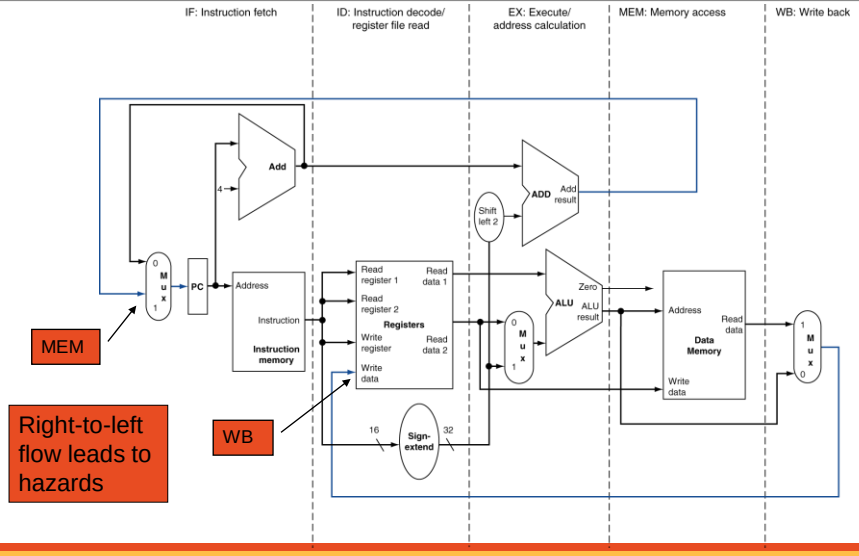
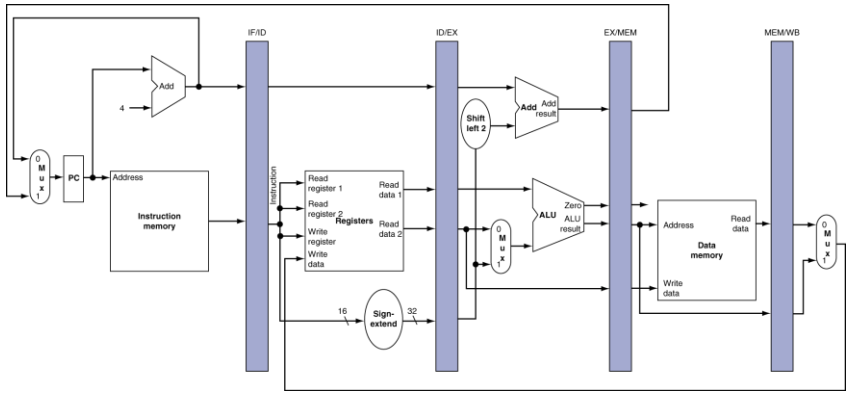
MIPS Pipelined Datapath



CSULB

Pipeline registers

- Need registers between stages
  - To hold information produced in previous cycle



