Chapter 4 Thermal Analysis of the MIPS Processor Formulated within DEVS Conventions

Alejandro Moreno Astorga

Universidad Nacional de Educación a Distancia, Spain **Eva Besada-Portas** Universidad Complutense de Madrid, Spain

José L. Risco-Martín Universidad Complutense de Madrid, Spain Luís de la Torre Universidad Nacional de Educación a Distancia, Spain

Joaquín Aranda Universidad Nacional de Educación a Distancia, Spain

ABSTRACT

The MIPS processor is used in computer architecture courses in order to explain matters such as performance analysis, energy consumption, and reliability. Currently, due to the desire for more powerful computers, it is interesting to learn how to reallocate certain components in order to achieve heat reduction with low cooling costs. DEVS is a general formalism for modeling and analysis of discrete event systems based on set theory and represents a basis for discrete event abstractions by formalizing the concept of activity which relates to the specification and heterogeneous distribution of events in space and time. The MIPS simulator is built upon known techniques for discrete event simulation and its definition within a formal language such as DEVS provides completeness, verifiability, extensibility, and maintainability. In this chapter, the authors carry out a thermal analysis of the MIPS processor using a DEVS simulator and show a register reallocation policy based on evolutionary algorithms that notably decreases the resulting register bank temperature.

DOI: 10.4018/978-1-4666-4369-7.ch004

Copyright ©2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

The MIPS processor is used in computer architecture courses in order to explain matters such as performance analysis, energy consumption, or reliability. The current integration scales introduce new phenomena's that significantly degrade the consistency of the chips. Electro migration, power consumption, performance, and temperature are some of the parameters to take into account.

Nowadays, due to the desire for more powerful computers, it is interesting to learn how to reallocate certain components in order to achieve heat reduction, with low cooling costs.

DEVS is a general formalism for modeling and analysis of discrete event systems based on set theory. This is a standard term in the field of simulation referring to a modular and hierarchical formalism that is commonly used to model and analyze different systems. DEVS was originally introduced by Zeigler (Zeigler, 1984a; 1984b) in 1976 to provide a computational basis to express behavior of widespread discrete event formalisms (event-scheduling, activity-scanning, process-interaction, and more) alongside with other basic systems formalisms (continuous state systems and hybrid continuous state and discrete event systems). DEVS can reproduce *Discrete Time System Specifications* (DTSS) and approximate continuous modeling paradigms (*Differential Equation System Specification* (DESS)). DEVS approaches continuous systems using numerical integration methods. Hence, simulation tools based on DEVS are potentially more general than others including continuous simulation tools (Kofman, 2004).

DEVS represents a basis for discrete event abstractions by formalizing the concept of activity which relates to the specification and heterogeneous distribution of events in space and time. This arrangement offers a new way to unify the computational representation of both continuous and discrete phenomena and to simulate them with the greater efficiency and flexibility afforded by object-oriented discrete event environments.

In this chapter, the MIPS simulator is built upon known techniques for discrete event simulation (DEVS). The definition of the MIPS processor within a formal language such as DEVS provides completeness, verifiability, extensibility, and maintainability. Moreover, DEVS conceptually separates models from the simulator, making possible to simulate the MIPS processor and its experimental frame using different simulators working in centralized, parallel or distributed execution modes. Also, models can be simulated with a simple ad-hoc program written in any language.

The simulation of the MIPS processor lets us generate a profile that stores the operations performed on each register. This profile allows the extraction of information such as the power consumed by each register.

40 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: <u>www.igi-</u> <u>global.com/chapter/thermal-analysis-mips-processor-</u> formulated/77799

Related Content

Motivated Learning for Computational Intelligence

Janusz A. Starzyk (2011). *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives (pp. 265-292).* www.irma-international.org/chapter/motivated-learning-computational-intelligence/53309

Intelligent Information Description and Recognition in Biomedical Image Databases

Khalifa Djemaland Hichem Maaref (2011). *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives (pp. 52-80).* www.irma-international.org/chapter/intelligent-information-description-recognitionbiomedical/53301

Using a 3D Simulation for Teaching Functional Skills to Students with Learning, Attentional, Behavioral, and Emotional Disabilities

Maria-Ioanna Chronopoulouand Emmanuel Fokides (2020). *Teaching, Learning, and Leading With Computer Simulations (pp. 209-233).*

www.irma-international.org/chapter/using-a-3d-simulation-for-teaching-functional-skills-tostudents-with-learning-attentional-behavioral-and-emotional-disabilities/235866

Overview on Agent-Based Social Modelling and the Use of Formal Languages

Cristina Montañola-Sales, Xavier Rubio-Campillo, José M. Cela-Espin, Josep Casanovas-Garciaand Adriana Kaplan-Marcusan (2014). *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications (pp. 333-377).* www.irma-international.org/chapter/overview-agent-based-social-modelling/77806

The DEVS Formalism

Rhys Goldstein, Gabriel A. Wainerand Azam Khan (2014). *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications (pp. 62-102).* www.irma-international.org/chapter/devs-formalism/77798