
CMSC 411
Computer Systems Architecture
Lecture 4
Basic Pipelining
Alan Sussman
als@cs.umd.edu

**PERFORMANCE
MEASUREMENT**

Administrivia

- Homework problems for Unit 1 posted and due next Thursday
- Office hours posted for instructor and TA
- Read Appendix C – Basic Pipelining
- Appendix K useful as a MIPS ISA reference
 - we'll only touch on parts of it in lecture as needed
- Appendix A useful as a reference on basic features of instruction sets
 - arithmetic operations
 - addressing modes
 - branches
 - and more ...

Summary of Technology Trends

- For disk, LAN, memory, and microprocessor, **bandwidth improves by square of latency improvement**
 - In the time that bandwidth doubles, latency improves by no more than 1.2X to 1.4X
- Lag probably even larger in real systems, as **bandwidth gains multiplied by replicated components**
 - Multiple processors in a cluster or even in a chip
 - Multiple disks in a disk array
 - Multiple memory modules in a large memory
 - Simultaneous communication in switched LAN
- HW and SW developers should innovate assuming **Latency Lags Bandwidth**
 - If everything improves at the same rate, then nothing really changes
 - When rates vary, require real innovation

Outline

- Moore's Law and Technology Trends
- Reliability and MTTF

Define and quantify dependability (1/3)

- How decide when a system is operating properly?
- Infrastructure providers now offer Service Level Agreements (SLA) to guarantee that their networking or power service would be dependable
- Systems alternate between 2 states of service with respect to an SLA:
 1. **Service accomplishment**, where the service is delivered as specified in SLA
 2. **Service interruption**, where the delivered service is different from the SLA
- **Failure** = transition from state 1 to state 2
- **Restoration** = transition from state 2 to state 1

Define and quantify dependability (2/3)

- **Module reliability** = measure of continuous service accomplishment (or time to failure).
2 metrics
 1. **Mean Time To Failure (MTTF)** measures Reliability
 2. **Failures In Time (FIT)** = $1/\text{MTTF}$, the rate of failures
 - Traditionally reported as failures per billion hours of operation
- **Mean Time To Repair (MTTR)** measures Service Interruption
 - **Mean Time Between Failures (MTBF)** = $\text{MTTF} + \text{MTTR}$
- **Module availability** measures service as alternate between the 2 states of accomplishment and interruption (number between 0 and 1, e.g. 0.9)
- **Module availability** = $\text{MTTF} / (\text{MTTF} + \text{MTTR})$

Example calculating reliability

- If modules have **exponentially distributed lifetimes** (age of module does not affect probability of failure), overall failure rate is the sum of failure rates of the modules
- Calculate FIT and MTTF for 10 disks (1M hour MTTF per disk), 1 disk controller (0.5M hour MTTF), and 1 power supply (0.2M hour MTTF):

FailureRate =

MTTF =

Example calculating reliability

- If modules have *exponentially distributed lifetimes* (age of module does not affect probability of failure), overall failure rate is the sum of failure rates of the modules
- Calculate FIT and MTTF for 10 disks (1M hour MTTF per disk), 1 disk controller (0.5M hour MTTF), and 1 power supply (0.2M hour MTTF):

$$\begin{aligned} \text{FailureRate} &= 10 \times (1/1,000,000) + 1/500,000 + 1/200,000 \\ &= (10 + 2 + 5) / 1,000,000 \\ &= 17 / 1,000,000 \\ &= 17,000 \text{ FIT} \\ \text{MTTF} &= 1,000,000,000 / 17,000 \\ &\approx 59,000 \text{ hours} \end{aligned}$$

CMSC 411 - 4 (from Patterson)

9

Outline

- **MIPS – An ISA for Pipelining**
 - 5 stage pipelining
 - Structural and Data Hazards
 - Forwarding
 - Branch Schemes
 - Exceptions and Interrupts
 - Conclusion

CMSC 411 - 4 (from Patterson)

11

MIPS ISA AND PIPELINING

CMSC 411 - 4 (from Patterson)

10

A "Typical" RISC ISA

- 32-bit fixed format instruction (4 formats)
- 32 32-bit GPR (R0 contains zero, DP take pair)
- 3-address, reg-reg arithmetic instruction
- Single address mode for load/store:
base + displacement
 - no indirection
- Simple branch conditions
- Delayed branch

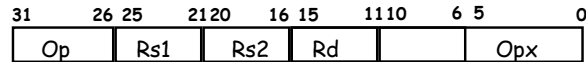
see: SPARC, MIPS, HP PA-Risc, DEC Alpha, IBM PowerPC, CDC 6600, CDC 7600, Cray-1, Cray-2, Cray-3

CMSC 411 - 4 (from Patterson)

