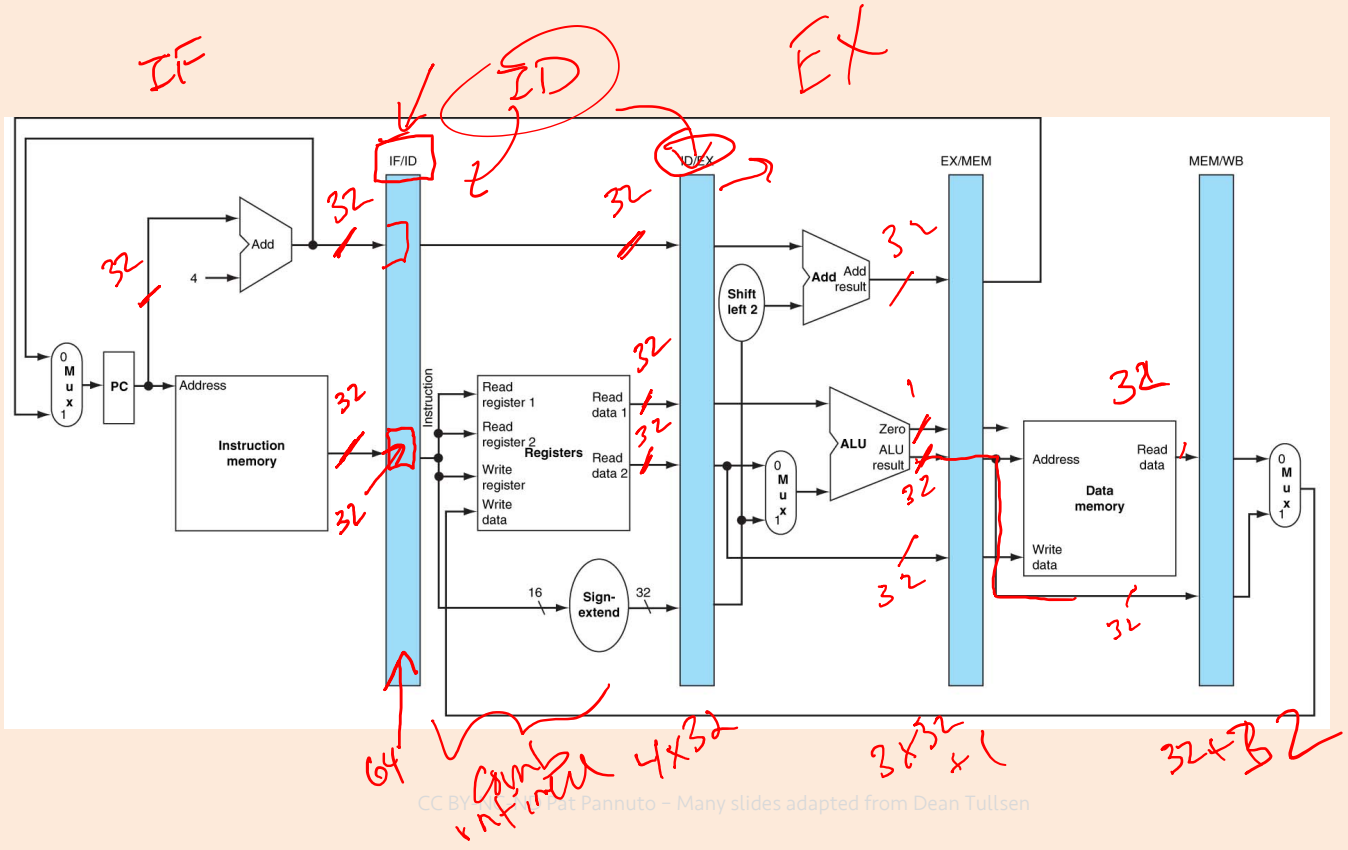
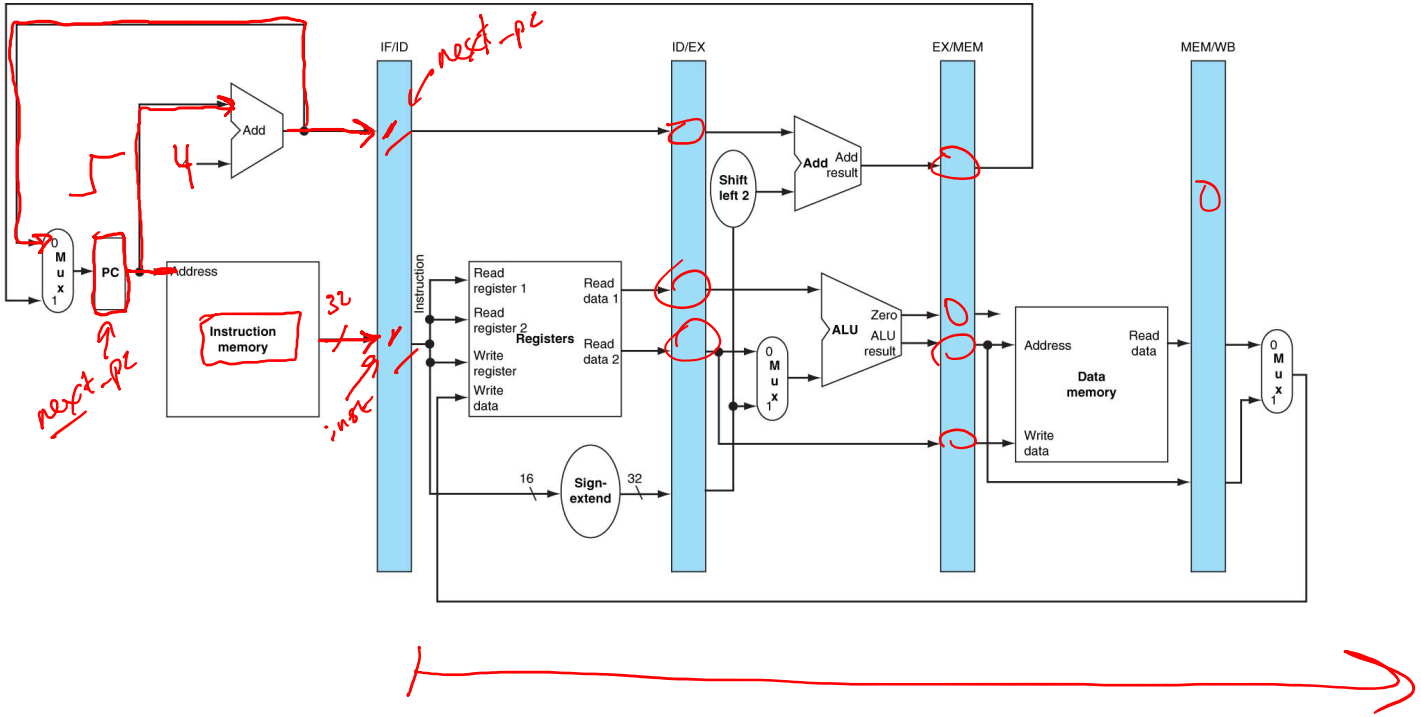
Poll Q: How many D flip flops are in this pipeline?