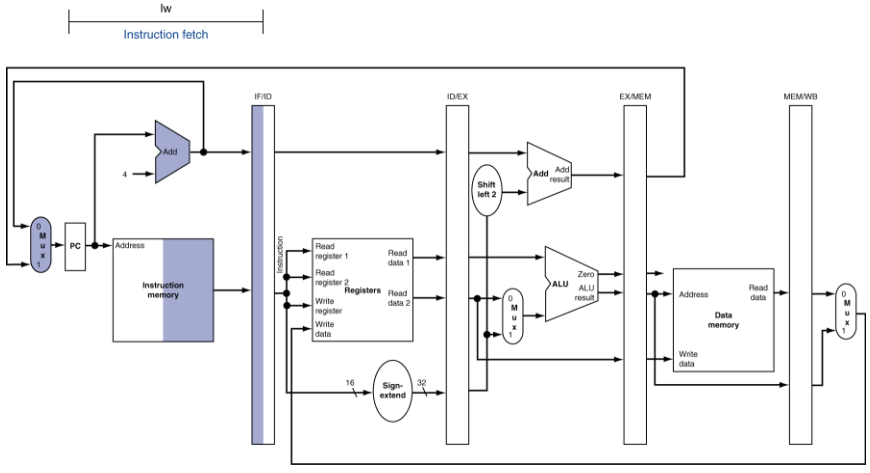
CSULB

Pipeline Operation

- Cycle-by-cycle flow of instructions through the pipelined datapath
  - "Single-clock-cycle" pipeline diagram
    - Shows pipeline usage in a single cycle
    - Highlight resources used
  - c.f. "multi-clock-cycle" diagram
    - Graph of operation over time
- We'll look at "single-clock-cycle" diagrams for load & store

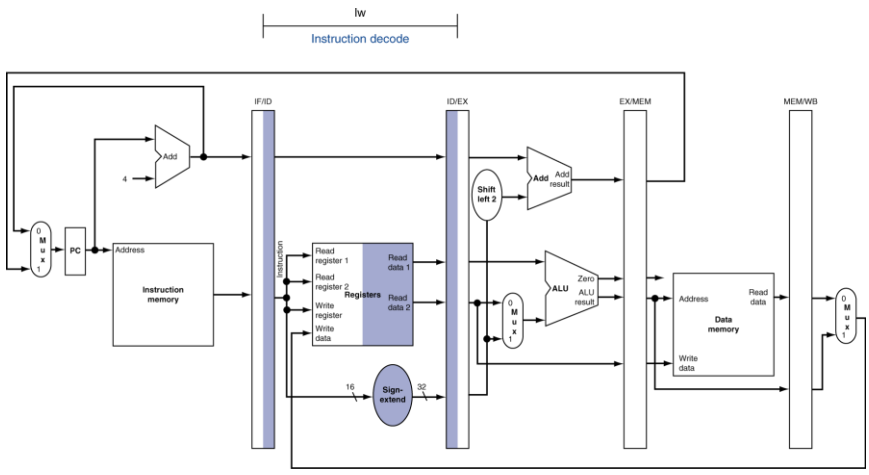
IF for Load, Store, …

```
lw $t0, 32($t1)
```



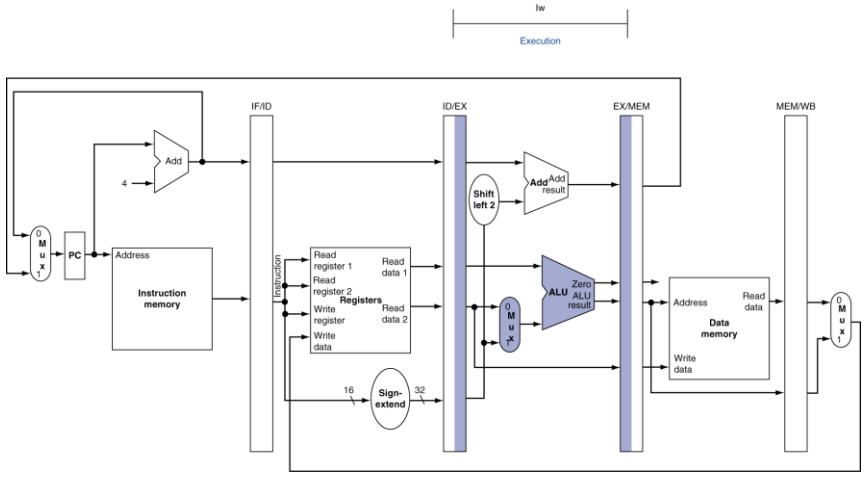
ID for Load, Store, …

```
lw $t0, 32($t1)
```



