Today

(brief brief) Review of (relevant to this course) architecture
Pipelines
Throughput / Latency
Because devices such as the hard drive and main memory have large capacities compared to the registers and ALUs, the constant movement of information among the different computer components experiences bottlenecks.

Q: HOW is information moved back and forth among components?

Q: What is the sequence of steps that are taken to “execute” a single program instruction?
Because devices such as the hard drive and main memory have large capacities compared to the registers and ALUs, the constant movement of information among the different computer components experiences bottlenecks.
Because devices such as the hard drive and main memory have large capacities compared to the registers and ALUs, the constant movement of information among the different computer components experiences bottlenecks.
Let us begin “small.”

**Q:** How might we “improve” the speed of a computer using the most naïve approach?

**Setup:** There are three instructions that we want to execute as part of a program ... how are they executed (concurrently?) so that their sum execution time is minimized
Q: What is a computer pipeline?
The acronyms IF, ID, EX, and WB are ...
The computer pipeline

Instruction Fetch
The instruction is fetched from physical memory

Instruction Decode
The instruction is decoded (for example if the instruction is to add two numbers, then fetch the values of R1 and R2 (registers) into the appropriate ALU)

Execute
Perform the calculation using the circuitry in the ALU

Write Back
Result of operation performed by ALU is written back to the register(s)
The computer pipeline

**Instruction Fetch**
- The instruction is fetched from physical memory.

**Instruction Decode**
- The instruction is decoded (for example, if the instruction is to add two numbers, then fetch the values of R1 and R2 (registers) into the appropriate ALU).

**Execute**
- Perform the calculation using the circuitry in the ALU.

**Write Back**
- Result of operation performed by ALU is written back to the register(s).

The instruction pipeline (orange, yellow, grey, green) may need to “fetch” information from L1, L2, main memory or the hard drive, so each instruction might require different amounts of time to complete.
The computer pipeline

**Goal**: calculate the “speed” or overall execution “time” for processing fully an instruction using the above pipeline

Assume that each step of the pipeline takes a single clock cycle to complete

**Q**: How many clock cycles does it take to complete one instruction?
**Goal**: calculate the “speed” or overall execution “time” for processing fully an instruction using the above pipeline.

Assume that each step of the pipeline takes a single clock cycle to complete.

**Q**: How many clock cycles does it take to complete one instruction?
The computer pipeline

Instruction “queue” Complete instructions

<table>
<thead>
<tr>
<th>IF</th>
<th>ID</th>
<th>EX</th>
<th>WB</th>
</tr>
</thead>
</table>

$I_1$

Clock cycle 2

**Goal:** calculate the “speed” or overall execution “time” for processing fully an instruction using the above pipeline

Assume that each step of the pipeline takes a single clock cycle to complete

**Q:** How many clock cycles does it take to complete one instruction?
### The computer pipeline

<table>
<thead>
<tr>
<th>Instruction “queue”</th>
<th>Completed instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>ID</td>
</tr>
<tr>
<td>$I_1$</td>
<td></td>
</tr>
</tbody>
</table>

Clock cycle 3

**Goal**: calculate the “speed” or overall execution “time” for processing fully an instruction using the above pipeline

Assume that each step of the pipeline takes a single clock cycle to complete

**Q**: How many clock cycles does it take to complete one instruction?
The computer pipeline

Instruction “queue”                                      Completed instructions

IF  ID  EX  WB

$I_1$

Clock cycle  4

Goal: calculate the “speed” or overall execution “time” for processing fully an instruction using the above pipeline

Assume that each step of the pipeline takes a single clock cycle to complete

Q: How many clock cycles does it take to complete one instruction?
We say that the pipeline latency of $I_1$ is 4 clock cycles.
The computer pipeline

Instruction “queue”

Completed instructions

I_2 \quad I_1

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

\[ I_2 \quad I_1 \]

Completed instructions

If the IF portion of the pipeline requires a single hardware component of the CPU, can more than 1 instructions be in the IF stage of the pipeline?