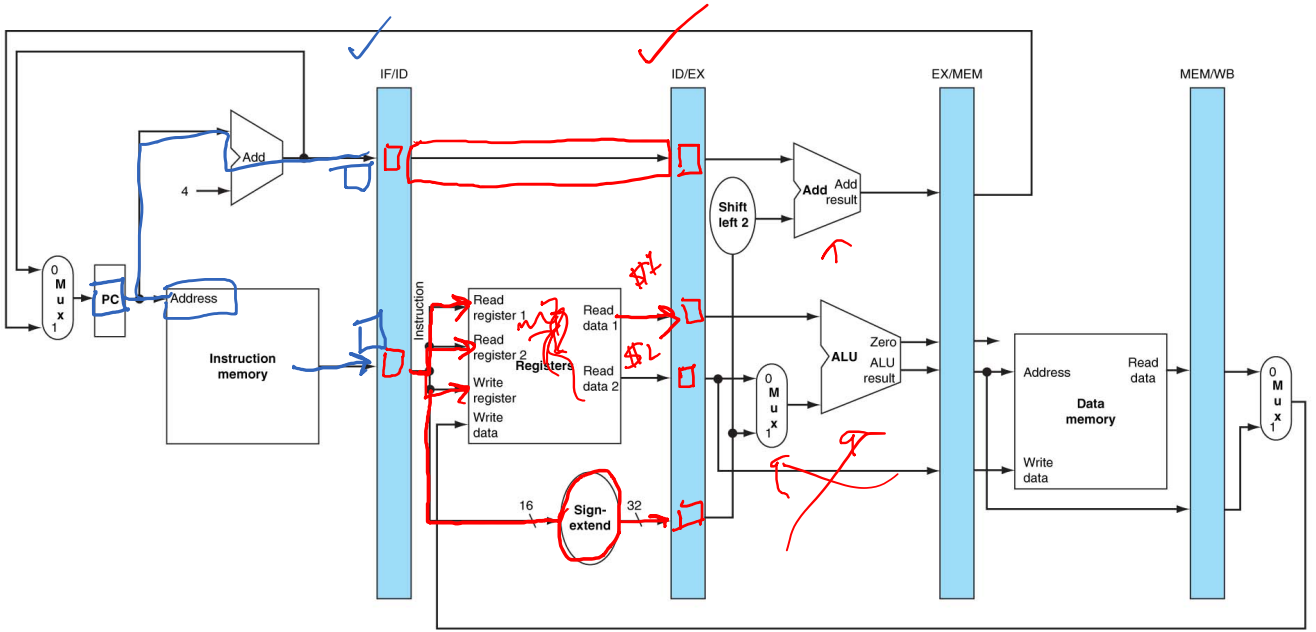
The Pipeline in Execution

add $10, $1, $2 | Instruction Decode/ Register Fetch | Execute/ Address Calculation | Memory Access | Write Back

