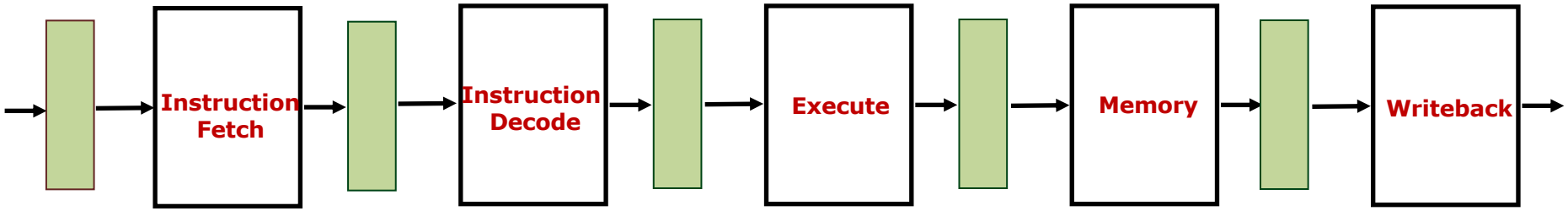
# CSE 520
# Computer Architecture II

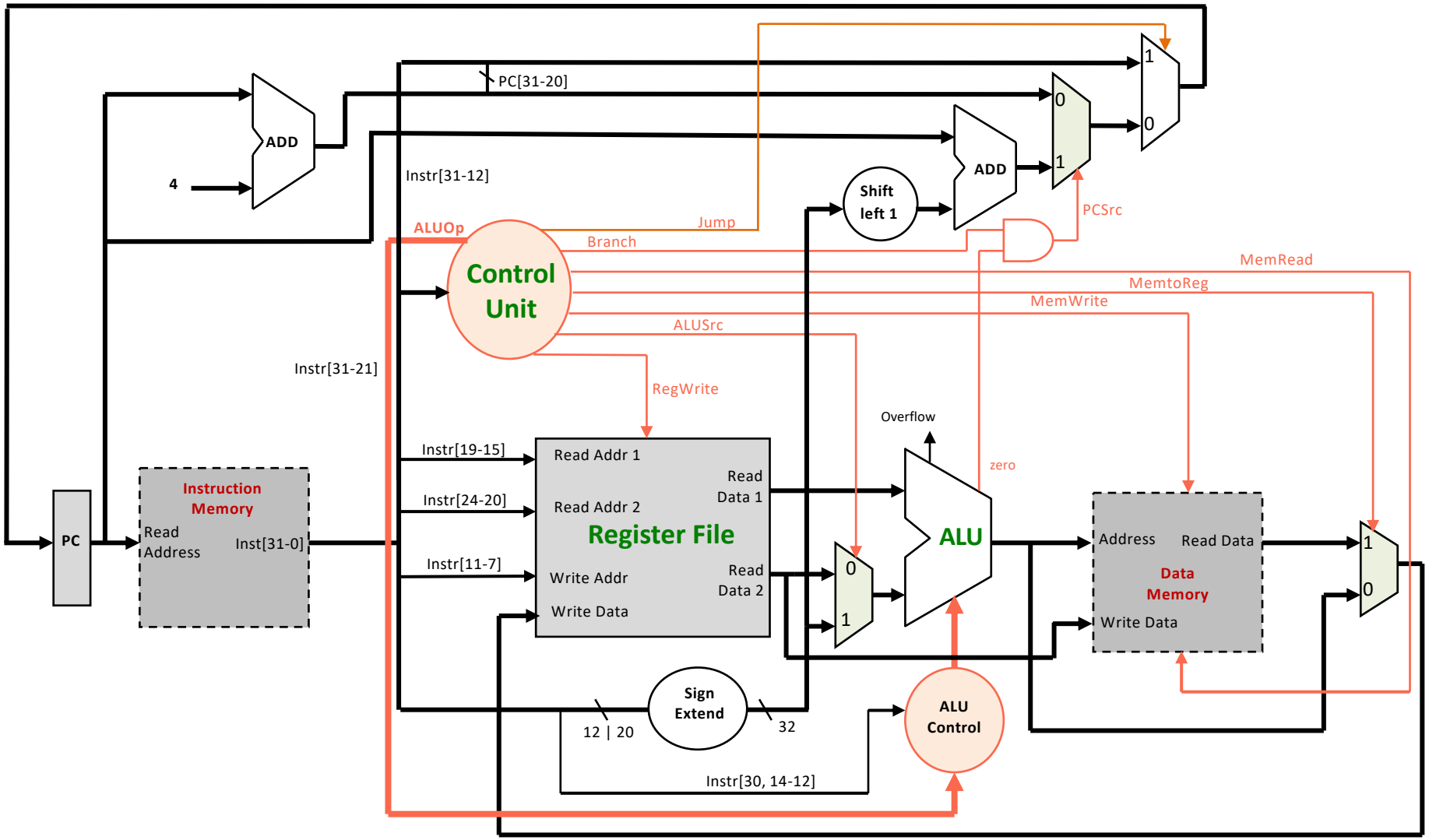
## Structural, Data and Control Hazards

Prof. Michel A. Kinsy

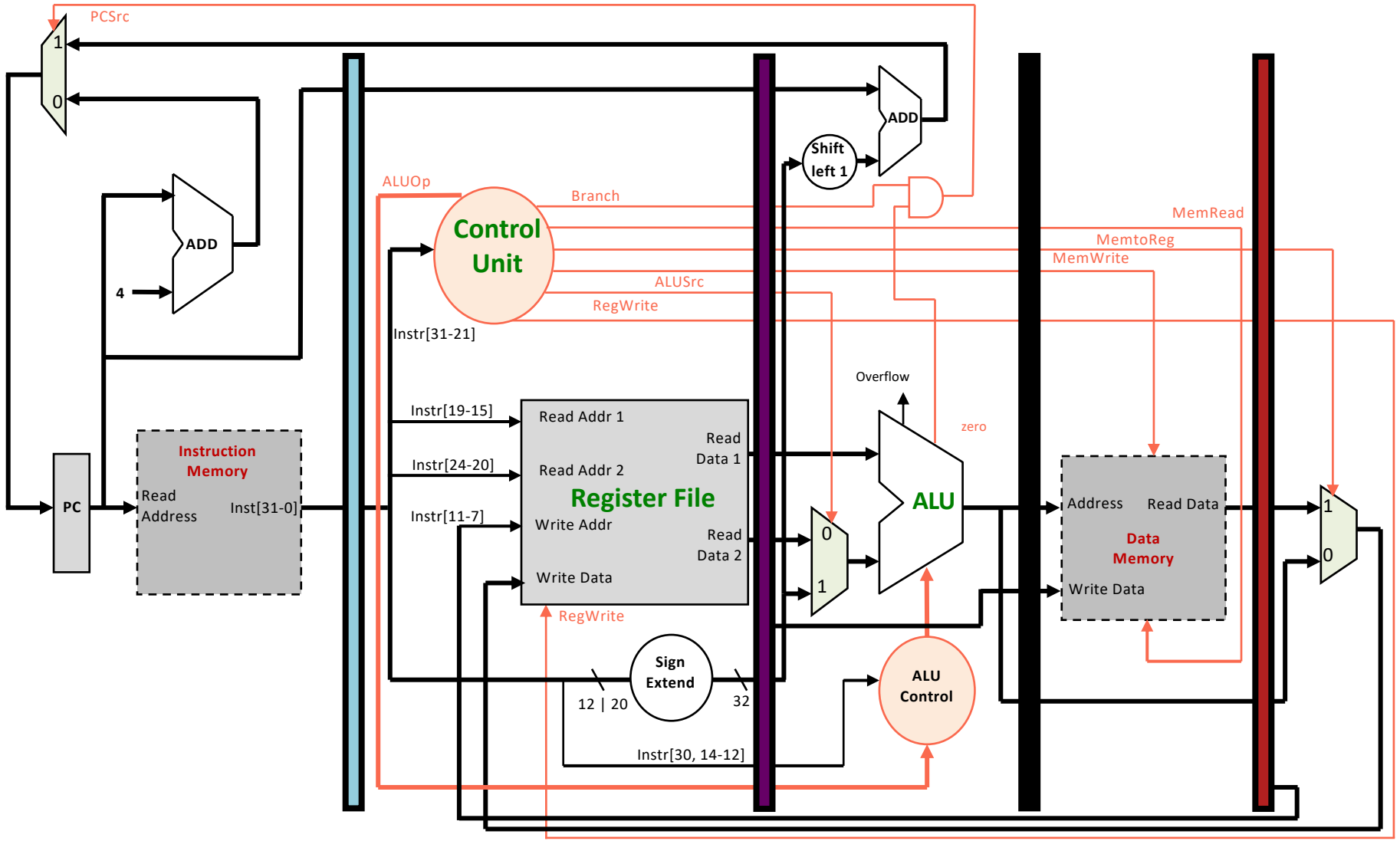
# 5-Stage RISC-V Pipelining

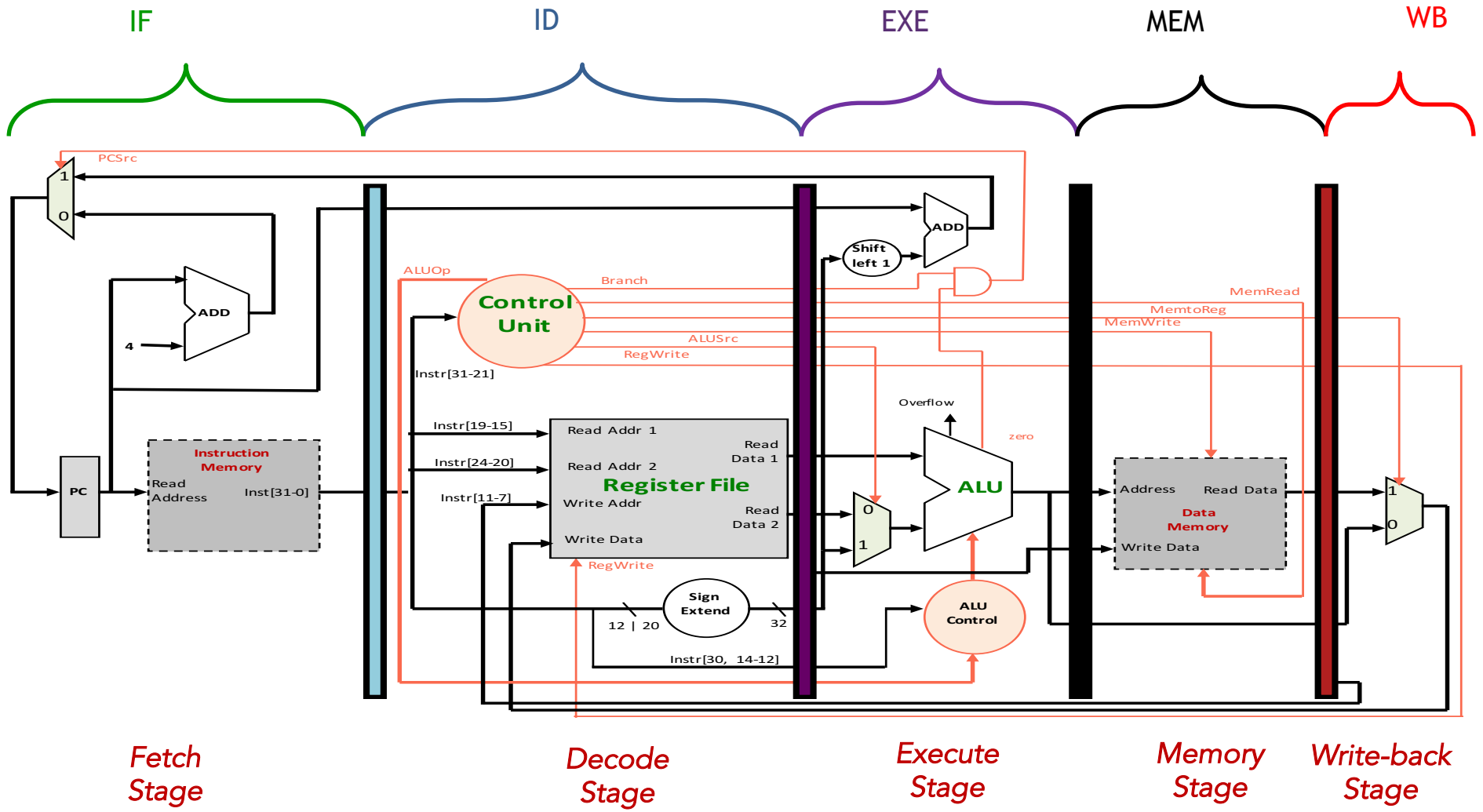
# Single Cycle RISC-V CPU



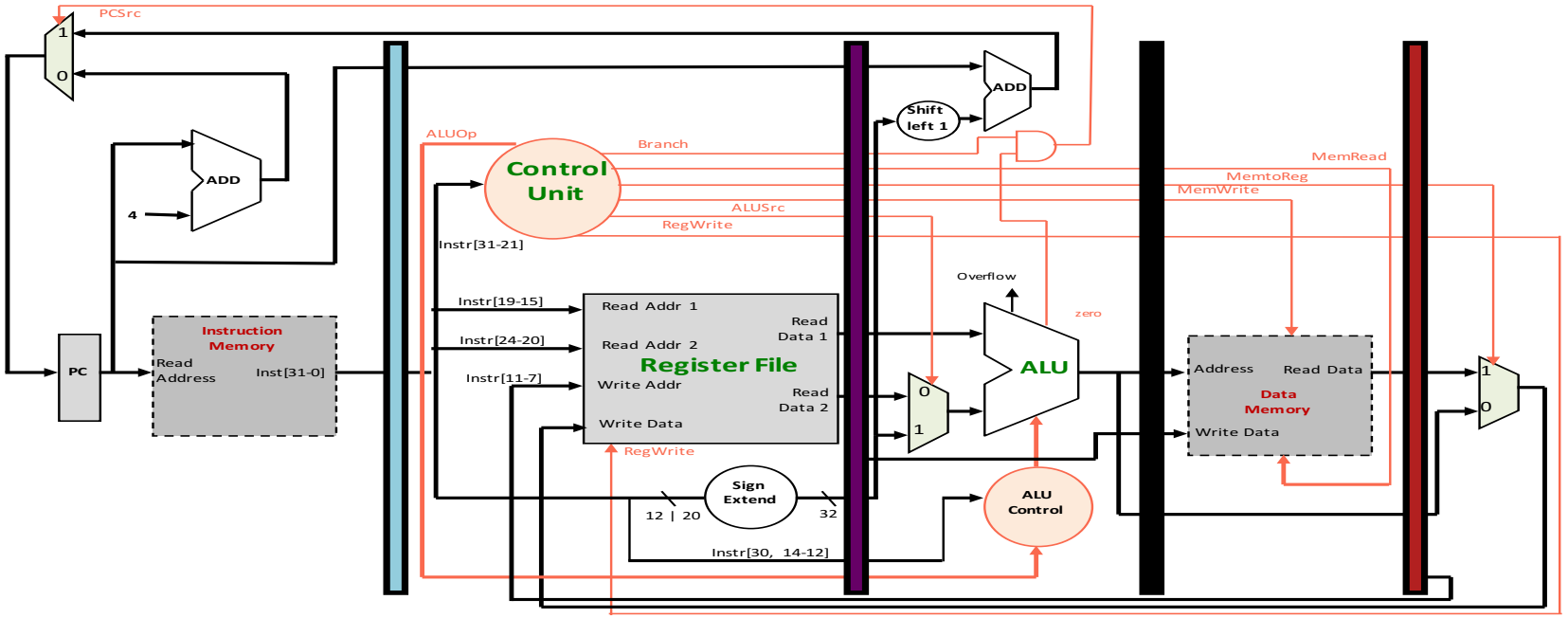
# Multi-Stage RISC-V CPU



# Pipelined Datapath



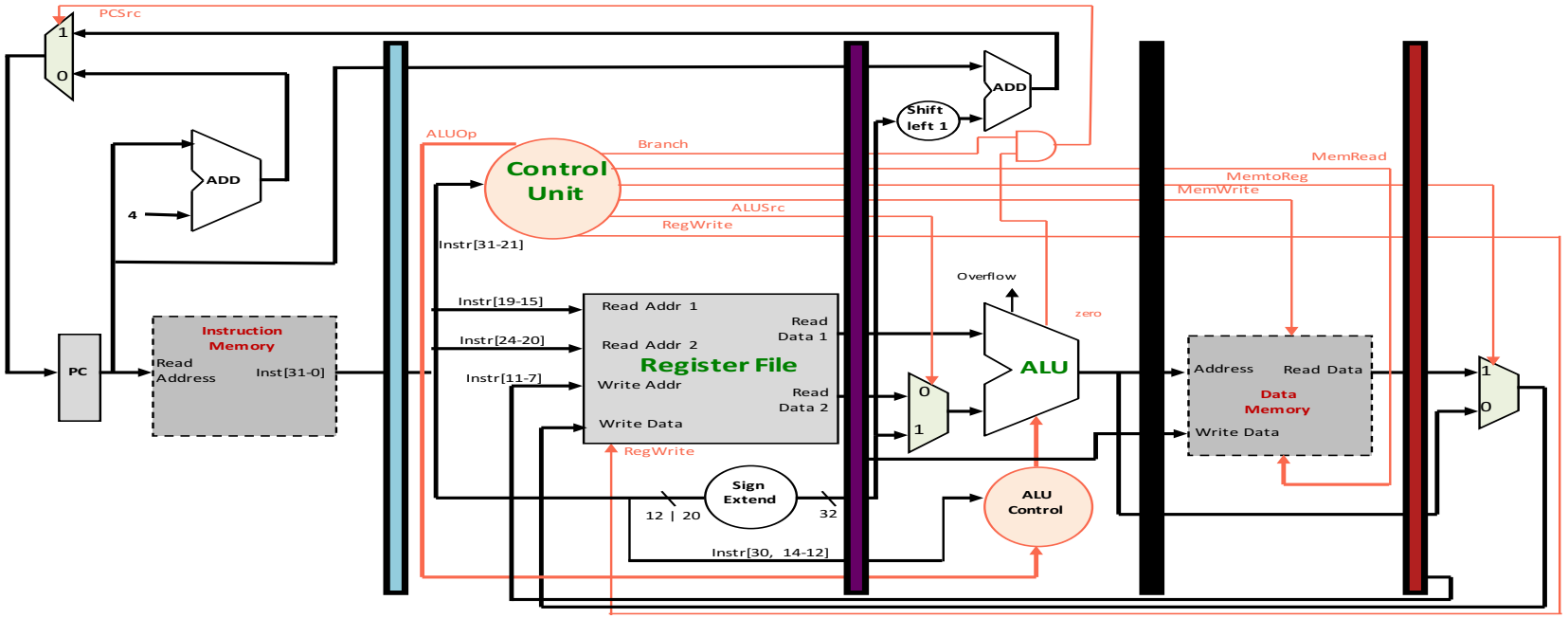
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | | | | | | | | |

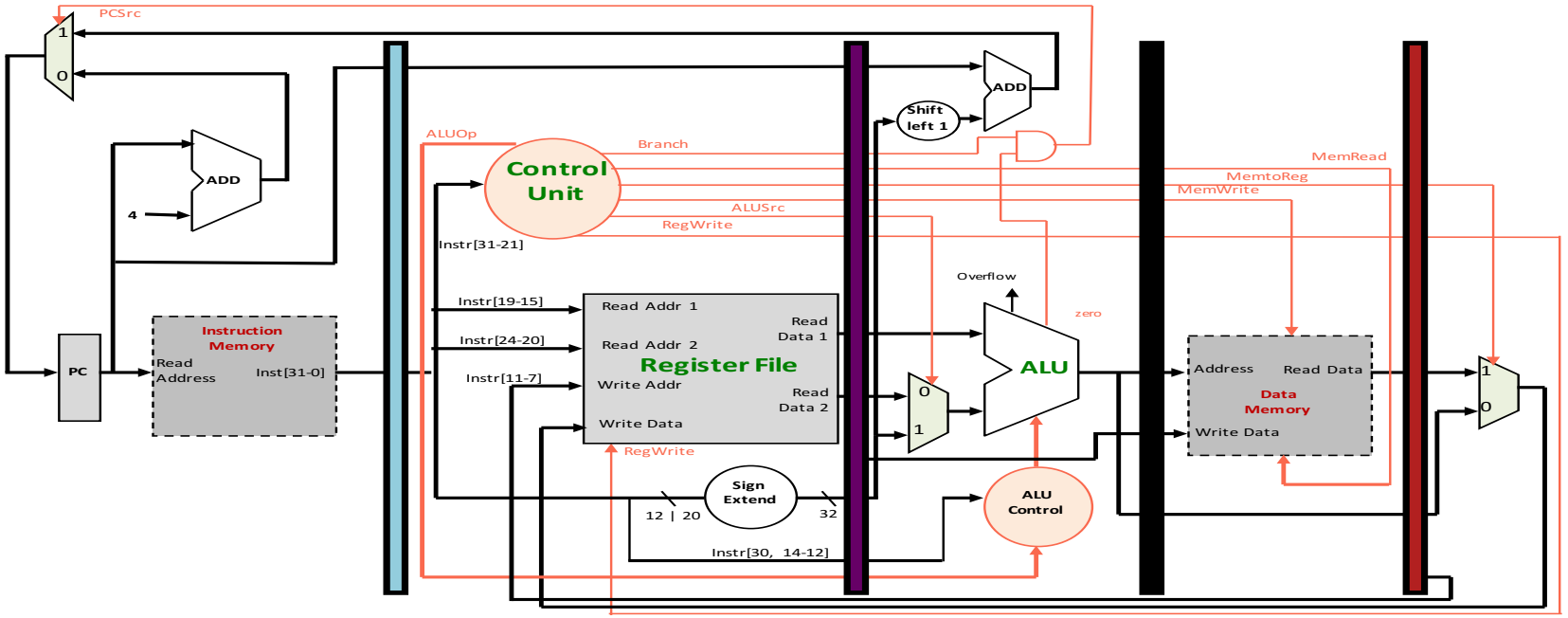
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | | | | | | | |
| Instruction2 | | $IF_2$ | | | | | | | |

