SOFHIA: A CAD Environment to Design Digital Control Systems

Ricardo J. Machado, João M. Fernandes, Alberto J. Proença
Dep. Informática, Universidade do Minho
4709 Braga codex, Portugal
rmac,miguel,aproenca@di.uminho.pt

Abstract

Petri Nets (PNs) prove to be an efficient methodology to model discrete-event systems with parallel activities. The main advantages lie on the graphical interface and on the availability of a set of techniques for formal analysis, including the validation and the test of the modelled system. A proposal to modify the normal PN behaviour is presented, which aims a fast specification of synchronous parallel digital systems, including both the data path and the control unit. A CAD environment, SOFHIA, was developed to model digital systems, to validate their properties and to simulate their behaviour. The environment includes the automatic generation of VHDL code to allow simulation and synthesis on existing CAD tools.

Keywords

Petri Nets, Digital Control Systems, VHDL, CAD Tools

SUMMARY

A new PN model, shobi-PN (synchronous, hierarchical, object-oriented and interpreted PN), was developed to support the use of hierarchy and to model the control unit and the data path in the specification of digital systems.

The SOFHIA CAD environment (Fig. 1) was developed to feed any ECAD package that accepts VHDL as input. This environment is appropriate for controller systems specified with the shobi-PN model. The hierarchical PN specification is directly and efficiently mapped to boolean equations. This approach simplifies the VHDL code debbuging, since there is a direct correspondence between the original PN and the produced VHDL code.

All the tasks needed for digital control systems design using shobi-PN-based specifications are completely supported by the SOFHIA environment. Among those tasks are: (1) formal verification of the properties of the model; (2) simulation; and (3) VHDL generation for the system synthesis.

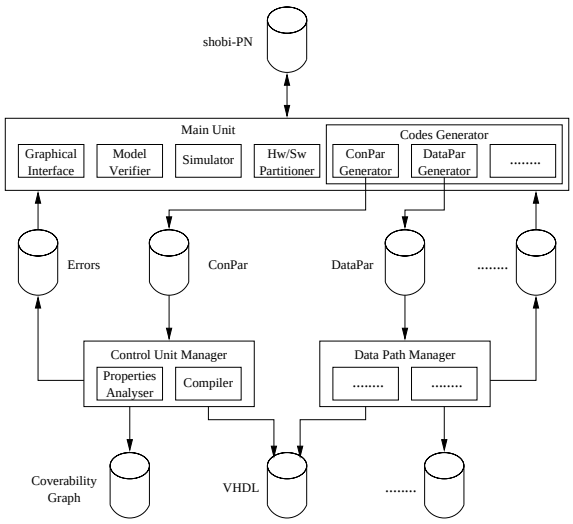

Figure 1 The SOFHIA CAD environment.

The SOFHIA CAD environment is structured in 3 independent blocks. The MU is responsible for the complete integration of the environment and for the dialog with the user. The CUM is responsible for formally verifying the properties of the model and its correctness. It is also the CUM that generates VHDL code for the control unit synthesis. The DPM generates VHDL code for the data path synthesis.

The implementation of the SOFHIA environment was based on two different software platforms: the ConPar tool (Fernandes *et al.* 1995a), that corresponds to the CUM, and the SCBA (Pina 1993) that was used as a development environment for the graphical interface and the simulator.

The Main Unit: The Graphical Interface allows the designer to use graphical icons for specifying the PN. The Model Verifier tests if the input specification fulfils the rules imposed by the shobi-PN model.

The Simulator module allows the behaviour of the PN to be simulated. The user can check the tokens' contents which helps in the verification of the correct output values. For large or complex PN specifications, the formal verification may demand too many computer resources, which makes simulation one of the possible solutions.

The Hw/Sw Partitioner selects the proper code generator block to generate descriptions of the system parts in intermediate languages. These descriptions will feed the CUMs and the DPMs to allow the parts to be synthesized in software and/or in hardware. This block allows the use of the SOFHIA environment for codesign.

The Codes Generator aims the generation of intermediate descriptions to

feed the CUMs and the DPMs blocks. The ConPar Generator (Fernandes *et al.* 1997) generates a file with the textual description of the specified PN in the intermediate ConPar language. This description is related to the control unit of the initial PN and it will feed the CUM. The DataPar Generator generates a file with the textual description of the specified PN in an intermediate language. This description is related to the data path of the initial PN and it will feed the DPM. This CAD environment has an open architecture, which allows the inclusion of multiple code generator blocks to ease the implementation of the system in several technologies.

The Control Unit Manager: Several CUMs may exist in the environment, depending on the number of final representations for implementing the control unit. The first already developed CUM accepts as input the specification of a control unit using SIPNs. This specification is written in ConPar, an intermediate language, both human- and computer-readable.

The Properties Analyser verifies if the input specifications are live and conflict-free, issuing a message to the user interface, whenever a problem occurs (Fernandes *et al.* 1995b).

The current Compiler version provides two VHDL code generation alternatives. To infer the initial and next markings, the user can select either a `BLOCK` or a `PROCESS` statement to be included into the generated VHDL file. In both cases, the remaining VHDL description is a set of concurrent signal assignments and concurrent `ASSERT` statements.

The Data Path Manager: Similarly, several DPMs may exist in SOFHIA, depending on the number of final representations to implement the data path. It may exist a DPM responsible for generating VHDL code for the data path from an intermediate description supplied by the DataPar Generator.

REFERENCES

Fernandes, J. M.; Pina, A. M. and Proença, A. J. (1995a) Concurrent Execution of Petri Nets based on Agents. In *1st Workshop on Object-Oriented Programming and Models of Concurrency*, Torino, Italy.

Fernandes, J. M.; Pina, A. M. and Proença, A. J. (1995b) Simulation and Synthesis of Parallel Controllers based on Petri Nets. In *VII SBAC-PAD Simpósio Brasileiro de Arquitetura de Computadores — Processamento de Alto Desempenho*, 481–92, Canela, Brazil. (in Portuguese).

Fernandes, J. M.; Adamski, M. and Proença, A. J. (1997) VHDL Generation from Petri Net Specifications of Parallel Controller. *IEE Proceedings-E: Computers and Digital Techniques.* Accepted for publication.

Pardey, J. and Bolton M. (1991) Logic Synthesis of Synchronous Parallel Controllers. *Proceedings of the IEEE International Conference on Computer Design*, 454–7.

Pina, A. M. (1993) SCBA — Agent-Based Concurrent Simulation. In *XX SEMISH*, Florianópolis, Brazil. (in Portuguese).