Digital Systems Design (Complex Engineering Problem)

CEP Statement:

You are tasked with designing and implementing a single-cycle MIPS processor on the Nexys A7 FPGA. The processor should be capable of executing a specific set of MIPS instructions. The processor should be designed to receive instructions from on-board memory, execute them, and display the results through the on-board LEDs and 7-segment displays.

Specific Requirements:

1. **Instruction Set Architecture (ISA) Subset:**
   - Implement the following MIPS instructions:
     - R-type: `add`, `sub`, `and`, `or`, `slt`
     - I-type: `addi`, `lw`, `sw`, `beq`
     - J-type: `j`
   - Implement a simplified 32-bit instruction and data memory.
   - Implement a register file with 32 registers.
2. **Single-Cycle Implementation:**
   - Design the processor as a single-cycle implementation.
   - Ensure that all instructions are executed within a single clock cycle.
   - Clearly identify and address the critical path.
3. **Verilog Implementation:**
   - Write synthesizable Verilog code for the processor.
   - Use modular design principles to improve code readability and maintainability.
   - Implement the control unit, ALU, register file, and memory modules.
4. **Nexys A7 Integration:**
   - Utilize the on-board memory of the Nexys A7 to store instructions and data.
   - Use the on-board LEDs to display the status of the processor (e.g., halt, error).
   - Use the 7-segment displays to display the contents of a selected register or memory location.
   - Implement a simple program counter (PC) that increments after each instruction.
   - Implement a way to load a simple program into the memory. This could be done through a text file that is read by the test bench, or through a predefined memory array in your verilog code.
5. **Testing and Verification:**
   - Develop a comprehensive testbench to verify the functionality of the processor.
   - Test all implemented instructions with various test cases, including boundary conditions.
   - Demonstrate the execution of a simple MIPS program on the Nexys A7.

Deliverables:

1. **Verilog Code:**
   - Well-documented Verilog code for all modules of the MIPS processor.
2. **Testbench:**
   - Verilog testbench for verifying the functionality of the processor.

    - Test cases covering all implemented instructions and corner cases.
3. **Nexys A7 Project:**
    - Vivado project with the implemented MIPS processor.
    - Bitstream file for programming the Nexys A7.
    - Clear instructions on how to load and run the program on the Nexys A7.
4. **Design Document:**
    - Detailed description of the processor architecture.
    - Explanation of the control unit logic.
    - Critical path analysis.
    - Explanation of the testbench and test results.
    - Explanation of how the program is loaded and ran.
5. **Demonstration:**
    - A live demonstration of the MIPS processor running on the Nexys A7, executing a simple program.

Grading Criteria:

- Correctness of the implementation.
- Functionality of the implemented instructions.
- Efficiency of the design.
- Clarity and readability of the code.
- Completeness and effectiveness of the testbench.
- Quality of the design document.
- Successful demonstration on the Nexys A7.

Extra Credit:

- Implement additional MIPS instructions.
- Implement a more complex program on the Nexys A7.
- Improve the performance of the processor.
- Implement a simple debugging interface.

Tools:

- Xilinx Vivado Design Suite.
- Nexys A7 FPGA board.

Important Notes:

- This assignment requires a thorough understanding of MIPS ISA and digital design principles.
- Pay close attention to the timing constraints and critical path analysis.
- Use modular design principles to simplify the implementation and debugging process.
- Start early and break the problem down into smaller, manageable tasks.
- Thorough testing and verification are crucial for ensuring the correctness of the design.

Course	**WK**	**PLO**	**WP**	**Blooms Learning Level**
DSD	WK3,WK4	3 (Design)	WP1,WP2,WP3,WP4	C5