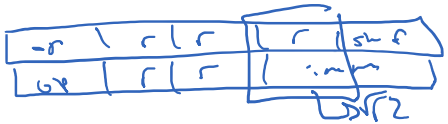


The Pipeline in Execution

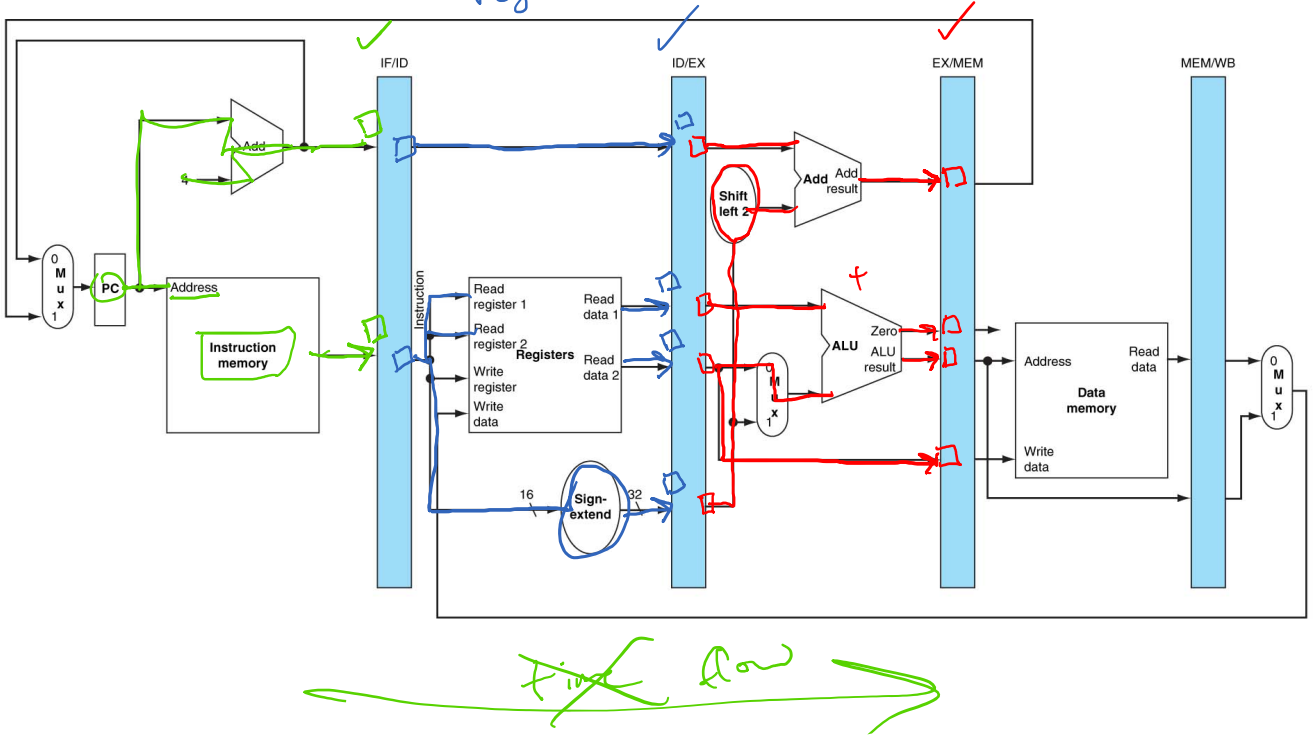
lw $12, 1000($4) **add $10, $1, $2** Execute/ Address Calculation Memory Access Write Back



The Pipeline in Execution

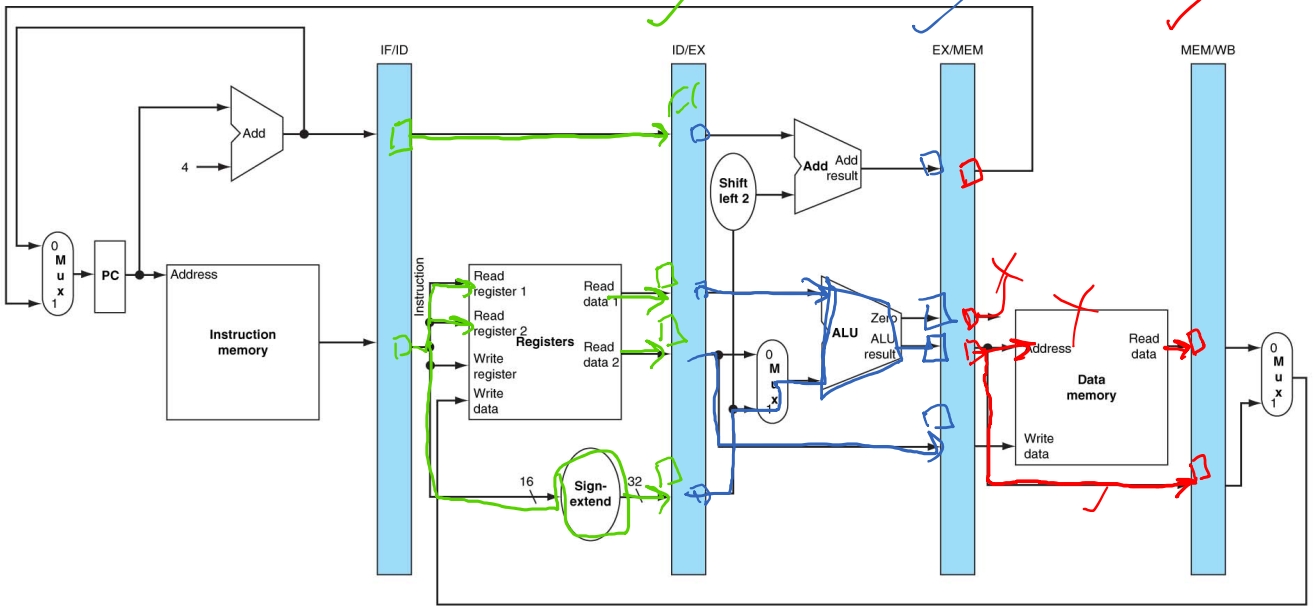


sub $15, $4, $1 | lw $12, 1000($4) | add $10, $1, $2 | Memory Access | Write Back



