Arch: Pipelining

Alex Chi

Update: April 16, 2020

Contents

1 What is pipelining?

- laundry example
- pipeline doesn't help latency

- helps throughput of entire workload
- potential speedup = #f pipeline stages
- 5-stage pipeline: instruction fetch, register file access, alu, data access, register write
- all instructions go through 5 stages

2 Pipeline Control

- EX
  - RegDst
  - ALUop
  - ALUSrc
- MEM
  - Branch
  - MemRead
  - MemWrite
- WB
  - MemToReg
  - RegWrite
- Control signals will be forwarded stage by stage

3 Data Hazard

- A situation that prevents starting instruction
- Structure hazard: conflict over use of a resource
  - e.g. instruction memory and data memory
- Data hazard: data not ready
  - forward (or bypassing): fetch data from previous stage
    - ex-decode
    - mem-decode requires stalling
      - compiler can reorder instructions
  - stall: stop issuing new instructions

4 Control hazard

-fetch next instruction depends on current one

- solution: flush the pipeline
- reduce control hazard
  - resolve target in ID stage
  - delayed branch (requires compiler to schedule)
  - branch prediction
    - flush instructions if wrong

5 Exceptions

- normal execution order of instruction is changed
- a.k.a. interrupt, fault, trap

5.1 Precise / Imprecise

- if instructions before fault complete and those after restart from scratch, precise

5.2 Steps

- force trap instruction into pipeline
- until trap is taken, turn off all writes for faulting instruction and for all instructions that follow in the pipeline
- after the exception-handling routine in the OS receives control, it immediately saves the PC of the faulting instruction

5.3 Exceptions in MIPS

- IF: page fault, misaligned memory, memory protection
- ID: undefined or illegal opcode
- EX: arithmetic exception
- MEM: page fault on data fetch, misaligned memory, memory protection
- WB: none

6 Handling Multi-cycle Operations

- complicated bypass or forwarding

- potential structural hazard
- multiple instruction may complete at the same time
- out-of-order completion, precise exception issue
- bypassing and forwarding now will stall more cycles
- as instructions go through different pipelines, there will be structural hazards
- also out-of-order completion (more of a data hazard), WAW
- ultimate solution: scoreboarding, tomasulo algorithm, speculation, reorder buffer

7 Super Pipeline

- e.g. Intel: 20+ stages