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(S4) \quad \text{add } S10, S1, S2 \quad \text{Execute/Address Calculation} \quad \text{Memory Access} \quad \text{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

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**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Execution Stage Control Lines</th>
<th>Memory Stage Control Lines</th>
<th>Write Back Stage Control Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RegDst</td>
<td>ALUOp1</td>
<td>ALUOp0</td>
</tr>
<tr>
<td>R-Format</td>
<td>i</td>
<td>i</td>
<td>0</td>
</tr>
<tr>
<td>lw</td>
<td>0</td>
<td>0</td>
<td>0</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>
Pipelined Control Signals

<table>
<thead>
<tr>
<th>Instruction</th>
<th>RegDst</th>
<th>ALUOp1</th>
<th>ALUOp0</th>
<th>ALUSrc</th>
<th>Branch</th>
<th>MemRead</th>
<th>MemWrite</th>
<th>RegWrite</th>
<th>MemtoReg</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-Format</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>lw</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sw</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>beq</td>
<td>x</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
</tbody>
</table>

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