ASSIGNMENT – MIPS SOFTCORE IMPLEMENTATION IN VERILOG

Goal

To extend the Verilog programme from prac 1 and to apply knowledge of MIPS from prac 2 to build a basic single cycle data path. You may work individually or in pairs.

Requirements

You are required to implement a single cycle data path for a 32-bit MIPS processor in Verilog. You may refer to figure 4.24 in Computer Organization and Design 4th edition (available electronically in the UTas library). You will be given a test bench file along with some modules and will need to implement the following modules yourself:

- Control (including the following instructions: r-type, lw, sw, beq, j, addi)
- ALU Control
- ALU (you may extend/modify your final ALU from prac 1)
- Register
- Single Cycle Datapath (including variables, wiring and multiplexing – some parts have already been implemented to get you started)

You must write a test programme in MIPS assembly, compile it to machine language in hexadecimal format (you may do this in MARS or any other MIPS compiler), load and run this programme in the provided **`MIPSSingleCycle_tb.v`** test bench file. Another test bench is also provided to assist you in testing your ALU; you may wish to extend this if you have other test cases you'd like to test.

Make sure you have a look at the test bench since it implements memory management and your data path will need to be built accordingly.

Submission

You must submit a single report file in PDF format. The report should include the following sections (you may add more sections and/or add subsections if needed):

- **Introduction:** Introduce the project and summarise the report. 250 words or less.
- **Technical Detail:** Provide a brief technical explanation to address criteria 2 and 3. Do not include code listings here.
- **Testing:** Briefly explain how you tested your implementation in terms of addressing all requirements. Refer to criterion 4
- **Conclusion:** Include one paragraph to summarise the report. Optionally you may add a paragraph to mention any other details or issues that were not addressed in the body of the report.
- **Code Listings:** Include all code files. Each file must have its own heading. Code must be formatted neatly and preferably be syntax highlighted[1]. Do not use screenshots for code listings. Refer to criteria 1 and 4.

[1] You can copy your code into Visual Studio Code (or any similar IDE) and then copy from the IDE into Word to get syntax highlighting.

ASSIGNMENT – MIPS SOFTCORE IMPLEMENTATION IN VERILOG

Assessment

You must produce a report and submit it to the assignment dropbox in MyLO.

The following criteria will be assessed in this assignment:

Criterion	Description	Learning Outcome
1	Simulate an advanced digital circuit.	ILO2
2	Calculate performance of advanced digital circuit.	ILO2
3	Present a suitable design to meet requirements.	ILO5
4	Evaluate the design to ensure it meets the requirements.	ILO5

Criterion 1 (30%): Simulate an advanced digital circuit.

For full marks you must demonstrate the following through your report. Marks will be deducted for each item not adequately demonstrated.

1. Provide full code listings.
2. Meet requirements through coding Verilog with correct logic.
3. Adequately document your code with comments.
4. Use appropriate datatypes (reg, wire, bit-widths, arrays).
5. Make appropriate use of constants.
6. Structure your code appropriately for ease of reading.

Criterion 2 (10%): Calculate performance of an advanced digital circuit.

For full marks you must demonstrate the following through your report. Marks will be deducted for each item not adequately demonstrated.

1. Present the usage of FPGA internal components
2. Comment on the performance requirements of your design (e.g. clock speed limitations, size requirements) and the suitability of the FPGA used in the Basys3 development boards. You may need to do some research to complete this point.

Criterion 3 (30%): Present a suitable design to meet requirements.

For full marks you must demonstrate the following through your report. Marks will be deducted for each item not adequately demonstrated.

1. Present a diagram to illustrate the hierarchy of modules – include inter-module dependencies (e.g. lines or arrows connecting modules).
2. Explain the hierarchy of modules, how they work together and interact.

3. Provide a correct description of all modules in the design (include the modules provided to you as well as those you have written yourself). Describe their inputs, outputs, types and functionality.

Criterion 4 (30%): Evaluate the design to ensure it meets the requirements.

For full marks you must demonstrate the following through your report. Marks will be deducted for each item not adequately demonstrated.

1. Develop a test programme in MIPS assembly that tests a range of values for all required instructions.
2. Explain how the test programme adequately tests for all requirements.
3. Present successful results from running the test programme via the given test bench (use your own MIPS assembly programme from point 1 above).