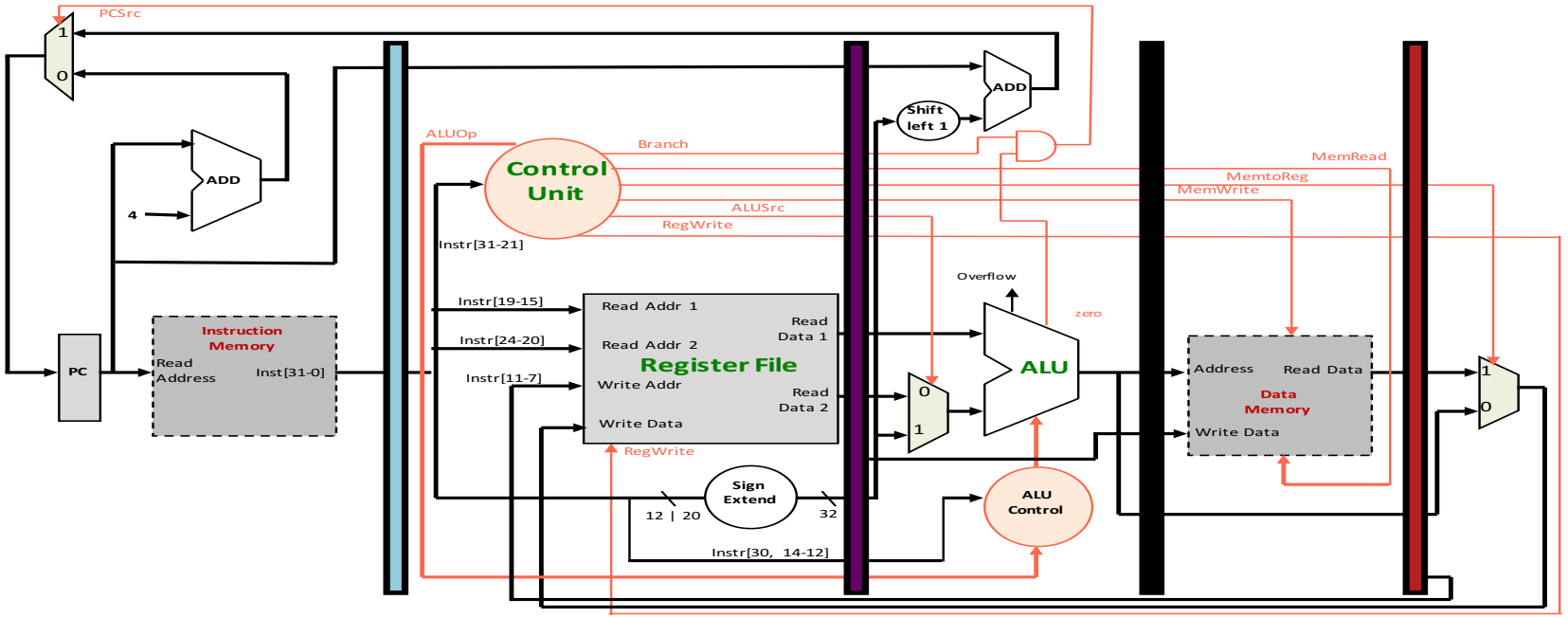
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | | | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | | | | | | |
| Instruction3 | | | $IF_3$ | | | | | | |

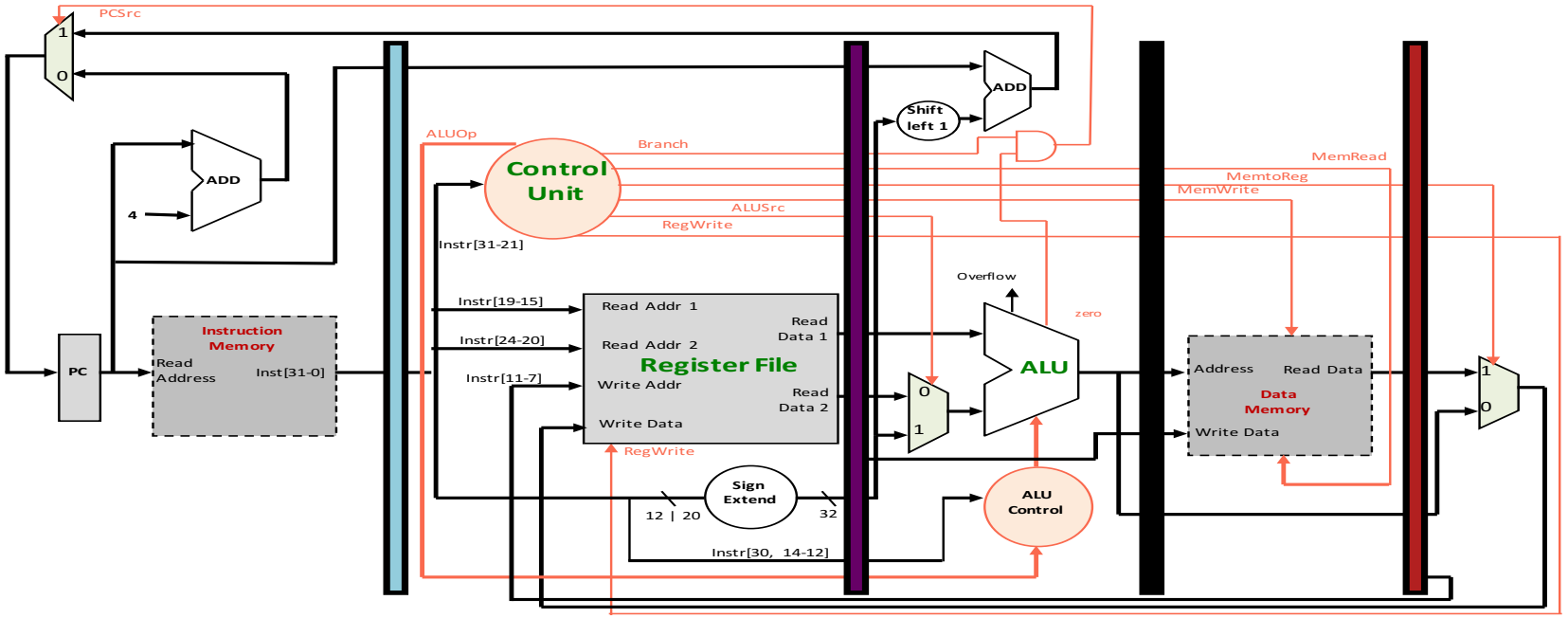
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | | | | | |
| Instruction3 | | | $IF_3$ | $ID_3$ | | | | | |
| Instruction4 | | | | $IF_4$ | | | | | |

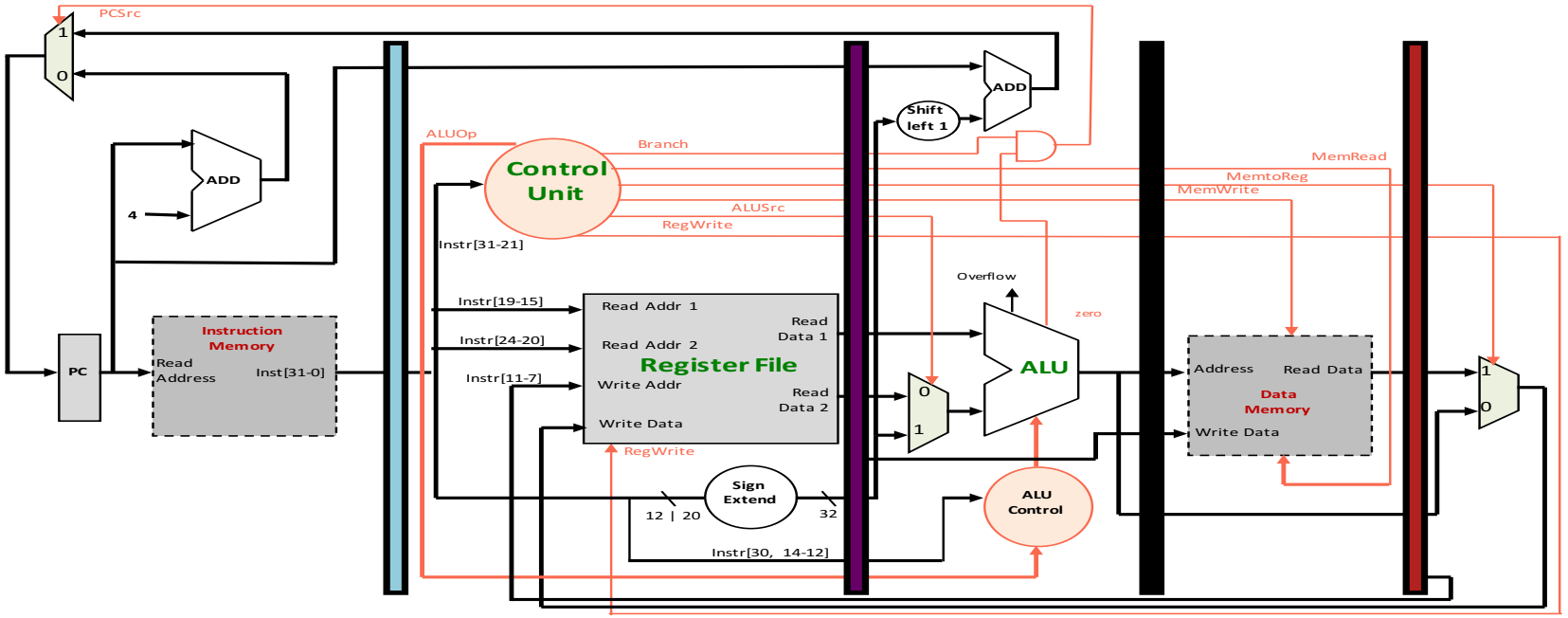
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | $MA_2$ | | | | |
| Instruction3 | | | $IF_3$ | $ID_3$ | $EX_3$ | | | | |
| Instruction4 | | | | $IF_4$ | $ID_4$ | | | | |
| Instruction5 | | | | | $IF_5$ | | | | |

