Flexible Simulation and Prototyping for RFID Designs

C. Angerer, B. Knerr, M. Holzer, A. Adalan, and M. Rupp
Vienna University of Technology, Institute of Communications and Radio Frequency Engineering
Vienna, Austria
(cangerer, bknerr, mholzer, aadalan, mrupp)@nt.tuwien.ac.at *

***Abstract* - This paper presents a flexible RFID simulation and prototyping system supporting both the 13.56 MHz domain and the 868 MHz domain, further referred to as HF and UHF domain, as well as several standards within these two frequency domains. The proposed prototyping system consists of an exchangeable analog RF front end supporting either HF or UHF, a protocol stack running on a DSP and a signal processing part implemented on an FPGA. Both software and configuration files for the reconfigurable hardware running on the prototyping board, are automatically generated from a powerful simulation environment, which is decomposed in a link layer model and a physical layer model. The link layer model covers the protocol stack and the command generation down to bit level, whereas the physical layer model additionally includes simulation of the modulation and coding schemes as well as the utilised receiver architecture. This allows for the evaluation of different protocols in general and to test the corner cases of the permitted parameter ranges in realistic scenarios in particular. Furthermore, this automated design flow ensures consistency of the simulation and the implementation with very small design cycles.**

I. Introduction

RFID technology shows a continuous growth in various application fields, like commerce, logistics, medical science, security, access control etc. The past trend in industry was going towards different standards for each kind of application. Some improvements in untangling these mostly conflicting standards were accomplished by EPCglobal, who created widely accepted standards for product identification. The EPC Class 1 Generation 2 standard was intended to resolve discrepancies among RFID systems used in different parts of the world [1].

Similar to a trend in other technologies, where one electronic device incorporates many different functionalities for which in the past separate devices were required, this development also emerges in the field of RFID. An urgent demand for tags and readers that are capable of operating at different frequencies, from HF to UHF, offering different application scenarios exists in this field. Moreover, as long as there are competing standards within every single frequency domain and consequently tags and readers that adhere to only one of those, the interoperability of the new tags and readers is another urgent demand. In short, the next generation of tags and readers shall be omnipotent. Therefore, a need emerges to focus on the interoperability between RFID systems just as much as on the common demands like for example small size, low power, and low cost. Some multi-frequency approaches and multi-standard systems for adjacent frequencies have already been developed by industry. Examples include Anadigm's RangeMaster [2], an RFID reader supporting various UHF protocols, in particular the EPCglobal Gen 1 and Gen 2 (class 0, 1, 2) and ISO18000-6 standards (ver. a,b), or combined LF and HF Readers which are offered by many manufacturers, e.g. by Deister Electronic [3] or Texas Instruments [4]. Some companies even provide a combination of technologies, such as Magnatec Technologie GmbH [5] that has combined RFID and GSM technology into an RFID scanning system that uses the GSM cell phone network for the communication between remote readers and a base station.

*This work has been funded by the Christian Doppler Laboratory for Design Methodology of Signal Processing Algorithms.

The rapid implementation and exploration of such multi-frequency, multi-standard systems is a tough challenge, since naturally multiple aspects need to be considered: first and foremost the different signalling technologies (e.g. inductive coupling vs. wave propagation), simulation on a multiple levels of abstraction (protocol stack vs. hardware implementation), cross platform verification, and the need for highly automated code generation. Research on simulation models and implementation of RFID systems has been carried out by various groups, addressing different problems.

Han et al. [6] presented a Matlab/Simulink model for the communication between a single reader and a single tag in the UHF frequency domain. They focus on modelling the RF and analog circuit and carefully analyse effects like oscillator phase noise, TX/RX coupling, I-Q balancing etc. Choi et al. [7] developed a PSpice and a digital VHDL simulation model for an HF reader and also present experimental results. The focus of their work is on multi-tag recognition and collision avoidance. Roy et al. [8] present an RFID reader implementation on an FPGA and an RF frontend in the UHF frequency domain. Their aim is to keep the implementation simple by the preferred usage of standard components.

