On Pipelining Dynamic Instruction Scheduling Logic

Jared Stark † Mary D. Brown ‡ Yale N. Patt ‡

Microprocessor Research Labs †
Intel Corporation
jared.w.stark@intel.com

Electrical and Computer Engineering ‡
The University of Texas at Austin
{mbrown,patt}@ece.utexas.edu

Palacharla, Jouppi, and Smith, in ISCA ’97:

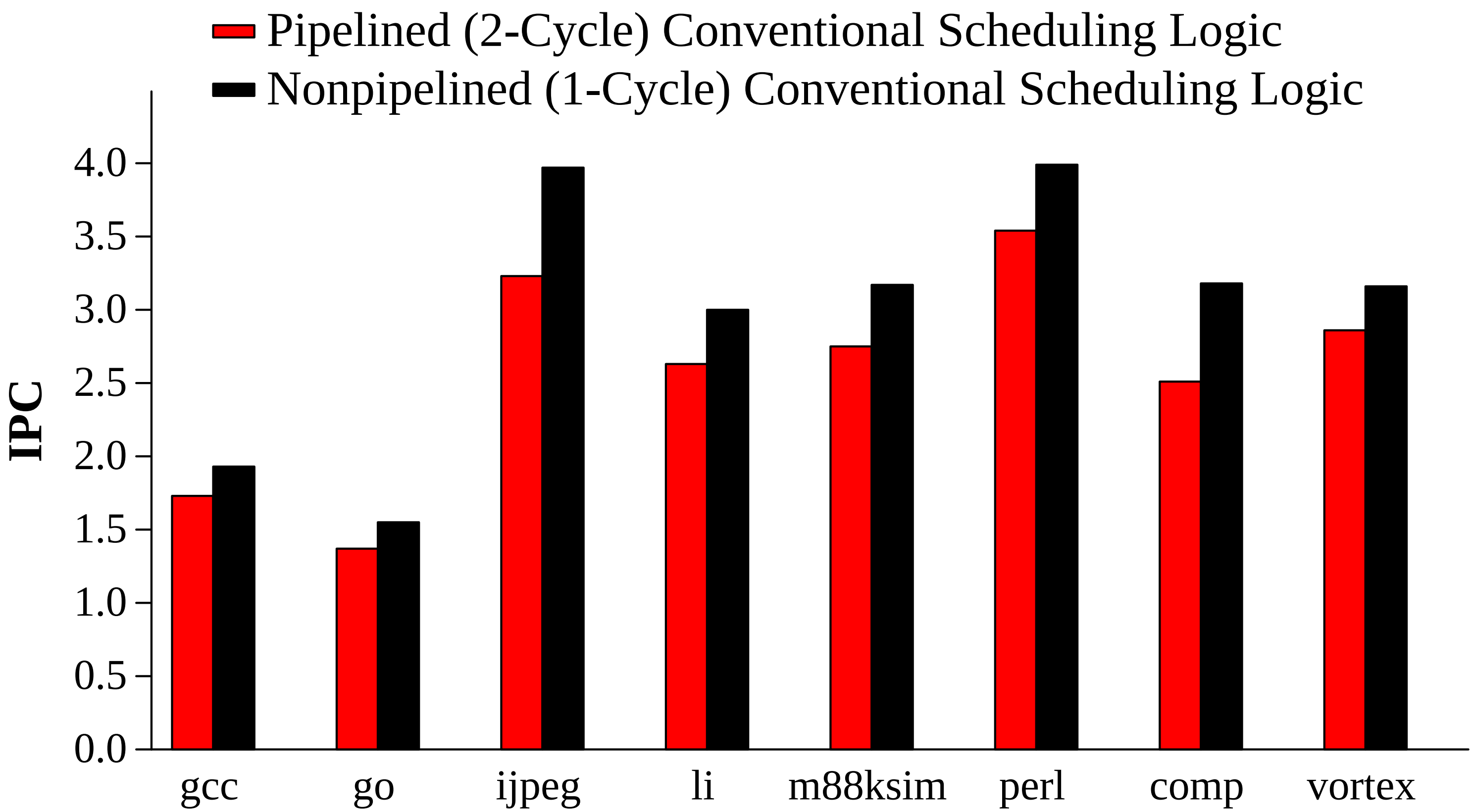
“Wakeup and select together constitute what appears to be an *atomic* operation. That is, if they are divided into multiple pipeline stages, dependent instructions cannot issue in consecutive cycles.”