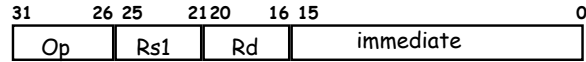
12

Example: MIPS

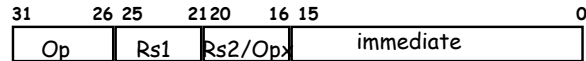
Register-Register



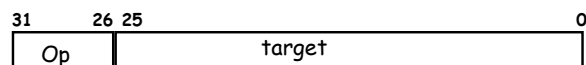
Register-Immediate



Branch



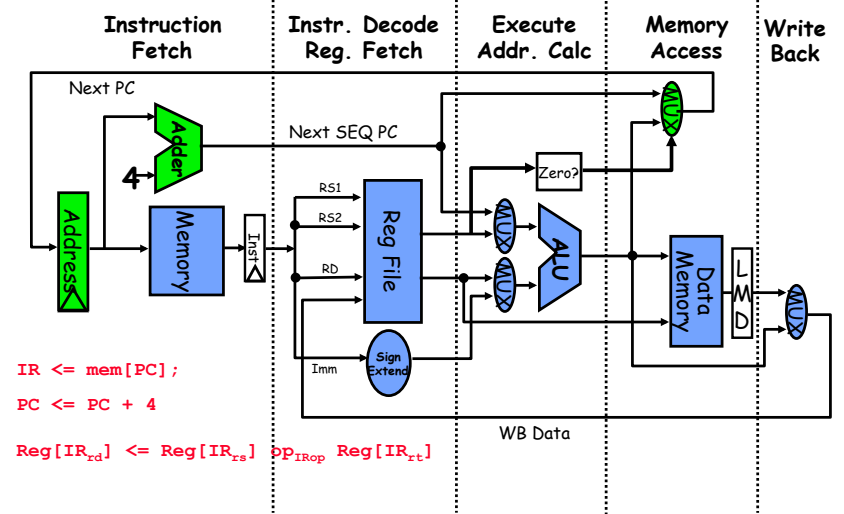
Jump / Call



CMSC 411 - 4 (from Patterson)

13

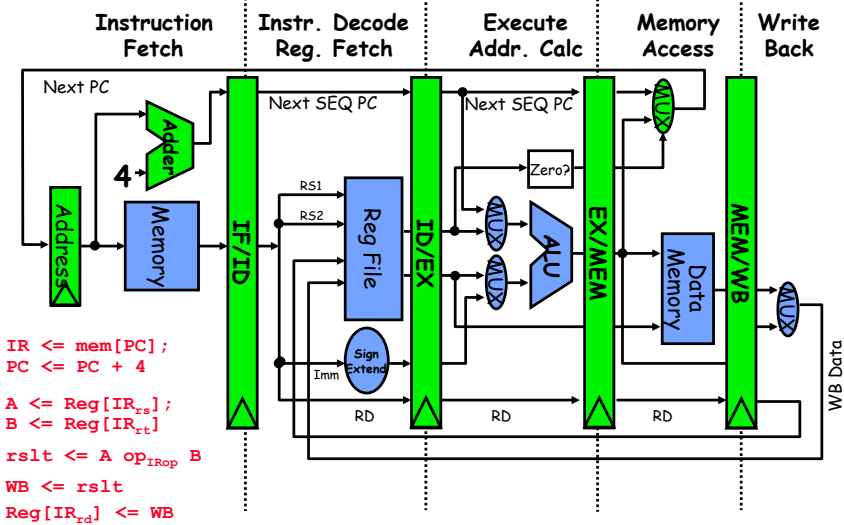
5 Steps of MIPS Datapath



CMSC 411 - 4 (from Patterson)

14

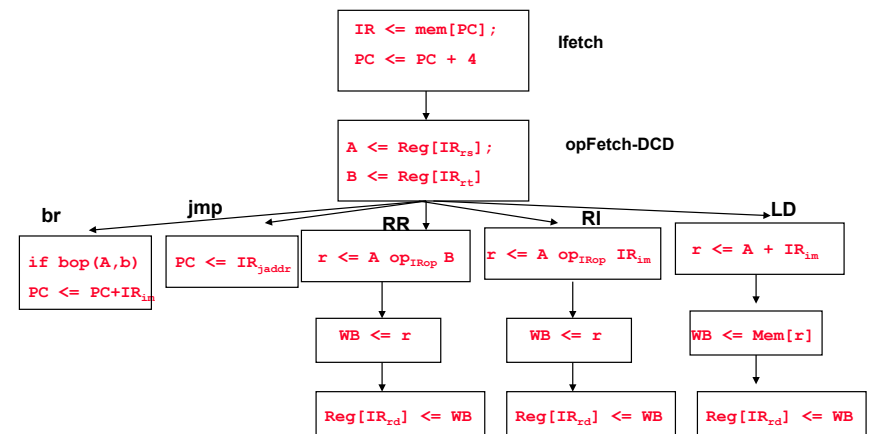
5 Steps of MIPS Datapath



CMSC 411 - 4 (from Patterson)

