

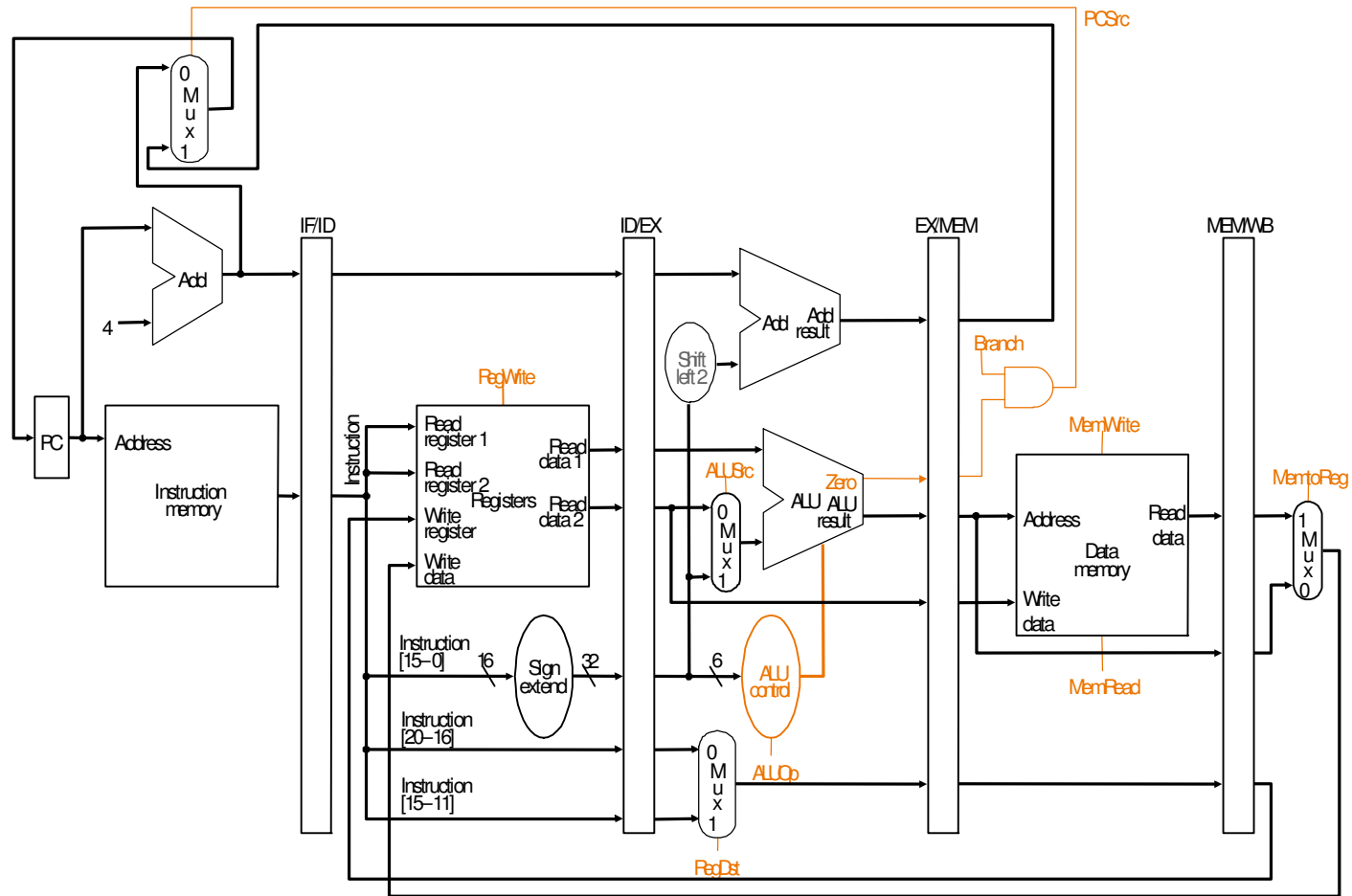
---

# **Chapter Six**

## **Pipelining**

### **Controle**

# Pipeline Control



- campo function para controle da ALU (EX)
- sinais de controle: os mesmos do capítulo 5, ciclo único
- PC e registradores de pipeline escritos em todos ciclos: não é necessário controle

