SystemC-based Power Simulation of Wireless Sensor Networks

Jan Haase, Markus Damm, Johann Glaser, Javier Moreno, Christoph Grimm
Vienna University of Technology
Institute of Computer Technology
Gusshausstraße 27-29/E384, 1040 Wien, Austria
EMail: {haase,damm,glaser,moreno,grimm}@ict.tuwien.ac.at

***Abstract*—Networks consisting of many autonomous sensors are gaining importance. Most wireless sensors have small batteries and must therefore be designed to consume very little power.**

In this paper, an approach for whole-system simulation for ultra-low power wireless sensor networks is proposed. To be able to estimate the power consumption of the whole network, the simulation framework must not only simulate the sensor nodes themselves, but also the overall system consisting of sensor nodes and other elements which can be much more sophisticated. It therefore includes an Instruction Set Simulator for better accuracy. To speed up the simulation Transaction Level Modeling (TLM) is used, with SystemC as the base for the simulation framework.[1]

I. Introduction

Wireless Sensor Networks (WSN) is a very active research field. However, there is still a wide gap between the requirements of most applications and the power consumption of platforms and nodes.

Applications of WSNs have huge differences in the requirements depending on the lifetime, the energy available, the kind of communication, security needs, etc. The application scenario described in this paper is an automotive WSN. The main constraint is, once more, energy consumption, as battery replacement is usually not feasible, e.g. in tire pressure sensors.

In Figure 1, a possible topology of a sensor network with wired and wireless connections is shown. Due to the multi-hop protocol used, some sensors would act as transition nodes. Thus the network perspective must also be taken into account.

Evaluation of architectures, techniques and protocols designed for energy saving is frequently done by virtual prototyping and simulation. Nevertheless, modelling wireless sensor networks and nodes involves the development of hardware, software, and network models. Co-simulation becomes necessary at the expense of simulation performance and interoperability issues.

To deal with co-simulation requirements and interoperability, a SystemC simulation framework utilizing Transaction Level Modeling (TLM) is being developed. SystemC permits simulating both, hardware and software, while TLM permits to efficiently simulate the communication model.

One specific energy optimization strategy proposed in this project is the development of ultra low power reconfigurable sensor node building blocks, designed to relieve the Micro Controller Unit of some of its tasks. This strategy makes the hardware/software co-simulation even more important.

This paper is structured as follows: In the next Section a view on related work is given. In Section III the ultra-low power reconfigurable blocks are explained. The proposed simulation framework is detailed in Section IV, followed by a short Section with first results. The paper then concludes in Section VI.

II. Related work

Power estimation is today an established technique for digital systems. Only very few works exist that go into the AMS domain and allow estimation of power consumption, e.g. [1]. In these works, power is estimated based on the use of functionalities in the specific case of

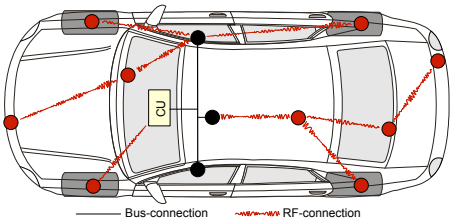

Fig. 1. A wireless sensor network in an automotive scenario with a central unit (CU) and an additional field bus

[1]This work is conducted as part of the Sensor Network Optimization by Power Simulation (SNOPS) project which is funded by the Austrian government via FIT-IT (grant number 815069/13511) within the European ITEA2 project GEODES (grant number 07013).

sensor networks. Estimation is done via measuring (or determining by circuit level simulation) the power consumption of a circuit when it is used. Power consumption in sensor networks then results by the duty cycle periods.

There are commercial solutions, like PowerOpt (ORINOCO) [2], to estimate power at the system level. In WSNs, network behaviour must be taken into account for the activity estimation [3].

Power optimization of WSNs has been previously analyzed in the Power Aware Wireless Sensor (PAWiS) project [4]. In fact, the PAWiS simulation framework is the starting point of our own framework. It is based on the OMNeT++ Simulation Environment [5] and abstracts both intra-node and inter-node communication as a number of modules that exchange messages among them. This approach is also used in this paper. Nevertheless, instead of OMNeT++, SystemC with TLM 2.0 is used (see Section IV).

