Part 1: Research Paper Exploration

1. Summary

Key Objectives

The research paper was written by *Ayaz Akram* and *Lina Sawalha* and was a survey of Computer Architecture Simulation Techniques and Tools. The main goal of the paper was to analyze a variety of Computer Architecture simulation techniques, and to classify each simulator into a different group based on its simulation model.

In comparison to previous surveys done on the same topic, Ayaz and Lina's survey aimed to be more detailed, as prior studies focused on simulators that were more focused on teaching or were not as comprehensive as Ayaz and Lina would have liked. Ultimately, they wanted their research to help novice computer architectures and system researchers and give them a proper guide to the available architecture simulation choices.

Methodologies

For this survey, the researchers chose to focus on six specific simulators (***gem5, MARSSx86, Multi2Sim, PTLsim, Sniper, and ZSim***).

To evaluate and classify each simulator, they chose to focus on the following key metrics:

- **Accuracy** *(Comparing the accuracy of the performance of the simulator to actual hardware)*
- **Performance** *(The speed of the simulator when emulating the target hardware)*
- **Level of Details** *(The number of details the simulator includes while representing the target hardware)*
- **Easiness of Development & Flexibility** *(The extent to which the simulator can be modified)*
- **User Friendliness** *(Ease of Use, and the ability to run different tests)*

The researchers do note that it is difficult to achieve optimal results in each category, as some of the metrics can contradict one another.

Findings

The researchers classified the simulators into three types:

- *Detail of the Simulation*
- *Scope of the Target*
- *Input to the Simulator*

They tested the accuracies of each simulator by comparing the simulation to the target hardware. The results displayed that ***Sniper*** was the most accurate simulation, as it had the least absolute error, and that ***ZSim*** was the fastest simulator for single core simulations.

The researchers do highlight the importance of ***validation,*** to properly ensure that the simulation is accurately representing the target hardware, and to ensure that the simulation does not include modding, specification or abstraction errors.

They do also note that the choice of simulator depends on goals of any given researcher, the purpose of their study and the specifications of the simulator.

2. Critical Analysis

Advantages of the Application of Simulators

- Allows one to assess a wide variety of design options, as simulation supports various ***ISA's*** *(Instruction Set Architectures)*.
- Its easy and effective in testing new research ideas.
- More practical than analytical models.
- Very easy to assess/analyze performance and power consumption of different processor models.

Challenges

- Validating the accuracy of a given simulator can be a bit difficult due to a scarcity in proper documentation.
- Contradictory metrics makes the comparison between simulation models difficult.

3. Application

This paper gave me insight into the important factors to consider when utilizing simulation models. I now understand the importance of ***validation***, as well as accounting for errors in accuracy and the discrepancy between the simulation and the real-life model, and most importantly properly assessing my research needs and which simulator would best suit them.

Part 2: Evaluating emulsiV

1. Overview

emulsiV is a visual simulator that's main purpose is to instruct students on the fundamentals of Computer Architecture. It was made for Virgule, which is a CPU core implementation modeled on the architecture of RISC-V.

As displayed on the website, the user is shown a simulated view of different components of the computer, such as the ***Memory (****which displays a table that shows the data in Rows and the assembly information in columns****)*** or the ***Datapath,*** which displays the pathway of a datapath, and plays animations to show how data moves between each of the functional units.

The main goals of emulsiV is to show students:

- *What data and instructions look like when stored in memory*
- *The steps executed to perform an instruction*
- *How data is transferred between functional units*
- *The standard input/output mechanisms and interrupts*

2. Advantages and Disadvantages

Advantages

- emulsiV is free to use: This is convenient for college and university students who might not have enough money to use more advanced simulators. Also, emulsiV was built to reflect the current state of the market and offers accurate information on the commercial standards of current processors.
- Easily accessible, given that one has internet access: If students have access to an internet connection, the website can easily be reached and requires no log in information. It is also open source.
- Simple Instructions: The website is easy to use, understand and implement, and its visual representation of how the CPU works provides clear information to the user on what is happening.

Disadvantages

- The scope of emulsiV is limited; It provides basic information about a typical CPU architecture and is independent of any specific hardware implementations. Therefore, it does not provide a fully accurate representation of how an actual CPU pipeline is organized, and the cycle is not fully accurate.
- It cannot serve as a general-purpose debugger, as it was primarily built to be an educational tool.

3. Comparison with Traditional Tools

If we compare emulsiV to WebRISC-V, we can see that both are open-source and very suitable to teaching students the basics of a computer architectural unit. For further comparison, the functionality of WebRISC-V is as follows:

- It provides a visual view of the schematics of a RISC-V processing unit.
- It can go back-and-forth in between execution of computer code, to better illustrate the inner workings of the processor's pipeline.
- Like emulsiV it is viewed on one webpage.
- It provides a more elaborate description of each element, along with its current state of operation, and this is shown to the user by placing the mouse on said element.

Overall, WEBRISC-V, like emulsiV, offers a simplified view of the CPU system, but is more detailed and comprehensive.

4. Practical Application

I would utilize emulsiV to help me visualize and understand the operations of a computer's processing system, and to gain a greater understanding of how the ***Memory*** and ***Datapath*** sections of the CPU operate.

References:

A survey of computer architecture simulation techniques and tools | IEEE Journals & Magazine | IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/8718630/ (Accessed: 19 July 2024).

Farseer (2022) *EMULSIV RISC-V simulator for Virgule Core*, *RISC*. Available at: https://www.riscfive.com/2022/08/21/emulsiv-risc-v-simulator-for-virgule-core/ (Accessed: 19 July 2024).

Emulsiv: A visual simulator for teaching computer architecture using the RISC-V instruction set: Guillaume Savation (2021) *RISCV International*. Available at: https://riscv.org/news/2020/06/emulsiv-a-visual-simulator-for-teaching-computer-architecture-using-the-risc-v-instruction-set-guillaume-savation/ (Accessed: 19 July 2024).

Mariotti, G. and Giorgi, R. (2022) *WebRISC-V: A 32/64-bit RISC-V pipeline simulation tool*. Available at: https://www.sciencedirect.com/science/article/pii/S235271102200070X (Accessed: 19 July 2024).

https://eseo-tech.github.io/emulsiV/