# Pipeline control

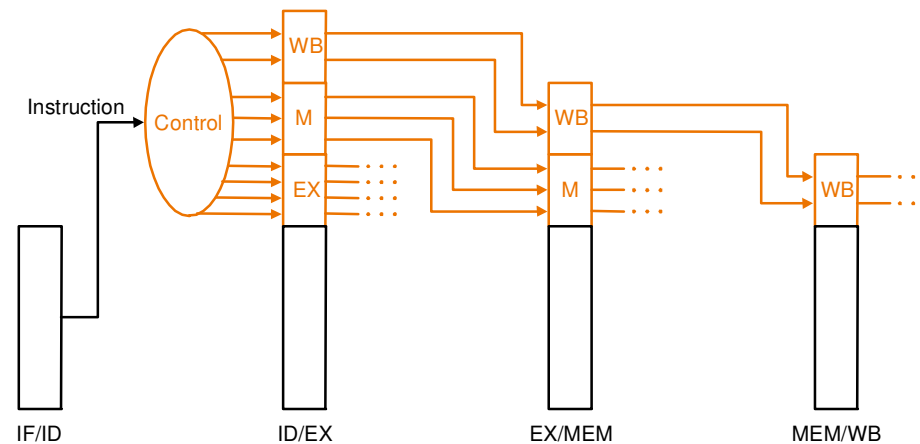
---

- **We have 5 stages. What needs to be controlled in each stage?**
  - **Instruction Fetch and PC Increment**
  - **Instruction Decode / Register Fetch**
  - **Execution**
  - **Memory Stage**
  - **Write Back**
- **How would control be handled in an automobile plant?**
  - **a fancy control center telling everyone what to do?**
  - **should we use a finite state machine?**

