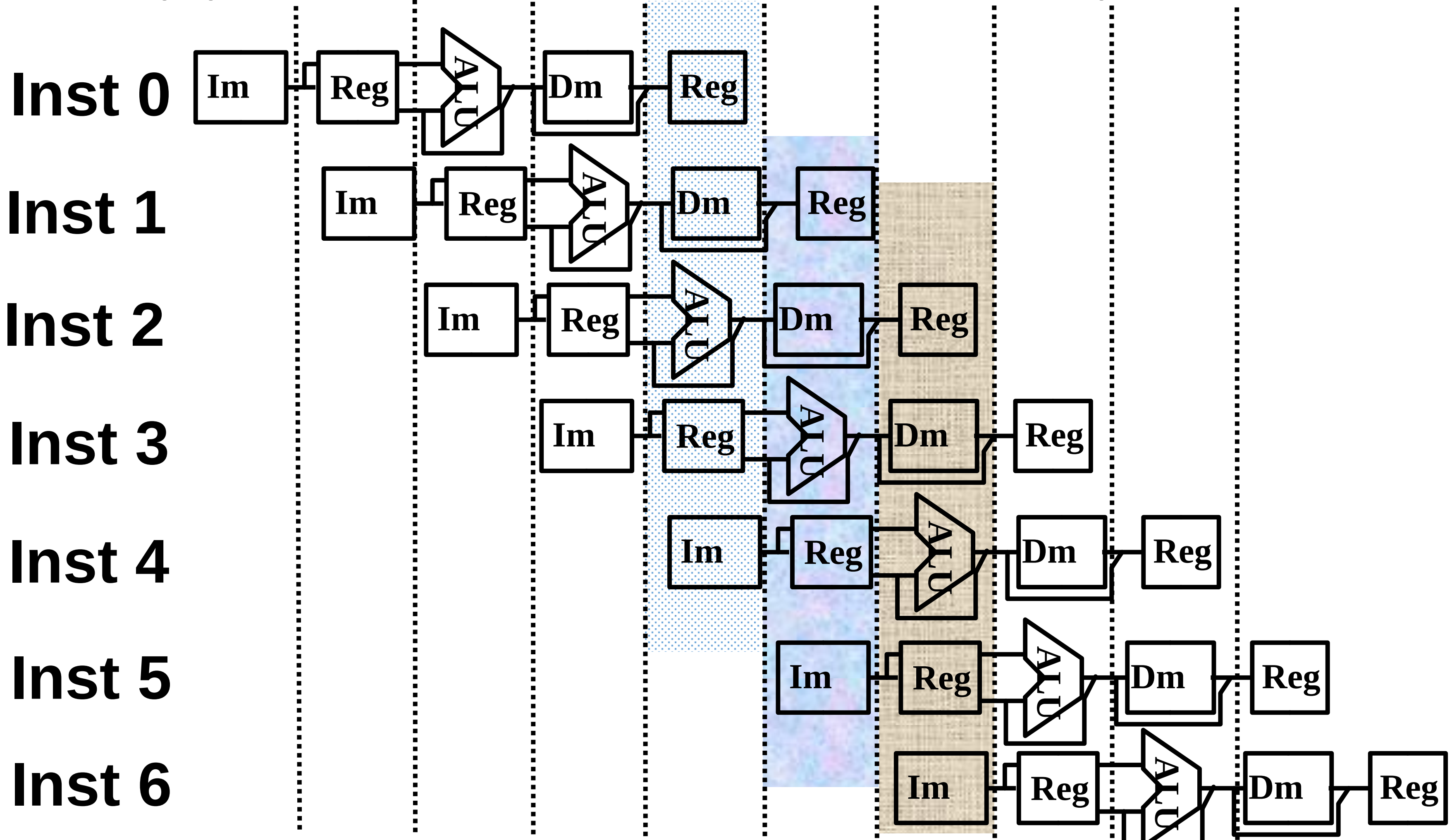
Understanding the performance of pipeline