Experiment

- Two Machines:
  - Pipelined (2-Cycle) Conventional Scheduling Logic
  - Nonpipelined (1-Cycle) Conventional Scheduling Logic

- Machine Configurations:
  - 10 Stage Pipeline (Minimum)
  - 8 Wide (Fetch, Execute, and Retire)
  - 128 Entry Instruction Window
  - 8 Kbyte Gshare Branch Predictor
  - 64 Kbyte I & D Caches

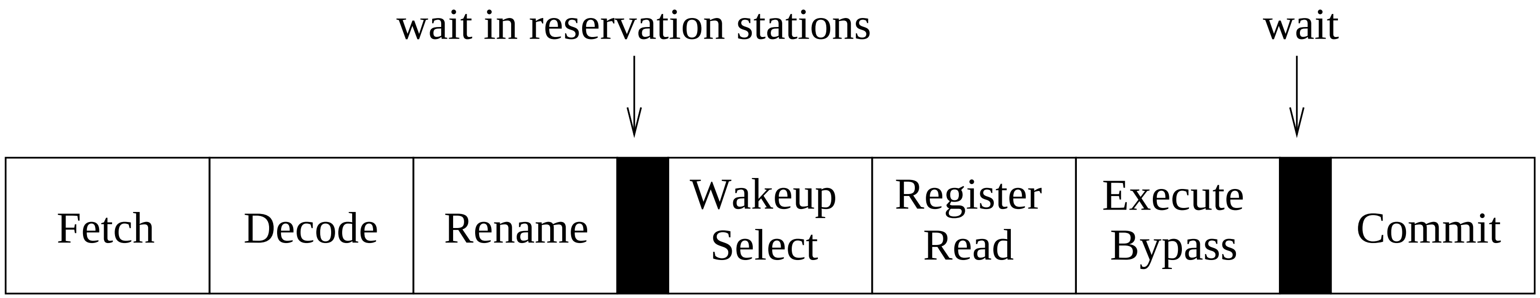
Effect of Pipelining the Scheduling Logic



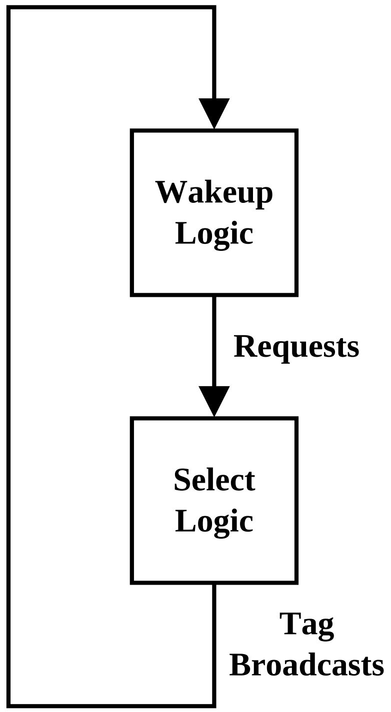
Outline

- Background
- Conventional Scheduling
- Pipelined Scheduling w/ Speculative Wakeup
- Experiments
- Conclusions

