ABSTRACT

SoCRocket is a design framework for rapid SoC development. It enables design, verification and evaluation of multiprocessor platforms based on a collection of open and freely available building blocks, including the LEON processor and the RTEMS operating system. Moreover, it provides a modular and standard-compliant tool-set for the creation, configuration, simulation, and performance analysis of virtual platform prototypes, supporting mixed abstraction levels to balance simulation, accuracy and speed. The framework is based on the industry standard specification language SystemC with TLM2. All modules are available in three abstraction levels: loosely timed (LT), approximately timed (AT) and RTL, so that mixed-abstraction and mixed-timing variants can be created quickly depending on the required use case.

KEYWORDS: SoCRocket, TLM, TLM2.0, SystemC, Space, System-Level-Design

1 Introduction

Increasingly large portions of electronic systems are being implemented in software and its development costs start dominating the overall system’s cost. Furthermore software is becoming a critical part of the development schedule, since deployment and testing on real target hardware prototypes is complicated, time-consuming and expensive.

Transaction Level Models (TLMs) can be used to describe timing and functionality of system components, as well as their communication interfaces at a high abstraction level. Embedded in a virtual platform, these models are sufficiently accurate to not only allow early software development and verification in a realistic environment, but also functional
verification of the modeled hardware. The capability of early design-space exploration is therefore a vital building block of full hardware/software co-design.

2 The Framework

To achieve these goals, we designed the SoCRocket Framework [1, 2] Written in SystemC/TLM, it meets the space industry's special requirements and builds the foundation for space-domain ESL design. To enable efficient construction of virtual platforms, we use SoCRocket to explore the following research aspects:

• **Infrastructure** – We evaluate the existing SystemC and TLM standard tools and available third-party tools. To lower the barrier of entry we adopted and extended these tools by simple-to-use APIs. Reusable well-defined APIs for building new components ease and simplify the design process.

• **Tooling** - The debugging of systems can be very time consuming and difficult. This is caused by insufficient insights into the hardware of the systems and the lack of full-synchronized stops of all cores in a system. To help software and hardware developers we extend the debugging capabilities of the SoCRocket platform with deep automated introspection support and direct debugger interfaces. Thus our platform can help to simplify the debugging process and will set the focus of the developer on finding the faults.

• **Analysis** - The common analysis approaches, like dumping signal values into VCD files, do not fit well for TLM style payload transactions. To overcome these shortcomings we do research on new methods for analysis. For example [3, 4].

• **Integration** – In addition to the infrastructure the utilities and tool chains to integrate the components in third-party software is crucial for the usage and acceptance.

• **Models** – Last but not least it is important to have a collection of models validated models. All our models are designed to simulate their corresponding counterparts from the Aeroflex Gaisler GRLib. Therefore every component is available in loosely timed, approximately timed and cycle timed.

3 References