Another feature of PAWiS is the environment and RF transmission model, the so called "air object". It registers the 3D position of each node and based on this information, it models the RF transmission considering attenuation, obstacles, noise and interference. The PAWiS framework supports also dynamic behaviour with a Lua scripting interface. However, there are some differences that must be overcome. The PAWiS framework does not include an Instruction Set Simulator (ISS), with the consequent lack of accuracy in the Micro Controller Unit (MCU) model. Therefore, we target to improve the PAWiS approach by including an ISS for better accuracy.

In [6] SystemC was successfully used to co-simulate the hardware and software parts of a wireless sensor network. Some preliminary tests were made with a pure SystemC model with excellent results. However, the lack of a network library for SystemC led to the usage of NS-2 [7] to finally model the network.

TLM has already been used successfully for network simulation. In [8], a protocol for cooperative MIMO mobile sensor networks was proposed. To model such a large and complex network, TLM and SystemC were used, resulting in a very fast and efficient simulator that permitted to evaluate the network performance.

III. Ultra low power approach

Sensor nodes normally consist of an MCU, the sensor itself, and a transceiver, as well as the associated subsystems, like memories or Analog Digital Converters (ADC). The widely used energy-saving concept at node system level is to switch components to an inactive low-power state when no activity is necessary. In the course of the PAWiS project [9] it was revealed that many simple control and processing tasks are still accomplished by microprocessors as software programs, which is a main source of energy waste.

A. Relieving the microprocessor

One main task of a sensor node is to periodically perform some measurements (e.g., temperature, strain, insolation) and generate a network packet to transmit the current value. This consists of some relatively simple sub-tasks: activate the sensor, wait until its value has settled, initiate the ADC conversion, and read the ADC value. If this value changed compared to the previous one, this value is transmitted via the network. This is the more complex part of the measurement task which consists of the generation of a network packet and the protocol handling.

Unfortunately, for every such simple activity the CPU has to leave its low-power sleep mode, even if its activity is over after the initial simple tasks. During this active state several waiting conditions (e.g., sensor settling time, ADC conversion time, listening interval) further deteriorate the power consumption. To avoid these periodic CPU activations, our proposal introduces dedicated *reconfigurable hardware blocks* which independently conduct the simple sub-tasks instead of the CPU. The CPU is only activated if any further (more complex) processing is required (see Figure 2). Therefore these logic blocks act as a "filter" for these events.

B. Using reconfiguration

Shifting the border in a software/hardware partitioning process towards hardware reduces the flexibility of the final application. While software can be modified by reprogramming the code memory, synthesized logic cores require a redesign of the chip.

There are three important reasons for flexibility:

1) Cover multiple different applications for a higher market potential.
2) Adopt to different external components across PCB design cycles.
3) Fix bugs without a chip redesign.

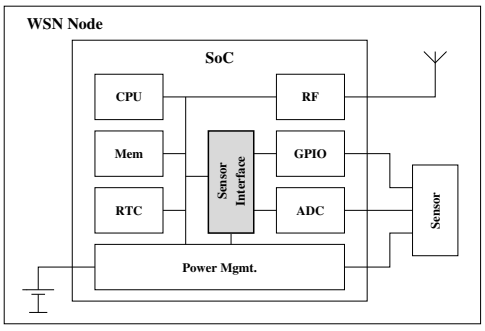

Fig. 2. WSN node with reconfigurable hardware block as Sensor Interface

Therefore we propose to make the introduced dedicated hardware blocks *reconfigurable*.

In contrast to FPGA cores (e.g., [10]), which consist of fine-grained structures, optimal for control dominated functions, we propose to also support multi-bit logic blocks and a multi-bit routing architecture for computational functions. To further reduce the power consumption and area requirement, the proposed approach requires the user to define an *application class* for which the reconfigurable block is inserted. This class describes the field of planned actual applications. Then the reconfigurable logic block is developed to be tailored to provide exactly those structures which are required to implement any of the desired applications. After the manufacturing of the SoC, the actual application is specified and implemented by configuring the reconfigurable block accordingly.

