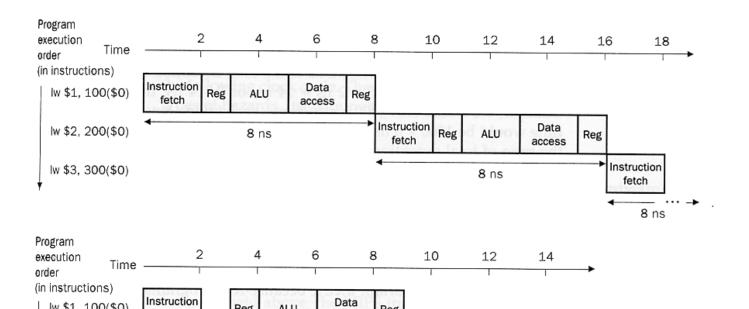
Pipelining

- What is it?
- Why does it improve performance?
- Basic data path
- Control
- What's so hard about this?



Reg

Data

access

ALU

2 ns

Reg

Data

access

2 ns

Reg

2 ns

access

ALU

2 ns

Reg

lw \$1, 100(\$0)

lw \$2, 200(\$0)

lw \$3, 300(\$0)

Reg

Instruction

fetch

2 ns

fetch

2 ns

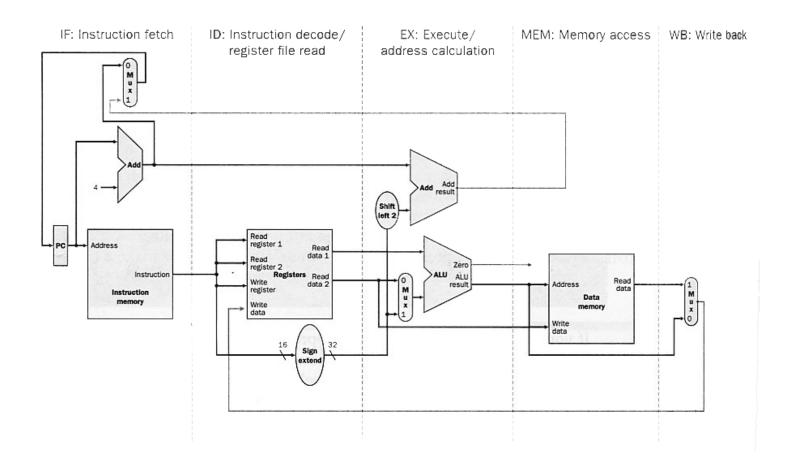
ALU

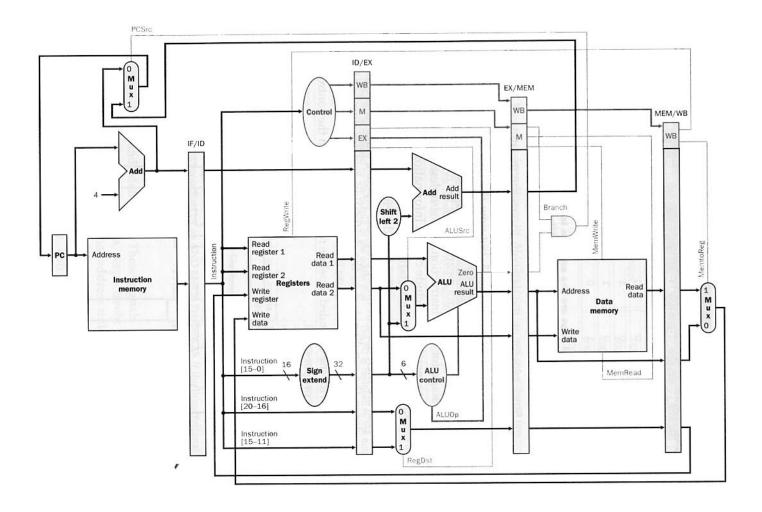
Instruction

fetch

2 ns

Reg





- Structural Hazards
 - A component is required to do more than one thing at a time
- Control Hazards
 - Hey, I wanted to branch!

```
beq $t0, $t1, skip
addi $t0, 1
skip: Iw $t1, N
```

- Data Hazards
 - dependencies

```
read-after-write (RAW)

addi $t0, $t0, 1

add $t2, $t0, $t1

write-after-read (WAR)

write-after-write (WAW)
```