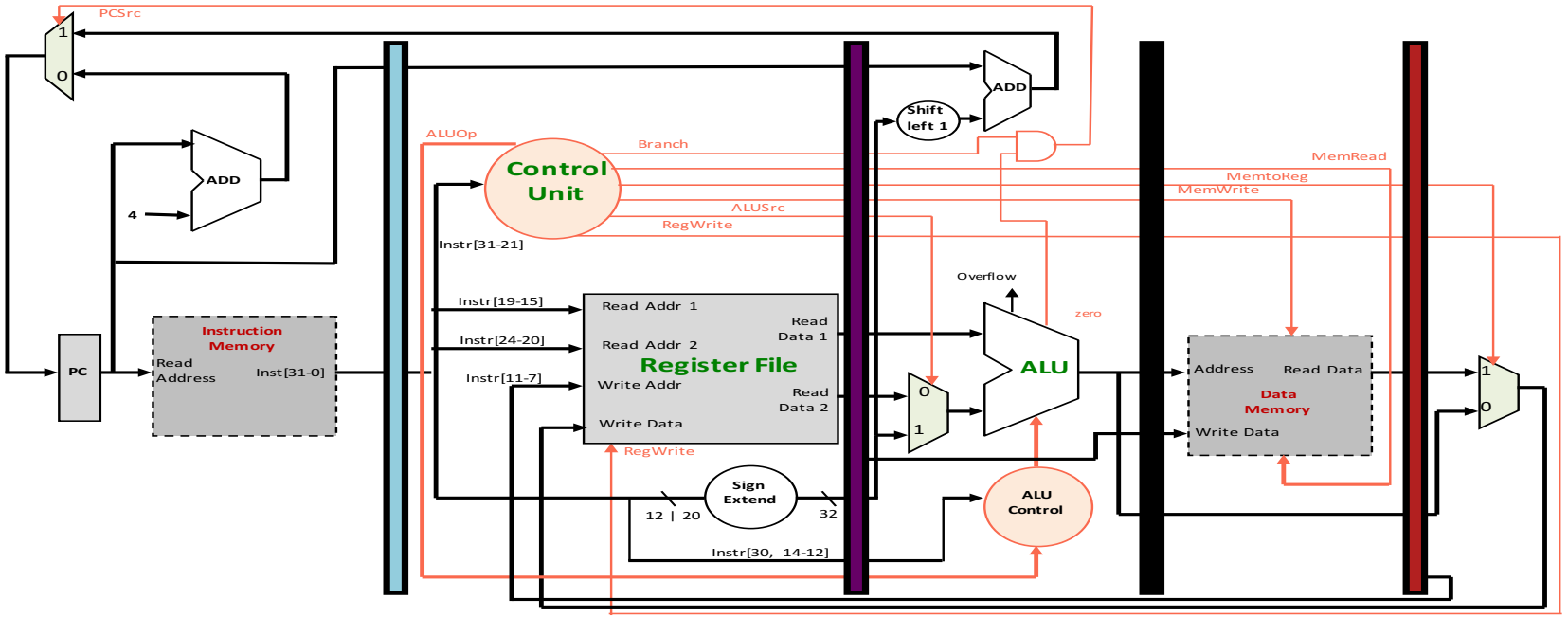
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | $MA_2$ | $WB_2$ | | | |
| Instruction3 | | | $IF_3$ | $ID_3$ | $EX_3$ | $MA_3$ | | | |
| Instruction4 | | | | $IF_4$ | $ID_4$ | $EX_4$ | | | |
| Instruction5 | | | | | $IF_5$ | $ID_5$ | | | |

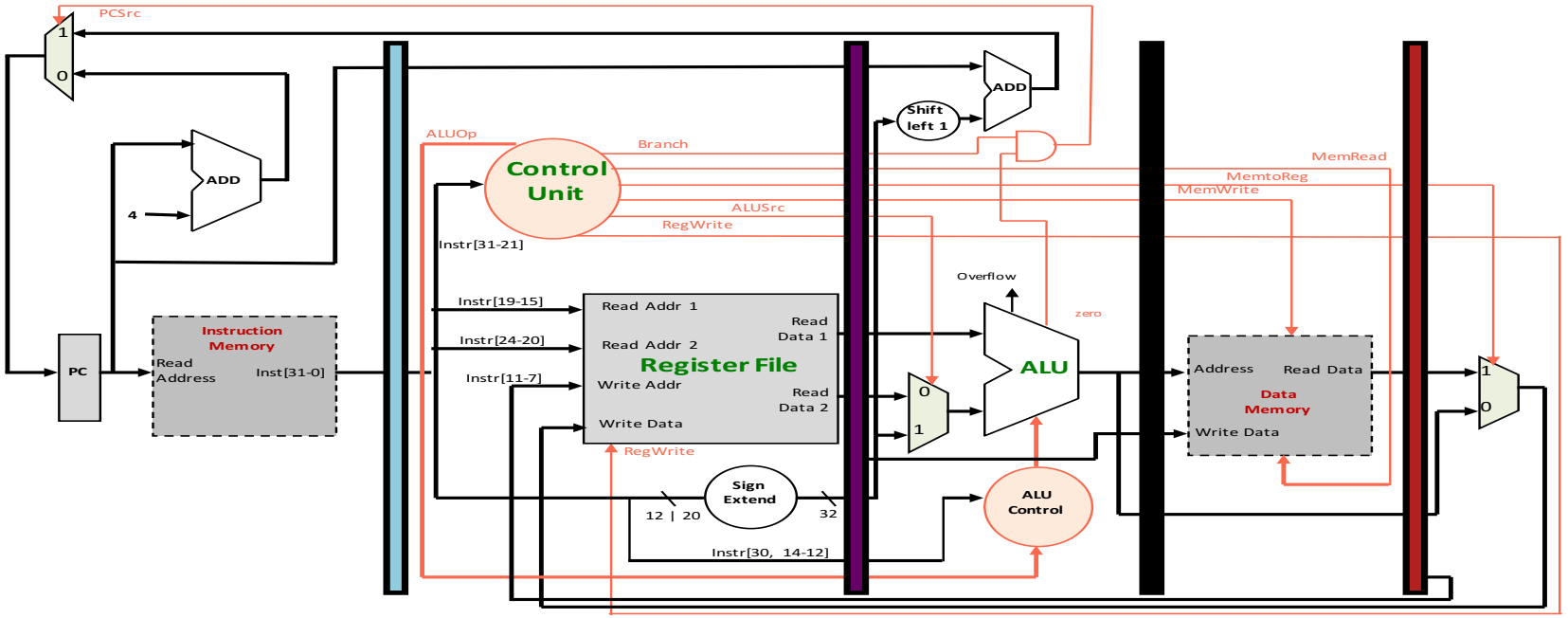
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | $MA_2$ | $WB_2$ | | | |
| Instruction3 | | | $IF_3$ | $ID_3$ | $EX_3$ | $MA_3$ | $WB_3$ | | |
| Instruction4 | | | | $IF_4$ | $ID_4$ | $EX_4$ | $MA_4$ | | |
| Instruction5 | | | | | $IF_5$ | $ID_5$ | $EX_5$ | | |

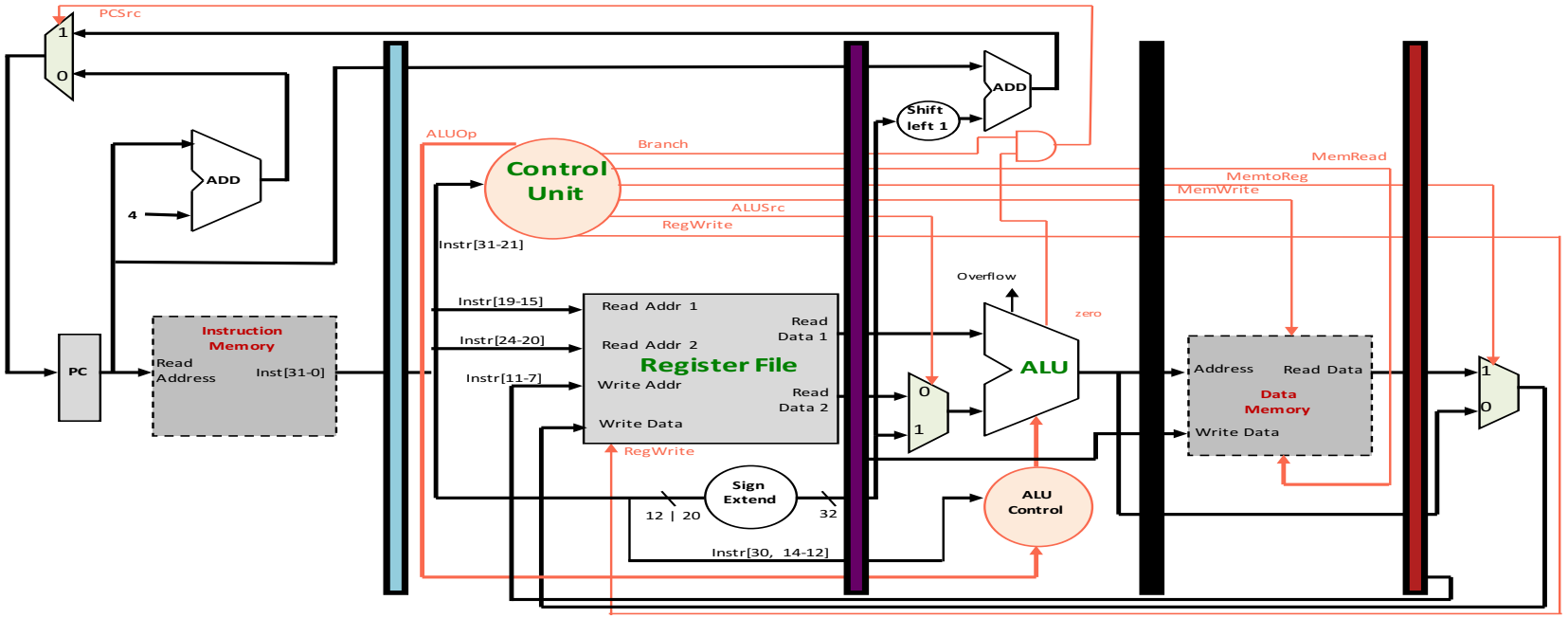
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | $MA_2$ | $WB_2$ | | | |
| Instruction3 | | | $IF_3$ | $ID_3$ | $EX_3$ | $MA_3$ | $WB_3$ | | |
| Instruction4 | | | | $IF_4$ | $ID_4$ | $EX_4$ | $MA_4$ | $WB_4$ | |
| Instruction5 | | | | | $IF_5$ | $ID_5$ | $EX_5$ | $MA_5$ | |

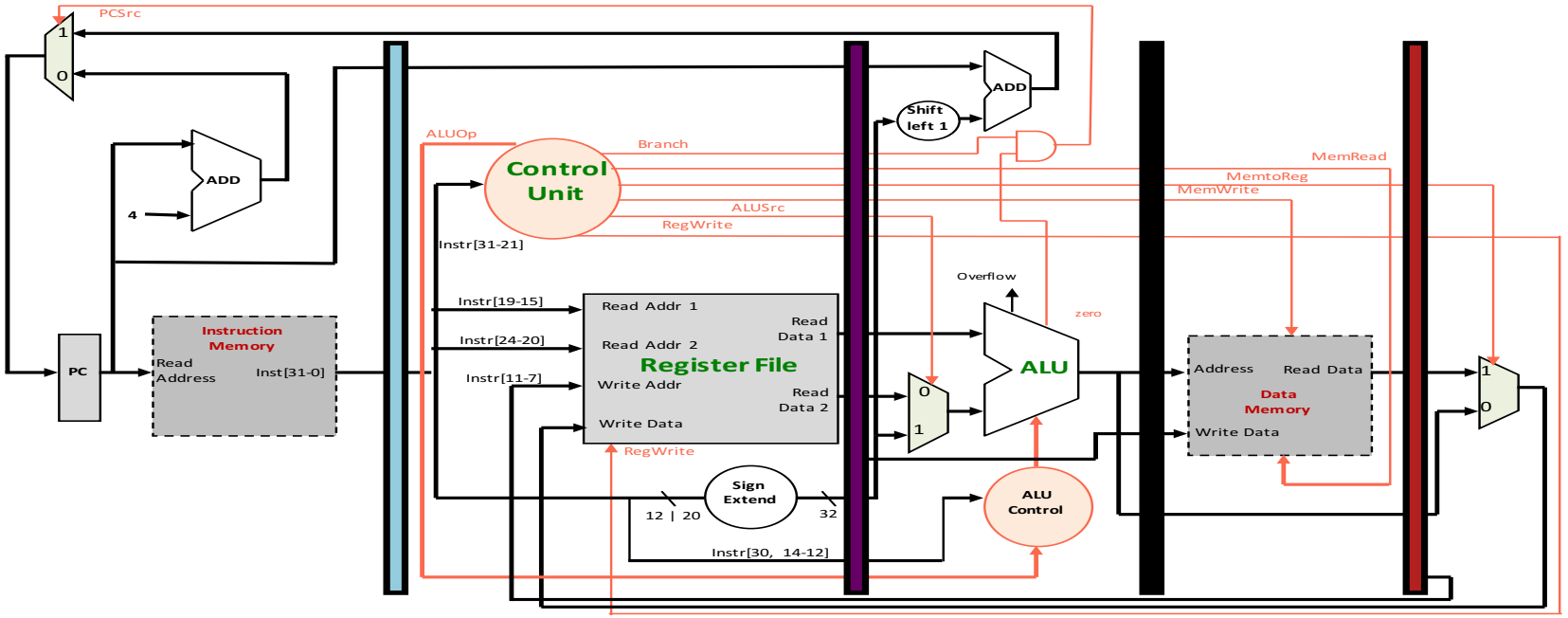
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| Instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | | |
| Instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | $MA_2$ | $WB_2$ | | | |
| Instruction3 | | | $IF_3$ | $ID_3$ | $EX_3$ | $MA_3$ | $WB_3$ | | |
| Instruction4 | | | | $IF_4$ | $ID_4$ | $EX_4$ | $MA_4$ | $WB_4$ | |
| Instruction5 | | | | | $IF_5$ | $ID_5$ | $EX_5$ | $MA_5$ | $WB_5$ |