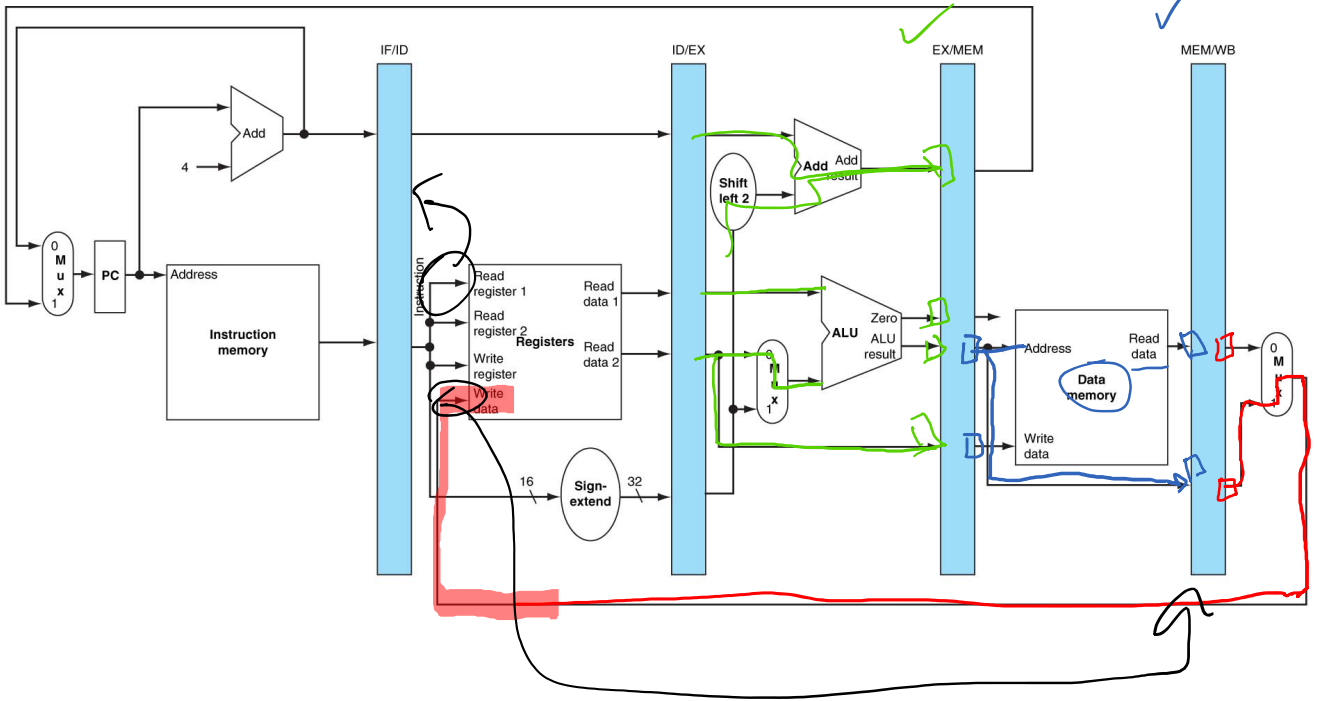
The Pipeline in Execution

Instruction Fetch | **sub $15, $4, $1** | **lw $12, 1000($4)** | **add $10, $1, $2** | Write Back



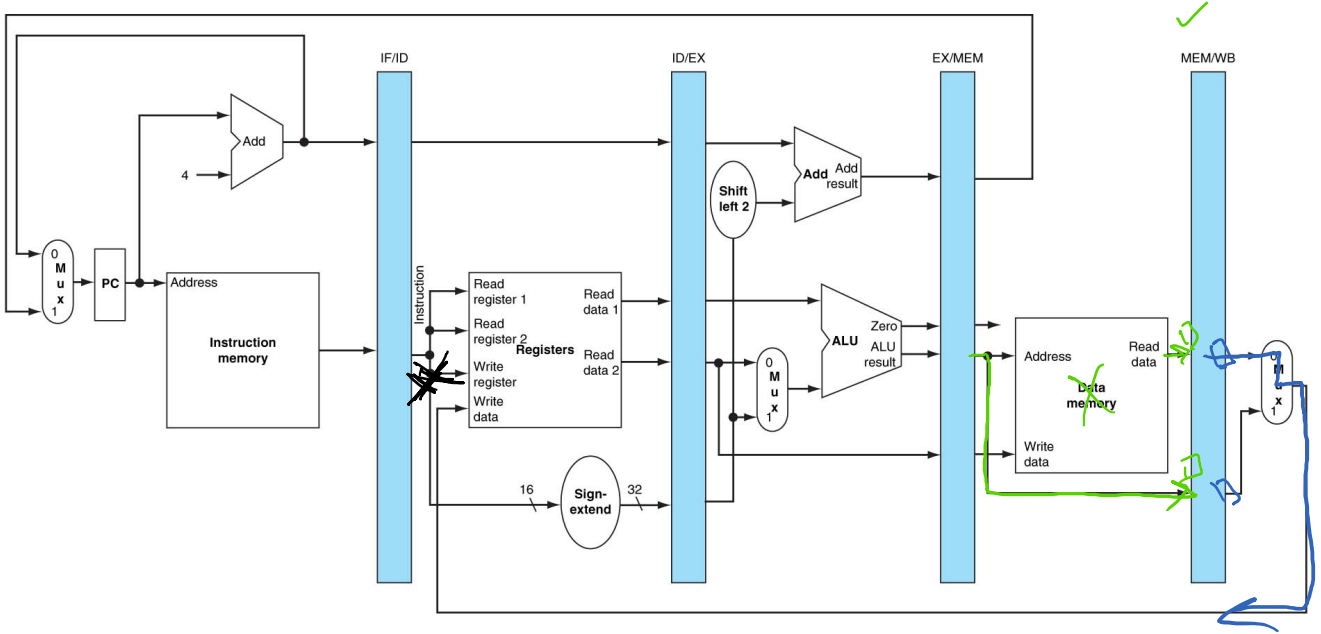
The Pipeline in Execution

Instruction Fetch | Instruction Decode/ Register Fetch | **sub $15, $4, $1** | **lw $12, 1000($4)** | **add $10, $1, $2**

