# **Address Translation Simulation**

This project simulates virtual-to-physical address translation in a computer system using techniques like **1-level page tables**, **2-level page tables**, and a **Translation Lookaside Buffer (TLB)**. The simulation also demonstrates the effects of TLB hits, page table hits, and page faults on access times.

## **Features**

* **Virtual Address Translation**:
  + Divides virtual addresses into page numbers and offsets.
  + Translates virtual addresses to physical addresses.
* **Support for TLB**:
  + Caches page table entries to optimize access time.
  + Simulates TLB hits and misses.
* **1-Level and 2-Level Page Tables**:
  + Supports different page table structures for translation.
* **Graphical User Interface (GUI)**:
  + Built using Tkinter for an interactive experience.

## **Requirements**

### **Software Prerequisites:** Python 3.x

**Python Libraries:** Install the required libraries using pip: numpy

**Project Files:** main.py, simulation.py, gui.py, requirements.txt

## **Usage**

### **Command Line Interface (CLI)**

Run the program: python main.py

1. Enter a virtual address (e.g., 1024) when prompted.

Run the GUI application: python gui.py

1. Enter the virtual address in the input box.
2. Click the **Simulate** button to view results.

## **Testing**

### **Sample Inputs**

* **Virtual Address**: 1024
  + Expected Output: Physical address with TLB or page table status.
* **Virtual Address**: 8192
  + Simulates translation using page tables or TLB.

### **Expected Results**

* If the page number is in the TLB or page table:
  + Returns the physical address and access time.
* If not mapped:
  + Reports a **Page Fault**.