The rationale for the described system in this paper is to respond to the need for a test environment for verification of future developments of RFID systems. Therefore, we focus on a rapid prototyping approach for implementing several different standards belonging to the HF and UHF frequency domains. A powerful simulation environment on two different abstraction levels, namely a link layer and a physical layer simulation model, is introduced. This allows for a fast evaluation of our implementations on a very abstract level. Furthermore, an automated design flow is presented, which generates `C/C++` code for the DSP and synthesisable VHDL code from both simulation models. Thus, new protocols can be rapidly transformed to the prototyping system. This prototyping system consists of a DSP, an FPGA, and an analog RF frontend including a DAC and an ADC. Our architectural approach is to map all signal processing and protocol state machines into the reconfigurable components of the reader, namely FPGA and DSP, and to keep only the layout of the RF frontends for HF and UHF fixed. These RF components are kept very simple and independent from the actually implemented standard. Within minimal design cycles different RFID reader solutions, supporting standards of the HF and UHF frequency domain, can be evaluated on different layers, from abstract simulation levels down to verification on hardware.

Additionally, the standards allow for a wide range of different parameters, like coding, link timing, modulation depths, slew rate, etc. Regarding these parameters, our implementation provides full control, meaning that the complete range of these parameters can be explored in order to comprehend the limitations of the system. For instance, the maximum read distance, the maximum number of detected tags per time unit, and achievable bit error rates are of major interest for a realistic scenario. The rapid prototyping approach has been tested with the draft version of the EPCGlobal HF Ver. 2 standard and the EPC RFID Class-1 Gen-2 UHF standard [1, 9] as well as the ISO 15693 standard [10].

The remainder of this paper is organised as follows: Section II. introduces our design flow and design methodology. Section III. presents our link layer and physical layer simulation models. In Section IV. the rapid prototyping board is introduced, while Section V.

and Section VI. show some simulation and measurement results. The last section concludes this paper.

II. Description of Design Flow

The design flow of our RFID prototyping system consists of three layers, a link layer model, a physical layer model, and a rapid prototyping board (Figure 1). The link layer model serves a simulation model of an RFID reader and several tags. This model implements the protocol state machine of the various RFID standards and allows for generating the bit stream of the communication between reader and tag. This layer does not consider any communication channels, coding, modulation, or board related restrictions and simulation of analog components. The communication between reader and tag operates at bit level. Thus, the validity of the generated commands and its corresponding bit sequences are verified at this layer. The reader is simulated using `C/C++`, whereas the responses of the tag are emulated with `SystemC`. This allows us for transferring the code of the reader directly to the DSP on the rapid prototyping board, only adjusting some global definitions, while the `SystemC` model of the tag is used as a testbench for this simulation.

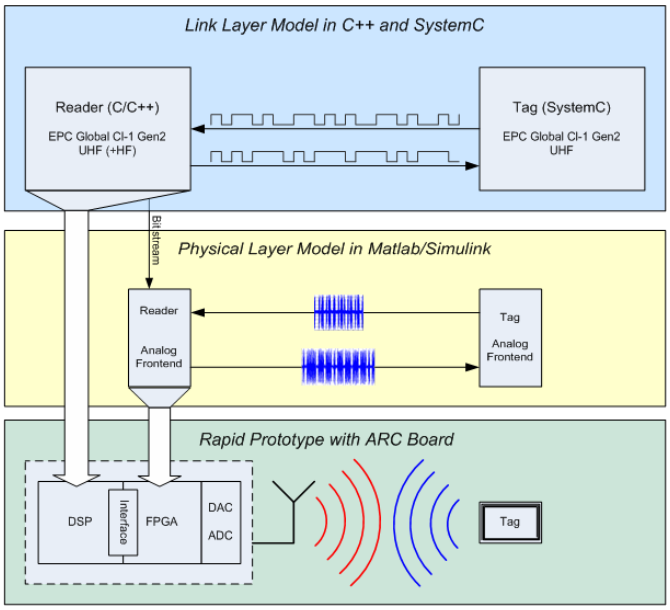

Figure 1 - Design flow for the rapid prototyping system

In a second step the model of the reader is refined by introducing a physical layer model. It additionally simulates the required modulation schemes and the utilised receiver architecture, extracting bits out of the analog receive signal. This physical layer model is constructed using Matlab/Simulink with the HDL Coder toolbox and the Xilinx' System Generator. An automatic VHDL code generation from the Simulink model is supported using these tools. This enables a push of a button solution and ensures consistency of the simulation model and the rapid prototyping design. Furthermore, the so-generated VHDL code can be directly imported into a Modelsim simulation environment for a cycle accurate VHDL simulation, which is fully equivalent to the later implemented FPGA architecture.

