System-Level Modeling of a Mixed-Signal System on Chip for Wireless Sensor Networks

Gilmar S. Beserra, Jose Edil G. de Medeiros, Arthur M. Sampaio, Jose Camargo da Costa
Department of Electrical Engineering, Faculty of Technology
University of Brasilia, Brasilia, Brazil
E-mail: {gbeserra,joseedil,camargo}@unb.br

***Abstract*—Due to the increasing advance on wireless communication and sensors, Wireless Sensor Networks (WSN) have been widely used in several fields, such as medicine, science, industrial automation and security. A possible solution is to use CMOS System on Chip (SoC) sensor nodes as hardware platforms due to its extremely low power, sensing, computation and communication capabilities. This work presents the modeling of a mixed-signal SoC for WSN using a system-level approach. The digital section was modeled using SystemC Transaction Level Modeling (TLM) and consists of a 32-bit RISC microprocessor, memory, interrupt controller and serial interface. The analog block consists of an Analog-to-Digital Converter (ADC) described in SystemC-AMS. An application was implemented to test the correctness of the model and perform the communication between the SoC and a functional level node model.**

I. Introduction

A Wireless Sensor Network (WSN) is typically composed by a large number of multi-functional sensor nodes that are deployed in a region of interest or very close to it. The nodes are small in size, but have sensing, data processing and communicating capabilities. They can communicate over a short distance and collaborate to accomplish a common task, such as environmental monitoring or industrial process control [1], [2].

A common challenge is the requirement for low energy consumption [3]. System on Chip (SoC) implementation of such nodes can provide both energy efficiency and adequate performance to meet the long deployment lifetimes and burst of computation that characterize WSN applications [4].

This work is part of a project that consists mainly of the design and implementation of a SoC for WSN applications. The main application of this project is to monitor environmental parameters at tropical forests, such as temperature and gas concentration. In order to develop and validate applications for the WSN before having a hardware platform available, a high-level modeling approach is needed. To raise the abstraction level is a commonly-accepted solution to increase the simulation speed by hiding low-level details of the architecture.

Transaction Level Modeling (TLM) is a high abstraction level in which modules can communicate with each other using transactions, a data structure (C++ object) that uses function calls, allowing a fast simulation. It is used for early development of the embedded software and also to validate the system functionality. It is based on SystemC, a C++ library used for the description of systems at different levels of abstraction [5].

The native SystemC simulation kernel does not allow the description and simulation of analog, continuous-time systems, which can be modeled using hardware description languages, such as VHDL-AMS [6] and Verilog-AMS [7]. A possible solution is to perform a co-simulation of mixed-signal systems using SystemC and Verilog/VHDL-AMS. However, there are some disadvantages related to simulation performance [8]. Because of that, SystemC-AMS extensions are being introduced to provide a framework for functional modeling, architectural exploration, integration, validation, and virtual prototyping of mixed-signal embedded systems [9]. AMS extensions fill a gap in the design ecosystem by inserting the missing abstraction at the architectural level. With AMS 1.0 and SystemC, it is possible to perform hardware/software (HW/SW) co-design of the system with all of the elements in place [10].

This work presents the SystemC TLM and SystemC-AMS modeling of a mixed-signal SoC for WSN using a system-level approach. The simulations involve the integration of TLM blocks with a SystemC-AMS module resulting in a mixed-signal virtual platform in which the software application can be developed and tested before having the hardware available. To validate the WSN application concept, the SystemC Network Simulation Library (SCNSL) [11] was used to allow the communication between a node composed by a SoC model and a functional level node model. With this methodology, a library with open-source high level models of the separated blocks was also generated for the design of new applications. In addition, more accurate analog models in VHDL and VHDL-AMS, for example, can be developed later from the high level models, as well as models in VHDL, Verilog or SystemVerilog for the digital modules.