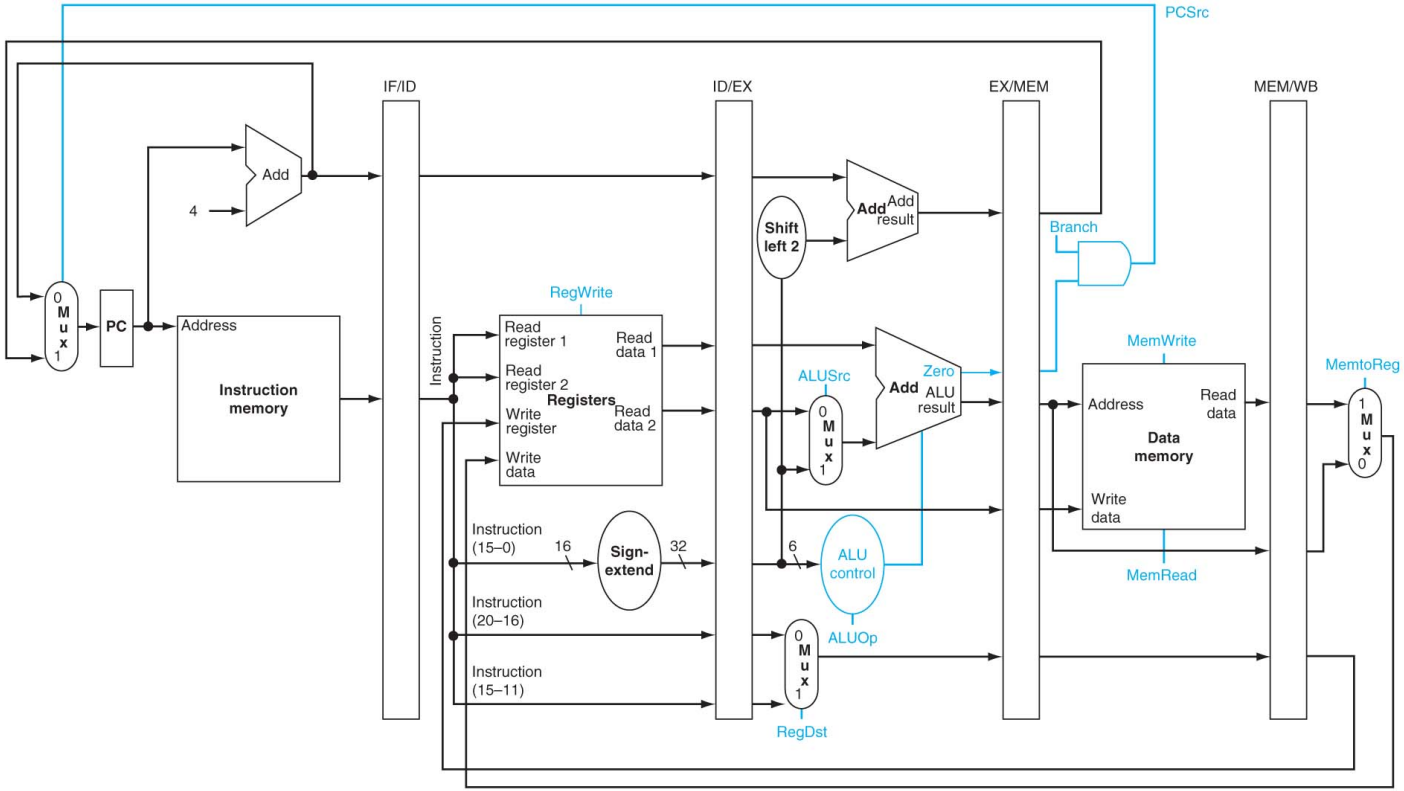


The Pipeline in Execution

Instruction Fetch | Instruction Decode/ Register Fetch | Execute/ Address Calculation | **sub $15, $4, $1** | **lw $12, 1000($4)**



The Pipeline, with controls But....



Pipelined Control

- I told you multicycle control was messy. We would expect pipelined control to be messier.

Pipelined Control

- I told you multicycle control was messy. We would expect pipelined control to be messier.
  - Why?

