Computer Architecture Project

Project Description

The project aims to design and implement a Programmable Interrupt Controller (PIC) based on the 8259-architecture using Verilog hardware description language. The 8259 PIC is a crucial component in computer systems responsible for managing and prioritizing interrupt requests, facilitating efficient communication between peripherals and the CPU.

Key Features:

1. **8259 Compatibility:** The Verilog implementation will closely emulate the behavior and features of the classic 8259 PIC, ensuring compatibility with existing systems and software.
2. **Programmability:** The project will include support for programming interrupt priorities and modes, allowing users to configure, using Command Words (ICWs) and Operation Command Words (OCWs), the PIC according to their specific requirements.
3. **Cascade Mode:** Implementing the cascade mode, where multiple PICs can be interconnected to expand the number of available interrupt lines, enhancing the scalability of the design.
4. **Interrupt Handling:** Efficient handling of interrupt requests, including prioritization and acknowledgment mechanisms, to ensure a timely and accurate response to various events.
5. **Interrupt Masking:** Implement the ability to mask/unmask individual interrupt lines to control which interrupts are currently enabled.
6. **Edge/Level Triggering:** Support both edge-triggered and level-triggered interrupt modes to accommodate different types of peripherals.
7. **Fully Nested Mode**: Implement Fully Nested Mode, allowing the PIC to automatically set the priority of the CPU to the highest priority interrupt level among the currently serviced interrupts.
8. **Automatic Rotation**: Extend the priority handling mechanism to support automatic rotation even in scenarios where lower-priority interrupts are being serviced.
9. **EOI**: Implement the EOI functionality, allowing the PIC to signal the end of interrupt processing to the CPU.
10. **AEOI**: Implement the EOI functionality, allowing the PIC automatically to signal the end of interrupt processing to the CPU.

11. **Reading the 8259A Status**: Implement the capability to read the status of the 8259A PIC.
12. **Simulation and Testing:** Comprehensive testbench development for simulating and validating the functionality of the Verilog-based 8259 PIC. This includes testing various interrupt scenarios and ensuring proper interaction with other system components.
13. **Documentation:** Thorough documentation detailing the design specifications, module functionalities, and usage guidelines to facilitate understanding and future development.

Required

The students should be divided into groups/teams of max 6 members.

Expected delivery

1. Each group should deliver the following.
   a. A **report** in pdf format describing the project details (e.g., block diagrams, signals description, and implementation).
   b. The **source code** of the project and the **testbench** used to verify it.
   c. **5-minute presentation** per each group. All the team/group members should participate in the presentation.
2. The report should contain.
   a. Cover with team member names and team members IDs.
   b. Table of contents.
   c. Block Diagram for your design.
   d. Description for all signals used in your design in tabular format (signal, description).
   e. Brief description of the testing strategy.
   f. Snap shots of simulation waves.
   g. A part that describes the work for each team member (who did what).
   h. Any additional information, such as enhancements made in the implementation.
   i. Any figure in the pdf should have a caption under it and any existing figure should be referred in the written paragraphs.
3. Presentation should contain brief description about the done work associated with waveforms screenshots and live demo.
4. Report, source code, and test bench (not the presentation) should be delivered in a zip file that will be uploaded on LMS.
5. The team should push their codes on **GitHub** repository. Each team member should contribute and push his/her part of the code on GitHub.
6. The team leader should attach the GitHub repository link of the team in the report.

Delivery deadline date

1. The deadline of the delivery date will be on **26th Dec.** at **11:59 PM**. The submission will be through LMS.
2. The presentations will be held on this preliminary date **27th Dec.** (the final date will be announced later).

Evaluation

1. 25% of the marks for individual contribution specially the GitHub repository contribution.

2. 75% of the marks for the project team.

Note: A team member without contribution on GitHub repo will get ZERO.