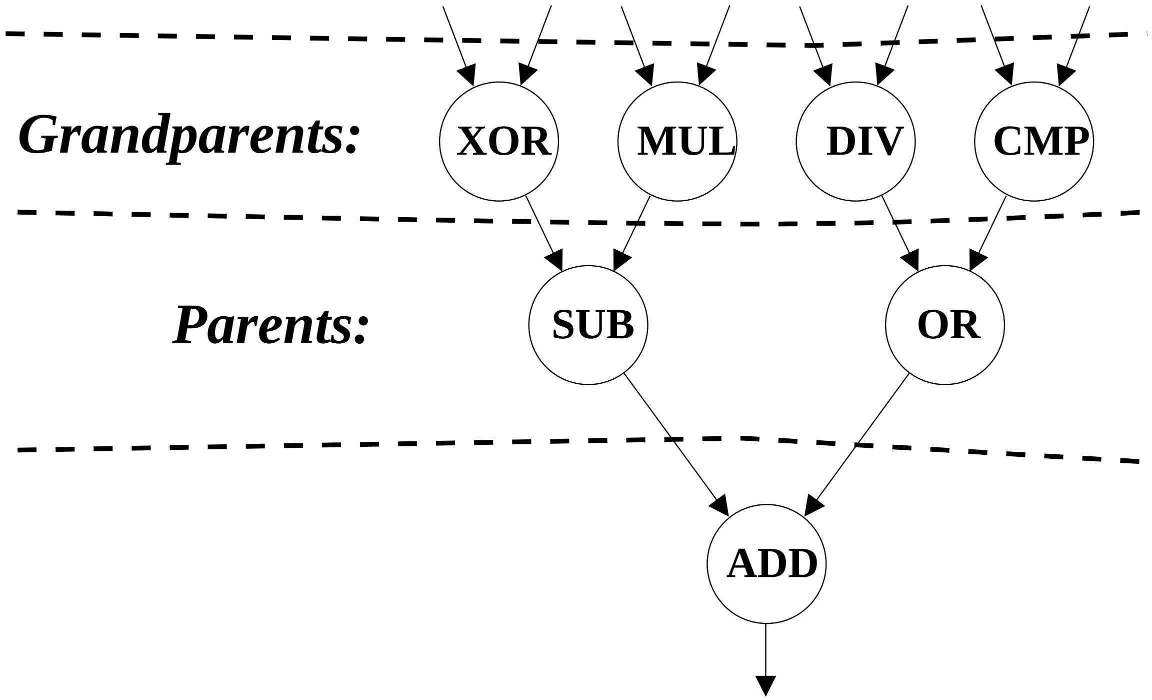
Pipeline for Out-of-Order Execution



Scheduling Loop

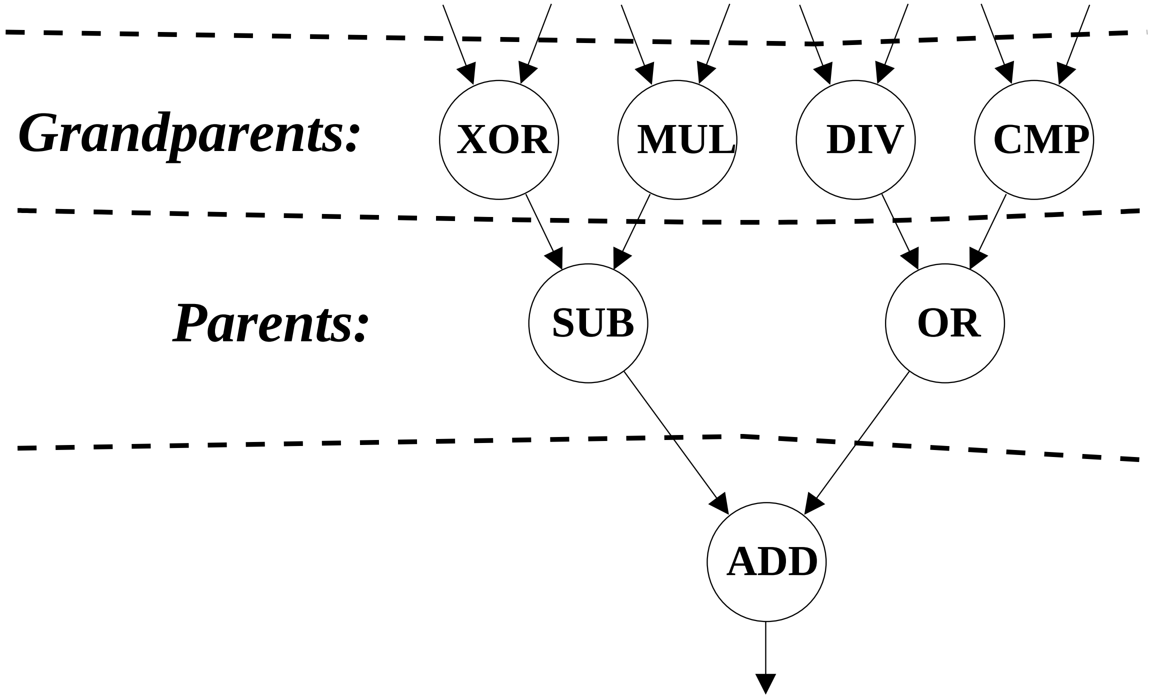


Terminology



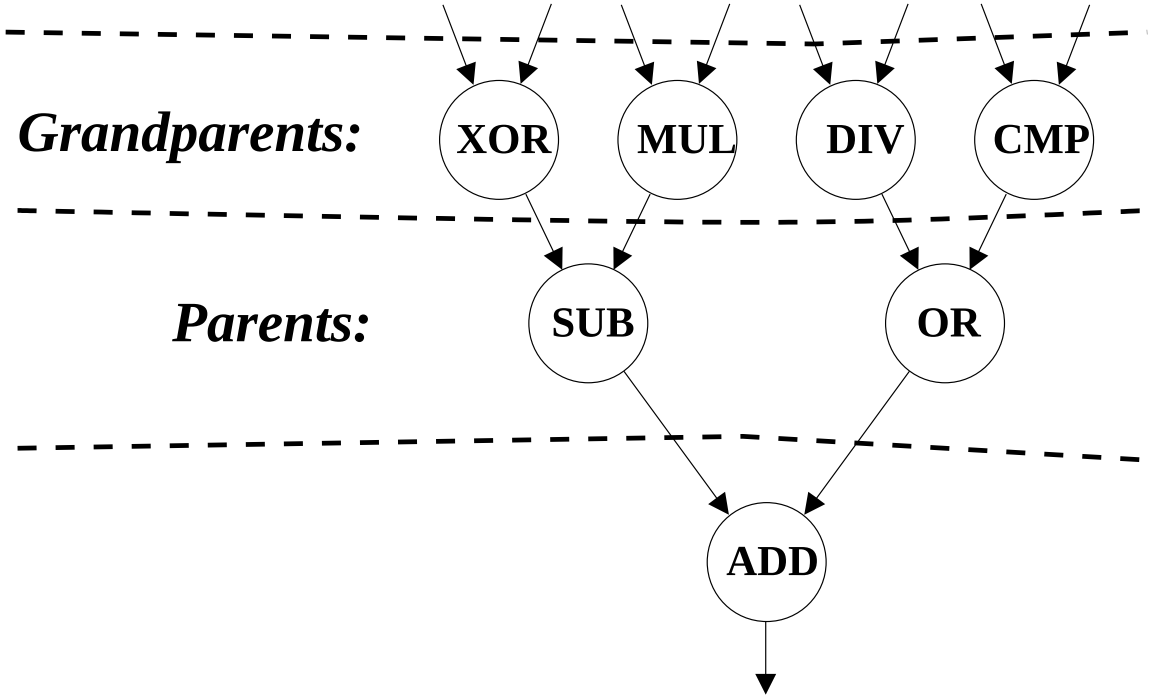
Conventional Scheduling

Wakeup when all parent tags have been broadcast.

