

# CSE 161 – Design and Architecture of Computer Systems

## Chapter Six

### Simple Questions

□ How many cycles will it take to execute this code?

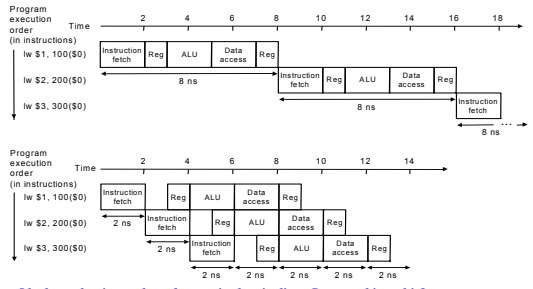
```
lw $t2, 0($t3)
lw $t3, 4($t3)
beq $t2, $t3, Label ← #assume not
add $t5, $t2, $t3
sw $t5, 8($t3)
Label:
...
```

□ What is going on during the 8th cycle of execution?  
 □ In what cycle does the actual addition of \$t2 and \$t3 takes place?



### Pipelining

□ Improve performance by increasing instruction throughput

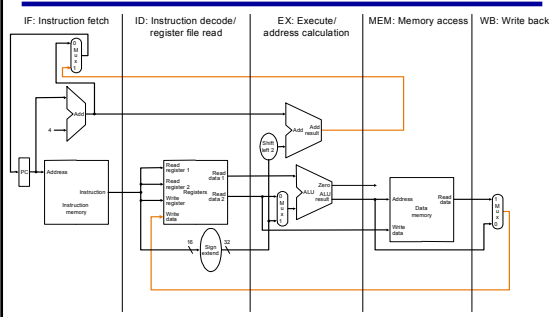


*Ideal speedup is number of stages in the pipeline. Do we achieve this?*

### Pipelining

- What makes it easy
  - all instructions are the same length
  - just a few instruction formats
  - memory operands appear only in loads and stores
- What makes it hard?
  - structural hazards: suppose we had only one memory
  - control hazards: need to worry about branch instructions
  - data hazards: an instruction depends on a previous instruction
- We'll build a simple pipeline and look at these issues
- We'll talk about modern processors and what really makes it hard:
  - exception handling
  - trying to improve performance with out-of-order execution, etc.

### Basic Idea



□ What do we need to add to actually split the datapath into stages?

### Pipelined Datapath

