
Instruction Issue Logic

Wen-mei Hwu
ECE

University of Illinois, Urbana-Champaign

Data Dependences

- Data flow dependence
 - consumer-producer relationship
 - register bypass and interlocks
- Data output and antidependences
 - reuse of registers at compile time
 - register renaming

Register Value Bypass

- Also known as forwarding
 - make values generated by function units immediately available (fig.)
 - costly due to the number of function units (10+ in typical processors)
 - simple bypass in in-order execution
 - more complex bypass in OOO.

Interlocking

- Allow instruction to execute only when data and resources ready
 - simple interlocking based on bypass logic for short pipelines
 - scoreboarding for deep pipelines
 - Tomasulo's Algorithm for out-of-order instruction dispatch

Tomasulo's Algorithm

- Invented for IBM 360-91 FPU
- First published in 1967(IBM Journal)
- Not for general CPU design until 1990's.
 - branch prediction and exception recovery problems solved

Tomasulo's Algorithm

- Register renaming
 - tags for values
- Out-of-order execution
 - reservation stations
- Data forwarding
 - common data bus

Tomasulo's Algorithm

- Instruction decode
 - fetch register file for value and tag
 - tag is handle for data currently being generated
 - determine RS to hold the decoded operations

Tomasulo's Algorithm

- Instruction Issue
 - insert operation and operands into reservation station entry assigned
 - mark destination register as not ready

Tomasulo's Algorithm

- Operation dispatch
 - identify operations ready for execution
 - determine highest priority operation for each port/function unit

Tomasulo's Algorithm

- Data forwarding
 - result value and tag distributed to RS entries for associative search
 - result value and tag delivered to destination register for potential update

P6-like Out-of-Order Execution

- Analyzed info and figures available on WWW in papers
- Drew on experience with previous Out-Of-Order design
- Actual P6 implementation may differ from this analysis

Out-of-Order Core

- Register Alias Table (RAT)
- Reservation Station (RS)
- Reorder Buffer (ROB)
- Retirement Register File (RRF)

Key Out-of-Order issues

- Register renaming
- Reservation Station
 - writing operations into RS
 - dispatching operations back-to-back
 - Selecting operations for dispatch
 - Selecting ports for dispatch

Key Out-of-order Issues

- Maintaining precise state
 - recovering from exceptions
 - recovering from mispredicted branches

Register Renaming

- Convert a logical register (EAX, EBX) into physical register (one of ROB's 40 entries or an RRF entry)
 - There is one entry in the retirement register file (RRF) associated with each logical register
 - A logical register may map to any reorder buffer (ROB) entry using RAT

Steps of Register Renaming

- Get physical register ids for each source register
- Allocate ROB entry for each operation and update RAT

Steps of Register Renaming

- Fix up register flows between operations
- Update RAT and operations with retirement information

Writing Operations Into RS

- Snoop result buses for values produced this cycle
 - Need to get operation's physical register ids into RS early enough to do snooping in this cycle (in parallel with read of RRF and ROB)

Reservation Station Entries

- Holds operations while waiting for execution
 - Snoops on result bus for operand status
 - Maintain valid bit to track each operand's progress

Reservation Station Entries

- Operation can execute when all three valid bits set to 1
 - Clears entry valid bit when it issues to execution unit

Dispatching Operations

- If wait until operands are written into RS (fig.)
 - No bypassing needed
 - Adds 2 cycles to each operation's effective latency

Dispatching Operations

- If notify RS dispatcher when starting execution of producer (fig.)
 - Full bypassing now required
 - Minimum effective latency -> 2 cycles (Effective for 2+ cycle ops)
 - May mislead RS dispatcher (I.e. cache miss)

Dispatching Operations back-to-back

- If notify RS dispatcher when starting dispatch stage 2 of producer
 - Now can have a 1 cycle effective latency
 - May mislead RS dispatcher more (resources unavailable, cache miss)

Dispatching Operations back-to-back (cont.)

- Cancel operations when detect that dispatcher was misled
 - Use bypass logic to detect when operands not truly available
 - Need to re-dispatch canceled operations later
 - Up to two cycles of work can be lost each time dispatcher misled

Selecting Operations for Dispatch

- Select the oldest ready operations (FIFO)
 - Most likely to have operations waiting for result
 - May be holding up retirement of mispredicted branch in ROB
 - Expensive to implement

Selecting Port for Dispatch

- By port availability (dynamic)
 - More complex selection logic
 - Which port should the op go to? (fig.)
 - Better load balancing across ports
- By entry location (static)
 - Less complex selection logic
 - Select port when placed in reservation station

Selecting Port for Dispatch

- Static (cont.)
 - Lost opportunities
 - Port selection could have big impact on performance
- On P6, only IEU ops affected (fig.)
 - Port contention effects may become more pronounced as units are added