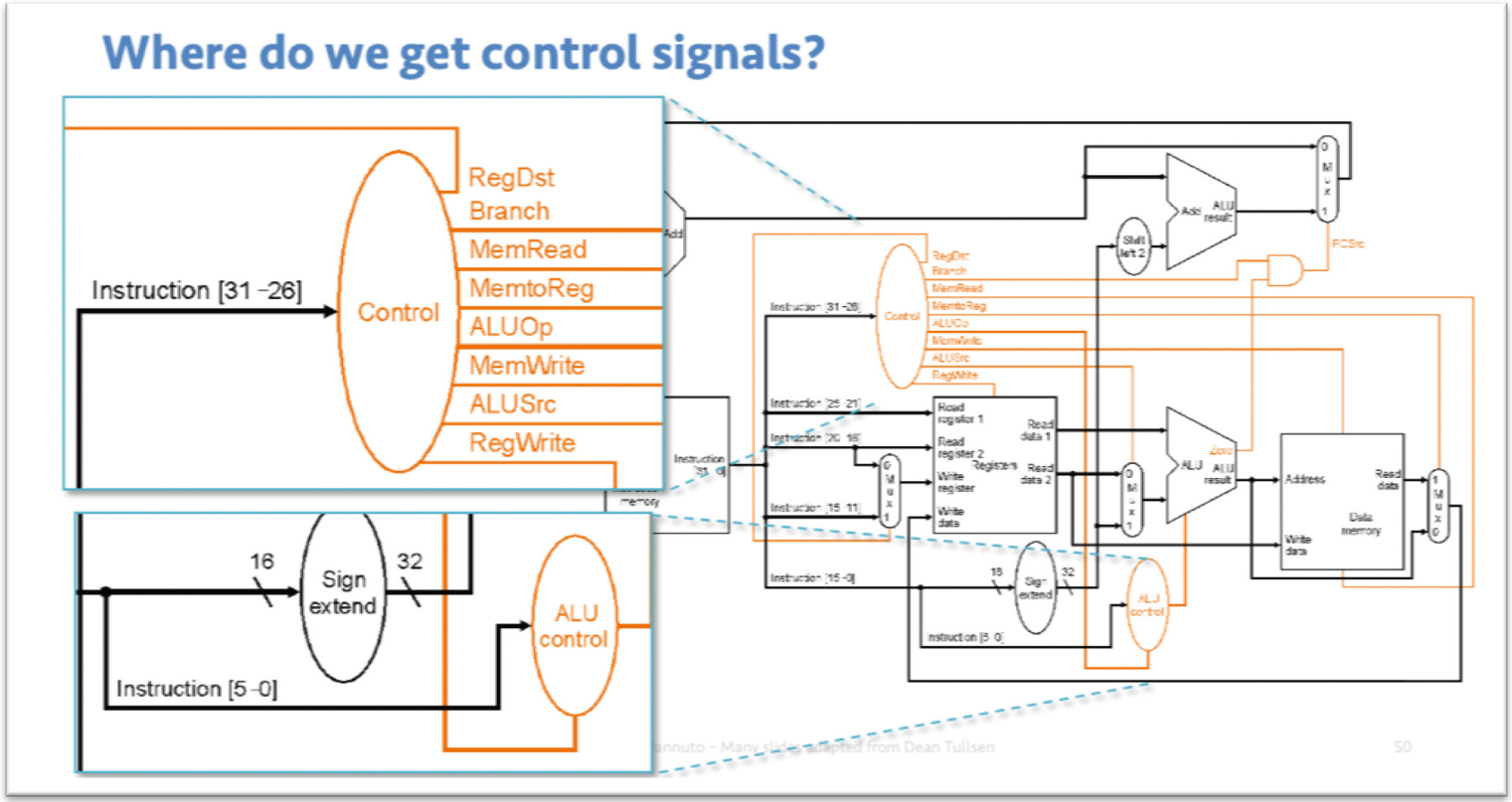
Pipelined Control

- I told you multicycle control was messy. We would expect pipelined control to be messier.
  - Why?
- But it turns out we can do it with just…

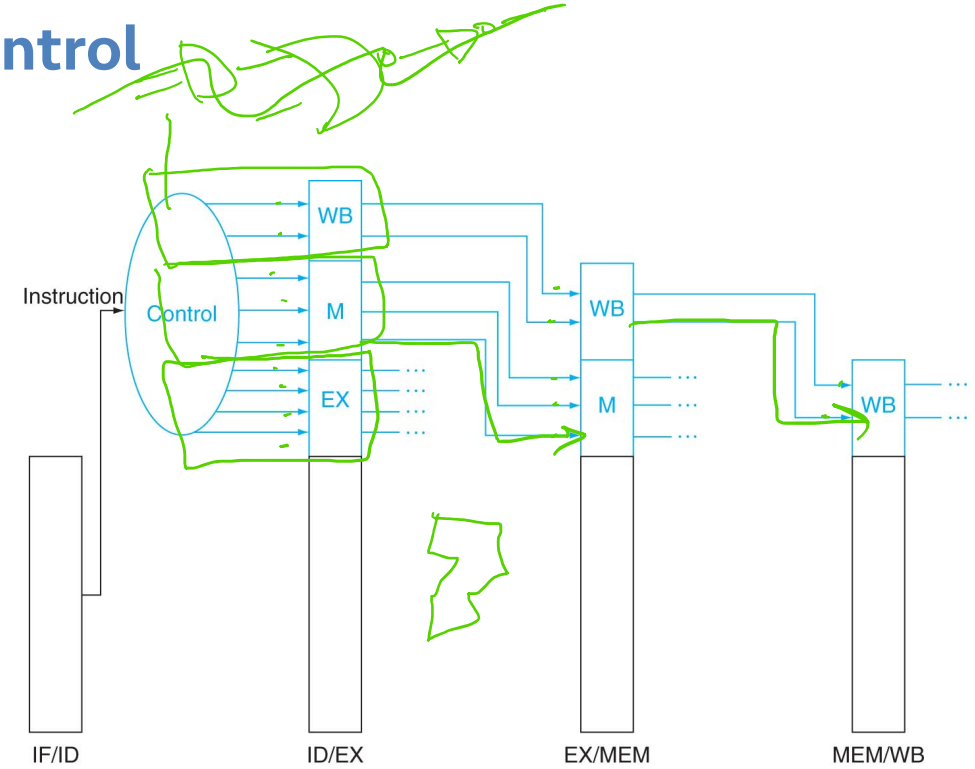
Pipelined Control

- I told you multicycle control was messy. We would expect pipelined control to be messier.
  - Why?
- But it turns out we can do it with just…
- Combinational logic!
  - Signals generated once
  - Follow instruction through the pipeline