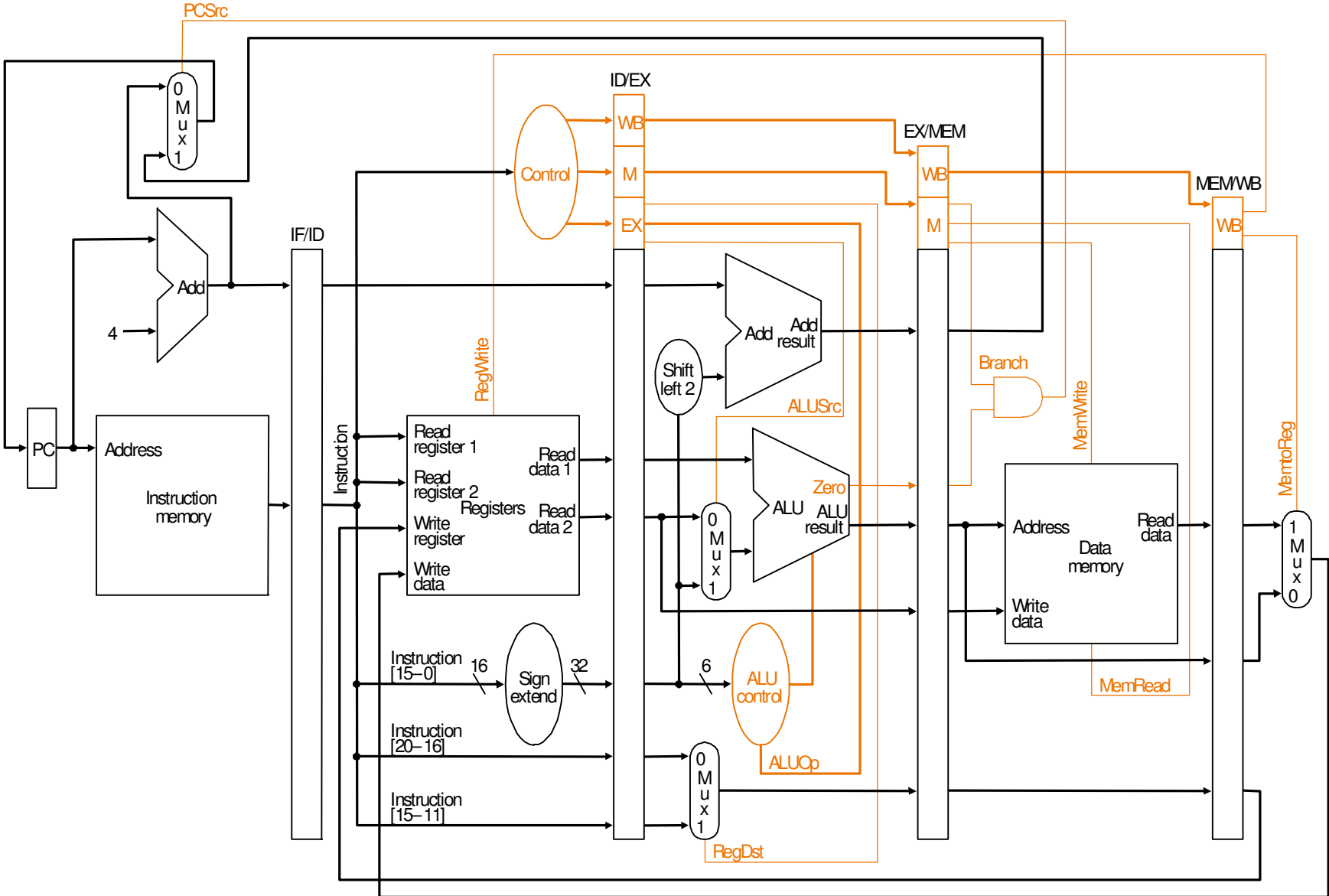
# Pipeline Control

- Pass control signals along just like the data

Instruction	Execution/Address Calculation stage control lines				Memory access stage control lines			stage control lines	
	Reg Dst	ALU Op1	ALU Op0	ALU Src	Branch	Mem Read	Mem Write	Reg write	Mem to Reg
R-format	1	1	0	0	0	0	0	1	0
lw	0	0	0	1	0	1	0	1	1
sw	X	0	0	1	0	0	1	0	X
beq	X	0	1	0	1	0	0	0	X