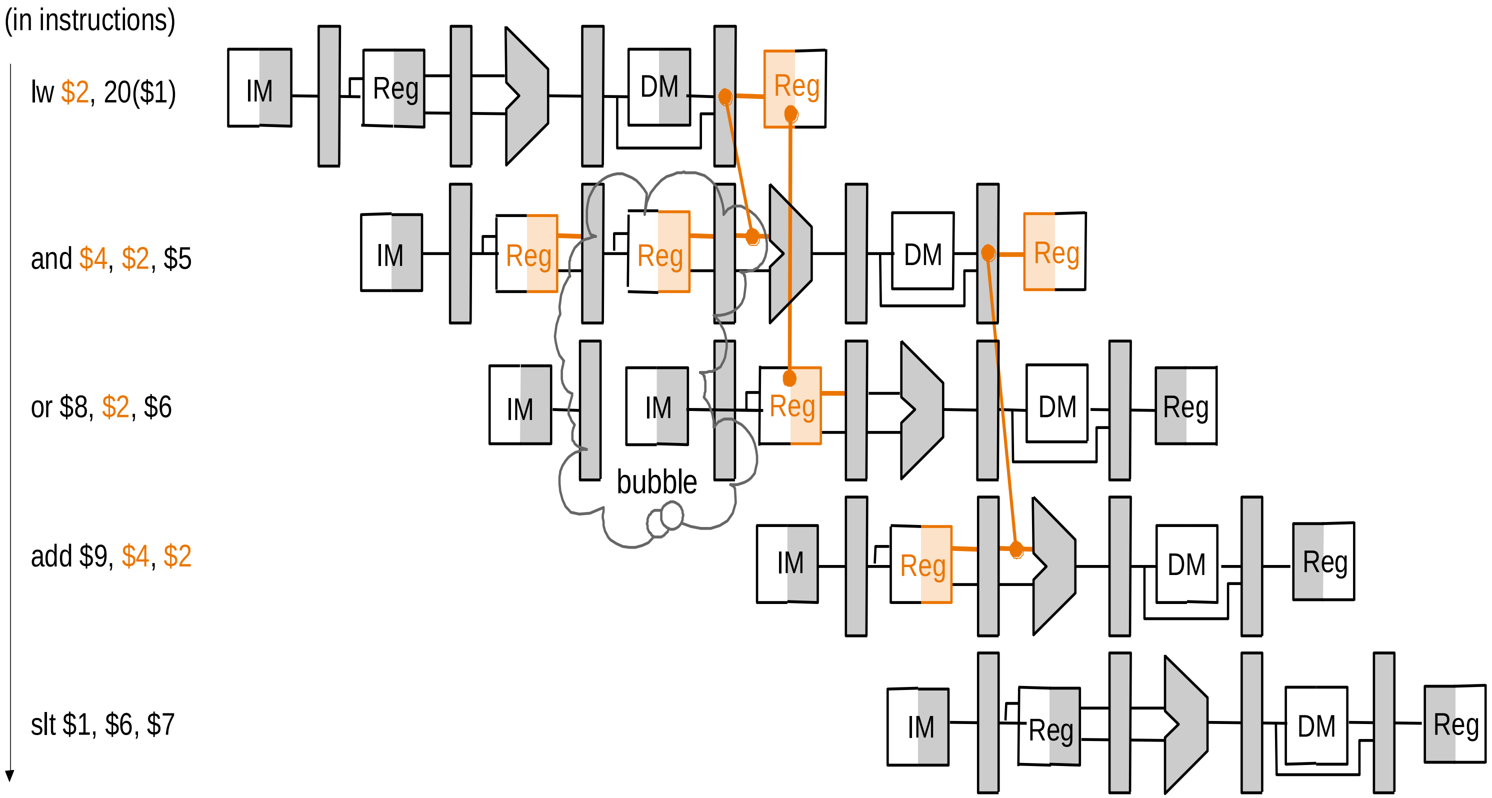
- Ideal CPI for pipelined processor: CPI = 1, since for each cycle, there will be one instruction taken into the pipeline and one instruction taken out of the pipeline (i.e., finished execution and quit)



Understanding the performance of pipeline

- CPI on data hazard (stalling)
- When shall a stall happen?
  - The lw instruction and the next one has a RAW error. E.g.,
  - lw $2, 20($1)

    add $4, $2, $5
- Effects on performance?
  - Effectively, 'lw' takes 2 cycles if RAW happens
  - 'lw' still takes 1 cycle if no data hazard happens.



Understanding the performance of pipeline

- CPI on control/branch hazard
  - Without early branch decision
    - 3 cycle penalty for branch taken, i.e., beq takes 4 cycles
  - With early branch decision
    - 1 cycle penalty for branch taken, i.e., beq takes 2 cycles
  - Normally when we talk about dealing with control/branch hazard, we use early branch decision.
  - j (jump) always takes 2 cycles, since branch always taken.