PDESIGNER FRAMEWORK

André Ricardo da Silva Souza, Millena de Andrade Almeida Gomes, Renato Marcelino de Oliveira, Williams Thiago de Oliveira Azevedo

Informatics Center (CIn)
Federal University of Pernambuco
Cidade Universitária 50740-540
Recife, PE, Brazil
{arss, maag, rmo, wtoa}@cin.ufpe.br

ABSTRACT

The PDesigner framework, a set of SystemC based tools to be used in the modeling of MPSoC platforms, is presented. Using the framework the designer is able to model MPSoCs by selecting components from the processor, devices, bus, memories and cache libraries, and to generate an executable simulator of the system. Also, PD framework has integrated the ArchC 2.0[1] language and tools, enabling processor descriptions to be exported to the framework processor library. Once in the library, the processor can be used as a component in the MPSoC platform. Two examples example of a MPSoC composed of the SPARCV8 and MIPS processors and, the AMBA and AVALON busses is given and simulation results are presented.

1. INTRODUCTION

The PDesigner framework, a set of SystemC based tools to be used in the modeling of MPSoC platforms, is presented. Its architecture, functionalities and interfaces that allow the communication with other tools, to increase the number of components in the library, are described. To long the paper is structured as follows: Section 2 describes the PD framework and its architecture. The design flow of a multiprocessor platform is given in section 3. An MPSoC modeling and simulation case study and results are discussed in section 4. Finally, some conclusions are drawn in Section 5.

2. THE PDESIGNER FRAMEWORK

PD framework is focused on processor and platform modeling and simulation. It has been developed on the Eclipse [2] standard workbench. The views and perspectives of the framework are depicted in Figure 1. The use of a graphical editor simplifies the design of platforms, since components can be represented by geometric shapes and selected from a component palette.

The user may connect the components just by dragging and dropping the connection field into a graphical interface to compose the architecture. When necessary the framework recognizes the interface of the components, if they already exist in the PDLibrary, and automatically generates the appropriate wrapper.

Also, the image of the overall modeled platform can be seen at the *Platform View*, which is an outline viewer containing a miniature of the platform image. The *Instance View* is a list of instantiated components that can be used to navigate through component instances and, in addition, to configure them through an editable *Properties View*.

The framework also contains a *Console View*, where all compilation, simulation and analysis information is displayed.

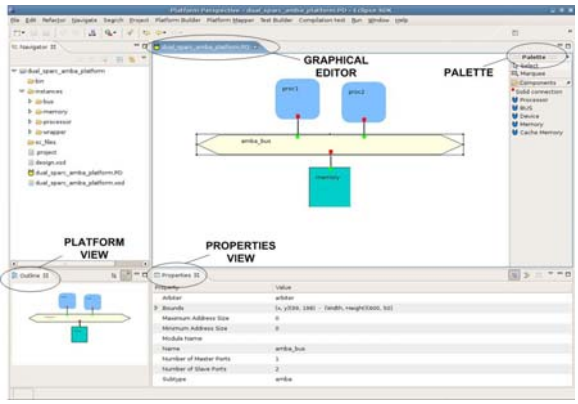

Figure 1 PD Framework

2.1. Architecture

PD is a set of tools build on Eclipse IDE, composed by three main plugins that allows component and platform modeling and an editable components library connected using a XML standard. The architecture of the framework as depicted in Figure 2.

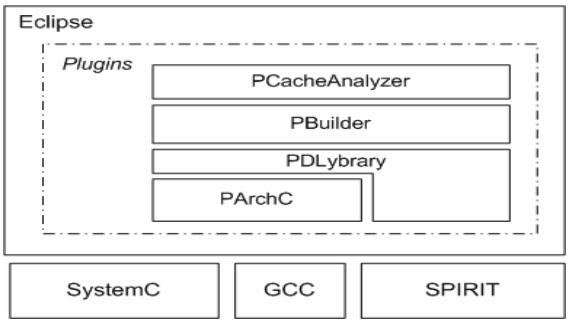

Figure 2 PD Framework Architecture

2.2. Platform designer Plugins

The PArchC Plugin allows designers to describe a processor at a graphical workbench for code edition, compilation and simulation. It also enables to validate the processor and export it encapsulated in the SPIRIT 1.2 standard [3] to the PDLibrary Plugin.