By the described design flow, design cycles can be reduced dramatically. The entire system is described using `C/C++/SystemC` and Matlab/Simulink, hence providing great flexibility and shortest evaluation time. Implementable and synthesisable code for the DSP and FPGA of the rapid prototyping board is generated automatically out of this simulation model.

III. Simulation Environment

The link layer model provides the facilities to simulate and test an RFID communication system on a high level of abstraction. This model is written in `SystemC V2.0.1` and incorporates a reader and several instances of tags to allow for test scenarios that verify the anti-collision protocol and the detection performance of a whole pallet with tags. Furthermore, the communication parameters like link frequency and decoding types can be adjusted during a simulation run depending on the size of the tag population or general channel characteristics. Additionally, the application scenario is programmable on a larger scale to enable the designer to create a more complex interaction without specifying any single command. For instance, assume a scenario in which a subset of the detected tag population has to be permanently killed, whereas the other two thirds just have to be memory locked distinguished by their manufacturer ID.

This environment establishes also a verification platform for the tag implementation. Here, a high level model of the tag can be replaced by its refined RTL description. Thus, the same test cases can be applied to the RTL implementation of the tag. Currently, a `SystemC V1.0` model of the tag has been verified as well as a corresponding VHDL implementation, which has been co-simulated (`C++/SystemC` and VHDL) in Modelsim. Both the SystemC and VHDL version of the tag have been provided by our industrial partner Infineon and represent low level (register transfer level) implementations of the standards EPCglobal Class 1 Gen 2 Tag for UHF and of a draft version for HF.

The link layer implementation of the reader consists of a sender, a receiver, and a control state machine. These core components of the reader have been implemented in `C/C++` to allow for direct cross-compilation onto the DSP of the rapid prototyping board.

The Matlab/Simulink model is a further refinement of the described link layer model. Combining the bits-to-send from the link layer model and a set of parameters, this Matlab/Simulink model forms the required transmit waveforms of the RFID interrogator. Figure 2 shows the transmit path, which consists of the protocol stack generating the bits (link layer model), the DSP to FPGA interface forwarding these bits, a TX state machine controlling the interface and switching to continuous carrier mode. After the pulse interval encoder (PIE), an amplitude shift keying (ASK) modulator, and a transmission filter, the signal is upconverted to 13.56 MHz. In receive direction, the bits are extracted from the received signal. The receiver basically consists of a filter, a moving average, a slicer, a symbol decoder and an FPGA to DSP interface. These units are controlled by a receive state machine and a synchronisation unit (Figure 2).

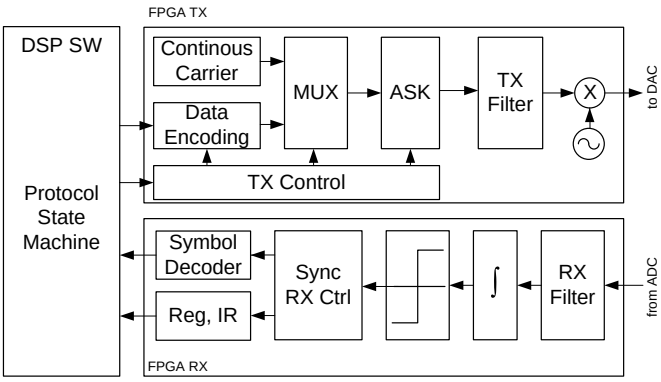

Figure 2 - FPGA transmit and receive paths

This Matlab/Simulink Model composed of blocks from the Xilinx' System Generator and the HDL Coder can be automatically transformed into synthesisable VHDL code. Furthermore, the generated VHDL code can be imported into a simulation with Modelsim. In transmit direction the input for this Modelsim simulation operates at bit level, which is provided by the link layer model. In receive direction a testbench is used, which generates the ADC input signal.