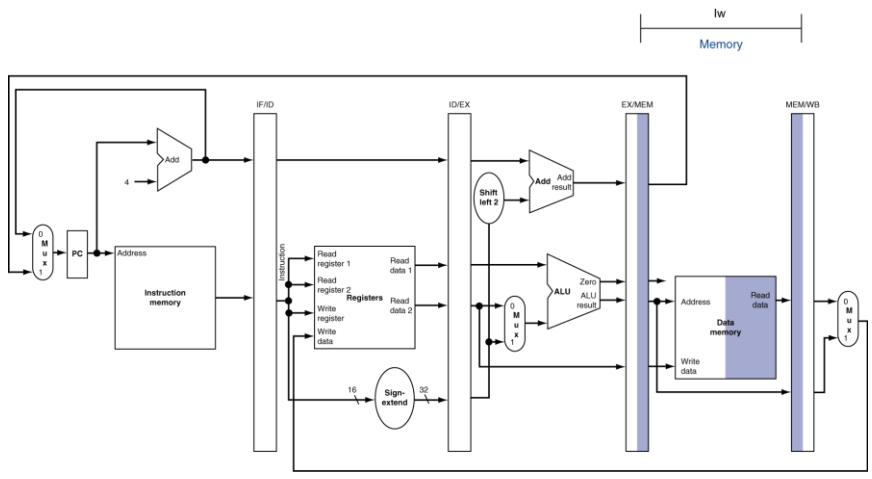
EX for Load

```
lw $t0, 32($t1)
```



MEM for Load

```
lw $t0, 32($t1)
```

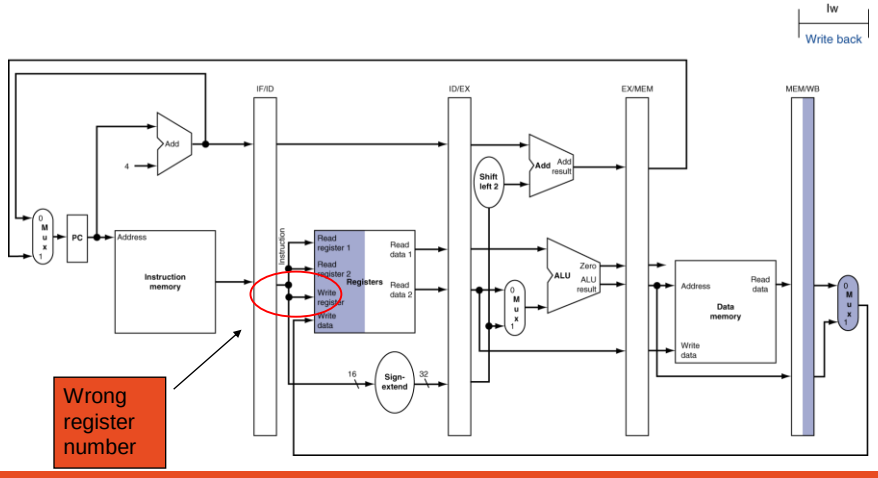


CSULB

WB for Load

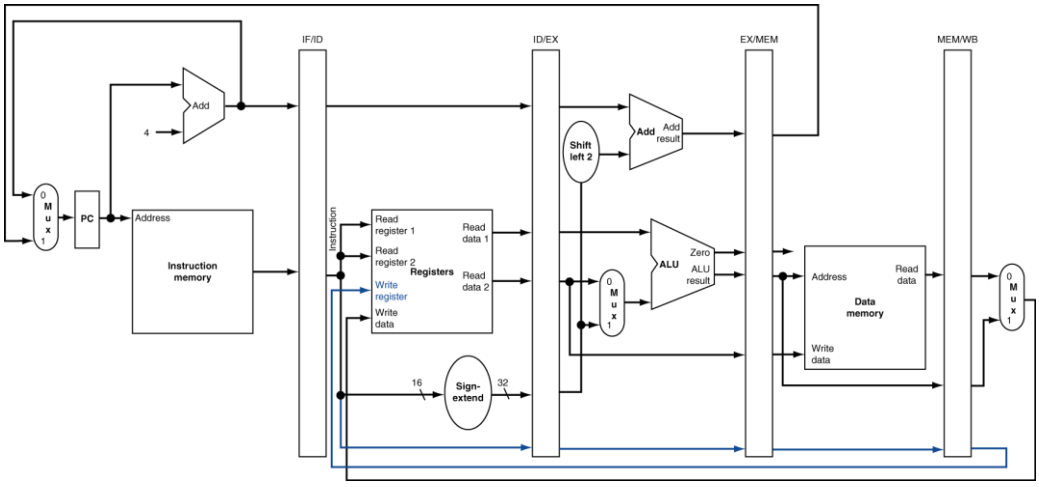
```
add $t2, $t3, $t4
```

```
lw $t0, 32($t1)
```