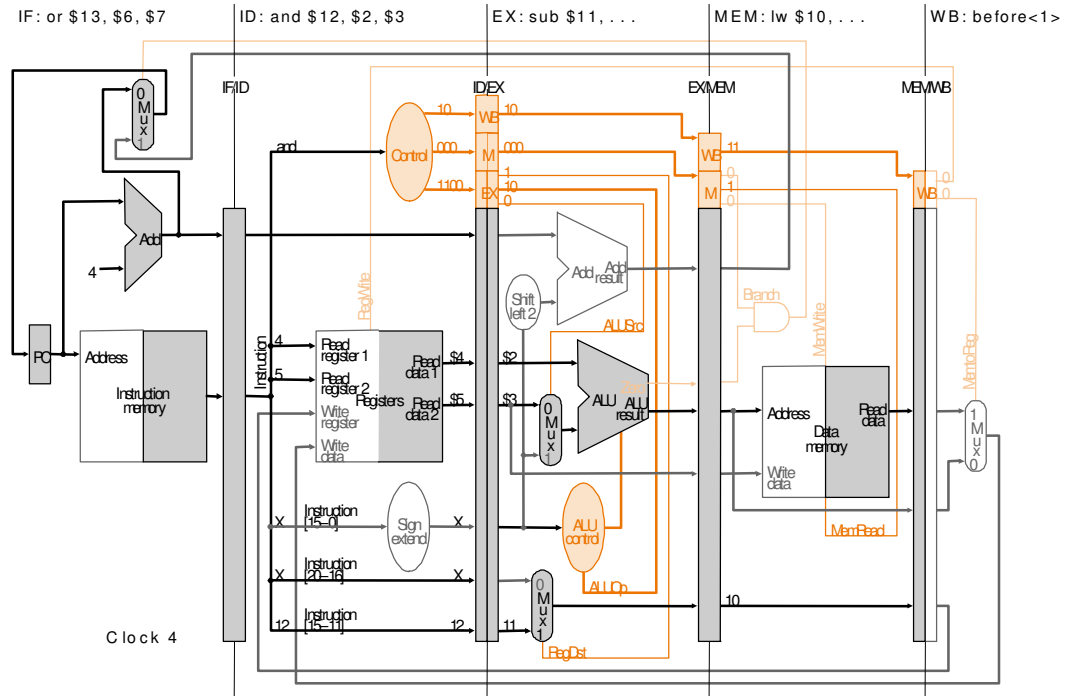
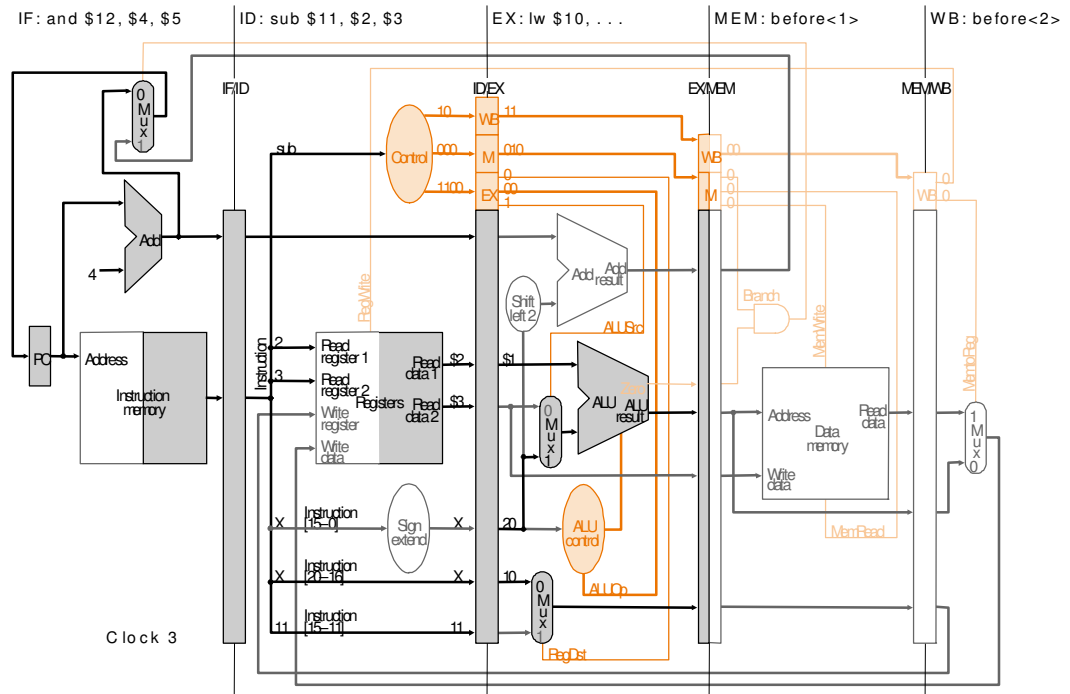
# Datapath with Control





