

CVA6 Student Contest

Télécom PaRISC

An adventure

- Lots of ideas
- Few able to be implemented on time
- Even less actually implemented
- Only one working as intended
- None improving the score
- One improving frequency accidentally

Our main ideas

- D\$ Prefetch*
- Going superscalar
- Reordering instructions

* actually the only feature that was kept in the final product

D\$ Prefetch



Going superscalar

Current scoreboard has room for improvement

- only contains issued instructions -> stalls ID
- issues one instruction per cycle

Going superscalar: how?

- allow unissued instructions in the scoreboard
- find **independent** non-issued instructions
- compute dependencies (= `rd_clobber`) and forwarding
- read operands and send to EX stage

Note: already multiple commit ports, no further action needed

Reorder buffer

- main idea: if an instruction is stuck between ID and IS stage, swap it with the next instruction
- careful about dependencies and control hazards (i.e. never swap branch instructions)
- problem: fetches instructions faster -> if branch predict is wrong, then fetches more useless instructions

Reorder buffer -> I\$ prefetch

- attempt to fix the aforementioned issue
- when the IF instruction queue is full (=> no I\$ request), prefetch instructions that the branch predictor dismissed

I\$ prefetch -> LRU replacement policy

- current I\$ (and D\$) uses a random replacement policy.
- attempt to replace it with a LRU policy

Comments on coremark

- focuses on heavy mathematical workloads
- not that memory intensive

Results!

	Frequency	Coremark	LUT	FF	DSP
Reference	42.9 MHz	112.2155	14807	9286	4
Final	43.9 MHz	111.8707	15059	9291	4