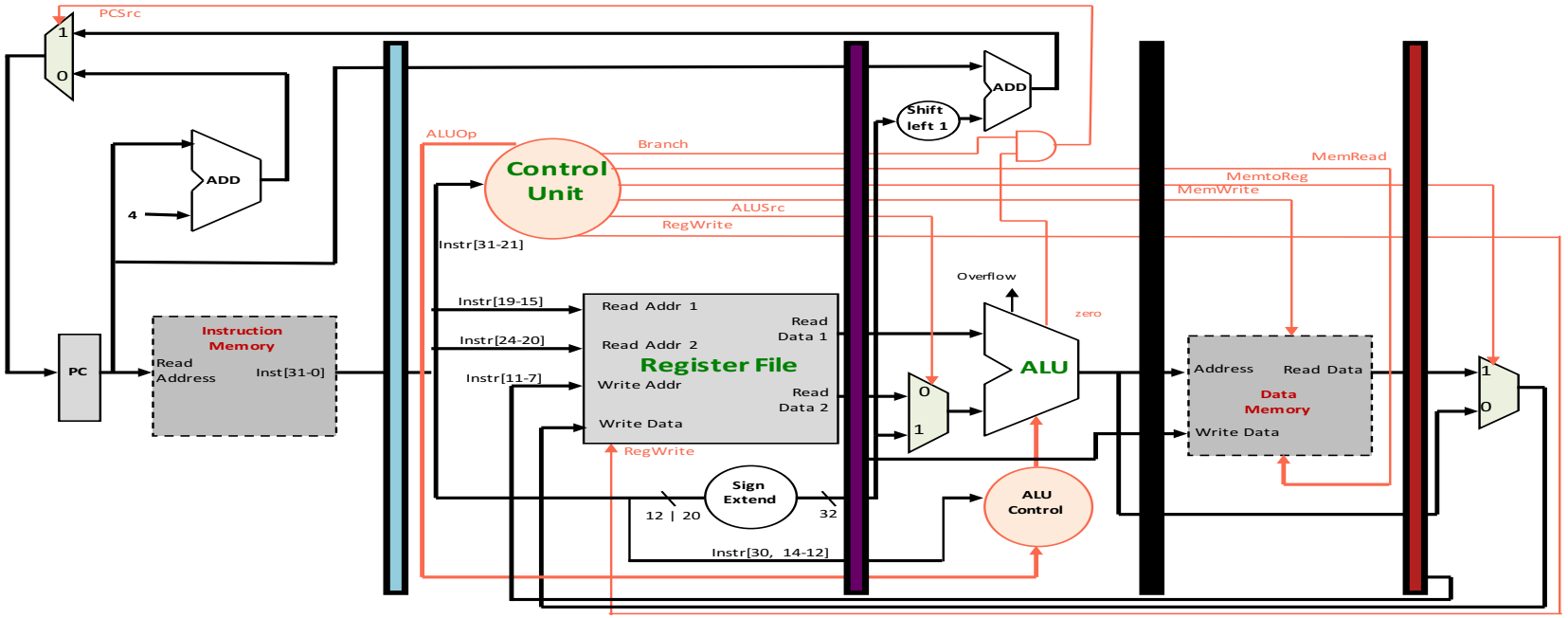
# 5-Stage Pipelined




*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

| *Time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . . |
|---|---|---|---|---|---|---|---|---|---|
| instruction1 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | | |
| instruction2 | | $IF_2$ | $ID_2$ | $EX_2$ | $MA_2$ | $WB_2$ | | | |
| instruction3 | | | $IF_3$ | $ID_3$ | $EX_3$ | $MA_3$ | $WB_3$ | | |
| instruction4 | | | | $IF_4$ | $ID_4$ | $EX_4$ | $MA_4$ | $WB_4$ | |
| instruction5 | | | | | $IF_5$ | $ID_5$ | $EX_5$ | $MA_5$ | $WB_5$ |

# 5-Stage Pipelined

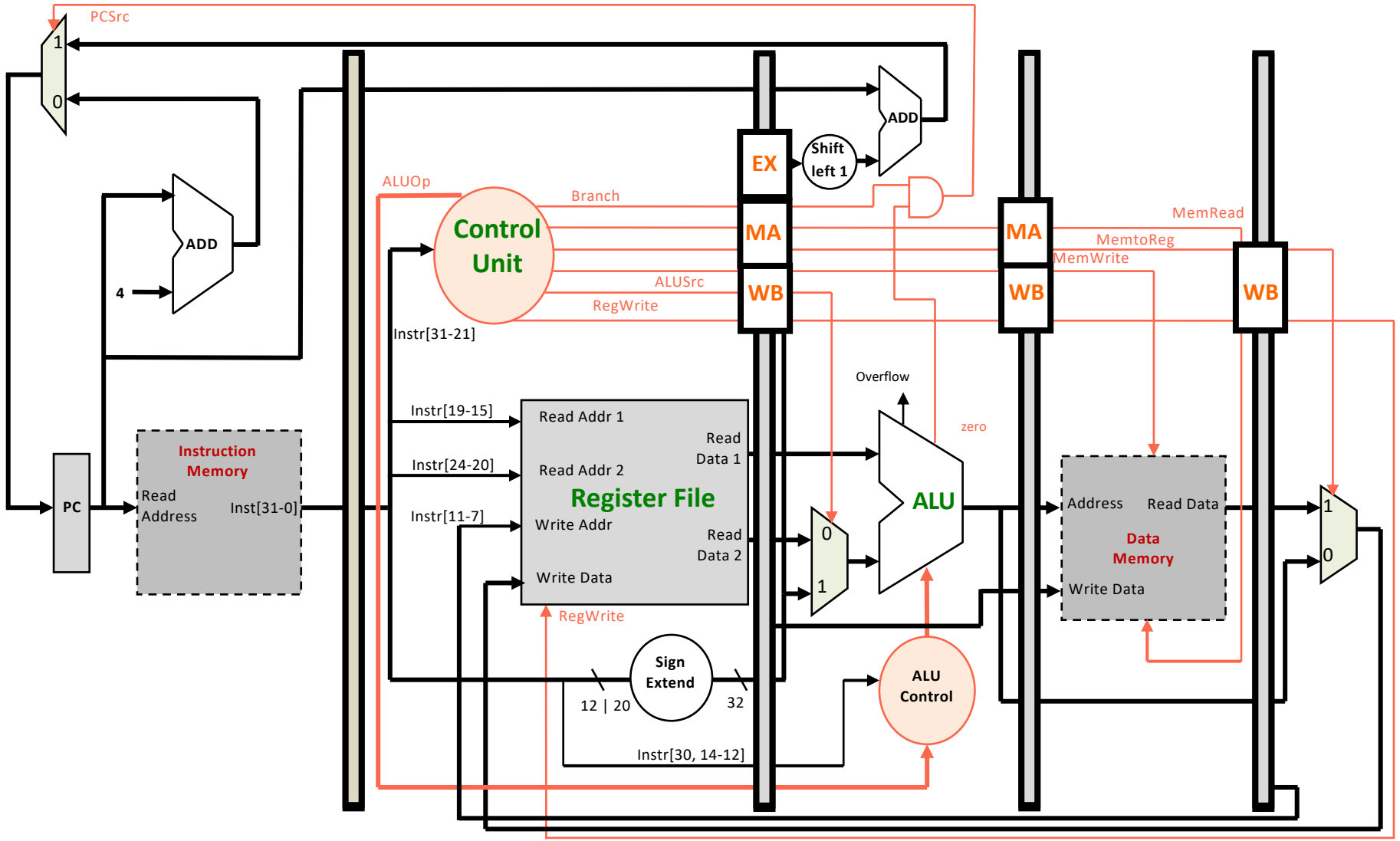



*Fetch Stage* *Decode Stage* *Execute Stage* *Memory Stage* *Write-back Stage*

*Resources*

| *time* | t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . | |
|---|---|---|---|---|---|---|---|---|---|---|
| *IF* | | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | | | | |
| *ID* | | | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | | | |
| *EX* | | | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | | | |
| *MA* | | | | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | | |
| *WB* | | | | | | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ |

# Multi-Stage & Control Signals

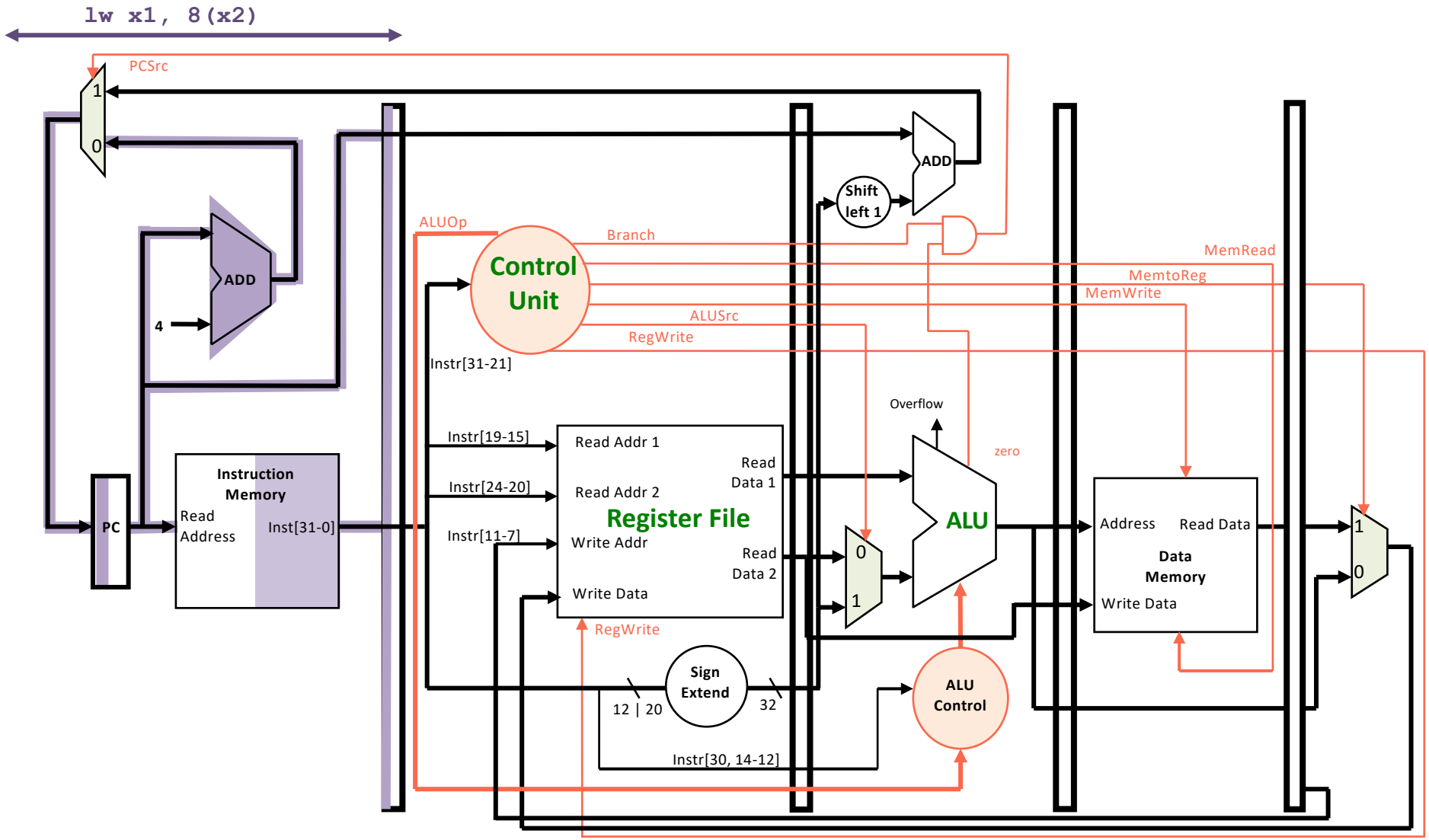


# Execution Simulation

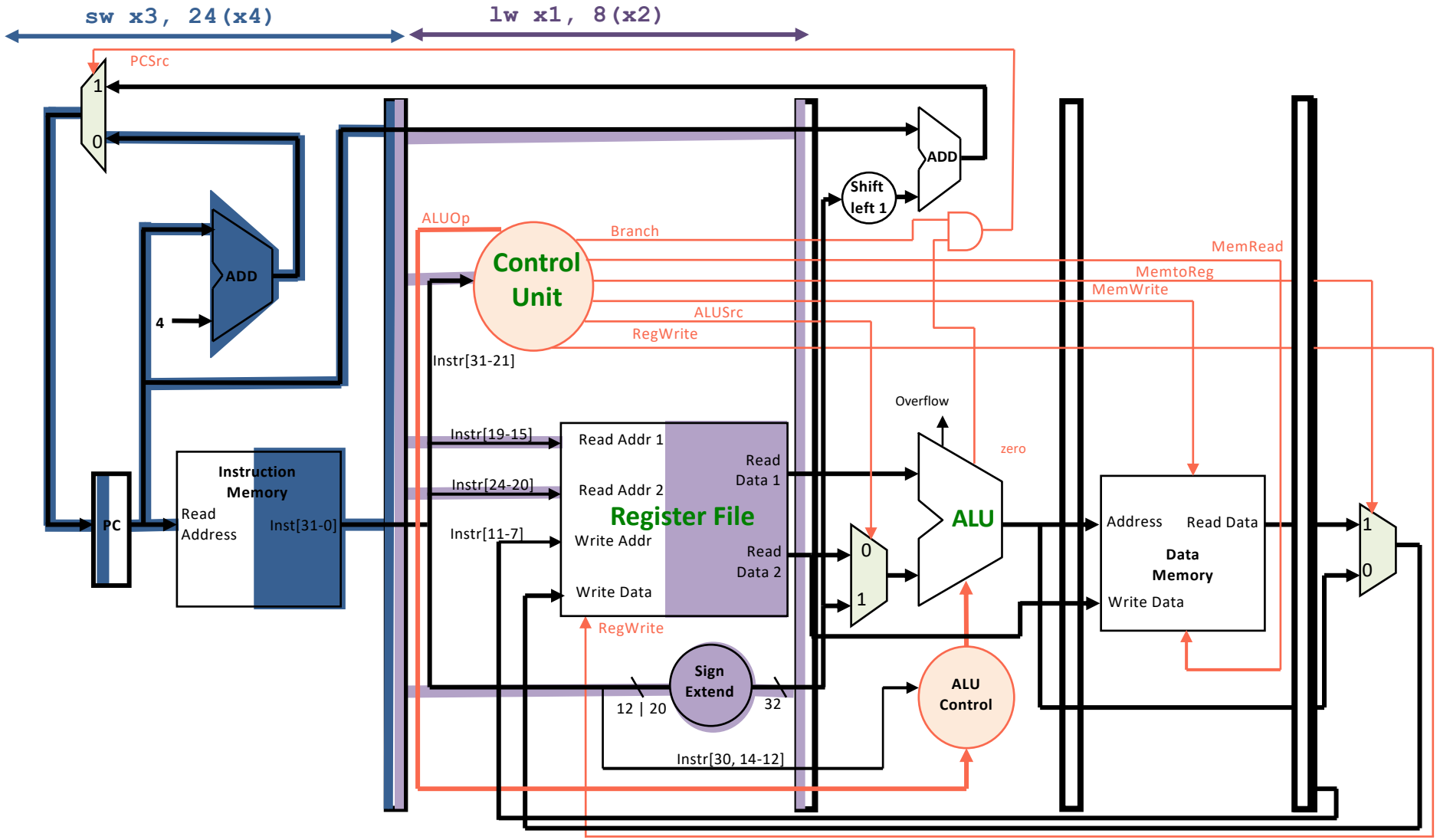
- Consider the following instruction sequence

```
lw  x1, 8(x2)
sw  x3, 24(x4)
add x5, x6, x7
sub x8, x6, x7
lw  x1, 4(x2)
```