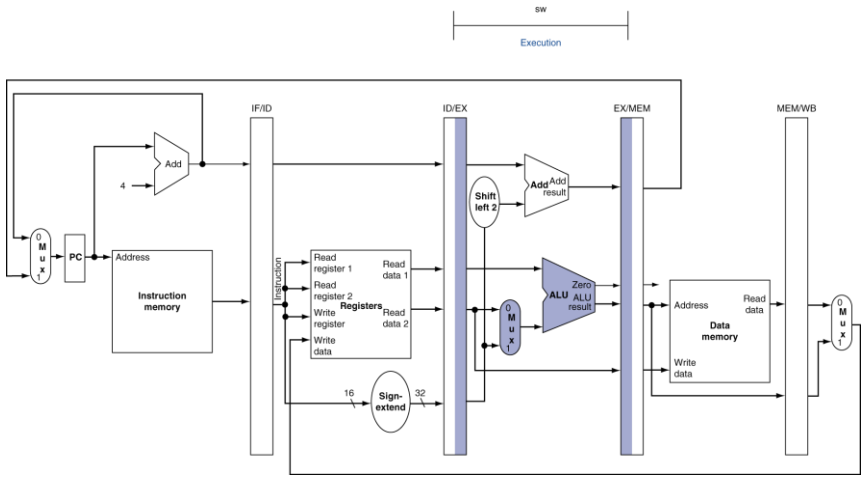
CSULB

Corrected Datapath for Load



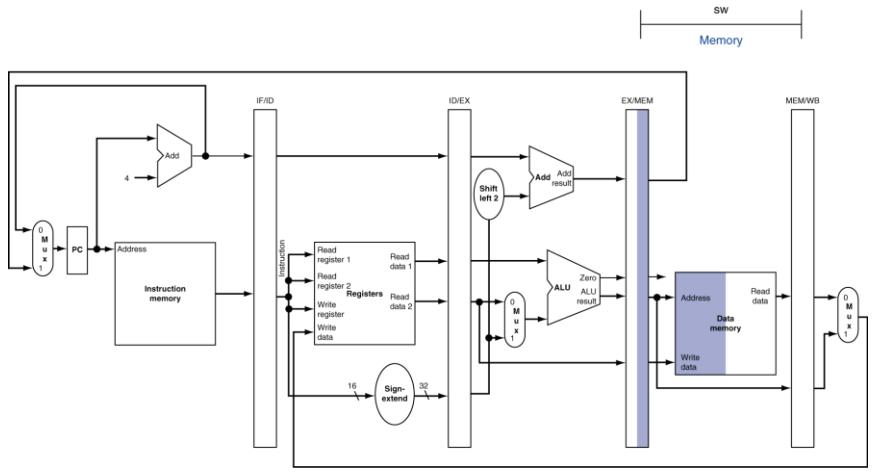
EX for Store

```
sw $t0, 32($t1)
```



MEM for Store

```
sw $t0, 32($t1)
```

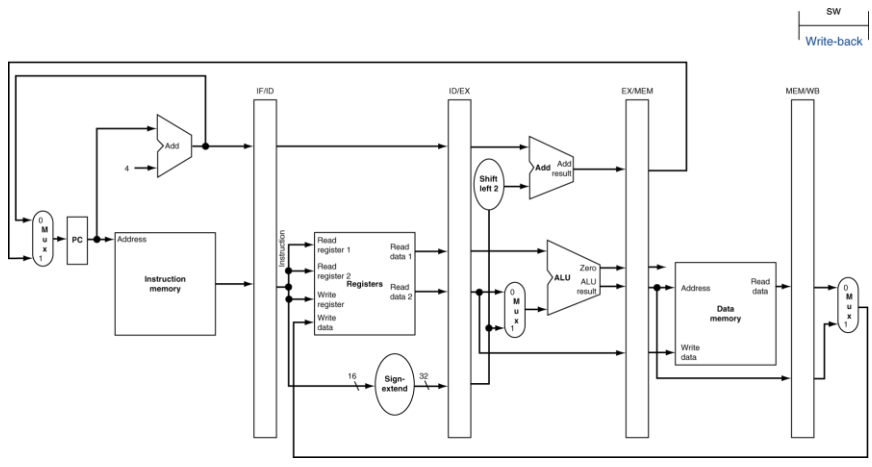


CSULB

WB for Store

```
add $t2, $t3, $t4
```

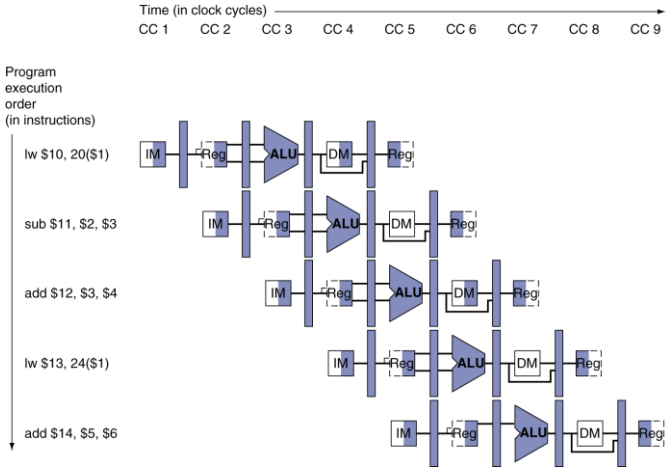
```
sw $t0, 32($t1)
```



CSULB