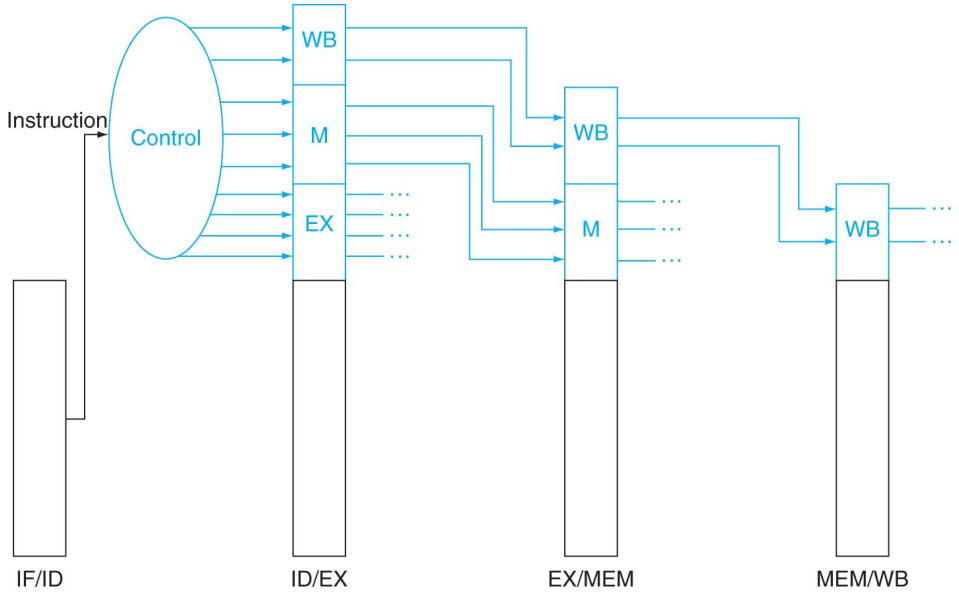
Recall: Control signals in the single-cycle machine