The paper is organized as follows. Section II shows some related work. Section III presents the methodology adopted during the system modeling. In Section IV, the SoC modules are presented, the modeling of the digital and analog parts are described, and the system integration is explained. Section V shows the results and discussions. Finally, Section VI presents the conclusions and future work.

II. Related Work

In literature, several system-level modeling approaches have been used to simulate embedded systems with network ca-

[1]978-3-9810801-7-9/DATE11/©2011 EDAA

pabilities at very early stages in the design process. One approach is to model system components using SystemC and combine them in the network environment using NS-2, a well-known network simulation. Although this approach combines the best characteristics of each tool, it also introduces a lot of complexity in the modeling task. Both SystemC and NS-2 simulation kernels have to be modified in order to allow the interaction between different simulators. This approach was used in [12], where a HW/SW/network co-simulation and co-design methodology is presented, based on the integration of SystemC and NS-2. In [13], the same approach is used to simulate both HW and SW parts, and also the network topology and communication infrastructures.

A model of WSN nodes with SystemC-AMS approach is simulated in [8]. Model refinements and simulation improvements are also presented. The WSN system consist of two nodes (N1, N2) that exchange information through a 2.4 GHz communication channel. Their focus is on evaluating the RF transceiver operation, showing that the advantages of SystemC-AMS are the capacity of interoperability and multi-frequency simulations.

Fummi presents the SCNSL in [11]. It was designed for HW/SW/Network co-simulation and allows to model network scenarios in which different kinds of nodes, or nodes described at different abstraction levels, interact together. With this simulator, devices are modeled in SystemC and their instances are connected to a module that reproduces the behavior of the communication channel. It also takes into account the propagation delay, interference, collisions and path loss by considering the spatial position of the nodes and their on-going transmissions.

III. Methodology

System design begins by specifying the required functionality. After the specification stage, the functions that will be executed in HW and SW are divided in order to optimize the architecture. Then, a high level modeling is performed in order to allow the concurrent software and hardware development. The next step is to describe the digital blocks in RTL and perform the logic and physical synthesis including DFT structures. The analog parts are implemented using the full custom methodology, that includes schematic generation, simulation, and layout generation. Those steps are followed by system integration, fabrication, and test. All the verification steps were ommitted for simplicity sake.

The contributions of this work focus on the high level system modeling and validation stages. Figure 1 shows the methodology used to model the components. The HW/SW partitioning is analyzed in order to define an optimized system architecture. After that, the connection among the modules is defined through an interface standardization, that is crucial to allow their compatibility and reuse in future projects. The development of the modules is performed using SystemC language. In this step, the analog parts can be described using SystemC-AMS extensions in order to interact with the remaining blocks. Architecture Description Languages (ADLs), such as ArchC [14], can be used for the development of processor models. In addition, the SCNSL library is used to allow the network environment modeling. IP blocks from other libraries or developed in previous projects can also be used in this stage. Finally, the modules are integrated, composing a simulation framework that can be used for both HW and SW engineers. With this virtual platform, the former are able to monitor the data flow and the communication among the modules in order to identify critical issues, and the latter are allowed to develop and test embedded software.

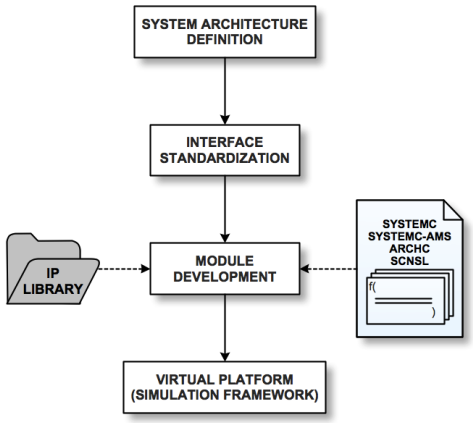

Fig. 1. Modeling design flow

By achieving the simulation of analog and digital components, the methodology can be applied again later, with fewer restrictions on how the blocks that are being integrated are modeled.

