Atelier Group

Aretmisia Atelier

A Pipelined Processor Simulated In Python (Based on MIPS Architecture)

Amir Mohammad Tavakkoli[1], Aref Sayareh[2], Amir Masoud Baghi[3]

Overview

Artemisia is a 32-bit pipelined processor which simulated the behavior of MIPS I processor, written in Python and based on a logical circuit simulation library, named as Lucretia, which was also developed by Atelier Group. This processor was intended as a course project for Computer Architecture in Spring 2020, offered by Shiraz University.

Lucretia (Logical Circuit Simulator)

This open-source Python library, which was also developed by Atelier Group, provides various gates and logical components with the aim of simulating hardware. This library offers simple and unified interfaces helping users to focus on productivity rather than complexity. Lucretia was developed based on real hardware design to simulate combinational as well as sequential circuits. The modular structure of this library provides extendibility for additional components and further modification. It is also worth noting that Lucretia is under active development, as of now. Just call the Logic, and it will do the magic!

Compiler

Alongside this project, a compiler was also developed. This compiler provides rich support for decoding common MIPS instructions, including R-Format, I-Format, and J-Format. In addition to regular compiler translation, it also detects and handles dependencies and hazards between instructions and reorders code to prevent unnecessary stalls and minimize the amount of clock cycles needed for execution. The compiler's performance was optimized through using Regex functionalities, and this also provides support for label translation.

[1] tavakkoli.amirmohammad@gmail.com
[2] arefsayareh@gmail.com
[3] bamirmasoud@gmail.com

Artemisia (Processor)

This processor is a pipelined CPU based on MIPS I. The components of this processor were developed in an independent manner, which were assembled in their respective order and in the right stages. As Keras Founder and Creator, François Chollet, intended his library components to be manipulated like Lego bricks, we also employed the same approach to Artemisia. Forwarding and hazard detection is fully implemented to speed up the execution of instructions, and when needed, stalls are also introduced. We implemented cache using the concept of main memory with 4 banks, thus, benefitting of reading a word in a single cycle. A GUI was developed using tkinter library to control the clock pulse and monitor the values of memory and registers.

Links:

Link to Lucretia GitHub Repository