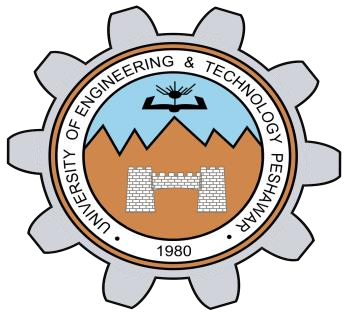
**Lab report 01**

Introduction to SPIM Simulator for the MIPS Assembly Language



**Computer organization and Architecture**

**CSE 304L**

**Fall 2024**

**Submitted by: Naveed Ahmad**

**Registration No.: 22PWCSE2165**

**Class Section: B Semester :5th**

*“On my honor , as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”*

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr. Ammad Khalil**

Oct 7, 2024

Department of Computer Systems Engineering

University of Engineering and Technology Peshawar

**DIFFERENCE BETWEEN COMPUTER ORGANISATION AND ARCHITECTURE**

**Compute organization.**

* Refer to the operational unit and they inter connection that relies on the architecture specification
* Deal with the physical and logical arrangement of hardware
* Focuses on the implementation of hardware and the physical system operation

**Examples**

1. Control signal
2. Interpret
3. Memory hierarchy
4. RAM size
5. IOP devices
6. Timers

**Computer architecture.**

* Refers to the abstract structure and behaver of the computer as seen by the programmers
* Focuses on the software interaction with the hardware and what functionality the hardware provides
* Involves the design and functional design of the system mainly from the programmers’ point of view

**Examples**

1. Instruction set design
2. Memory addressing
3. Instruction execution
4. CPU functionalities and capabilities

**Types:**

1. **CISC ARCHITECTURE**

* CISC Stands for*Complex Instruction Set Instruction*
* Provides a rich set of instruction allowing the possessor to perform complex operation. In a single instruction the goal is to reduce the number of instructions per program for making each instruction more capable
* instruction more complex often performing multiple operation in a single instruction.
* Variable instruction length.
* Access the memory directly

1. **RISC**

* *Reduced Instruction Set Computing*
* Simplify the instruction set of the CPU making each instruction in a single clock cycle
* RISC processors are highly pipelining
* Load and store

**DIFFERENCE BETWEEN RISC AND CISC ARCHITECTURE:**

|  |  |
| --- | --- |
| ***RISC*** | ***CISC*** |
| *Fewer and simple instruction* | *Many and complex instruction* |
| *Fixed instruction length* | *Variable instruction length* |
| *Each instruction executes in one cycle* | *Instruction can take multiple clock cycle* |
| *Load and store architecture* | *Direct memory access with most instruction* |
| *Highly optimised for pipelining* | *More difficult to pipelining due to complex city* |
| *Simple hardware with few transistor* | *More complex design* |
| *MIPS, arm, power PC* | *Intel* |

**MIPs:**

* Stand for micro processor without interlock pipeline stages
* Follow the RISC design principle
* MIPS processors are developed in early 1988
* Famous for their simplicity efficiency and pipeline design
* They are widely use in embedded system and high-performance application
* They are fixed length instruction
* Load store architecture
* Each register 32 bit

