Implementation of Speed Control of Brushless DC Motors using Iopt petri Net Models

R.Nithya,ME Embedded System Technologies ,Sri Sai Ram Engineering College ,Chennai,India

nithragu@gmail.com ,

S.Usha,Dept of Electronics and Communication ,Sri Sai Ram Engineering College,Chennai,India

usha.ece@sairam.edu.in

Abstract: **The current integrated circuit technologies are approaching their physical limits in terms of scaling and power consumption, during this context, the electronic design automation trade is pushed towards finding more challenging issues in terms of performance, scalability, adaptability, and reduced development time. The combination of an intuitive graphical modeling language, in conjugation with automatic model checking and code generation tools is proposed. The proposed framework contributes to significantly reduce development time, both during the specification, system design phase and during the test. To demonstrate the proposed approach, the Brushless DC Motor closed-loop speed controller is designed with IOPT Petri Net models. The IOPT Petri Net model is an integrated development environment offered by IOPT-Tools. IOPT- Nets is a Web based Petri net class specifically designed to support the implementation of embedded system controllers without the need to manually write software or hardware description programs. The IOPT Web service includes an interactive graphical editor to plan IOPT models, a model checking framework consisting of a query system and state space generator, and automatic code generation tools to provide software or hardware controller implementations. The BLDC Motor speed controller designed rely on several subsystems, as well as noise filter, Quadrature decoder, PWM generator and a BLDC commutation manager. The sub-systems were modeled by adopting IOPT models, analyzed using the model checking tools, leading to the automatic creation of VHDL modules for every sub-system.**

I.PETRI NETS

Petri nets are graphical and mathematical modeling tool applicable to several systems. They are a promising tool for describing and learning information processing systems that are characterized as being synchronous,asynchronous, distributed, parallel, nondeterministic, and random. As a graphical tool, Petri nets can be used as a visual communication aid nearly like block diagrams, flow charts and networks. In addition, tokens are utilized in these nets to simulate the synchronous and dynamic activities of systems. As a mathematical tool, it is possible to set up state equations, mathematical models and other algebraic equations governing the behavior of the systems.

Petri net was introduced by Carl Adam Petri in 1962. Petri net model is a graphical illustration used to design a complex system. A diagrammatical tool to model concurrency and synchronization in distributed systems. It is terribly like State Transition Diagrams. Utilized as a visual communication to model the system behavior. It is based on strong mathematical foundation. Petri net is a bipartite graph. It consists of two parts a net structure and an initial marking. A net (structure) contains two forms of nodes: places and transitions.

There are directed arcs from transitions to places and directed arcs from places to transitions in a net. Places are diagrammatically represented by circles and transitions by boxes or bars. A place can hold a token which is denoted by black dots, or a positive integer representing their number. The allocation of tokens over the places of a net is termed a marking that corresponds to a state of the modeled system. The initial token distribution is hence referred

II. IOPT PETRI-NETS

IOP T stands for input output place transition tool. IOPT tools are used to generate the code automatically for the model designed. IOPT tools have been developed

by many members of the R&D Group on Reconfigurable and Embedded Systems (GRES). The mathematical properties of IOPT Petri nets are applied to notice the design errors during the early design stages, contributing to minimize the time consumed during the check and validation stages and reducing price and time-to market.

The IOPT Petri net class inherits all characteristics from P/T (place-transition) nets, and a collection of non-autonomous extensions planned to the design of embedded system controllers, providing support for communication with physical devices, together with the controlled systems and user interfaces. The Web interface and the automatic code generation provides a easy way to create embedded system controllers, ready to be used even by persons without deep knowledge of hardware and low-level software code.