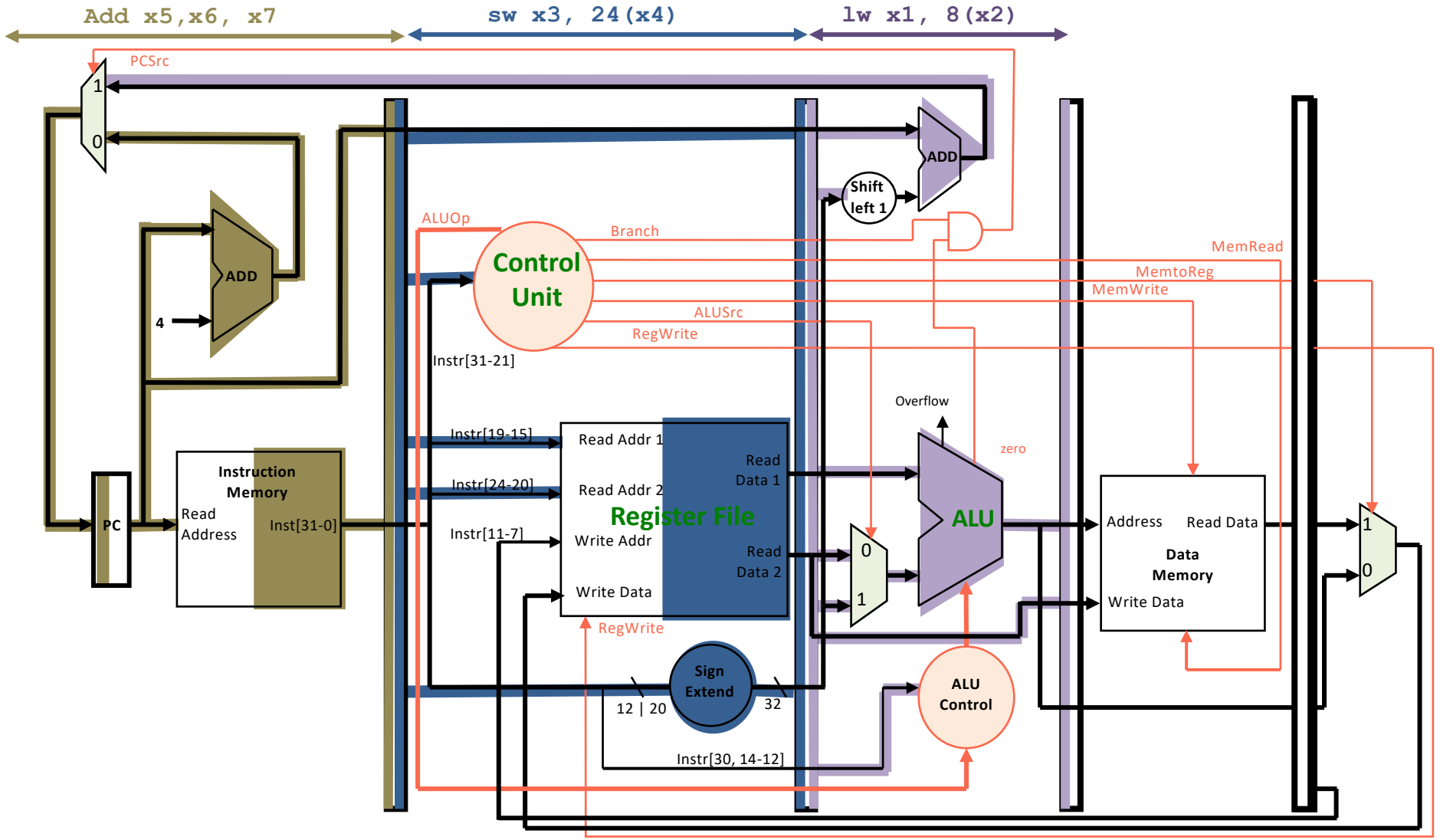
# Execution Simulation: Cycle 1



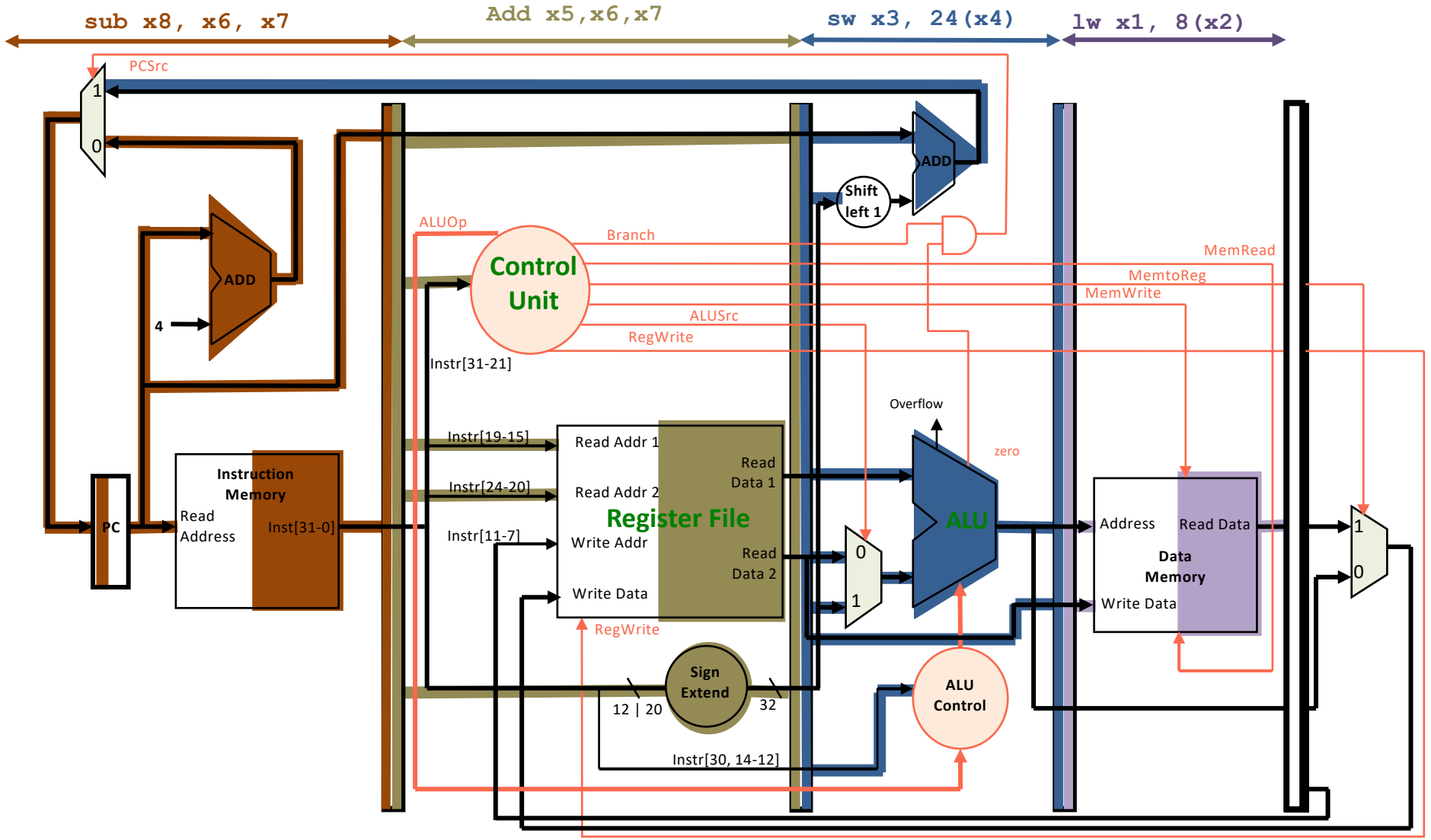
# Execution Simulation: Cycle 2



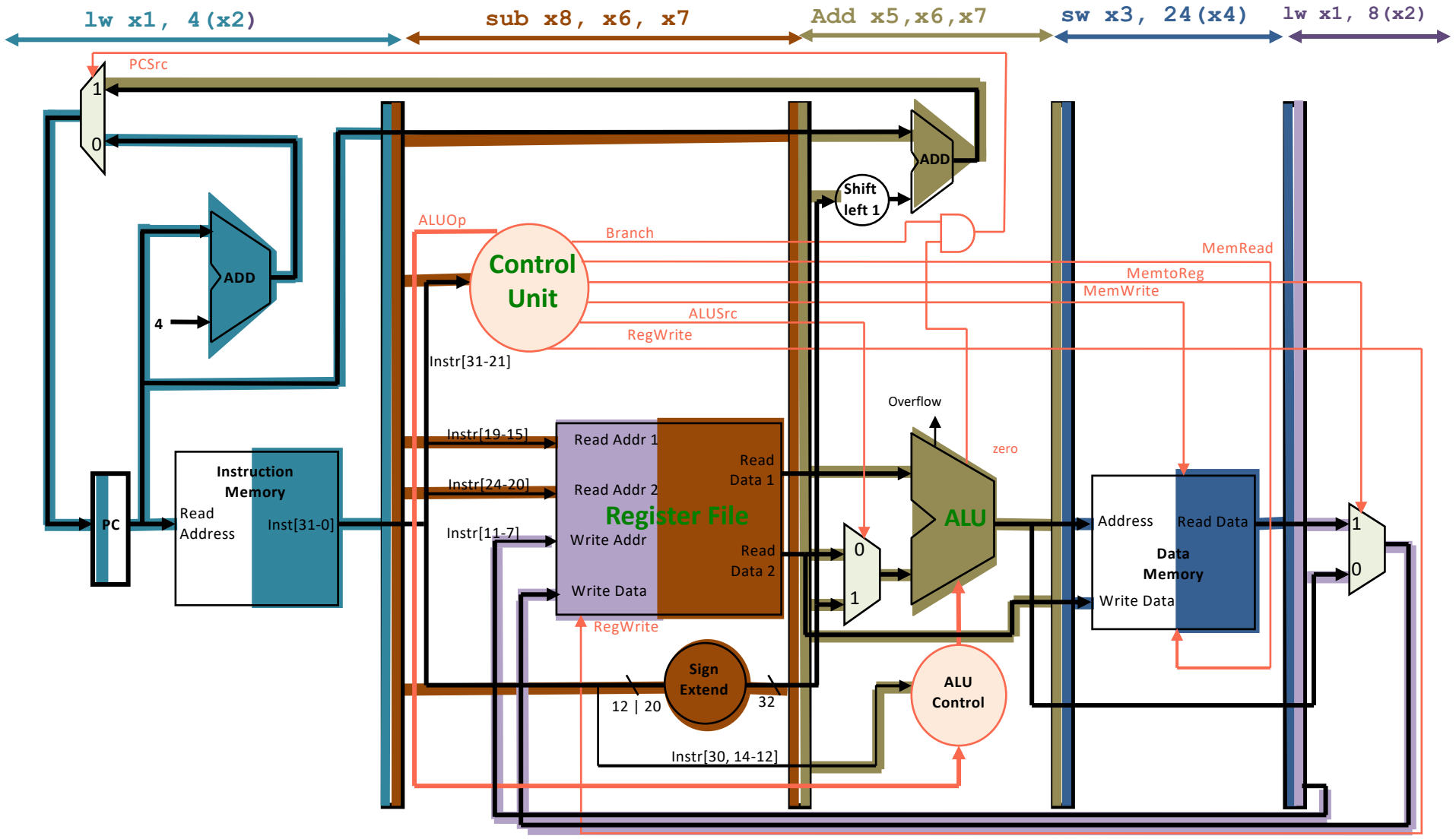
# Execution Simulation: Cycle 3



# Execution Simulation: Cycle 4



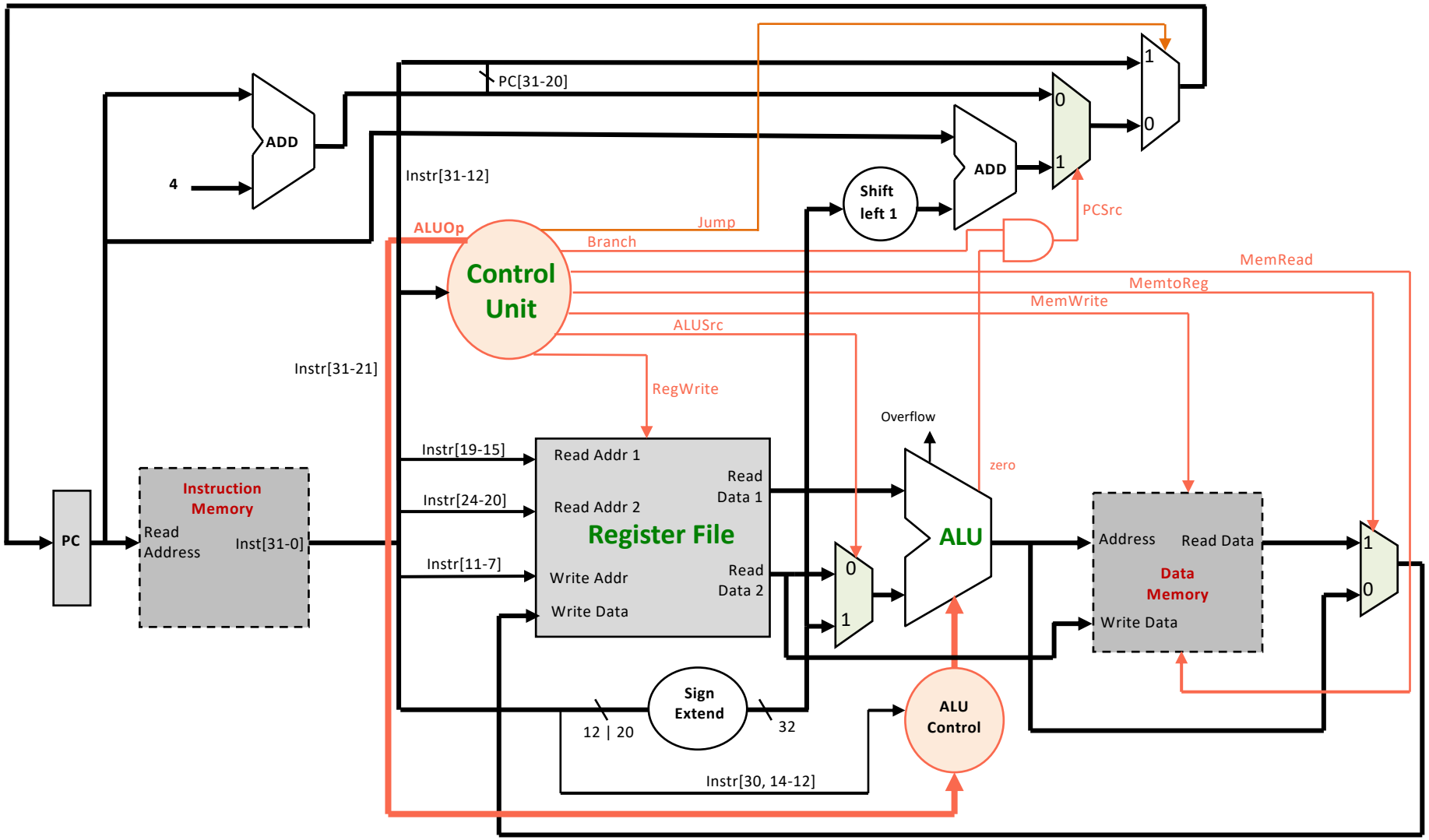
# Execution Simulation: Cycle 5



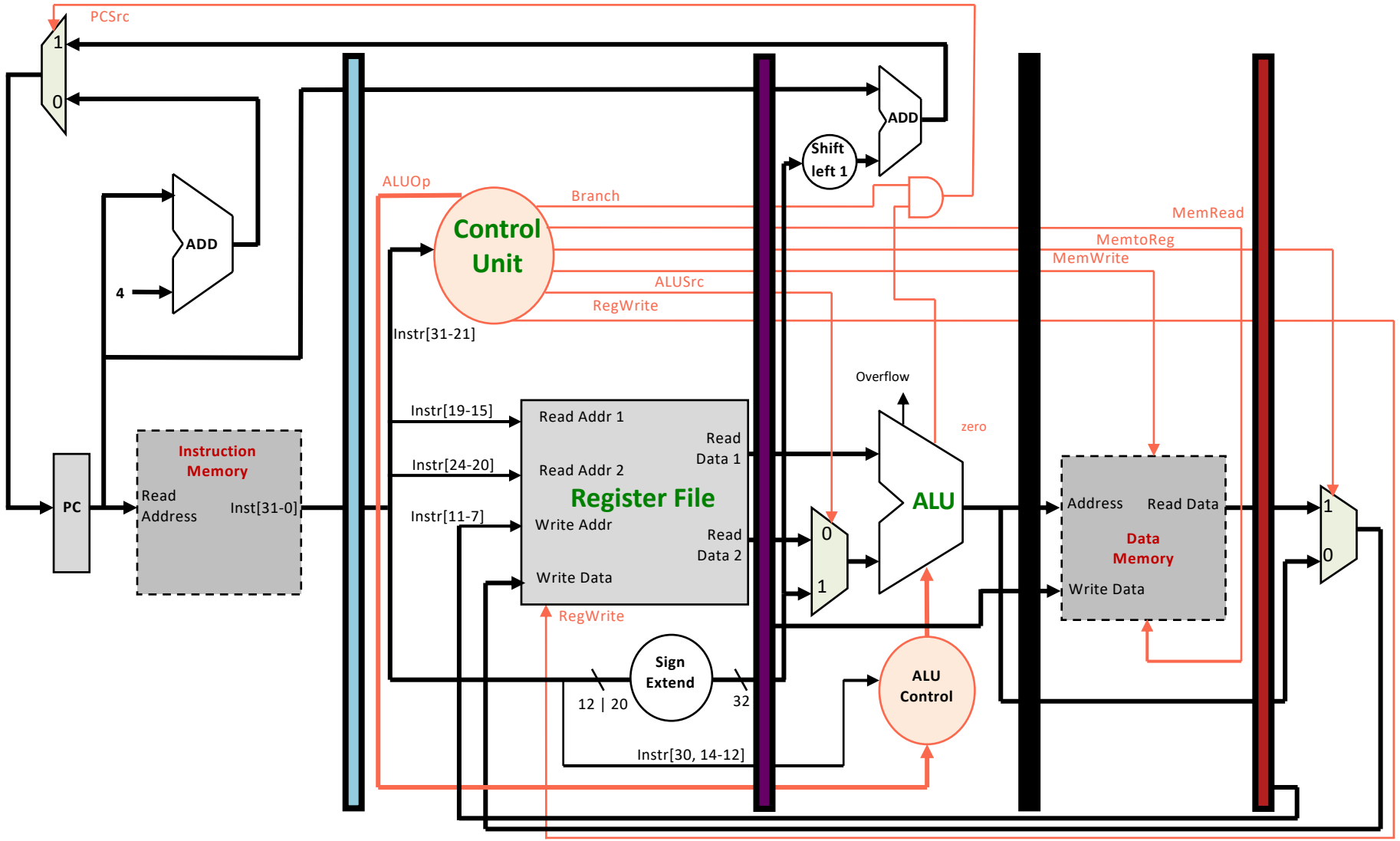
# Instruction Interactions

- An instruction in the pipeline may need a resource being used by another instruction in the pipeline
  - Structural hazard
- An instruction may depend on something produced by an earlier instruction
  - Dependence may be for a data calculation
    - Data hazard
  - Dependence may be for calculating the next address
    - Control hazard (branches, interrupts)

