

SystemC Hardware/Software Design Language

Mikroelektronik / Prof. Dr. Marcel Jacomet

New design languages are required to cope with the hardware/software co-design methods needed for complex system designs. The SystemC design language seems to have quite some potential compared with the classical design languages: it supports the hardware/software co-design approach, and places special focus in the design of communication channels between hardware blocks using the Open Core Protocol (OCP) standard recently defined. The goal of this project is to design a first communication example using the SystemC language to learn its potential for complex designs.

SystemC

SystemC offers a revised approach for the SoC-design based on the C++ programming language. It extends the capabilities of C++ to enable hardware description. SystemC adds such important concepts as concurrency, events and data types. Compared to VHDL, with SystemC, interfaces between blocks are not simply described by signals, but by communication methods and protocols. This drastically increases the design abstraction and thus the design efficiency. This capability is provided via a class library that provides new mechanisms to model system architecture with hardware elements, concurrency and reactive behavior.

Transaction Level Modeling

SystemC 2.0 introduces a set of features for generalized modeling of communication and synchronisation called transaction-level modeling (TLM). Communication is modeled as channels and transaction requests using interface method calls of these channel models. Unnecessary details of communication are hidden in the TLM and can be worked out later on. By using TLM we simplify the design effort and also gain simulation speed.

Open Core Protocol

For the hardware communication framework we decided to use the Open Core Protocol (OCP) which provides a highly configurable protocol set. OCP builds the interface between an Intellectual Property (IP) core and an on-chip-channel. It is completely bus-independent and can express a large variety of communication behaviors. The OCP fosters the goal of IP design reuse. It optimizes die area by configuring into the OCP IP only those features needed by the communicating cores.

Communication intensive design example

The goal of this project was to learn the potential of SystemC in combination with the OCP by designing a communication intensive hardware/

software project. As design example we chose a dc-motor controller with an USB interface. For implementing the HW/SW co-design methodology we used an OCP-transaction-level channel for the communication.

The SoC consists of an IP core microprocessor modeled in VHDL and two hardware blocks: PWM (pulse width modulator) hardware block for controlling a dc-motor and an encoder hardware block for sensing the dc-motor speed. The microprocessor sends the reference values to the PWM block which in turn generates the PWM signals. The actual motor speed values are sensed by the encoder block. The feedback control algorithm is implemented in Matlab/Simulink which communicates in real time with the microprocessor.



Pfyl Thomas
1981
079 347 49 40
thomas_pfyl@bluewin.ch