IV. Simulation

To be able to have a simulation of the concepts from the preceding Section, in this Section means for the modelling and simulation of whole systems are described.

A. Requirements

The MCU is the most complex subsystem within the sensor node architecture. In order to be able to compare the efficiency of different algorithms, a cycle-accurate model is preferable. An ISS provides this level of abstraction but at the expense of reducing simulation speed.

The hardware level of the simulation framework requires hardware models which can be modelled as finite state machines and interact with the ISS. Finally, the network level requires a topology and traffic model to estimate which nodes are more likely to run out of energy and which protocols distribute load in an optimal way so that the overall network lifetime is maximized.

Summarizing, the simulation framework has to meet the following requirements:

- The simulation of the single sensor nodes has to provide a detail level that is sufficient for determining meaningful information on power consumption.
- It must be possible to simulate a mixed wireless / wired sensor network, where the wired part would be a field bus in the targeted application scenario.
- The simulation should also give the designer information on how a chosen protocol or certain protocol parameters might influence the overall power consumption as well as the power consumption of a chosen node.

B. Using TLM for the simulation framework

As described in the preceding Subsection, modelling and simulation on a high detail level is required, including an ISS. This requires simulation speedup on the network level to keep the total simulation time within a reasonable range. Therefore, we plan to utilize TLM by using SystemC together with TLM 2.0 as a base for the simulation framework (see [11], [12]). This allows for an easy connection to the C++-based in-house ISS of our industry partner Infineon. The integration of an ISS into a TLM 2.0 SoC model has been demonstrated in [13].

TLM is mainly intended for systems using bus communication, where the main idea is to bundle the low-level events occuring in bus communication ("pin wiggling") into a single data structure called *transaction*. For example, a read request from a CPU to a Memory module together with the resulting data transfer can be captured by one transaction.

The basic idea of our approach is to model the "air object" (see Section II) within the simulation framework similar to a bus in TLM (see Figure 3). While the bus model in a TLM system handles address decoding and establishes communication from the so called *initiator* to the intended *target*, a shared medium as used by wireless transmission or a wired network like the CAN bus requires the addressing to be handled by every listening node itself. This is necessary to appropriately model the functionality and this power consumption of the RF receiver. Additionally, buses like the CAN bus use message IDs instead of addresses, where several nodes might be interested in the message. This is also relevant for network broadcasts.

Therefore, it is obvious that we have to alter the TLM approach for our purposes. This will include most likely adding additional parameters to the transaction via the generic payloads extension mechanism (e.g. signal

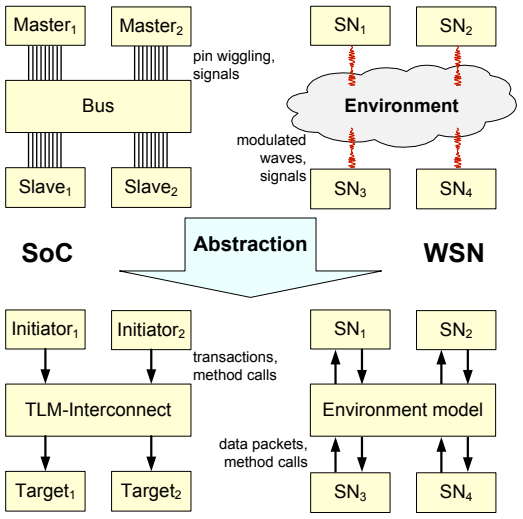

Fig. 3. Similar abstractions for SoCs and WSNs (SN: Sensor Node)

strength), but could also extend to introduce new phases to capture typical wireless communication steps like carrier sensing. A successful adaptation of TLM 2.0 to wireless sensor network yields multiple benefits:

- With regard to the mixed wireless / wired sensor network scenario, we stay within a modelling paradigm where bus modelling is straightforward.
- We can benefit from the general simulation performance enhancement prospects of TLM (although it might be problematic to apply the loosely timed coding style for WSN).
- Using TLM to model the sensor node internals enables seamless integration of node and network simulation.

Moreover, a whole new (somewhat orthogonal) view on power consumption in a wireless sensor network is achieved by associating power consumption information to transactions. The influence of a single transaction on the power consumption of the whole system or a certain critical node is investigated to optimize WSN protocols.