Moreover, observers were developed which track the generated ouput signals. The Modelsim simulation includes all interfaces to the neighbouring components of the FPGA (DSP and ADC/DAC). These interfaces are defined by the physical architecture of the rapid prototyping board and have been manually designed. The generated VHDL code from the physical layer model however can easily be embedded using a wrapper module, without touching these manually designed blocks. Hence, these interfaces are independent from the currently implemented reader and do not need to be adapted. This allows for an automatic transformation from the Matlab Simulink model to the rapid prototyping board.

IV. Prototyping Board

The rapid prototyping hardware consists of a Texas Instruments TMS320C6416 fixed-point DSP with an ethernet interface to a PC, a Xilinx Virtex II FPGA, two 16 bit DACs, and two 14 bit ADCs. The RF frontend amplifies and transmits the generated TX signal and downconverts the RX signal using a carrier suppression bridge circuit, an envelope detector and a common mode rejection module. For the UHF frequency domain, the RF frontend provides additional upconverters from and downconverters to 13.56 MHz. The frontend was devoloped in collaboration with the radio frequency engineering group of our institute [11]. The FPGA, DAC, and ADC use a common clock frequency of 40 MHz. A block diagram of the components of the rapid prototyping board is shown in Figure 3.

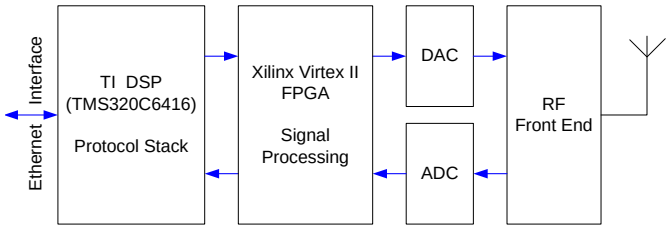

Figure 3 - Block diagram of rapid prototyping board

To demonstrate the feasibility of the proposed design flow simulation results and measurement results are presented in the following sections.

V. Simulation results

The Matlab/Simulink design introduces a comfortable framework to examine the channel characteristics and RF frontend properties to speed up the development of powerful receiver structures and shall later on serve as an exact model for the physical layer to minimise the effort for real measurements. In Figure 4 the simulation setup of a the fundamental receiver structure is depicted. The simulation model consists of the generator for the noisy receive signal and the receiver components RX filter (order 8), moving average (length of the