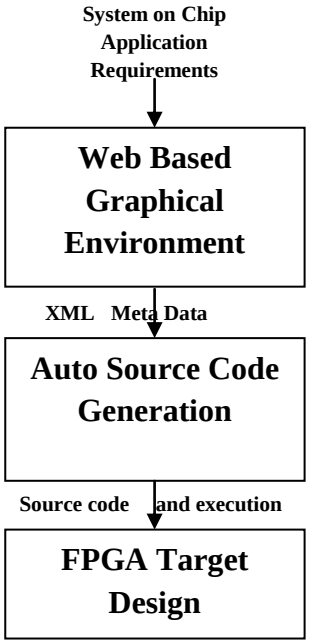
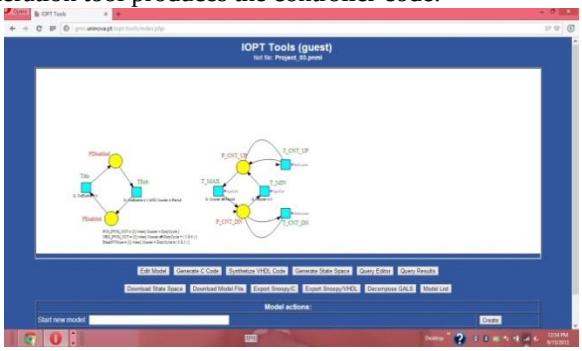


Fig 1: IOPT-Tools Web User Interface

The Web based new model-checking tools, consists of a state-space generator, a query system and a automatic code generator, were added to associate existing tool framework that already had tools to import and edit automatic code generators, controller models and automatic hardware synthesis tools. The new tools allows the automatic analysis of system properties and the also the quick detection of errors during early design stages, significantly reducing debug and validation time. All tools share the similar Web based user interface, publicly accessible on the research group Web page (http://gres.uninova.pt).

Model definition and specification benefit from the graphical nature of Petri nets, where design often starts from a set of known use-cases. Complex concurrent systems can be easily modeled as collection of individual sub-systems that are later joined with the support of a net addition operation. To detect design flaws, the framework contains a model checking tool comprehending a state-space generator and a query system used to extract information from the resulting state-space graphs. The query system is necessary to analyze real-world embedded system controllers that frequently lead to very large state-space graphs with millions.
In addition, the model-checking tools always check important system properties, as the existence of deadlocks and conflicts between net transitions. Maximal place bounds are calculated to determine the size of the memory elements, or data types, used in controller implementations. The automatic code generation tools compile IOPT controller models into software source-code implementing the model's execution semantics, minimizing the amount of handwritten code that is reduced to simple interface code, dependent on the target devices. Low level details are hidden from the high-level models and the error-prone and time consuming coding tasks are almost eliminated.

The tools are offered under a Web-based user interface, implementing an integrated development environment, displayed in Figure 2. Users can upload IOPT Petri net model files, perform model visualization and edition, state space generation and execute model-checking using a query editor and a query results filter page. Finally, the automatic code generation tool produces the controller code.



2

Fig 2: IOPT-Tools Web Page

System modeling and controller design is done using the Snoopy IOPT Petri net editor, or other Petri net editor supporting the PNML standard, with capability to support the IOPT extensions. Controller implementation was performed using several automatic code generation tools, generating C software code or

VHDL hardware descriptions according to the target device.

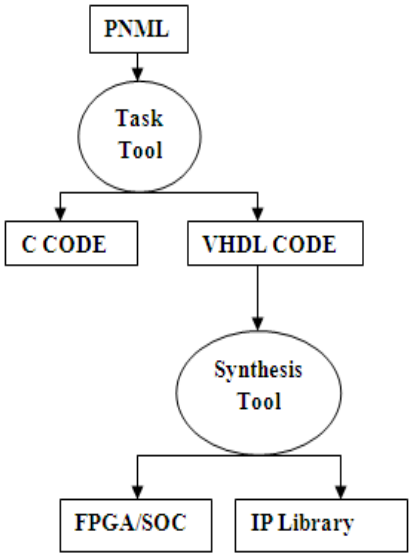


Fig 3: Automatic Code Generation Tool

An Animator tool creates system synoptic and graphical interfaces to interact with users. The interfaces are designed by specifying a set of rules relating the system state and input values with the position and movement of graphical objects on screen, without requiring writing any software code. Another tool, GUIGen4FPGA, implements those graphical user interfaces directly in FPGA hardware, including a high resolution video generator and pointing and touch device interfaces, enabling the automatic creation of full embedded system controllers comprehending the system controller and a user interface.