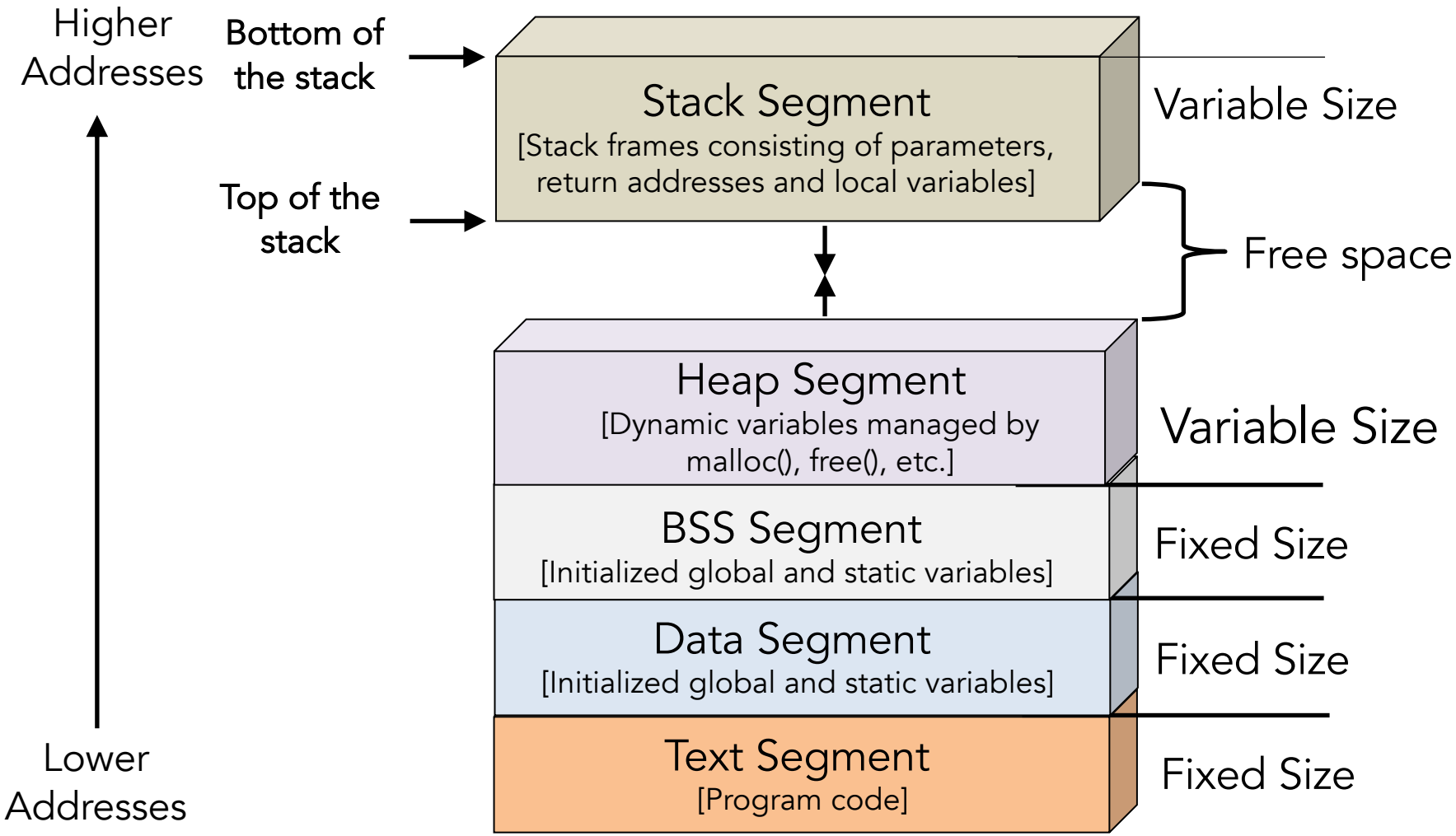
# Single Cycle RISC-V CPU



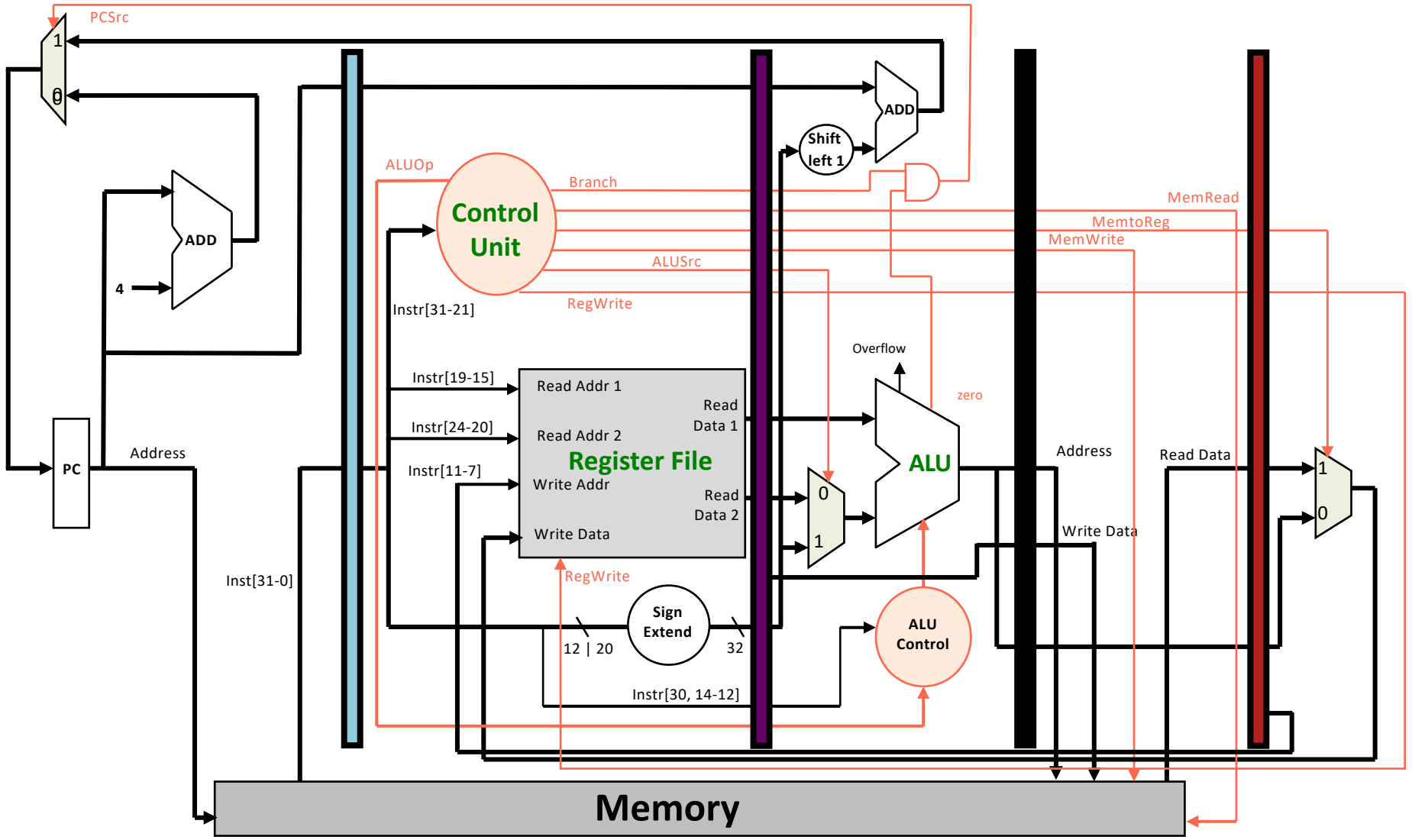
# Multi-Stage RISC-V CPU



# Program memory management



# Multi-Stage RISC-V CPU

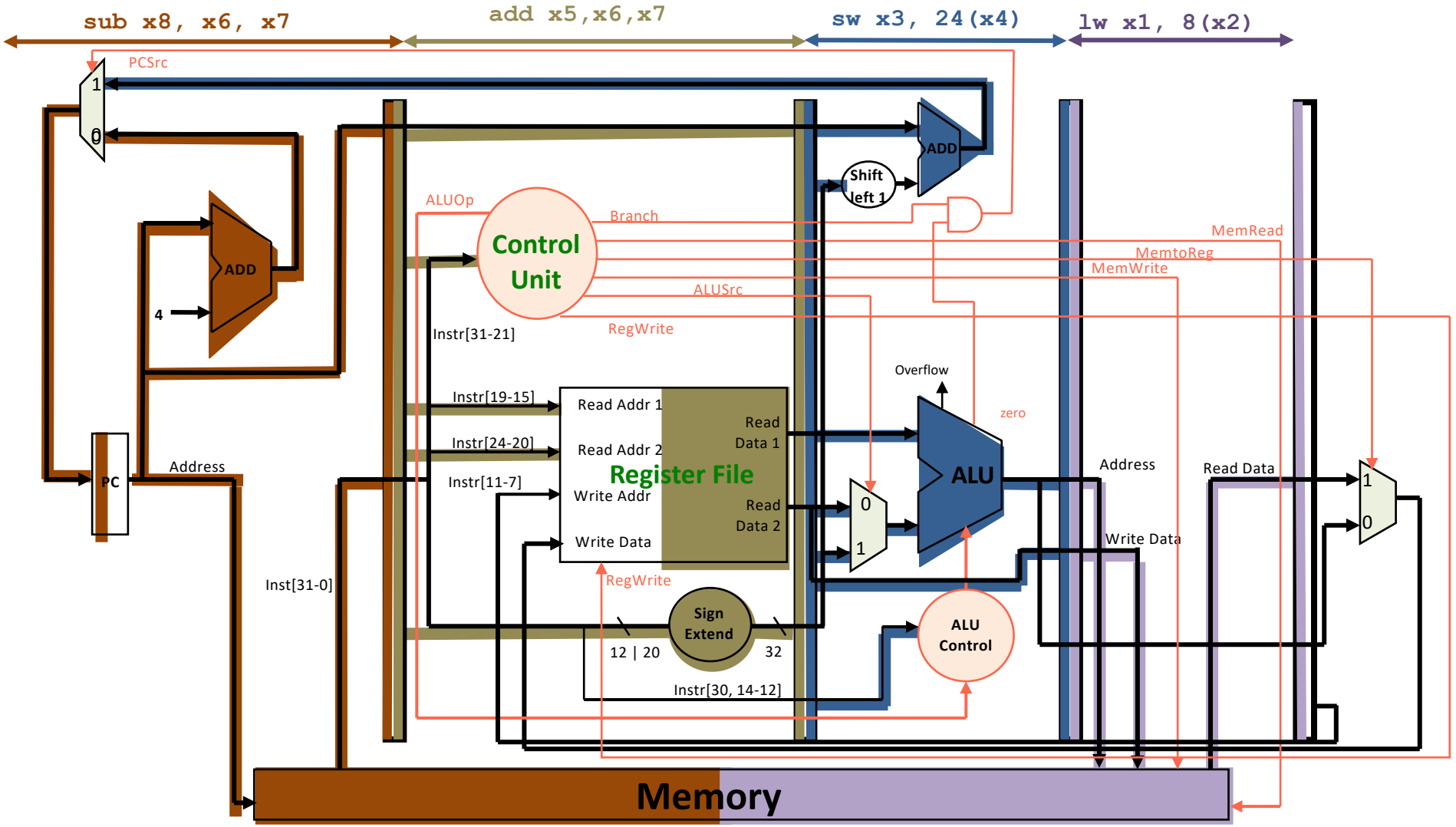


# Structural Hazard

- Consider the following instruction sequence

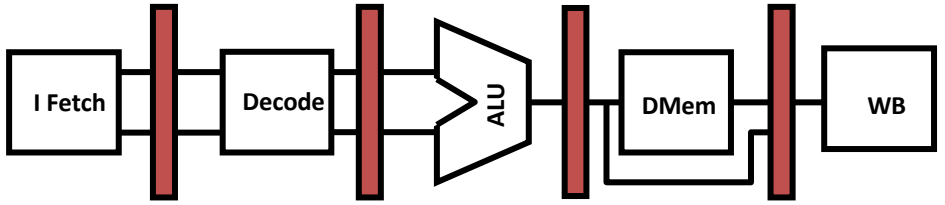
```
lw  x1, 8(x2)
sw  x3, 24(x4)
add x5, x6, x7
sub x8, x6, x7
lw  x1, 4(x2)
```