15

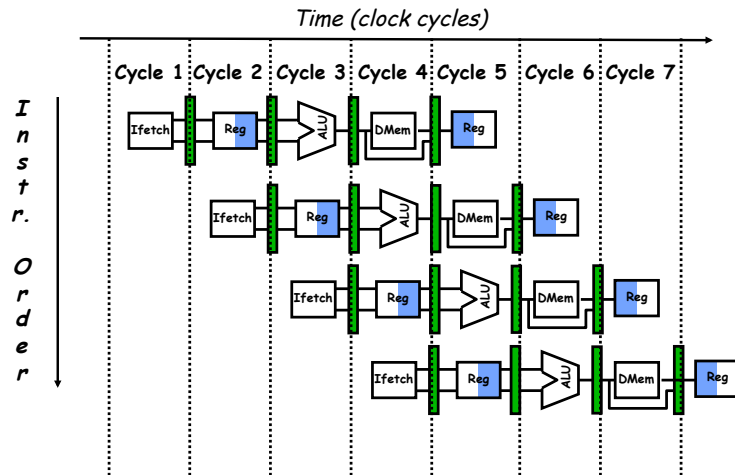
Inst. Set Processor Controller



CMSC 411 - 4 (from Patterson)

16

Visualizing Pipelining



CMSC 411 - 4 (from Patterson)

17

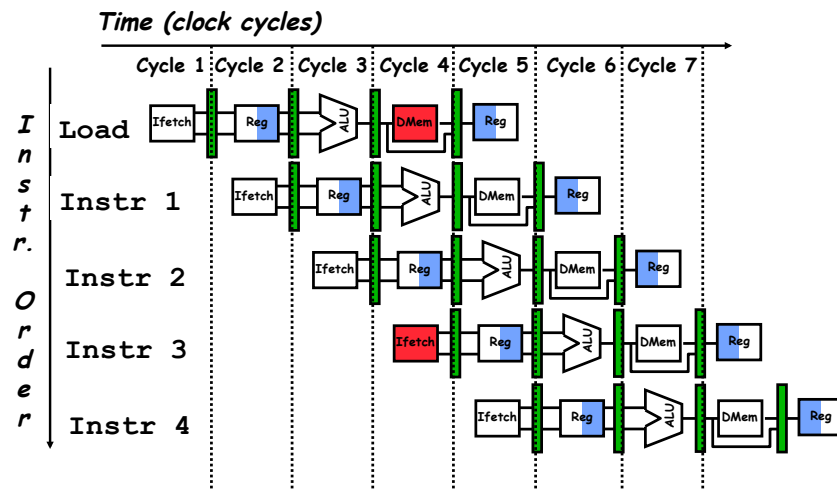
Pipelining is not quite that easy!

- Limits to pipelining: **Hazards** prevent next instruction from executing during its designated clock cycle
 - Structural hazards**: HW cannot support this combination of instructions (single person to fold and put clothes away)
 - Data hazards**: Instruction depends on result of prior instruction still in the pipeline (missing sock)
 - Control hazards**: Caused by delay between the fetching of instructions and decisions about changes in control flow (branches and jumps).

CMSC 411 - 4 (from Patterson)

18

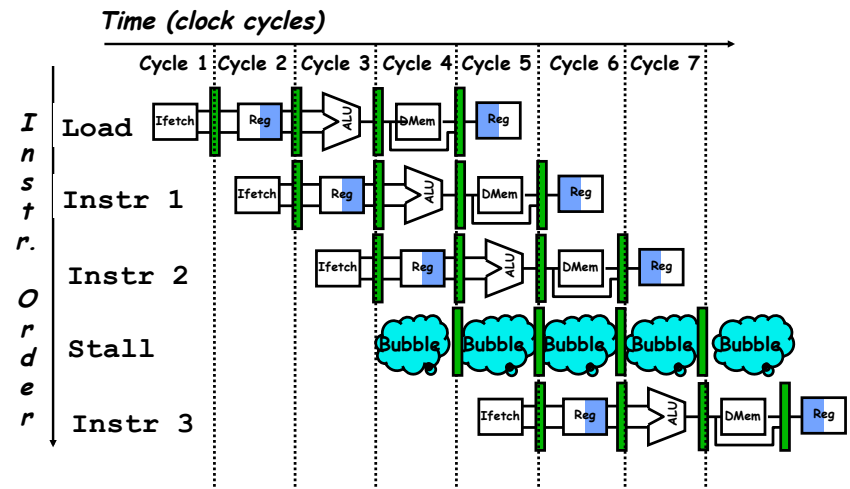
One Memory Port/Structural Hazards



CMSC 411 - 4 (from Patterson)

19

One Memory Port/Structural Hazards



How do you "bubble" the pipe?

CMSC 411 - 4 (from Patterson)

20

Speed Up Equation for Pipelining

$$CPI_{\text{pipelined}} = \text{Ideal CPI} + \text{Average Stall cycles per Inst}$$

$$\text{Speedup} = \frac{\text{Ideal CPI} \times \text{Pipeline depth}}{\text{Ideal CPI} + \text{Pipeline stall CPI}} \times \frac{\text{Cycle Time}_{\text{unpipelined}}}{\text{Cycle Time}_{\text{pipelined}}}$$

For simple RISC pipeline, ideal CPI = 1:

$$\text{Speedup} = \frac{\text{Pipeline depth}}{1 + \text{Pipeline stall CPI}} \times \frac{\text{Cycle Time}_{\text{unpipelined}}}{\text{Cycle Time}_{\text{pipelined}}}$$