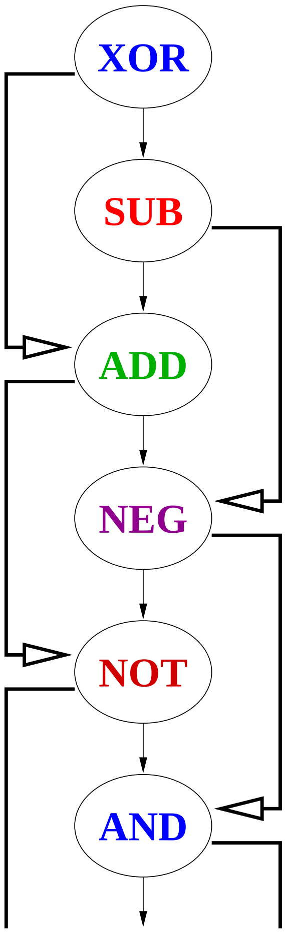


Pipelined Scheduling w/ Speculative Wakeup

Wakeup when all grandparent tags have been broadcast.

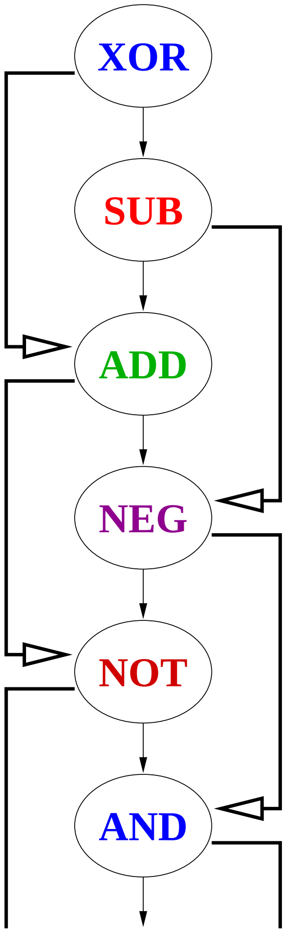


Pipelined Scheduling w/ Speculative Wakeup (continued)



