

Phase 3- Execute, Writeback **Deadline:** Sun., Nov. 30, 11:59 PM and Commit

Fall 2025

(Upload it to Gradescope.)

Notes

- You will be implementing these modules and reusing them throughout the period of the project
- You are allowed to use AI tools for writing code as long as you understand what you're implementing and properly indicate what has been written by the tool.
- This phase (and all future phases) can be done in groups. Your group should have been listed under this document: [Honors Group](#)

Steps:

- The execute stage:
 - a. Before executing, note that we are doing read after issue, so there should be a register read step. You don't need a separate pipeline stage between reg read and execute.
 - b. ALU Unit (consider only one FU that takes 1 cycle)
 - c. Address Generation + LSU
 - i. Partial LSU, we will only be implementing the Load instruction
 - ii. Tips:
 - You should implement your memory unit as a BRAM. Consider that access to BRAM takes 2 cycles.
 - d. Branch Unit
 - i. Compute the branch outcome and destination. This needs to invoke the misprediction/flushing if misprediction is detected. For now, you don't need to implement the flushing/recovery mechanism (i.e., phase 4), so just create a flag that is connected to the output.
 - e. Tips:
 - i. Remember, if something is inside these units, there should be a flush wire to set these data as invalid / rewrite these data (in case recovery is needed).

- The writeback stage:
 - a. Tips:
 - i. The writeback should fill up the data in the PRF (i.e., data should be forward to PRF). The PRF can be updated out-of-order. In phase 4, we will introduce a checkpointing mechanism in case recovery is needed.
 - ii. You need to implement the wakeup logic in RS too (i.e., dependent instructions should be updated). Assume that each FU (ALU, LSU, and Branch) can update the RS and PRF separately.
- The commit stage:
 - a. Tips:
 - i. Always remember, the commit is in order. Hence, it should look at the ROB, check whether an instruction is at the head, and then commit.
 - ii. Note that the commit stage shouldn't stall (i.e., in terms of a valid-ready handshake, it should have valid but not ready).
 - E.g: if a commit comes into the rename, the rename must accept the commit, and not stall it.
 - This is to prevent deadlocks (Google it up)
 - iii. The major part of the commit is issuing a recovery. This will be implemented in Phase 4. This also includes misspeculation (i.e., ROB needs to be flushed).
 - iv. For now, commit is pretty straightforward (i.e., just removing from ROB) and broadcasting it to rename and PRF.

What to submit

1. You need to upload all your code to a GitHub repository and share the link on Gradescope (assuming you have already done that). Share the commit number in your report.
2. A short report (a PDF file) that explains how each person in the group contributed to this phase.
3. **A short diagram showing your block diagram design (for all stages). Include this in your report!**