IV. SoC Description and Modeling

Figure 2 shows the block diagram of the modeled study case. It shows the SoC architecture, that consists of a 32-bit RISC microprocessor, ADC, memory, an RF interface defined for network communication and a simple bus.

The key concept in this methodology is to model enough detail to enable most of the embedded software development. In this level of abstraction, the focus is not on cycle-accurate simulation but on sufficient timing information to be capable of interrupt handling. Memory mapping correctness, register accuracy and bit-true operations constitute a major requirement for software development [15].

A. CPU and Bus Modules

Our intention is to obtain a low-power high-efficiency processor, taking advantage on the MIPS existing development tools. After the processor description, the ISS generated by the ArchC tool was adapted and customized for the SoC. The processor is a 32-bit RISC that implements the instruction set and the registers of the MIPS architecture. The code was modified to allow a more realistic timing annotation method and also to implement a sleep mode.

The processor model also includes an interrupt target port. All transactions directed to this port will interrupt the processor and call and interrupt handler. The interrupt manager

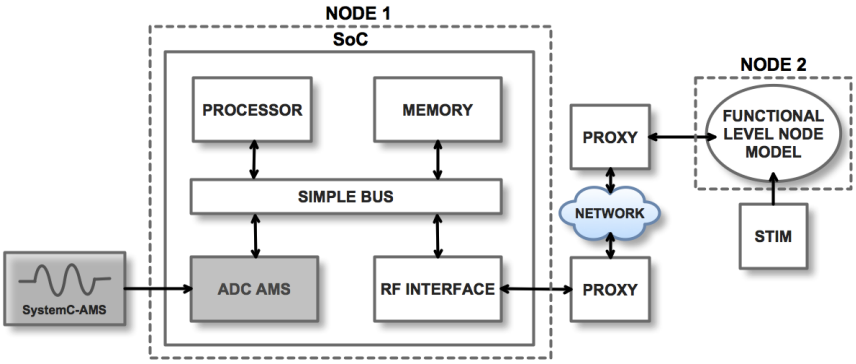

Fig. 2. Modeled case study

receives all the interrupts generated in the system, calls the processor interrupt routine and stores the status registers indicating the events that have occurred. The programmer can implement different interrupt handling routines in software using the information in the status register.

The modeled bus consists of an interconnection structure that reads the address of the transactions initiated by the processor module and routes it to the appropriate target module based on the memory mapping addresses. All communication between modules is implemented using TLM 1.0 instead of TLM 2.0 in order to use the same protocol defined by ArchC in our models.

B. Analog-to-Digital Converter (ADC)

As the SoC applications involve sensing the environment, an ADC is included as an interface with external sensors. In the abstract model, the ADC is described as a module that reads an analog value and converts it into a digital word.

SystemC-AMS offers modeling in Timed Data Flow (TDF), which allows faster simulations as the scheduling is done in advance. The TUV AMS Library provides building blocks that focus on the field of communication and radio frequency systems, im particular on signal sources, modulation/demodulation blocks, filters, measurement and observation parts. They can be subdivided into three categories: signal sources, basic blocks for signal processing and signal analysis units [16].

From the categories listed above, a general ADC and a sine generator blocks were chosen and customized for being integrated into the SoC model in order to validate the WSN applications. An End-Of-Conversion (EOC) flag and an additional output to convert the TDF output into a DE (Discrete Event) output were created.

C. TLM and SystemC-AMS Integration

To perform this step, some adaptations were necessary. A wrapper was used to connect the SystemC-AMS ADC into the TLM SoC model.

In order to connect the ADC into the simple bus, an *sc_export* was created. In addition, an *sc_port* was added and connected into the processor to generate the interrupts. Both of them use the *tranport()* method implemented on the *ac_tlm_protocol* from ArchC.

D. SCNSL and Communication

In order to allow early software development for the networked environment, we designed the model of our system using SCNSL. A case study in which one SoC sends data to a functional level node model was written in order to validate the viability of the methodology.