Q: How many clock cycles does it take to complete two instructions?
Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

| IF | ID | EX | WB |

I₂   I₁

Completed instructions

Clock cycle  2

Q: How many clock cycles does it take to complete two instructions?
Q: How many clock cycles does it take to complete two instructions?
Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”  

Completed instructions

IF  ID  EX  WB

\[ I_2 \quad 5 \quad I_1 \]

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

Completed instructions

<table>
<thead>
<tr>
<th>IF</th>
<th>ID</th>
<th>EX</th>
<th>WB</th>
</tr>
</thead>
</table>

\[ I_2 \quad I_1 \]

Clock cycle \( 6 \)

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

| IF | ID | EX | WB |

Completed instructions

\[ I_2 \quad I_1 \]

Clock cycle 7

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

Completed instructions

IF  ID  EX  WB

I₂  I₁

Clock cycle  8

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

Completed instructions

\[ \text{Instruction queue: } I_2 \quad I_1 \]

Clock cycle \( 8 \)

This is what a non pipelined architecture machine does

Q: Can we do better? How? Why?
The computer pipeline

Instruction “queue”

Completed instructions

\[ I_2 \quad I_1 \]

Clock cycle

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

Completed instructions

<table>
<thead>
<tr>
<th>IF</th>
<th>ID</th>
<th>EX</th>
<th>WB</th>
</tr>
</thead>
</table>

I₂  I₁

Clock cycle  1

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”                Completed instructions

| IF | ID | EX | WB |

\[ I_2 \quad I_1 \]

Clock cycle 2

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”         Completed instructions

\[ \text{IF} \quad \text{ID} \quad \text{EX} \quad \text{WB} \]

\[ l_2 \quad l_1 \]

Clock cycle   3

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”  Completed instructions

IF  ID  EX  WB

\[ I_2 \quad I_1 \]

Clock cycle 4

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

Completed instructions

IF  ID  EX  WB

$I_2$  $I_1$

Clock cycle  5

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Instruction “queue”

| IF | ID | EX | WB |

Completed instructions

\[ I_2 \quad I_1 \]

Clock cycle  5

This is what a pipelined architecture machine does

Q: How many clock cycles does it take to complete two instructions?
The computer pipeline

Another pictorial representation to show how the instructions are pipelined through the 4-step pipeline

We say that the **throughput** of this pipeline is 1 instruction every clock tick (imagine sitting at the “exit” of this pipeline and observing the instructions being completed)
The computer pipeline

In the “real” world, however ...

Q: What simplifications have we assumed, and what are the ramifications?
The computer pipeline

Each step of the pipeline does not take the same amount of time.

Q: What is the pipeline latency* for processing 1 instruction?

Q: What is the pipeline latency* for processing 2 instructions?

Q: What is the pipeline latency* for processing 3 instructions?

(in class exercise)

* Here “pipeline latency” means “total time that the pipeline is busy working”
The computer pipeline

\[ 6\text{ns} + 3\text{ns} + 5\text{ns} + 8\text{ns} = 22\text{ns} \]

Q: How many clock cycles does a second (queued) instruction require?
The computer pipeline

Remember that each step of the pipeline can be executing a single instruction only, so instruction 2 must “wait” for the IF stage until instruction 1 is done with the IF stage

2 instructions : $22 + 6 = 28$ns

Is this correct? Why or why not?
Instruction 1 and 2 are attempting to use the WB stage of the pipeline at the same time.
This is not allowed.

How do we fix this?
The computer pipeline

2 instructions : $20 + 2 + 8 = 30\text{ns}$

Instruction 2 is delayed by instruction 1

Q: What is the pipeline latency for processing 3 instructions?
The computer pipeline

Instruction 3 is delayed by instruction 2, which was delayed by instruction 1

Q: Is the maximum number of stalls that an instruction experiences always 1?
The computer pipeline

For the 4-stage pipeline shown above, and for 3 instructions queued ...

- What is the pipeline latency for instruction 1?
- What is the pipeline latency for instruction 2?
- What is the pipeline latency for instruction 3?
- How many stalls (ns) does instruction 1 experience?
- How many stalls (ns) does instruction 2 experience?
- How many stalls (ns) does instruction 3 experience?
- What is the pipeline throughput for the 3 instructions?

Assume that there is only one of each architecture component capable of performing the IF, ID, EX and WB stages

(In-class exercise, on-the-board explanation)
Stalls
Data and structure hazards
instruction dependencies