III.SYSTEM MODEL

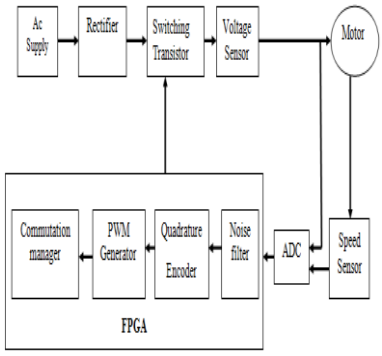


Fig 4: System Model

DESCRIPTION: The speed of the motor is controlled using PWM (pulse width modulation) technique. The input AC supply is given to the rectifier which is used to converts AC to DC. The rectifier output is given to the Switching transistor. The switching transistor consists of.

As the FPGA requires the digital input ADC is used .The input to the ADC is from speed sensor and the output voltage of the switching transistor. The output of ADC is digital which is 8 bit .The FPGA block consists of four sub-block they are noise filter ,Quadrature decoder, PWM generator, BLDC commutation Manager. The output of ADC is given to the FPGA in which the VHDL code is interfaced to generate the PWM signal. Noise filter is used to remove the high frequency noise, Quadrature decoder is responsible for counting the pulses, PWM generator will produce the PWM signal for certain pulses, the BLDC commutation manager consists of commutation table to produce the six PWM output signal which is fed back to the switching circuit. Based on the PWM pulses the output voltage will get increase or decrease .This output voltage will control the speed of the motor.

IV.PROTOTYPE IMPLEMENTATION

The simple closed loop system BLDC motor control model designed for real-time justification. The designed models are inserted in the IOPT-Tools Web service (http://gres.uninova.pt). To examine for deadlocks, unreachable desired states the model checking tools were applied.

The VHDL code synthesis tools were applied to every model, producing VHDL codes for designed modules that were appended to a Xilinx ISE project.

The Test FPGA board uses Spartan3AN X700 FPGA with 50MHz clock with required motor driver circuits.

Although the VHDL code created by the automatic code generator is not optimal and includes several dead sections preceded by conditions that are always false or always true, the optimizing tools provided by FPGA vendors can easily remove the dead sections and produce a virtually optimal result. As a consequence, the FPGA resource consumption of the whole project is very small, corresponding to approximately 3% of the resources available on the Spartan3 FPGA.

V. CONCLUSION

The main goal of the work is to control the speed of BLDC motors using IOPT Petri net models. IOPT models provide a very simple and intuitive way to specify the desired system behavior. By dividing entire systems in small components that can later be inter-connected, complexity can be kept at very low

levels and sub-systems are easily validated with the model- checking tools. Codes are generated for each subsystem and these codes are interconnected to single module. The dead sections of automatic code generation need to be improved on IOPT tools.

REFERENCES

[1] FPGA based Speed Control of Brushless DC Motors using IOPT Petri Net models. Pereira.F, Gomes,L. Industrial Technology (ICIT), 2013 IEEE International Conference on Digital Object Identifier: 10.1109/ICIT.2013.6505810 Publication Year: 2013 , Page(s): 1011 - 1016

[2] L.Gomes, J.Barros, A.Costa, and R.Nunes, "The Input-Output Place-Transition Petri Net Class and Associated Tools," in Proceedings of the 5th IEEE International Conference on Industrial Informatics (INDIN'07), Vienna, Austria, July 2007.

[3] F.Moutinho, L.Gomes, "From models to controllers integrating graphical animation in FPGA through automatic code generation," in IEEE International Symposium on Industrial Electronics (ISlE 2009), Seoul Olympic Parktel, Seoul, Korea, July 5-8 2009.

[4] Ming-Fa Tsai, Tran Phu Quy, Bo-Feng Wu, Chung-Shi Tseng , "Model construction and verification of a BLDC motor using MATLAB/SIMULINK and FPGA control," Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on, vol., n., pp.1797-1802, 21-23 June 2011doi: 10.1109/ICIEA.2011.5975884

[5] R. Nunes, L.Gomes, J.P.Barros, "A graphical editor for the input-output place-transition petri net class", Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on , vol., no., pp.788-791, 25-28 Sept. 2007

[6] Pereira.F, Moutinho.F, Gomes.L , "Model-checking framework for embedded systems controllers development using IOPT Petri nets," Industrial Electronics (ISIE), 2012 IEEE International Symposium on , vol., no., pp.1399-1404, 28-31 May 2012 doi: 10.1109/ISIE.2012.6237295

[7] W. Reisig, "Petri nets: an introduction." NY, USA: SpringerVerlag New York, Inc., 1985.