Pipelined Control

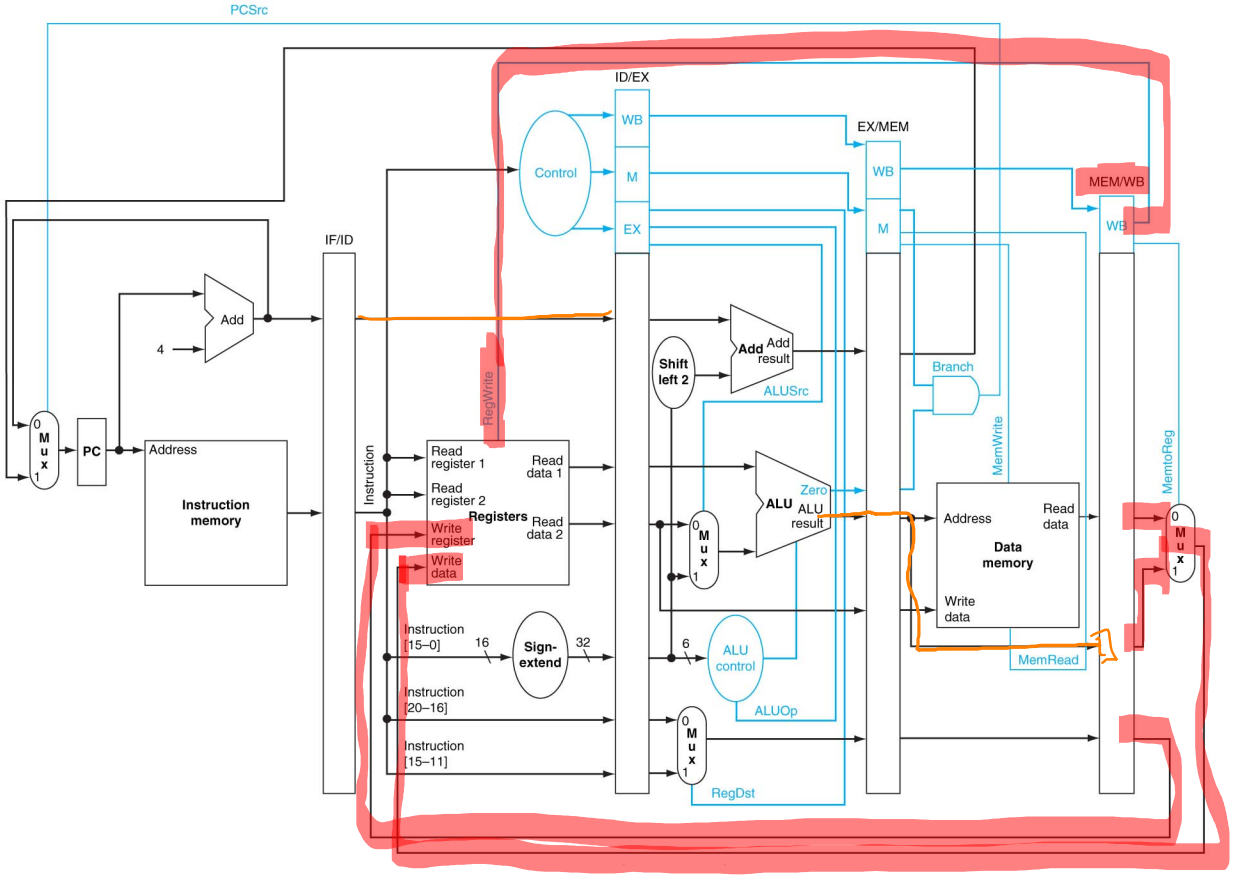


Pipelined Control



So, really it is combinational logic and some registers to propagate the signals to the right stage.

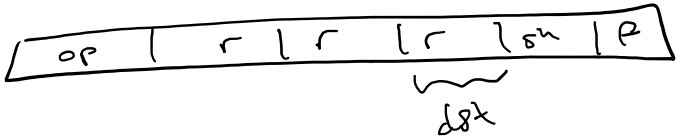
The Pipeline with Control Logic



Pipelined Control Signals

	Execution Stage Control Lines				Memory Stage Control Lines			Write Back Stage Control Lines	
Instruction	RegDst	ALUOp1	ALUOp0	ALUSrc	Branch	MemRead	MemWrite	RegWrite	MemtoReg
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

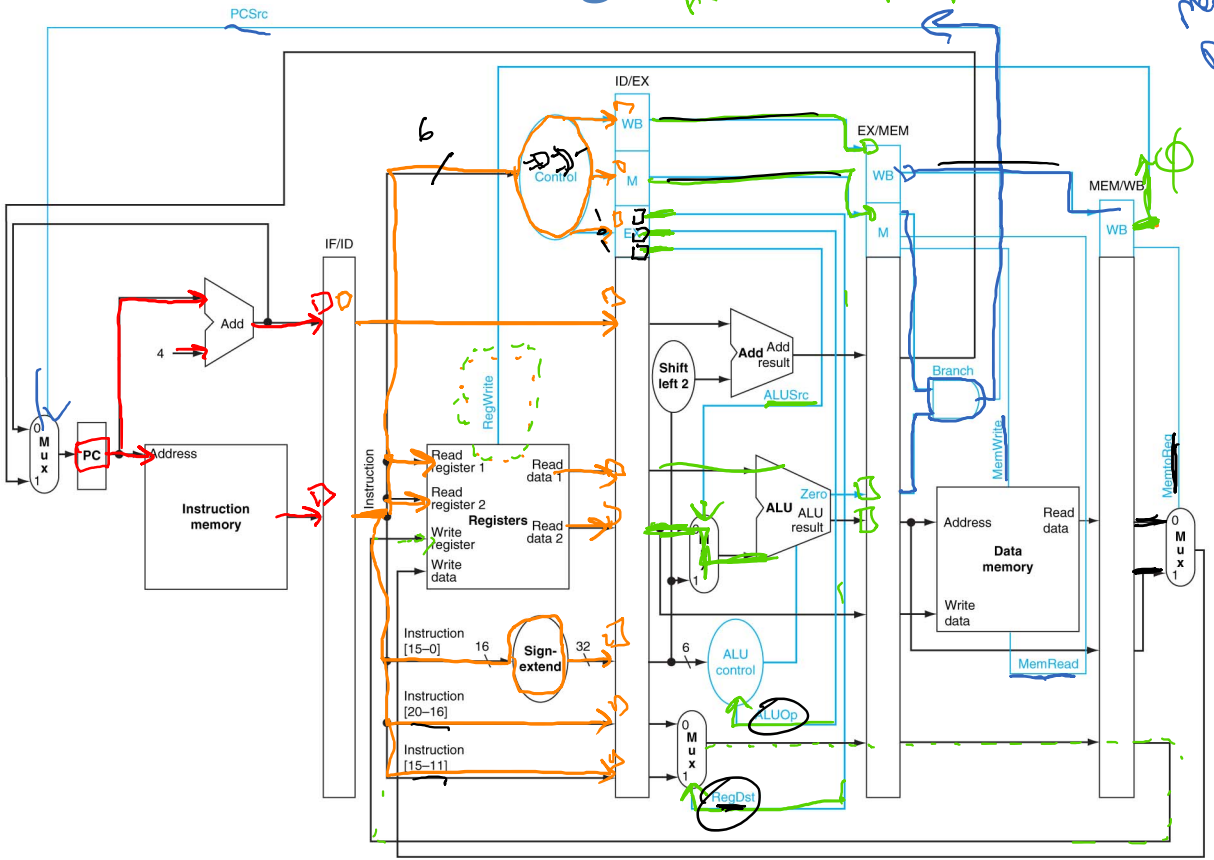
Pipelined Control Signals



	Execution Stage Control Lines				Memory Stage Control Lines			Write Back Stage Control Lines	
Instruction	RegDst	ALUOp1	ALUOp0	ALUSrc	Branch	MemRead	MemWrite	RegWrite	MemtoReg
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

Let's just do one.

The Pipeline with Control Logic



Is it really that easy?

- What happens when…

  add $3, $10, $11

  lw $8, 1000($3)

  sub $11, $8, $7