**QtSpim simulator:**

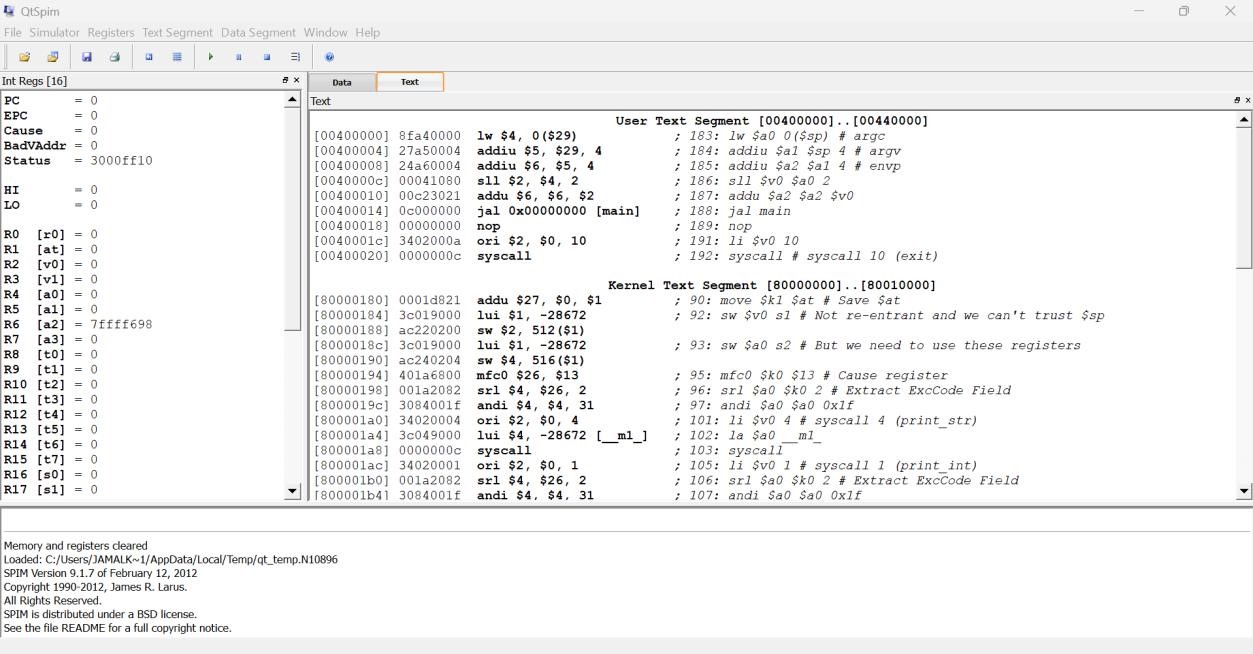
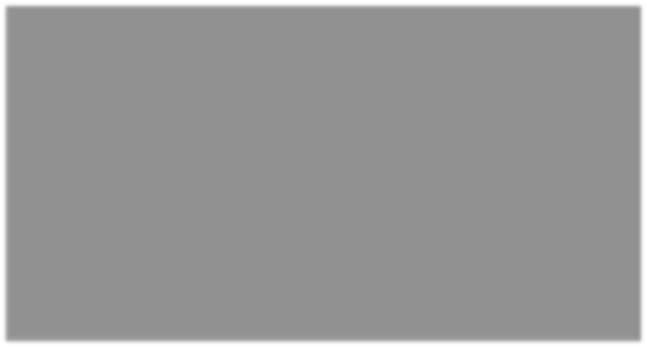
QtSPIM is a simulator that runs MIPS assembly language programs. It helps you understand how the MIPS processor works by simulating its operations.

**KEY POINTS:**

1. **Simulates MIPS Processor:** QtSPIM simulates the behavior of a MIPS processor, which is commonly used in computer architecture education.
2. **Runs Assembly Code:** You can write and run MIPS assembly language programs on it.
3. **Registers and Memory**: It shows how data is stored in registers and memory when the program runs.
4. **Simple Interface**: QtSPIM has a simple interface where you can load, run, and step through your assembly programs.
5. **Helps Debugging**: You can find and fix errors in your code by observing the changes in registers and memory.

**Source File Name:**

* Here **source file name** is the name of the file where we write our MIPS assembly code. It usually ends with .asm or .s. we load this file into QtSPIM to run our program.



* In the QtSPIM screenshot, the Data and Text segments refer to different parts of memory where the program's components are stored:

**TEXT SEGMENT (RIGHT SIDE):**

* This section contains the **program instructions**, or the MIPS assembly code, that the CPU will execute. It’s the part where the logic of the program is defined (e.g., addiu, lw, syscall).

**DATA SEGMENT:**

* This section is not fully visible in the screenshot but typically holds the **data** the program uses, such as variables and constants. It includes static or global variables and memory used during program execution.
* ***QTsipm*** is used for MIPs architecture. QT refers to QT farmwork which is used for graphical user interface
* *SIMPs* is revers version of MIPs.
* *In simple term computer architecture defines what a computer does while computer organization explains hoe it does it.*