The network and proxy blocks shown in Figure 2 come from SCNSL library. The former is the core of the network simulator. It reproduces the behavior of the channel and manages the packet forwarding from the source node to destination nodes. Transmission delay, path loss, collisions and the state of destination nodes are taken into account. The latter constitutes an interface between the custom SystemC modules and the network core.

The RF block implements both ArchC and SCNSL interfaces to achieve interaction with both domains (*ac_tlm_transport_if* and *tlm_transport_if*, respectively). The main thread waits for a transmit command from MIPS processor and then starts the communication.

V. Simulation and Discussion

With the model described in the previous item, it is possible to simulate the communication between the SoC and a functional level node model. In order to validate the platform functionality, a simple application was developed. It consists of a MIPS cross-compiled program that is loaded into the memory in the beginning of the simulation.

The developed code is quite simple. First, the interrupts are enabled. Then, the processor is set to the sleep mode. An infinite loop runs while there are no interrupt requests. When the ADC performs a conversion, or when the functional node sends a package through the wireless network, the microprocessor stops the loop and runs an interrupt treatment routine. In this routine, the processor is set back to the normal mode and performes actions according to the interrupt cause. The results can be checked by getting the simulation and debug information, as shown in Figure 3, where *node0* and *Node1* represent the SoC and the funcional level nodes, respectively.

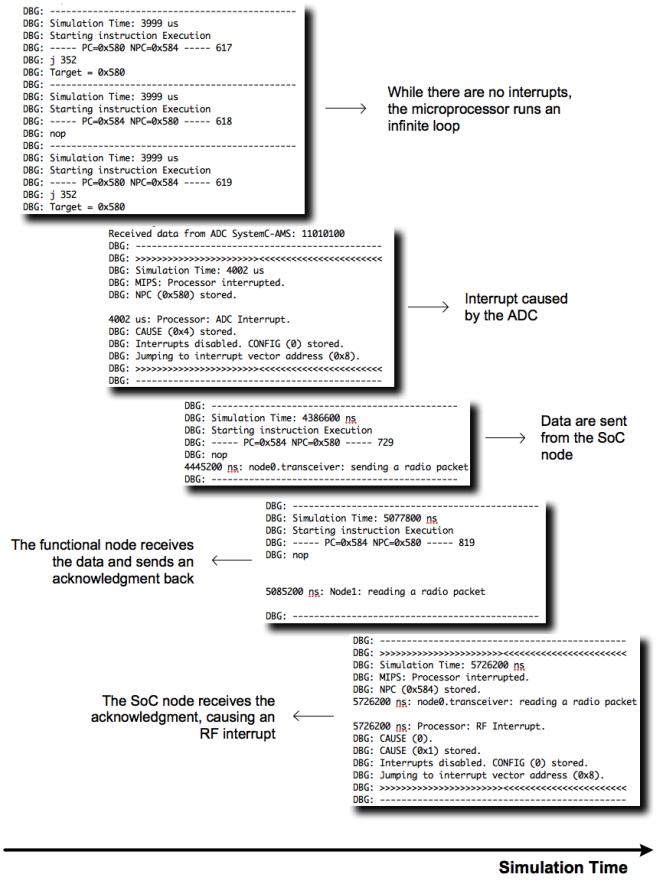

Fig. 3. Debug information

This approach is different from the other works showed in Section II, since it combines the use of SystemC-AMS to describe an ADC and SCNSL to allow network communication between the nodes, while other tools such as NS-2 are usually adopted. Although SystemC-AMS blocks are used in the nodes implemented by [8], the communication between them does not use the resources offered by SCNSL that makes easier to perform simulations with a larger number of nodes.

Other simulations can be performed to validate the WSN applications in which tasks such as analog signal acquisition, digital data processing and communication are carried out. The model has mixed-signal blocks integrated using a methodology that includes SystemC, TLM, ArchC, SystemC-AMS and SCNSL. A library with all the blocks was generated, allowing design reuse in future projects. All the tools are open-source.