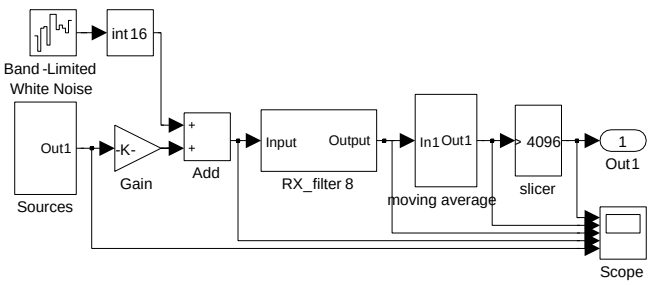

Figure 4 - Simulink Model of Receiver

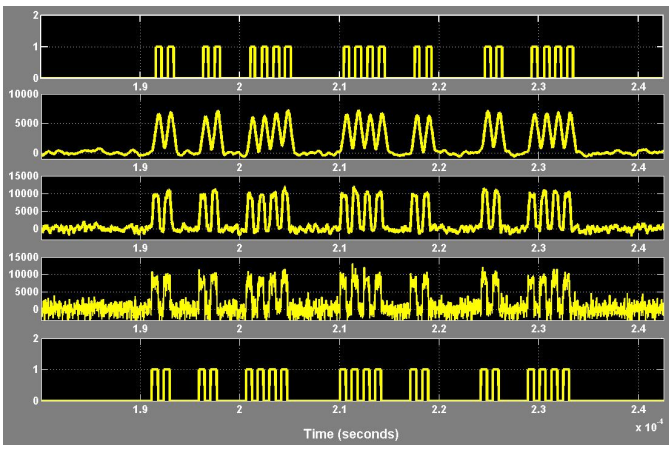

Figure 5 - Simulation Results of Simulink Model

half period of the link frequency) and slicer (with programable decision threshold). The sampling frequency and bitwidths are matched to those values on the prototyping board. Utilising the filter design tools of Matlab and the simulation capabilities of Simulink, these blocks can be evaluated and adapted with a very small design effort. Figure 5 shows (from bottom to top) the RX bits, the noisy version of the RX signal, the filtered and the integrated RX signal, and finally the decided RX bits. The RX signal is an FM0 encoded sequence at a link frequency of 847 kHz, as it is expected at the ADC after the envelop detector on the RF frontend. We use the HDL Coder toolbox of Simulink to generate synthesisable VHDL code from the blocks of interest and embed them into our receiver design. Hence, the hardware realisation of the receiver design can be verified using Modelsim. Consequently, it is ensured that the simulation results adhere perfectly to the expected outcome on the prototyping board.

VI. Measurement results

Several different implementations of various standards have already been carried out on our rapid prototyping hardware. Parameterised implementations provide the full range of system characteristics in order to allow for a profound analysis of the system and its limitations. Typical parameters of interest are for instance: transmit power, link timing, modulation, coding, data rate, additional pilot sequences and symbol characteristics.

Figure 6 shows two measurements of the board output for the transmit sequences of a query command, that originates from the latest draft of the EPCGlobal HF Version 2 standard with different parame-

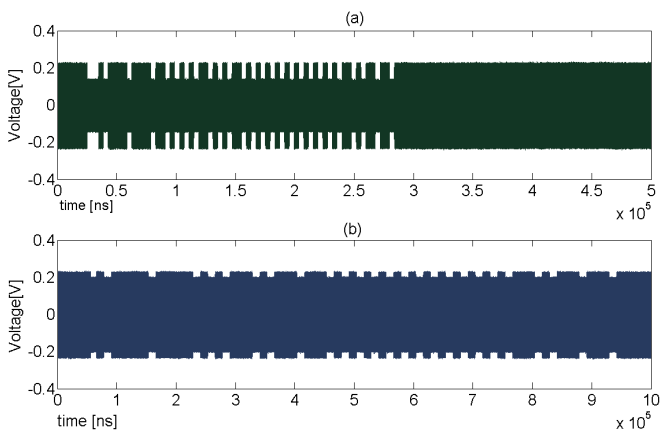

Figure 6 - Pulse Intervall Encoded Transmit Sequences

ters (within the range of the standard). The signals are already upconverted to the carrier frequency of 13.56 MHz. The query sequences modulate their information onto the continuous carrier via pulse interval encoding (PIE) with different modulation depths and different data rates.

The parameters for the sequence in Figure 6a: modulation depth: 30%, pulse width: $4\mu s$, type A reference interval (tari): $8\mu s$, reader to tag calibration symbol (RTCal) and tag to reader calibration symbol (TRCal): $20\mu s$.

The parameters for the sequence in Figure 6b: modulation depth: 15%, pulse width: $12\mu s$, tari: $25\mu s$, RTCal and TRCal: $75\mu s$.

In a second example a reader compliant to the ISO/IEC 15693 standard has been implemented and measurements with off-the-shelf tags have been carried out. Figure 7 shows the inventory command of

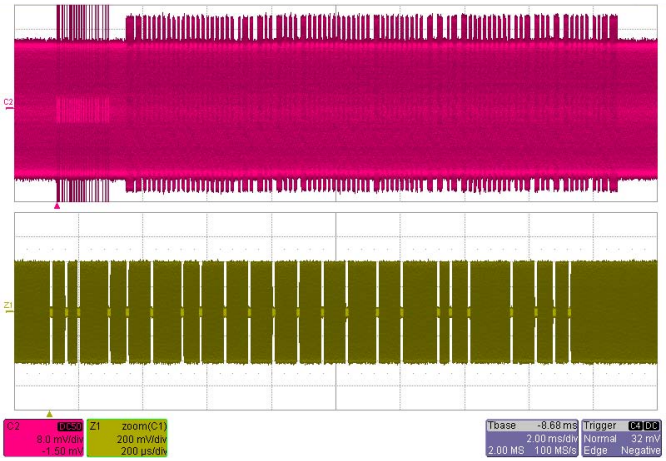

FIGURE 7 - INVENTORY COMMAND AND TAG ANSWER

the reader (1 out of 4 encoding) and the corresponding tag response. The upper signal (pink) is measured after the carrier suppression module. It shows the inventory command in the first three milliseconds and the following Manchester encoded tag response. The lower screen shot (green) is a zoom in the inventory command from the reader (first three milliseconds of the upper screen shot), measured at the antenna coil. The tag to reader distance in this measurement is about 10 cm.