# Structural Hazard



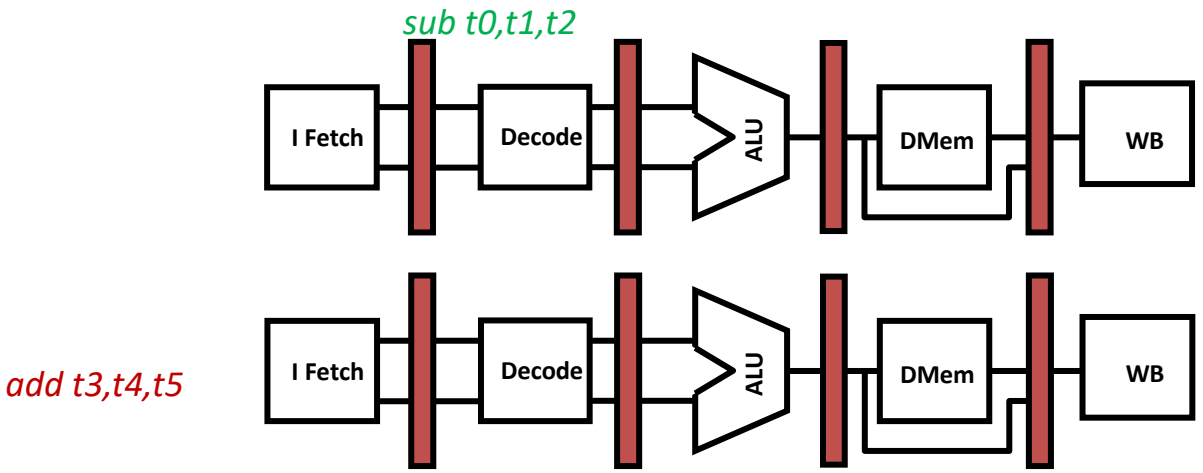
# Instruction Interactions

*sub t0,t1,t2*



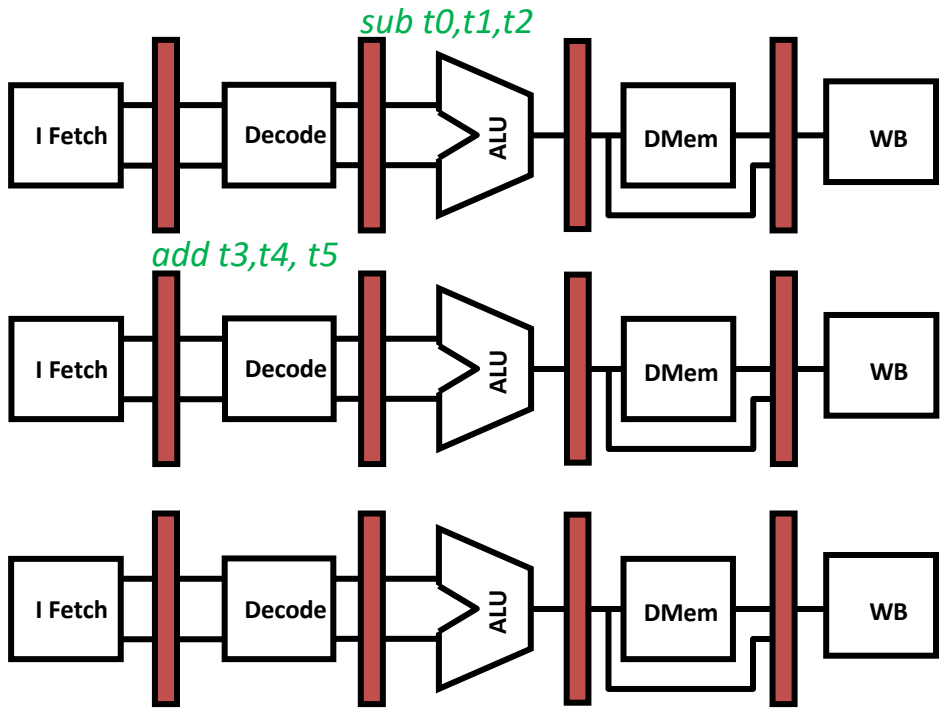

*sub t0,t1,t2*
*add t3,t4,t5*
*add t6,s2,s3*
*sub t1,s3,s2*
*lw t2,16(t4)*
*add t5,t5,s2*
*sw t0,16(s3)*

# Instruction Interactions



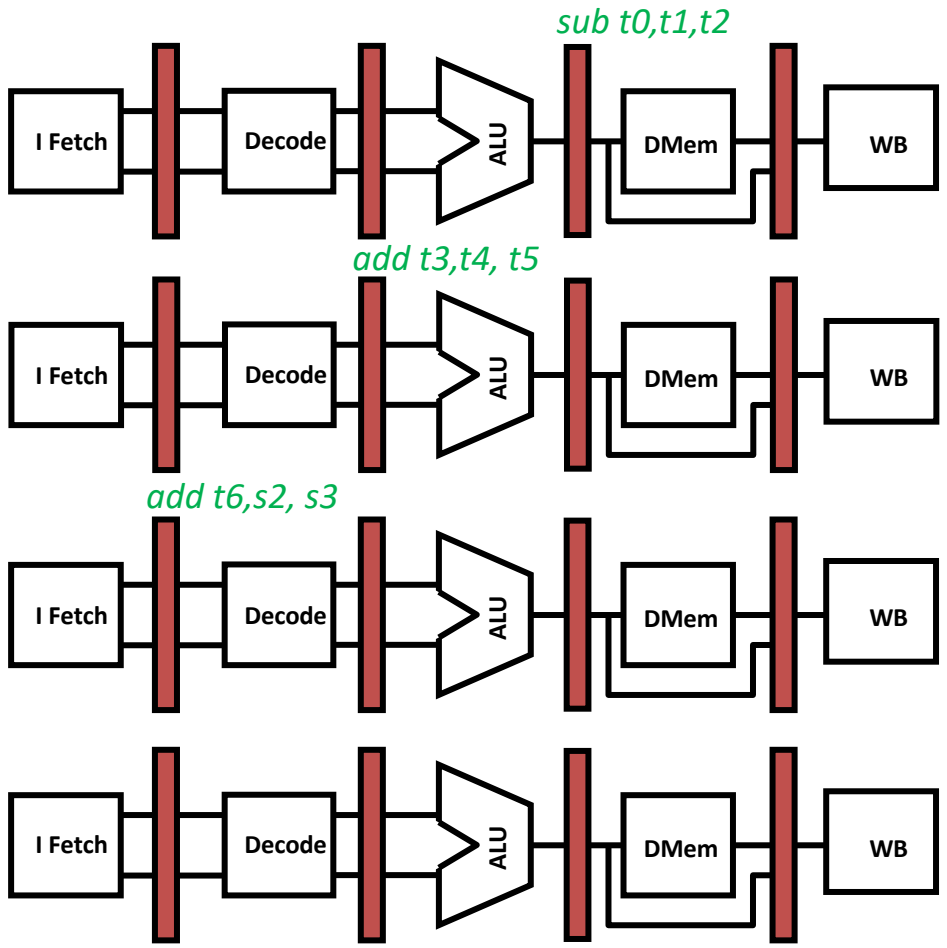

*sub t0,t1,t2*
*add t3,t4,t5*
*add t6,s2,s3*
*sub t1,s3,s2*
*lw t2,16(t4)*
*add t5,t5,s2*
*sw t0,16(s3)*

# Instruction Interactions




*sub t0,t1,t2*
*add t3,t4,t5*
*add t6,s2,s3*
*sub t1,s3,s2*
*lw t2,16(t4)*
*add t5,t5,s2*
*sw t0,16(s3)*

# Instruction Interactions




*sub t0,t1,t2*
*add t3,t4,t5*
*add t6,s2,s3*
*sub t1,s3,s2*
*lw t2,16(t4)*
*add t5,t5,s2*
*sw t0,16(s3)*

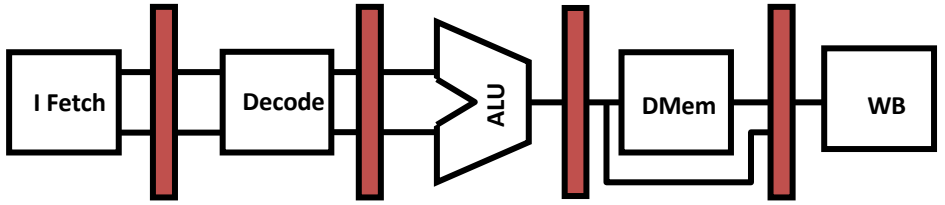
# Instruction Interactions

- Supposed that we have the following set of instructions:

```
sub t0,t1,t2
add t3,t0,t4
add t5,t0,t6
sub s2,s3,t0
lw   s4,16(t0)
add s4,s4,s2
sw   s4,16(t0)
```

