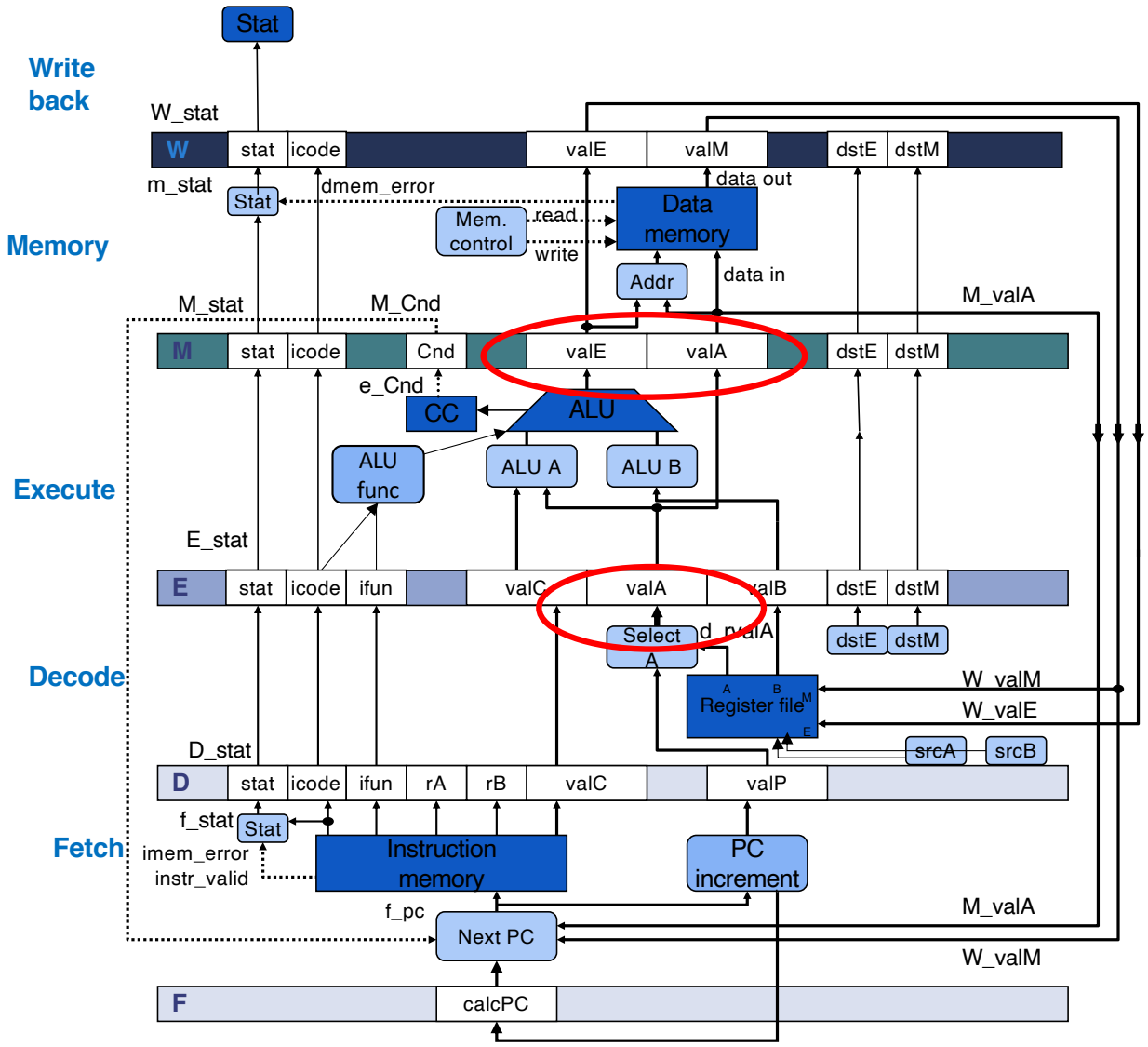
Today

- Topics
  - Stalling the processor to deal with hazards
  - What are bubbles in the pipeline?
- Learning outcomes
  - Given a sequence of instructions, determine for how many cycles the processor will stall or how many bubbles are inserted into the pipeline
- Reading
  - 4.5.5 (until “Avoiding Control Hazards”)

Recall

Consider the following sequence:

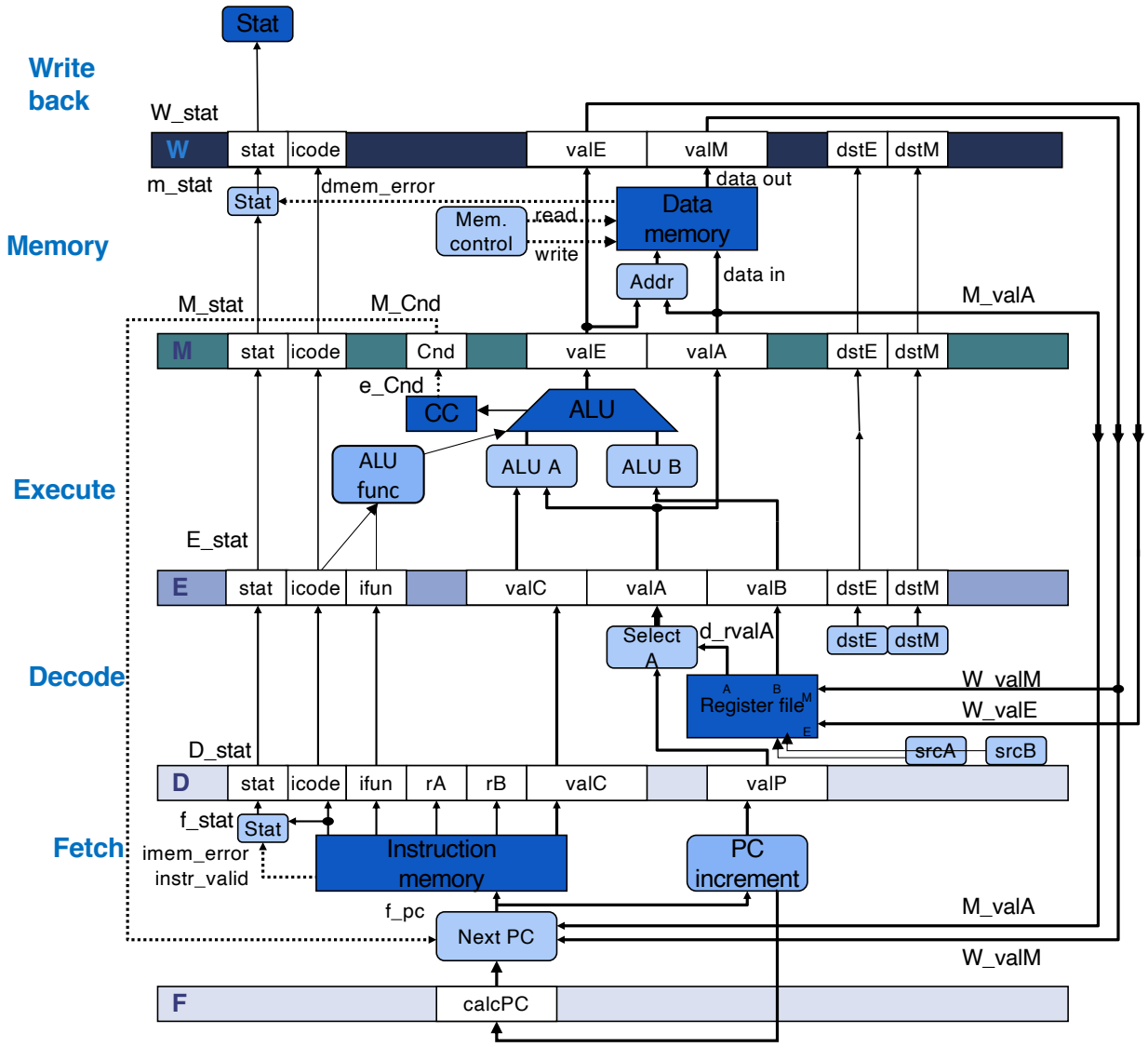
ADDQ %rax, %rbx
ADDQ %rbx, %rcx



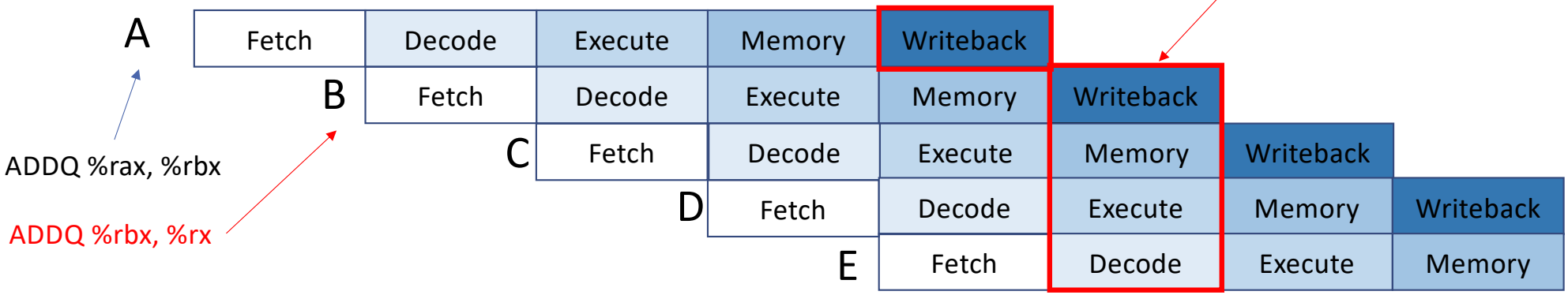
Stalling

Stall: Make instruction wait until its data is available.

Q: For how long do we need to stall so that the second add gets the right data?

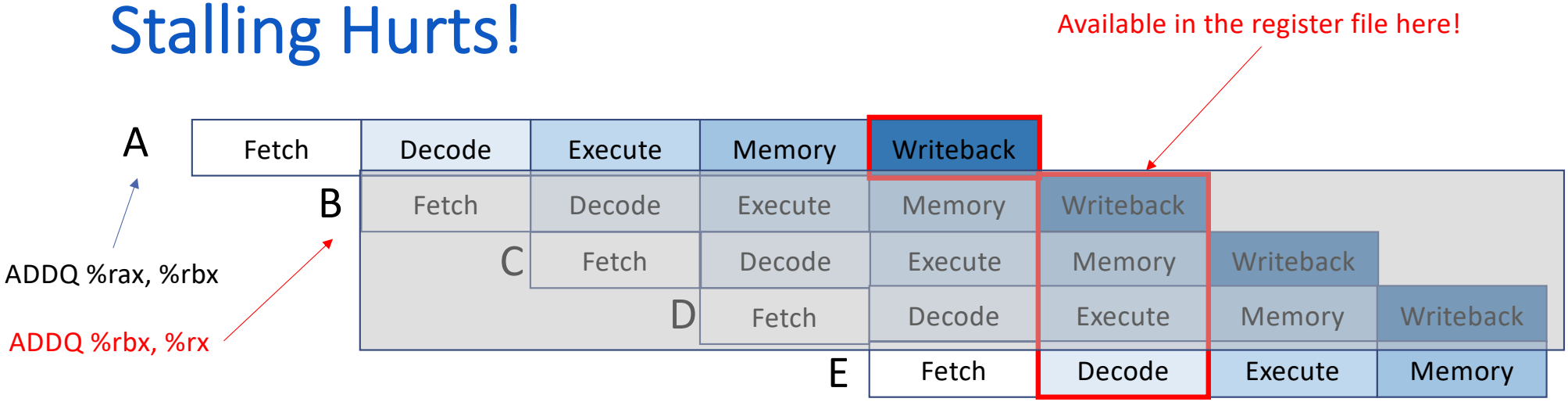


When will %rbx be written?



When will Instruction A's new value get written into %rbx?

Stalling Hurts!



Good news: We get the right answer!
Bad news: We wasted three cycles!

Performance that accounts for stalling

- So far, we've described the performance of our processor (latency, throughput).
- But what about the performance of our program?
- We use cycles per instruction (CPI) to describe how efficiently a program is executing on a processor.
- If we never stalled, what CPI would a program achieve?

Performance that accounts for stalling

- So far, we've described the performance of our processor (latency, throughput).
- But what about the performance of our program?
- We use cycles per instruction (CPI) to describe how efficiently a program is executing on a processor.
- If we never stalled, what CPI would a program achieve?
  - 1 – this is the best we can do on the y86.
  - Real processors are superscalar, which means they can retire more than one instruction per cycle, and therefore, could achieve < 1 CPI.