Cycle 2:	XOR Select	
Cycle 3:	ADD Wakeup	SUB Select
Cycle 4:	ADD Select	NEG Wakeup
Cycle 5:	NOT Wakeup	NEG Select
Cycle 6:	NOT Select	AND Wakeup
Cycle 7:		AND Select

Why *Speculative* Wakeup?



Cycle 1:	XOR Wakeup	SUB Wakeup
Cycle 2:	XOR Select	~~SUB Select~~
Cycle 3:	ADD Wakeup	SUB Select
Cycle 4:	ADD Select	NEG Wakeup
Cycle 5:	NOT Wakeup	NEG Select
Cycle 6:	NOT Select	AND Wakeup
Cycle 7:		AND Select

Experiments

- Baseline:

  Pipelined (2-Cycle) Conventional Scheduling

- Speculative Wakeup:

  Pipelined (2-Cycle) Scheduling w/ Speculative Wakeup

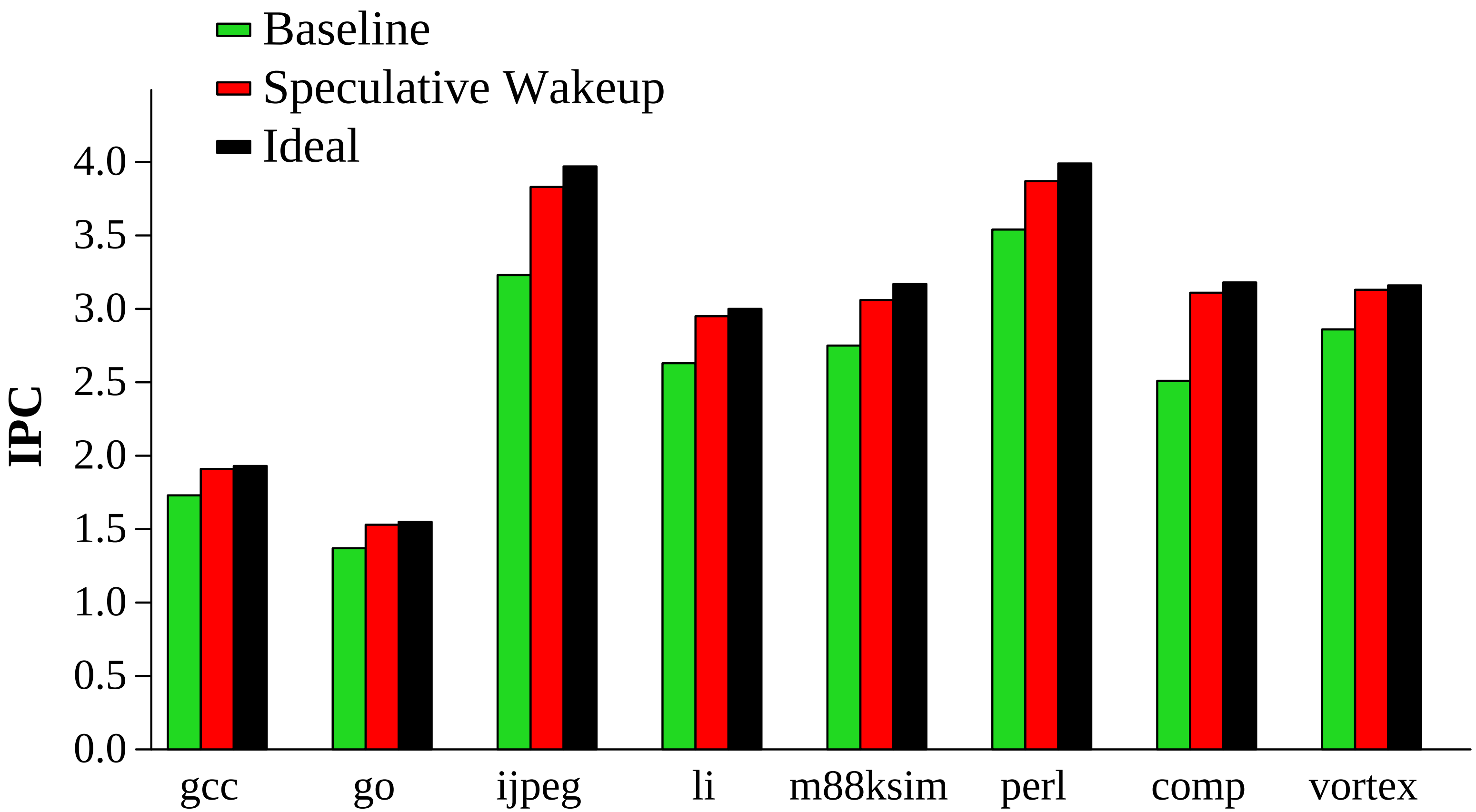
- Ideal:

  Nonpipelined (1-Cycle) Conventional Scheduling

Machine Configurations

- Pipeline Depth: 10 Stages (Minimum)
- Issue Rate: 8 Instructions / Cycle
- Branch Predictor: 8KB Gshare
- ICache/DCache: 4-Way 64KB, Hit/Miss: 2/9+ Cycles
- Execution Core: 8 Functional Units (4 Load/Store)
- Window Size: 128 Instructions (16 per Functional Unit)

Results


Baseline
Speculative Wakeup
Ideal
IPC
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
gcc
go
ijpeg
li
m88ksim
perl
comp
vortex

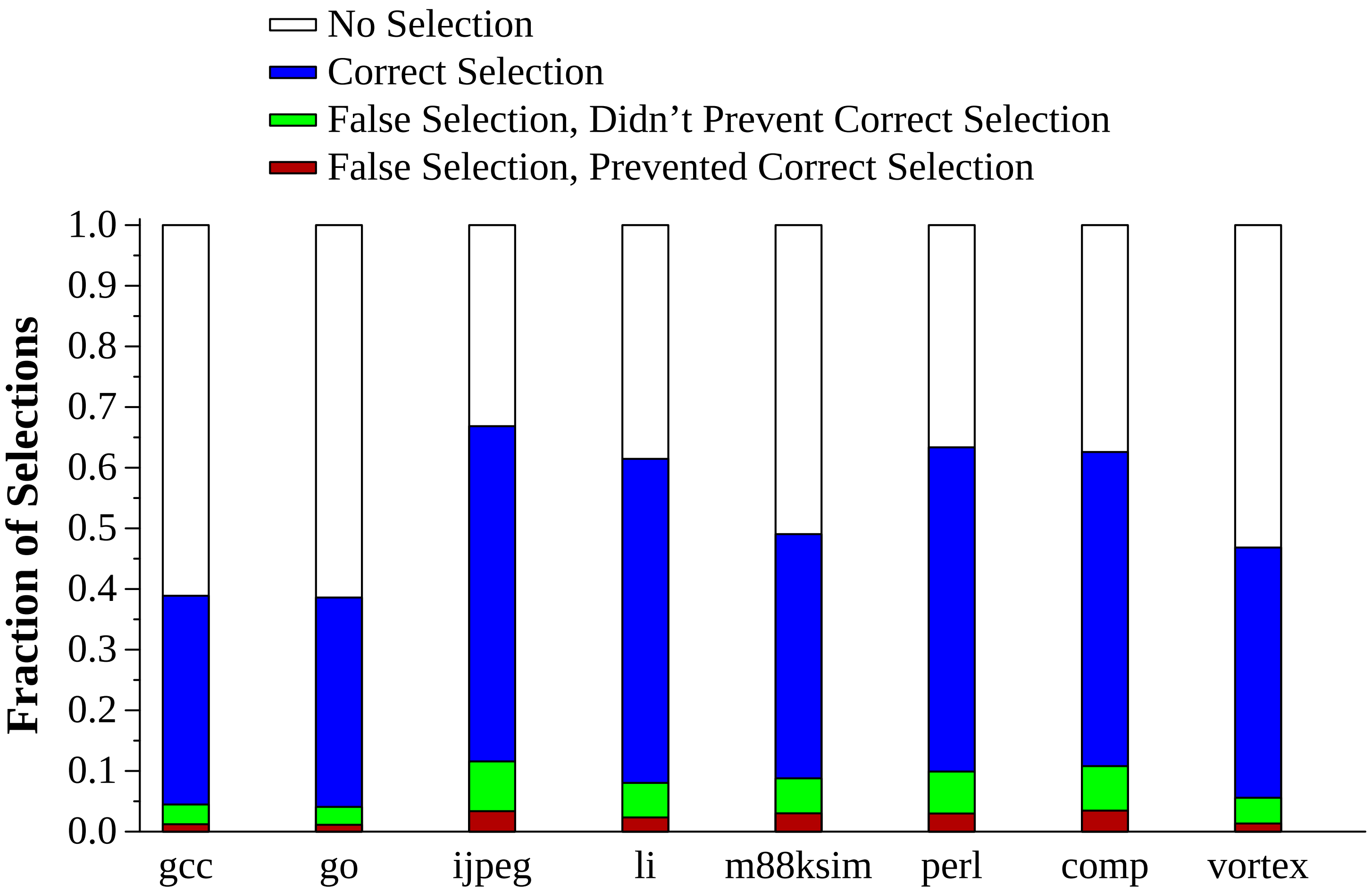
Conclusions

Scheduling logic can be pipelined with little impact on IPC.

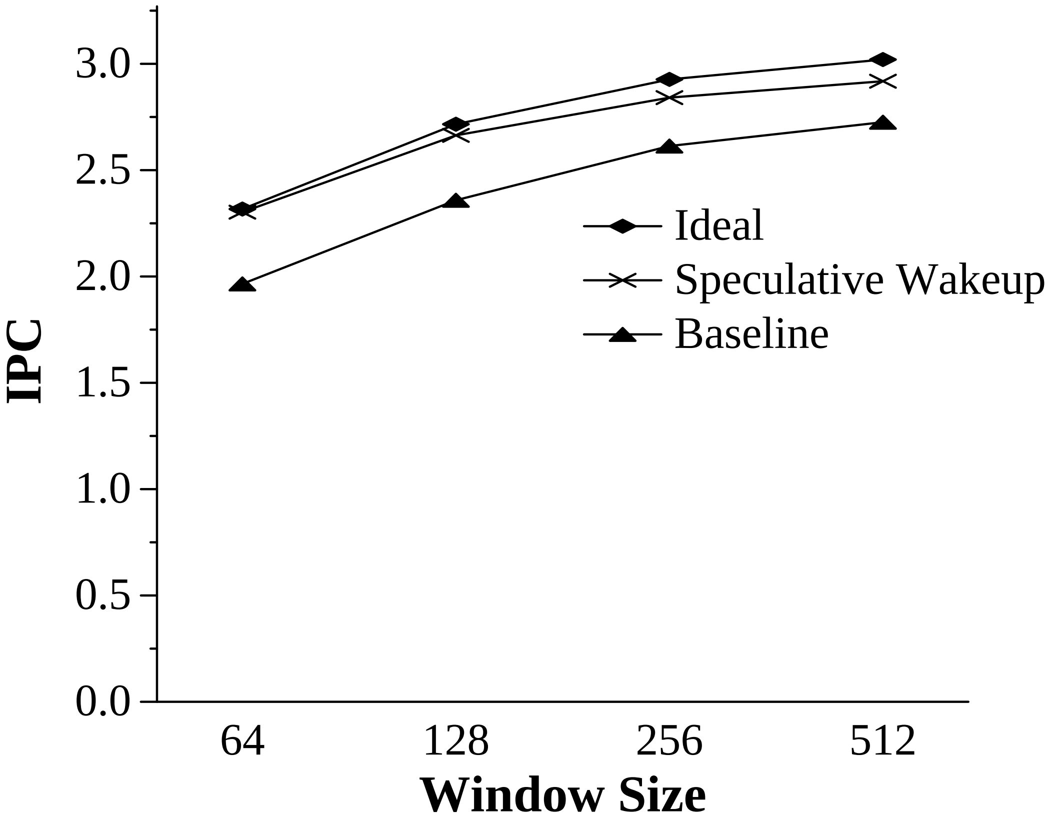
Compared w/ Baseline (Pipelined Conventional) Scheduling:

- Pipelined Scheduling w/ Speculative Wakeup: + 13% IPC
- Ideal (Nonpipelined Conventional) Scheduling: + 15% IPC

Selection Outcomes



Varied Window Size (8-Wide Superscalar)



Varied Window Size (16-Wide Superscalar)