The PBuilder Plugin allows the designer to model, simulate and analyze multiprocessor platforms with the PDLibrary components, specified in several languages, especially in SystemC and ArchC, to simulate and to analyze them using a friendly graphical interface.

The PCacheAnalyzer allows the designer to do a previous cache analyses trough a table of cache configuration informing the miss rate, cache size, line size and associativity of the configuration.

2.3. SPIRIT Schemas

The PD framework complies with the SPIRIT 1.2 standard [1] and tools, to describe and distribute generated components. The library communicates through XML files implementing the Loose Generator Interface (LGI).

Design, component, bus interface, configurator and generator SPIRIT Schemas are used to describe and configure the components used in the platform project.

3. DESIGN FLOW

The designer, using the library components, is able to model the desired platform graphically by inserting, connecting and configuring components. Components that use different protocols can be connect, the necessary wrappers being generated automatically when available in the library. The wrapper generation is transparent to the designer.

The designer is also able to model and export its own components to the PDLibrary using the ArchC Plugin as shown in Figure 3.

The platform high level representation is converted to a SystemC code by the PDBuilder plugin to allow its simulation and analysis.

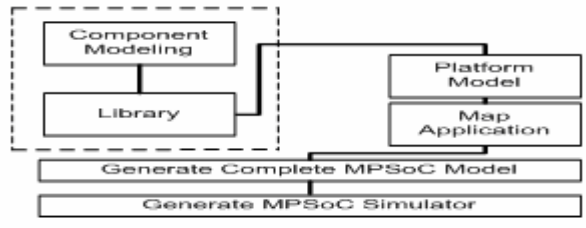

Figure 3 Platform Project Design Flow

4. CASE STUDY

In order to demonstrate the use of the proposed framework, a platform with two processors SPARCV8, an interconnection structure AMBA and a memory was specified. The platform is depicted in Figure 4.

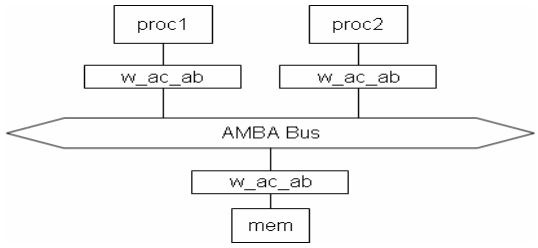

Figure 4 Multiprocessor platform

The platform specified was built in the PD framework, connecting the SPARCV8 processors to an AMBA Bus, using a SimpleBus Memory connected to it, inserting wrapper automatically to allow those connections.

For simulation, a producer/consumer application was mapped in the platform, the producer running on proc1 and the consumer running on proc2. The communication between the processors was based on a shared memory.

Simulation results are summarized in Table 1.

	proc1	proc2
Simulation Time	18,17 s	18,17 s
Number of instructions	1410975	2135380
Simulation Speed	77.65 K instr/s	117.52 K instr/s

Table 1 Simulation Statistics

5. CONCLUSIONS

The PDesigner, a framework for modeling, simulation and analysis of MPSoC platforms was presented. Using the proposed framework the modeling effort is reduced to a minimum, since processors and platforms are modeled in a unified environment.

Besides modeling support, the proposed approach allows the automatic generation of platform simulators in SystemC at distinct abstraction levels. The main advantage of this feature is that the designer does not need to make any change on the platform description to obtain simulators. It hides all the simulation scheme details from the designer by just issuing command line options or clicking the corresponding buttons in a graphical framework. As an additional advantage of the approach is the ability to explore distinct platforms with small effort.

6. REFERENCES

[1] R. Azevedo, S. Rigo, M. Bartholomeu, G. Araújo, C. Araújo and E. Barros."The ArchC architecture description language and tools", In the International Journal of Parallel Programming, Vol. 33, No. 5, October 2005

[2] Eclipse Framework available at www.eclipse.org

[3] SPIRIT Consortium available at www.spiritconsortium.org