Weekly Report: Week of 04/18 - 04/25

Project: Superscalar Processor Implementation

Meeting with Martin (04/18):

We met with Martin on Thursday to discuss the initial steps for implementing our superscalar processor. Martin provided valuable insights and tips on how to approach the project.

Progress Made:

1. Fetch and Decode Stage:
    - Implemented function (notLineBoundary function) to handle line cache for instructions.If the instruction 1 PC is a multiple of 60 (assuming 60 is the last PC of our cache line), we return False; otherwise, we proceed with fetching two instructions.
    - Added the Superscalar FIFO and used it to enqueue instruction 1 and instruction 2.
    - Create OneTwoWords struct to receive the two instructions from one line. Ins 1 being a word, inst 2 being a Maybe type, depending if inst 2 is valid we would use it or not.

Questions and Implementations:

- Memory Requests: Explored options for making two memory requests for two instructions in the same cycle. Options included modifying the cache to handle dual requests or using a two-ported BRAM for the cache.
- Decode Stage: Discussed handling dependencies between instructions in the decode stage, including the need for stalls when dependencies exist.

Next Steps:

1. Execute Stage:
    - Explore possibilities of duplicating ALUs and outline misprediction scenarios for both instructions.
2. Writeback Stage:
    - Increment number of ports on register file and scoreboard to handle writebacks from two instructions.

Action Items for Next Week:

1. Refine decode stage implementation to handle dependencies and stalls effectively.
2. Continue discussions on execute and writeback stages, addressing any remaining questions and confirming implementation strategies.

Conclusion:

The week was productive in laying the groundwork for the superscalar processor implementation. We are progressing as planned and addressing key questions to ensure a smooth development process.