Efficient Loosely-Timed SystemC TLM-2.0 Modeling: A Hands-On Tutorial

Educational/Workshop

Tuesday, December 6 — 2:15 - 3:45 PM

Nils Bosbach, *RWTH Aachen University*, +49 241 80-27985, bosbach@ice.rwth-aachen.de
Lukas Jünger, *MachineWare GmbH*, +49 241 47592028, lukas@mwa.re
Rainer Leupers, *RWTH Aachen University*, +49 241 80-28301, leupers@ice.rwth-aachen.de

Preperations

In this workshop, the participants get a hands-on experience in the use of the Virtual Components Modeling Library (VCML). To participate in the practical part, you will have to bring a laptop with you.

Please download and install the required software:

- *Oracle VM VirtualBox*: `https://www.virtualbox.org/`
- Virtual machine image file `DVCon_2022_tutorial_lubuntu.ova`
  - `https://mwa.re/dvcon`
  - Password: dvcon2022

Introduction

The increasing complexity of Systems-on-a-Chip (SoCs) requires simulation during an early design stage. To simulate complete SoCs, Virtual Platforms (VPs) are needed. VPs can be used for rapid prototyping, hardware/software co-design, and debugging. SystemC offers a standardized interface to construct such simulations. In this workshop, we present VCML as a tool that builds on top of SystemC/TLM-2.0 to provide a framework to quickly develop and integrate new models for system-level simulations.

VCML[1] contains a set of SystemC/TLM-2.0 modeling primitives and component models that can be used to swiftly assemble system-level simulators for embedded systems, i.e. VPs. Its main design goal is to accelerate VP construction by providing a set of commonly used features, such as TLM sockets, Interrupt ports, Input/Output (I/O) peripherals, and registers. Based on these design primitives, TLM models for frequently deployed components are also provided, such as memories, memory-mapped buses, Universal Asynchronous Receiver/Transmitters (UARTs), etc. VCML is licensed under the Apache-2.0 license.

At the Institute for Communication Technologies and Embedded Systems (ICE) of RWTH University, we developed the ARMv8 VP *An ARMv8 Virtual Platform (AVP64)*[2]. The VP is based on VCML and uses Quick Emulator (QEMU) as Instruction-Set Simulator. The architecture of the VP is show in Figure 1.

The platform consists of an QEMU-based CPU and different peripherals, which are all part of VCML. The interconnection of the components is realized using TLM-2.0-based protocols for memory accesses and interrupts which are defined in VCML as well.

[1] `https://github.com/machineware-gmbh/vcml`
[2] `https://github.com/aut0/avp64`

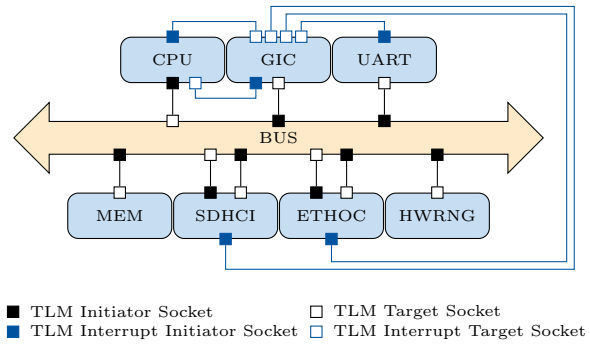

Figure 1: AVP64 architecture.

Workshop

In this workshop, we will demonstrate the strengths of VCML by showing the steps to implement a PL011 UART virtual model based on VCML. The tutorial addresses participants who are interested in system-level simulation and virtual prototyping. C/C++ programming capabilities are beneficial to participate in the practical session.
In the introductory presentation, the design principles of VCML are presented. The structure of the library is introduced and the integration of SystemC/TLM-2.0 is shown. Features that are needed for the implementation of the UART model are presented.
A 60-minute practical session is used to implement the PL011 UART model based on a given template. The model will be integrated into AVP64 to evaluate the functionality. Linux can be booted on the VP and the implemented model can be used for user interaction. In a final discussion, the results of the implementation are presented.
The participants can expect to learn

- the basics of VCML and how it can be used to build a VP
- how new models can be implemented
- how models can be connected to a VP
- how models and target software can be debugged