**General Structure of Report for CPU:**

*Section 1:* CPU design and architecture

1. Outline the CPU structure (i.e Harvard implementation, what that means etc).
2. Break the CPU down into blocks and give a general overview of how it functions, and how each component functions. (goal is to simulate a report to clients, so specifics and optimisations need to be demonstrated and talked about as well, why this implementation is efficient/cost effective, what benefits etc.)
3. Have diagram(s) of the CPU and components for a visual aid.
4. Talking about clock cycles, how specific functions would work, differences in implementation vs simulation (i.e in Verilog everything compiles immediately but in real life there would be a time delay involved).
5. How it functions (How instructions are handled by the cpu, storage, ease of maintenance, aspects relating to component replacement/ compatibility etc.)

*Section 2:* CPU Testing

1. Methodology: go over how specifically the tests were done. What was used (testbench in iverilog, testcases which checked for general functionality and edge cases, simple vs complex testing to test base capabilities vs required maximum capabilities etc.)
2. Go over the stesting process. For each instruction, we can have a two stage process:

Stage 1:

Ad hoc development testing: Where we have developed an instruction, we then go through tests on that instruction alone, making test cases but doing simple tests as well to ensure functionality in a vacuum. Once the instruction has no failures in the isolated testing, it is considered complete and ready for the main testing phase of all instructions sequentially.

Stage 2: Sequential Testing of everything at once. Instructions run through a series of test cases developed for each instruction. Once completed, a pass or fail is given depending on whether the instruction passed or failed a test. From this point, two options exist:

Option 1: All pass, in which case testing is completed and the instruction set works.

Option 2: One or more fail, in which case the instructions in question are examined, test cases are reviewed to ensure that there wasn’t an error with the test case, and the Sequential testing is run again. This step repeats until there are no fails produced.

NOTE: Important to make distinction between instructions: Some are **Required** for testing:

These include: lui, addiu, jr (for jr $0), any other explicitly important instructions…

I.E: Split instruction set into those required for testing the other instructions, and those not.

*Section 3:* Modelling the CPU as a physical device:

1. Run the Cyclone test to see how it functions, obtain Area Timing Summary.
2. Interpret the result, compare to existing CPUs.
3. Reach a conclusion on the performance of the CPU compared to others. Speculate and speak on reasons that the CPU performs particularly well or particularly poorly in comparison to other CPUs. Cite examples (instructions) that are well optimised or poorly optimised as reasons for the performance simulated.

General: Citation of any external sources used.

Flowchart reference:

Diagram

Description automatically generated