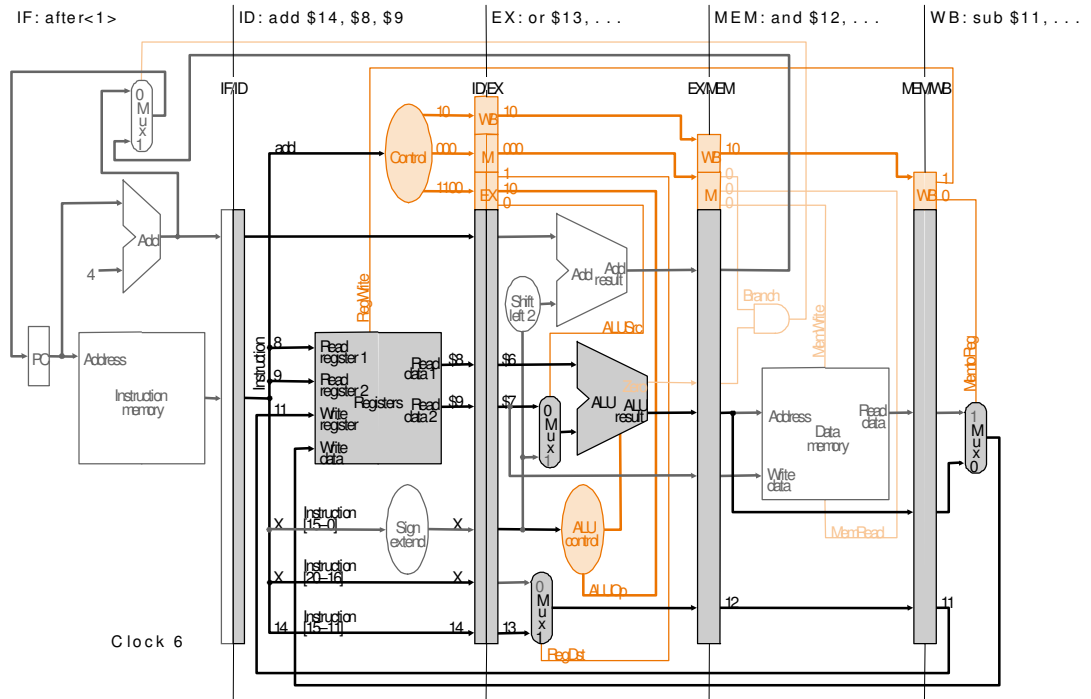
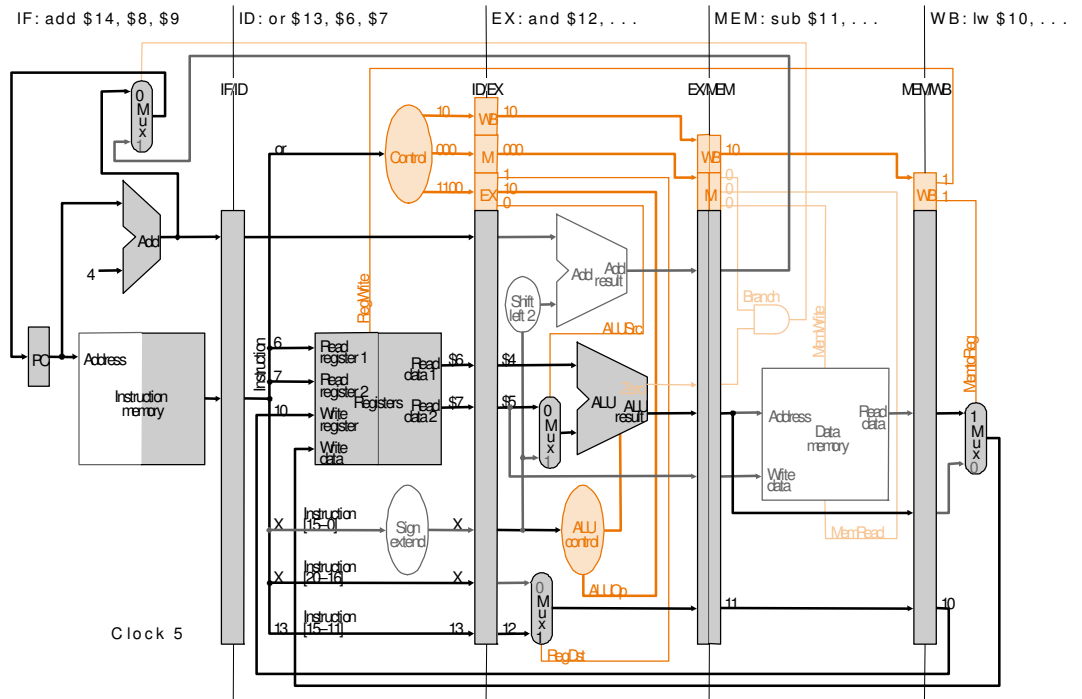
# Exemplo (pag. 471)

```
lw    $10, 20($1)
sub   $11, $2, $3
and   $12, $4, $5
or    $13, $6, $7
add   $14, $8, $9
```



# Exemplo (pag. 471)

```
lw    $10, 20($1)
sub   $11, $2, $3
and   $12, $4, $5
or    $13, $6, $7
add   $14, $8, $9
```





# Exemplo (pag. 471)

```
lw    $10, 20($1)
sub   $11, $2, $3
and   $12, $4, $5
or    $13, $6, $7
add   $14, $8, $9
```