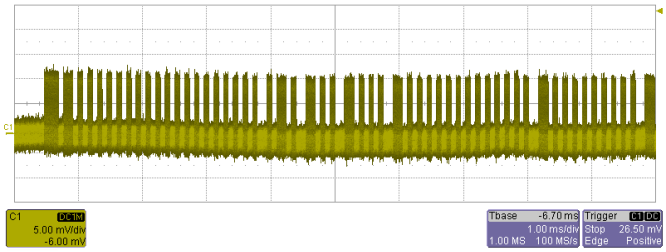

FIGURE 8 - ADC INPUT AFTER ENVELOPE DETECTOR

Figure 8 shows the ADC input of the tag answer (same as in Figure 7) after the envelope detector.

VII. CONCLUSIONS

A rapid prototyping system for RFID readers has been presented. With our exchangeable RF frontends the HF as well as the UHF frequency domain are supported. Powerful simulation environments offer shortest design and system verification cycles. These simulation environments are split into a `SystemC` based link layer and a Matlab/Simulink physical layer simulation for the evaluation of the protocol stack of the reader and the signal processing parts, respectively. Furthermore, we present a design flow that automatically generates synthesisable VHDL code from the physical layer model and `C/C++` code for the DSP from the link layer model, respectively. By means of this automated code generation, consistency of the behaviour of the simulation models and the executed code on the rapid prototyping board is fully ensured. The rapid prototyping hardware is independent from the actually implemented HF or UHF standard, since all standard dependent properties are mapped into the automatically reconfigurable hardware components, namely DSP and FPGA. Three different standards of two frequency domains have been implemented and examined by simulations and measurements and a complete communication link for off-the-shelf tags (ISO/IEC 15693) has been established in order to substantiate the service capability of the proposed rapid prototyping design methodology.

Future work will focus on the development of a framework supporting multiple receive and transmit antennas to enable MIMO RF-ID systems. By this means it is expected to greatly improve the detection range, the data rate, and to incorporate precise object localisation of existing and future RF-ID standards.

ACKNOWLEDGEMENTS

We would specially like to thank our industrial partner Infineon Technologies for enabling this work and supporting us with many advices and discussions. They also provided a `SystemC` testbench of the tag in order to test the link layer model. Moreover we would like to thank Austrian Research Centers, who supported us with a *SmartSim* rapid prototyping board.

REFERENCES

[1] EPCGlobal. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID. `http://www.epcglobalinc.org`.

[2] Anadigm. `http://www.anadigm.com`.

[3] Deister Electronic GmbH. `http://www.deister.com`.

[4] Texas Instruments Incorporated. `http://ti.com`.

[5] Magnatec Technologie GmbH. `http://www.magnatec.de`.

[6] N. Choi, H. Lee, S. Lee, and S. Kim. Design of a 13.56 MHz RFID System. In *The 8th International Conference on Advanced Communication Technology, Vol. 1, p.840 - 843*, October 2006.

[7] Y. Han, Q. Lin, and H. Min. System Modelling and Simulation of RFID. March 2006. `http://www.autoidlabs.org`.

[8] N. Roy, A. Trivedi, and J. Wong. Designing an FPGA-Based RFID Reader. *XCell Journal*, pages 26–29, Second Quater 2006.

[9] EPCGlobal. EPC Global HF Air Interface Version 2, Document Version 0.1, November 2006.

[10] ISO / IEC. ISO / IEC 15693, Identification Cards - Contactless Integrated Circuit Cards - Vicinity Cards, January 2000.

[11] A. L. Scholz L. W. Mayer. Efficient Measurement Methods for UHF Transponder Antennas. In *The First International EURASIP Workshop on RFID Technology*, September 2007.