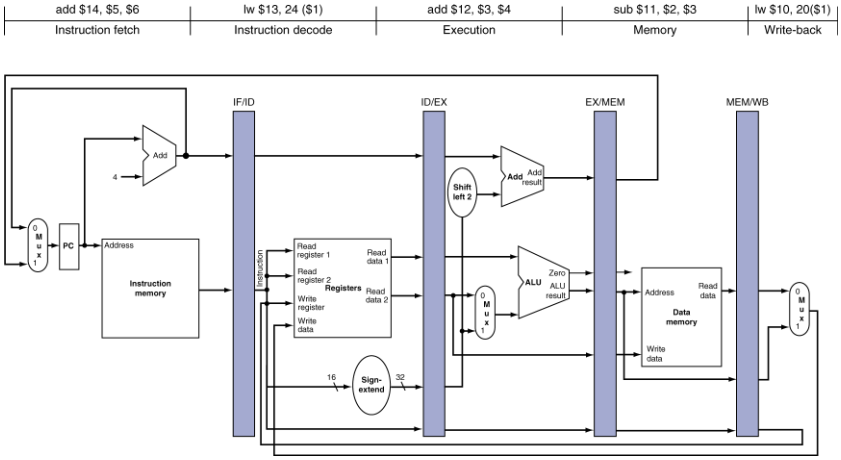
Multi-Cycle Pipeline Diagram

- Form showing resource usage

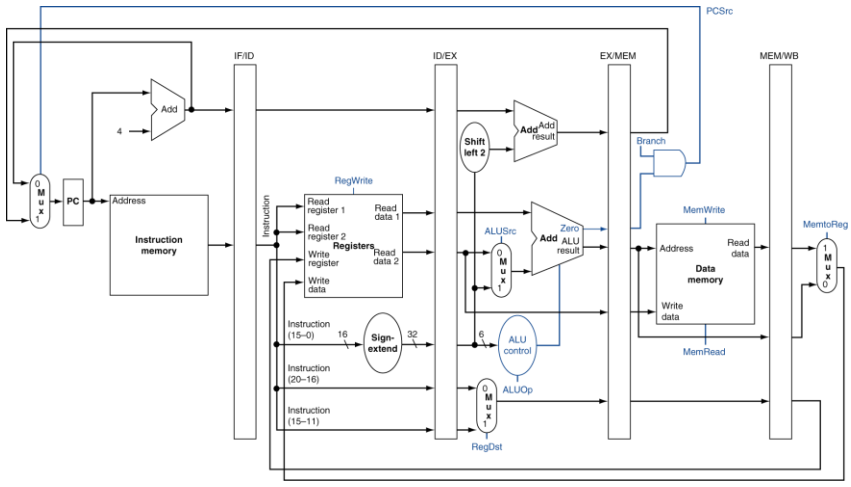


Single-Cycle Pipeline Diagram

- State of pipeline in a given cycle



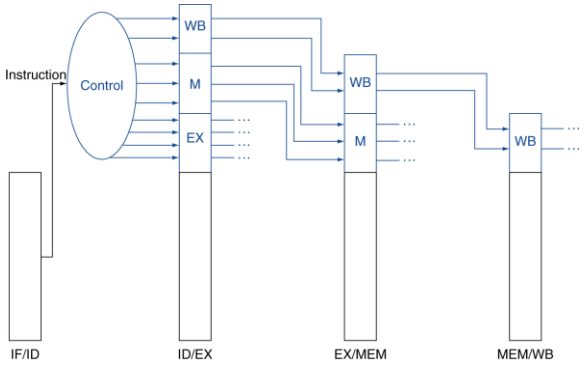
Pipelined Control (Simplified)



CSULB

Pipelined Control

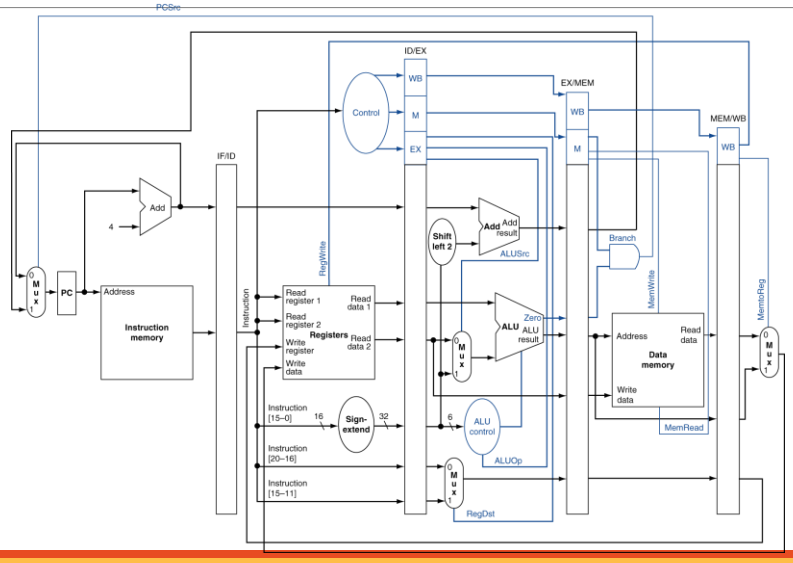
- Control signals derived from instruction
  - As in single-cycle implementation



Instruction	Execution/address calculation stage control lines				Memory access stage control lines			Write-back stage control lines	
	RegDst	ALUOp1	ALUOp0	ALUSrc	Branch	Mem-Read	Mem-Write	Reg-Write	Memto-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

CSULB

Pipelined Control



10/26