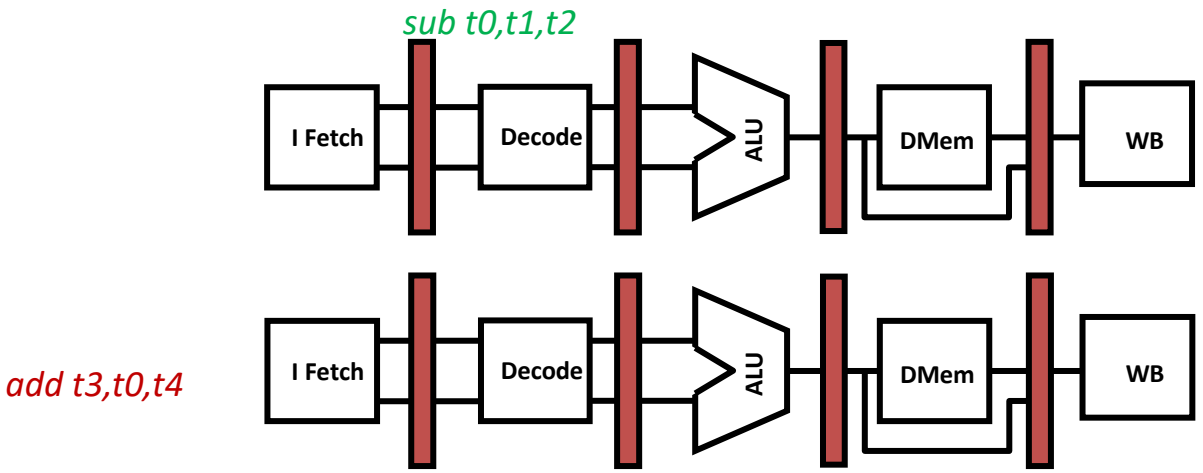
# Instruction Interactions

*sub t0,t1,t2*



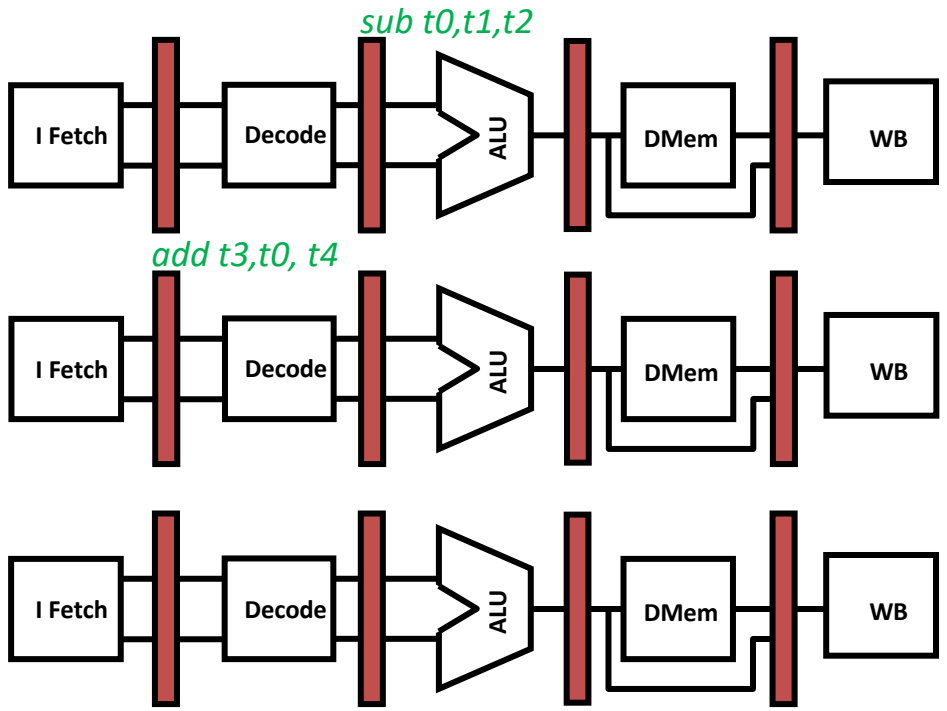

*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions



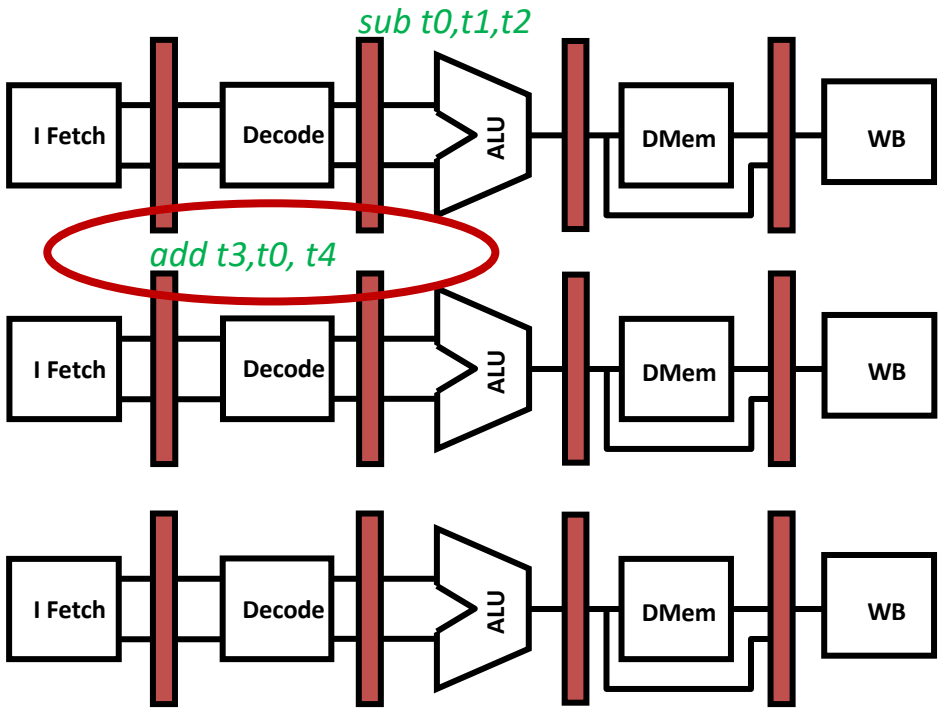

*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions



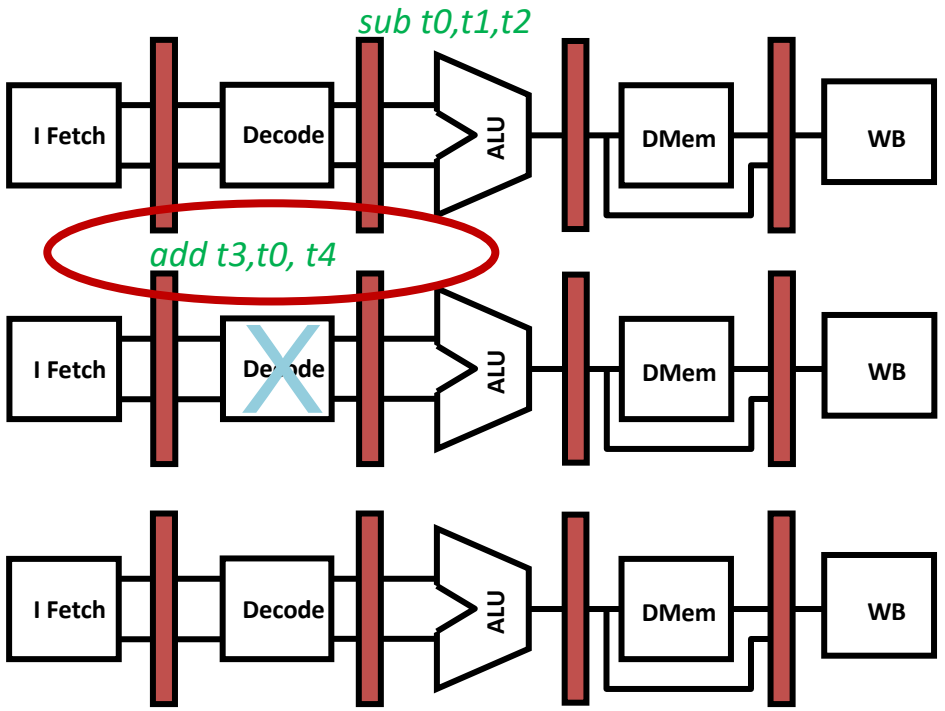

*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions



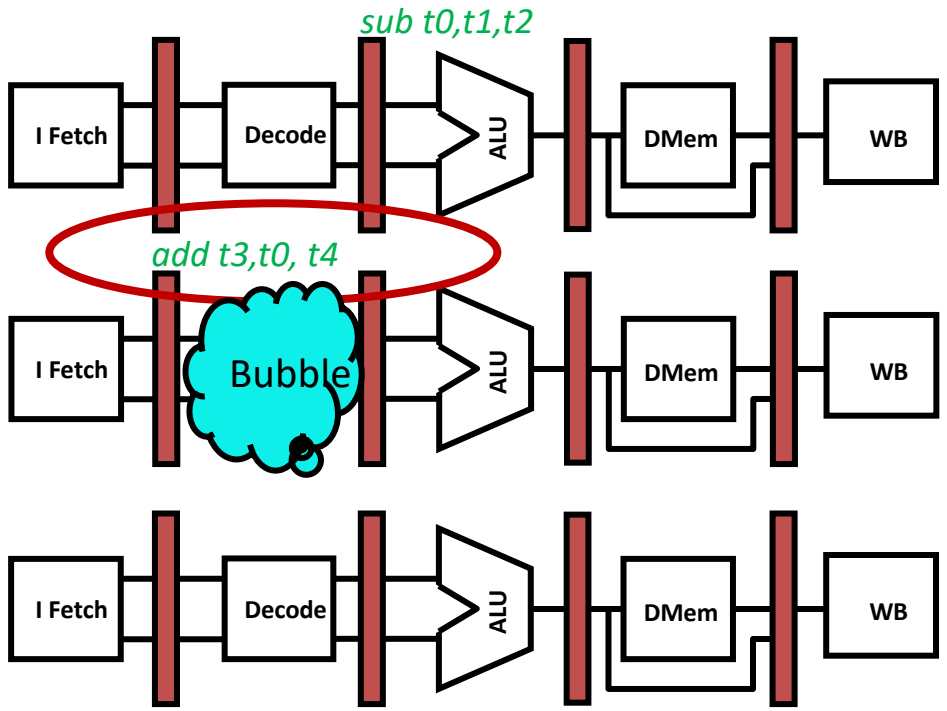

*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions



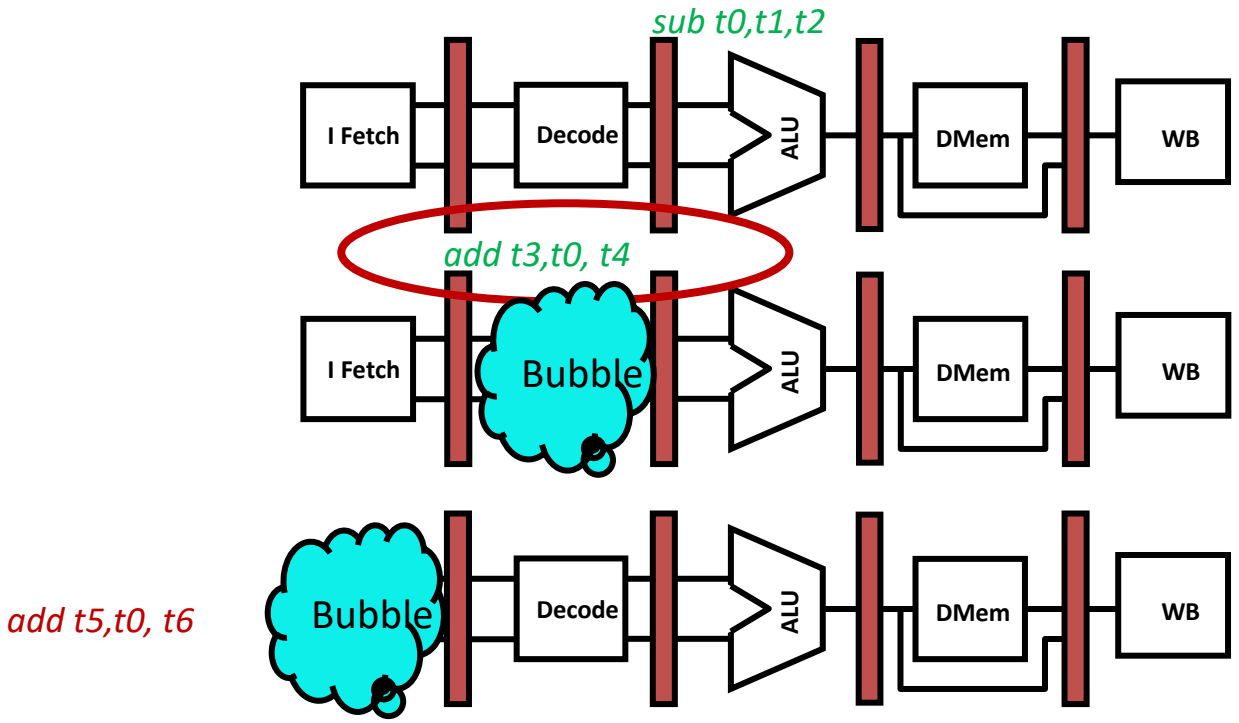

*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions




*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

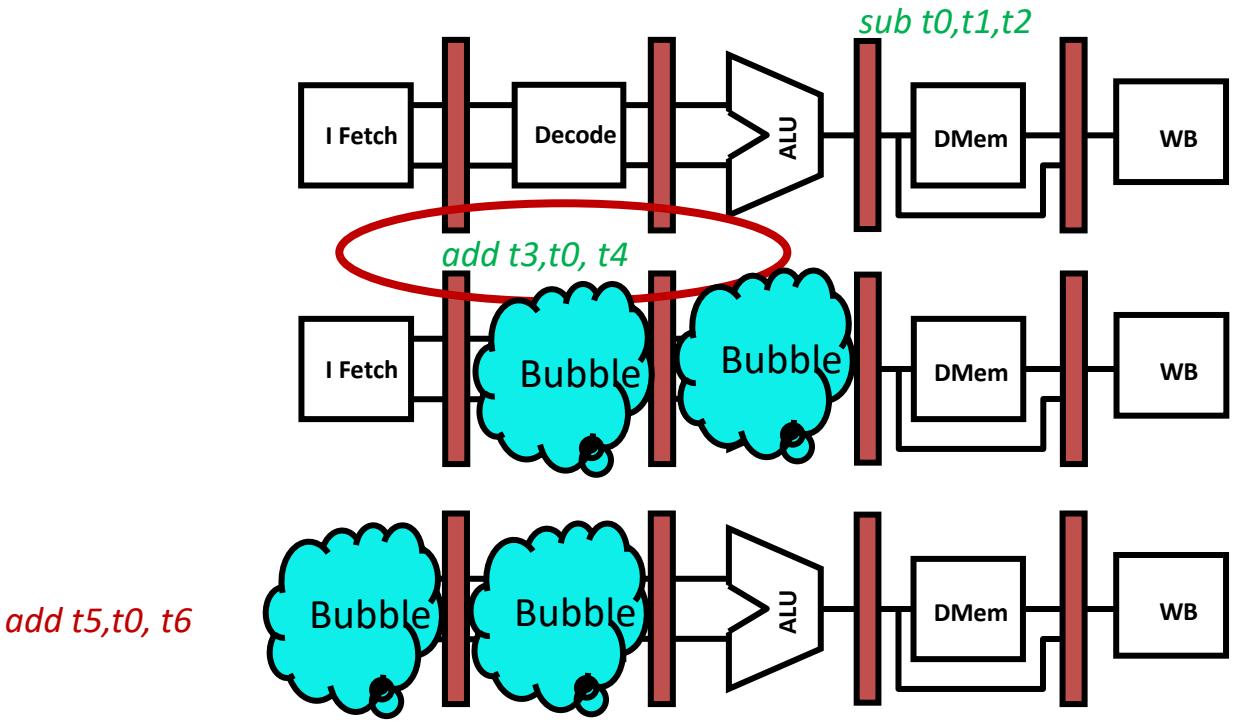
# Instruction Interactions




*add t5,t0, t6*

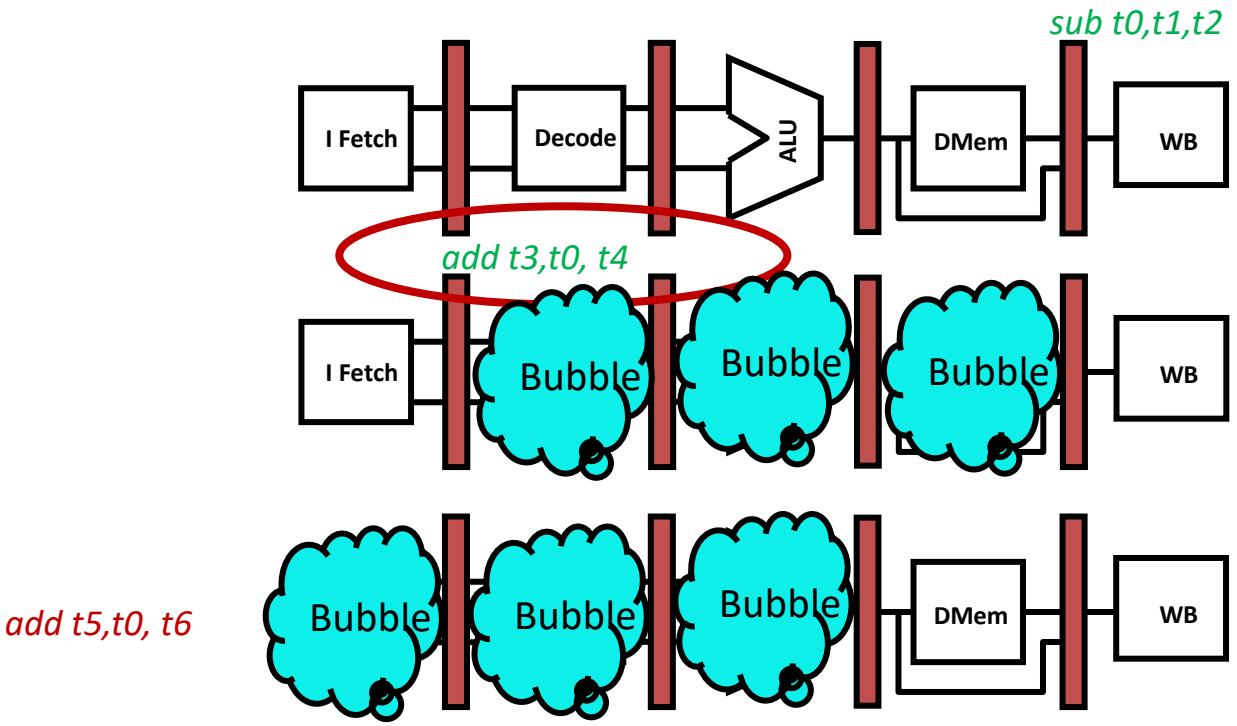
*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions




*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

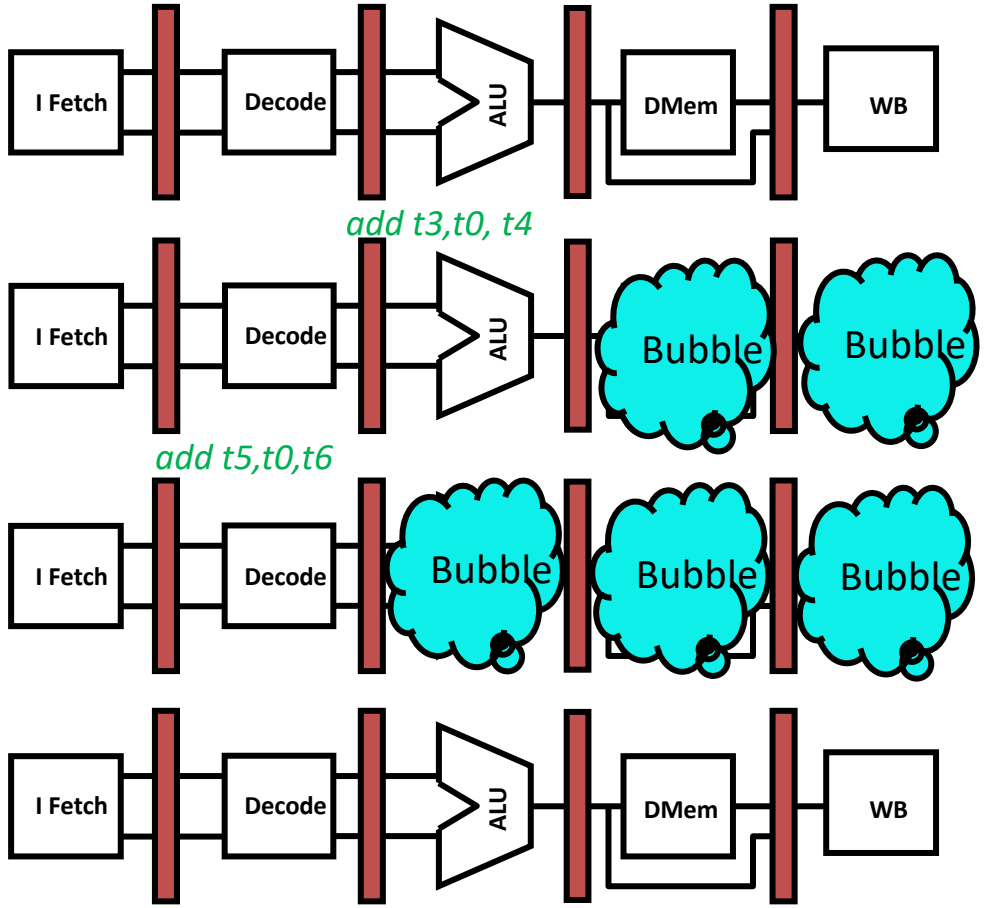
# Instruction Interactions




*add t5,t0, t6*

*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Instruction Interactions




*sub t0,t1,t2*
*add t3,t0,t4*
*add t5,t0,t6*
*sub s2,s3,t0*
*lw s4,16(t0)*
*add s4,s4,t7*
*sw s4,16(t0)*

# Stalled Stages and Pipeline

| | *time* t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 . . |
|---|---|---|---|---|---|---|---|---|
| ($I_1$) r1 ← (r0) + 10 | $IF_1$ | $ID_1$ | $EX_1$ | $MA_1$ | $WB_1$ | | | |
| ($I_2$) r4 ← (r1) + 17 | | $IF_2$ | $ID_2$ | $ID_2$ | $ID_2$ | $ID_2$ | $EX_2$ | $MA_2$ |
| ($I_3$) | | | $IF_3$ | $IF_3$ | $IF_3$ | $IF_3$ | $ID_3$ | $EX_3$ |
| ($I_4$) | | | | | | | $IF_4$ | $ID_4$ |
| ($I_5$) | | | | | | | | $IF_5$ |

*stalled stages*

*Resource Usage*

| | *time* t0 | t1 | t2 | t3 | t4 | t5 | t6 | t7 | . . . |
|---|---|---|---|---|---|---|---|---|---|
| IF | $I_1$ | $I_2$ | $I_3$ | $I_3$ | $I_3$ | $I_3$ | $I_4$ | $I_5$ | |
| ID | | $I_1$ | $I_2$ | $I_2$ | $I_2$ | $I_2$ | $I_3$ | $I_4$ | |
| EX | | | $I_1$ | nop | nop | nop | $I_2$ | $I_3$ | |
| MA | | | | $I_1$ | nop | nop | nop | $I_2$ | |
| WB | | | | | $I_1$ | nop | nop | nop | |

# Stalled Stages and Pipeline

- Pipelining was introduced to improve performance

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Cycles}}{\text{Instruction}} * \frac{\text{Time}}{\text{Cycle}}$$

- But with stalling, CPI will go up
- So we need a way to reduce stalling cycles!

# Next Learning Module

- CPU: Hazard Resolution