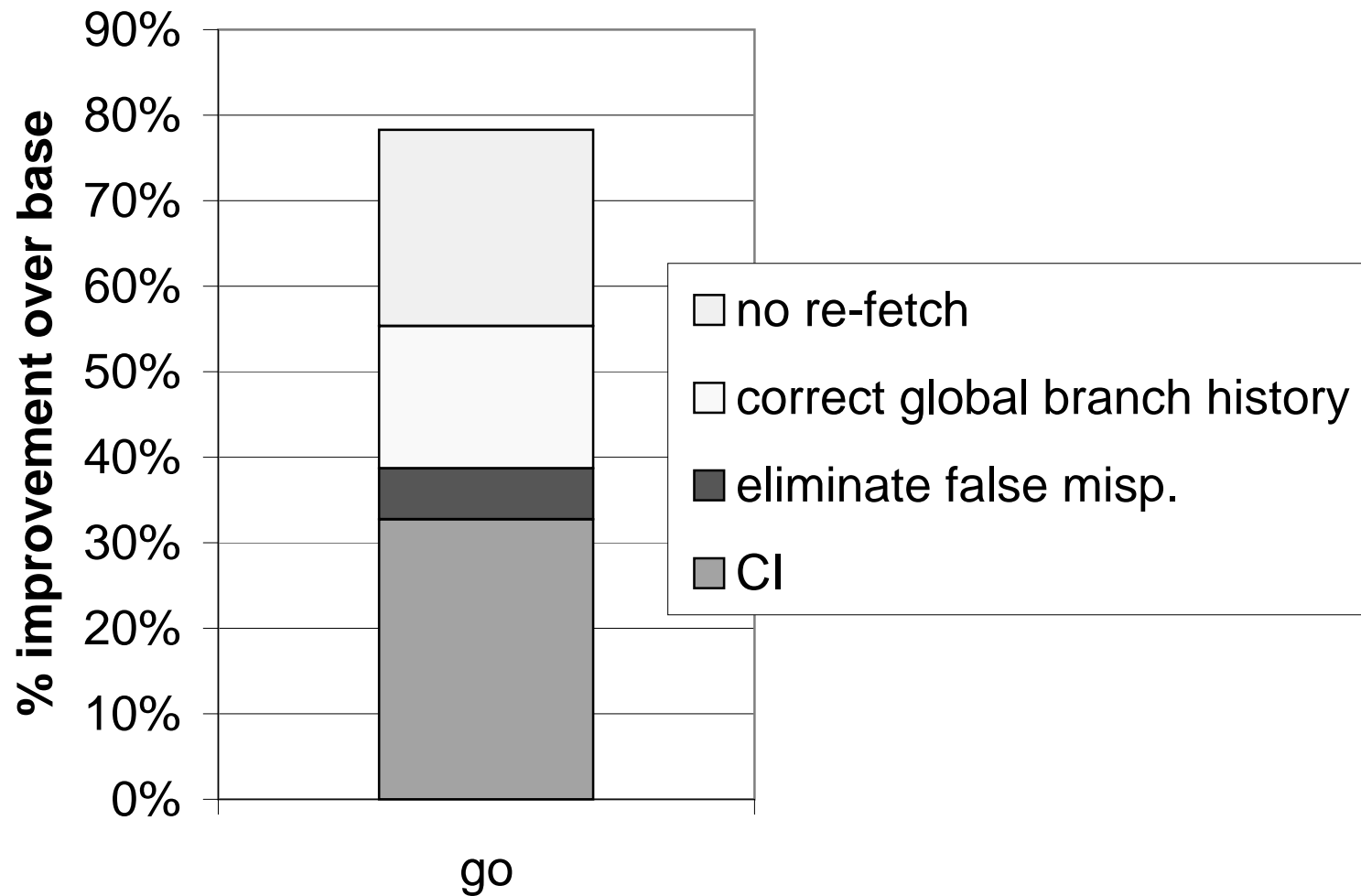
Implementation Study



Other Limiting Factors

- Factors exposed by implementation study
 - False mispredictions
 - Incoherent global branch history
 - Re-fetching/Re-dispatching control independent instructions
 - ...and many others. (See Technical Report and revisions.)

Implementation Study Revisited



Closing Remarks

- Control independence has been gaining momentum
 - Several limit studies
 - Instruction reuse
 - CMU studies
 - DMT
 - Multiscalar/Spec MT
 - Trace processors
 - Michigan studies
 - Instruction recycling in SMT
- We'll see even more control independent designs in the future => misprediction bottleneck
- Now's the time to understand the primary and secondary factors affecting control independence