V. First results

As a proof of concept, the above mentioned sensor interface example was implemented as a hardware circuit utilizing the proposed energy optimization approach. Instead of the production of a chip, the circuit was realized with an FPGA. Note the stack of reconfigurability, where the underlying technology (FPGA) is reconfigurable and the implemented circuit itself is reconfigurable, too. Due to the usage of a commercially available FPGA instead of a full- or semi-custom chip design, no optimized multi-bit cells were implemented. Because of the early state of the research for the simulation framework, no simulations for the reconfigurable blocks were performed yet.

To show the actual power reduction, a comparison of the very same task in three different implementations is performed. An implementation utilizing the proposed approach with a reconfigurable state-machine is used as first case. The second implementation is performed as a hard-coded state machine. Both are realized with an FPGA. The third implementation is a software program running on a microcontroller.

The energy consumption values of a single sensor measurement performed by the three different implementations are summarized in Table I. The FPGA implementation with a reconfigurable state-machine consumes ≈2.7 times more energy. The microcontroller implementation requires 30 times the energy for every measurement compared to the reconfigurable FPGA implementation. This gap stems from the high power consumption of the microcontroller (1.2 mA), even when waiting for the completion of the ADC conversion or until the sensor output has settled.

TABLE I
Energy consumption of a sensor measurement performed by different implementations.

Implementation	Energy
FPGA: Reconfigurable	5.61 nJ
FPGA: Hardcoded	2.09 nJ
Microcontroller	168.30 nJ

VI. Conclusion

A novel approach for the design and optimization of wireless sensor nodes was presented. The node internals as well as the communication among the nodes including the environment are modeled with SystemC and TLM. Extensions to the TLM framework are required to map the broadcast characteristic of the wireless medium to the message transmission. As one optimization strategy a new approach is presented to relief the microprocessor of power-intensive, yet simple tasks by accompanying the microprocessor with coarse grained reconfigurable hardware blocks. The comparison of power consumption of the sensor interface implemented in an FPGA to the implementation with an ultra-low-power microcontroller shows a large reduction by a factor of 30.

References

[1] J. Glaser, D. Weber, S. A. Madani, and S. Mahlknecht. Power Aware Simulation Framework for Wireless Sensor Networks and Nodes. *EURASIP JES*, 2008.

[2] PowerOpt. http://www.chipvision.com.

[3] W. Nebel. System-level Power Optimization. In *DSD. Euromicro Symposium*, 2004.

[4] D. Weber, J. Glaser, and S. Mahlknecht. Discrete Event Simulation Framework for Power Aware Wireless Sensor Networks. In *INDIN*, 2007.

[5] A. Varga. *OMNeT++ Discrete Event Simulation Sytem User Manual*.

[6] F. Fummi, D. Quaglia, F. Ricciato, and M. Turolla. Modeling and simulation of mobile gateways interacting with wireless sensor networks. In *DATE*, 2006.

[7] S. McCanne and S. Floyd. NS Network Simulator - version 2. Website. http://www.isi.edu/nsnam/ns.

[8] Q. Zhang, W. Cho, G. Sobelman, L. Yang, and R. Voyles. *Ubiquitous Intelligence and Computing*, pages 508–516. Springer Berlin/Heidelberg, 2006.

[9] S. Mahlknecht and M. Rötzer. Energy Supply Considerations for Self-sustaining Wireless Sensor Networks. In *EWSN*, 2005.

[10] S. Wilton, N. Kafa, J. Wu, K. Bozman, V. Aken'Ova, and R. Saleh. Design Considerations for Soft Embedded Programmable Logic Cores. *IEEE Journal of Solid-State Circuits*, 40(2):485–497, February 2005.

[11] Open SystemC Initiative. *OSCI TLM2.0*. http://www.systemc.org.

[12] J. Aynsley. OSCI TLM2 User Manual. Technical report, Open SystemC Initiative, 2008.

[13] J. Bennett. Building a Loosely Timed SoC Model with OSCI TLM 2.0, 2008. Embecosm Application Note 1, Issue 1.