With this model, it is possible to run and develop applications before having the hardware platform available, saving time and reducing design costs. In addition, more nodes can be added to the network and routing algorithms can be tested.

VI. Conclusion

This work presented a simulation of a mixed-signal SoC for WSN using SystemC TLM for the digital blocks and SystemC-AMS for an ADC. The networked environment was modeled using SCNSL in order to run WSN applications. The main advantage is to use a single tool to model hardware, software and network interactions, simplifying the modeling task.

A case study in which one SoC node performs data acquisition from the ADC and also receives data from a functional level node model via a wireless network is presented. Although its simplicity, the presented example illustrates the proposed modeling methodology. The combined use of SystemC-AMS and SCNSL differs from the other approaches showed in Section II to model WSN and brings the advantages of fast simulation speed from the former and network capabilities from the latter.

Acknowledgment

The authors would like to thank CNPq and inct NAMITEC - National Institute of Science and Technology of Nano and Microelectronic Systems (Brazilian government agencies) for financial support.

References

[1] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. Wiley-IEEE Press, 2009.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] J. Glaser, J. Haase, M. Damm, and C. Grimm, "Investigating power-reduction for a reconfigurable sensor interface," in *Proceedings of the Austrian National Conference on the Design of Integrated Circuits and Systems (Austrochip 2009), Graz, Austria*, vol. 7. Citeseer, 2009.

[4] M. Hempstead, G. Wei, and D. Brooks, "An accelerator-based wireless sensor network processor in 130nm CMOS," in *Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*. ACM, 2009, pp. 215–222.

[5] The Open SystemC Initiative website. [Online]. Available: http://www.systemc.org

[6] "IEEE Standard VHDL Analog and Mixed-Signal Extensions," *IEEE Std 1076.1-2007 (Revision of IEEE Std 1076.1-1999)*, pp. c1 –328, 15 2007.

[7] The Verilog-AMS website. [Online]. Available: http://www.vhdl.org/verilog-ams

[8] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux, "Efficient and refined modeling of wireless sensor network nodes using SystemC-AMS," in *Research in Microelectronics and Electronics, 2008. PRIME 2008. Ph. D.* IEEE, 2008, pp. 81–84.

[9] M. Damm, J. Haase, and C. Grimm, "Co-Simulation of mixed HW/SW and Analog/RF systems at architectural level," in *Behavioral Modeling and Simulation Workshop, 2008. BMAS 2008. IEEE International*. IEEE, 2009, pp. 84–89.

[10] Open SystemC Initiative. (2010, March) SystemC AMS Extensions Users Guide. [Online]. Available: http://www.systemc.org/downloads/standards

[11] F. Fummi, D. Quaglia, and F. Stefanni, "A SystemC-based framework for modeling and simulation of networked embedded systems," in *Specification, Verification and Design Languages, 2008. FDL 2008. Forum on*. IEEE, 2008, pp. 49–54.

[12] F. Fummi, M. Poncino, S. Martini, F. Ricciato, G. Perbellini, and M. Turolla, "Heterogeneous co-simulation of networked embedded systems," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 3. IEEE, 2004, pp. 168–173.

[13] N. Drago, F. Fummi, and M. Poncino, "Modeling network embedded systems with NS-2 and SystemC," in *Circuits and Systems for Communications, 2002. Proceedings. ICCSC'02. 1st IEEE International Conference on*. IEEE, 2002, pp. 240–245.

[14] The ArchC Architecture Description Language website. [Online]. Available: http://archc.sourceforge.net

[15] F. Ghenassia, *Transaction-level modeling with Systemc: TLM concepts and applications for embedded systems*. Springer Verlag, 2005.

[16] J. Ou, P. Brunmayr, F. Muhammad, J. Haase, and C. Grimm. (2010) TU Vienna SystemC AMS Communications Library Documentation. [Online]. Available: http://www.systemc-ams.org