Where is the delay?

1) Instruction fetch
2) Decode and register read
3) ALU (compute)
4) Use memory
5) Register write

Candidate for a 5-stage pipeline

An analogous example: car wash

Soak
Brush
Rinse
Dry
# Pipelining the car wash

<table>
<thead>
<tr>
<th>Soak</th>
<th>Brush</th>
<th>Rinse</th>
<th>Dry</th>
</tr>
</thead>
</table>

---

# Pipelining the car wash

<table>
<thead>
<tr>
<th>Soak</th>
<th>Brush</th>
<th>Rinse</th>
<th>Dry</th>
</tr>
</thead>
</table>
What are the requirements for efficient pipeline operation?

What is the effect on throughput and delay?
What do we need to add to actually split the datapath into stages?

Inter-stage buffers

- Add buffers between consecutive stages (store data at the end of a clock cycle, and use it at the beginning of the next cycle)
- The cycle time should be long enough to allow the signals in each stage to propagate from the input to the output buffers of the stage
Pipelined execution of an instruction

Cycle 1

Cycle 2
Pipelined execution of an instruction

Cycle 3

Cycle 4
Pipelined execution of an instruction

Cycle 5

Pipelined execution of consecutive instructions

Cycle 2
Pipelined execution of consecutive instructions

Cycle 3

Instruction memory
IF/ID
Registers
ALU
Data memory
ID/EX
EX/MEM
MEM/WB

lw
add

Pipelined execution of consecutive instructions

Cycle 4

Instruction memory
IF/ID
Registers
ALU
Data memory
ID/EX
EX/MEM
MEM/WB

lw
add
Reconsider the Pipelined datapath (slide 25)

Can you find a problem with the above datapath?
What instructions can we execute to manifest the problem?

Corrected datapath
We have 5 stages, each having its own control signals

- Pass control signals along just like the data

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Execution/Address Calculation stage control lines</th>
<th>Memory access stage control lines</th>
<th>Write-back stage control lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-format</td>
<td>Reg</td>
<td>ALU Op1</td>
<td>ALU Op0</td>
</tr>
<tr>
<td>lw</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sw</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>beq</td>
<td>X</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Pipeline Control

- We have 5 stages, each having its own control signals.
- Pass control signals along just like the data.
Pipeline control

Datapath with Control
representation of pipelining

<table>
<thead>
<tr>
<th>Cycle 1</th>
<th>IF stage</th>
<th>ID stage</th>
<th>EX stage</th>
<th>MEM stage</th>
<th>WB stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle 2</td>
<td>add $4, $5, $6</td>
<td>add $4, $5, $6</td>
<td>add $4, $5, $6</td>
<td>add $4, $5, $6</td>
<td>add $4, $5, $6</td>
</tr>
<tr>
<td>Cycle 3</td>
<td>and $1, $2, $3</td>
<td>and $1, $2, $3</td>
<td>and $1, $2, $3</td>
<td>and $1, $2, $3</td>
<td>and $1, $2, $3</td>
</tr>
<tr>
<td>Cycle 4</td>
<td>lw $3, 300($0)</td>
<td>lw $3, 300($0)</td>
<td>lw $3, 300($0)</td>
<td>lw $3, 300($0)</td>
<td>lw $3, 300($0)</td>
</tr>
<tr>
<td>Cycle 5</td>
<td>sub $7, $8, $9</td>
<td>sub $7, $8, $9</td>
<td>sub $7, $8, $9</td>
<td>sub $7, $8, $9</td>
<td>sub $7, $8, $9</td>
</tr>
</tbody>
</table>

Can help with answering questions like:
- how many cycles does it take to execute this code?
- what is the ALU doing during cycle 4?

**Pipeline throughput**: Because the pipeline completes the execution of one instruction every clock cycle (after an initial delay to fill up the pipeline), then the CPI = 1.