A Flexible Dual Frequency Testbed for RFID

Christoph Angerer, Martin Holzer, Bastian Knerr, Markus Rupp
Institute of Communications and Radio Frequency Engineering
Vienna University of Technology
Gusshausstrasse 25/389, 1040 Wien, Austria
{cangerer, mholzer, bknerr, mrupp}@nt.tuwien.ac.at *

ABSTRACT

This paper presents the setup of a testbed developed for the fast evaluation of RFID systems in two frequency domains. At the one hand the 13.56 MHz and at the other hand the 868 MHz frequency domain are supported. The suggested design flow for configuring the testbed is highly automated and supports rapid evaluations of different designs and implementations within shortest time. Several layers of abstraction for rating existing and future RFID standards are provided, from abstract simulation models down to the implementation on the rapid prototype. This includes a fast setup for measuring and rating existing RFID standards, or running them side by side, as well as the rapid exploration of future RFID designs and various aspects of those prototypes. It allows for verifying both protocol stack- and signal processing of RFID systems. In order to demonstrate this concept, we briefly describe an implemented example system, and our overall hardware layout in detail. Measurement results on two existing RFID standards corroborate the concept.

Keywords

RFID, reader, testbed, rapid prototyping, dual frequency

1. INTRODUCTION

Radio frequency identification (RFID) is an automatic identification technology such as the bar code or magnetic strip technology [5]. An RFID system consists of two basic components: the RFID reader also called interrogator and one or several RFID tags. The task of the interrogator is to read a unique identification number from the tag, which is usually attached to some object. The tag can be powered by a battery (active tag) or receives its energy from the electromagnetic field (passive tag) provided by the reader. In the case of a passive tag the reader has to emit a strong continuous carrier wave to the tag in order to supply it with energy. This task has to be fulfilled during the time of interrogation and also during the time of tag to reader communication, which results in a strong carrier interference with the tag response at the receiver of the reader. The main challenge in RFID reader design is to detect and decode the signals from the tag in the presence of this strong interference.

RFID is a very fast emerging and developing technology with a wide range of applications in different fields, like commerce, logistics, medical, security and access control as well as many others. With the tremendous technological increase since the very first beginnings, the spectrum of applications has been growing enormously, leading to many different standards in several distinct frequency bands for supporting these applications. However, most of these standards are conflicting each other and this variety also yielded to many implementations, only suitable for a single application but not compatible with other RFID equipment.

Therefore, todays and future RFID systems are required to support several of these standards, including the support of different frequency ranges. Hence, this demands a multi standard, multi frequency RFID system that provides access to a wide variety of RFID applications. Additionally, complexity of future systems is increasing dramatically which calls for an early exploration of the systems to figure out any possible technological or design limitations, and to estimate the upcoming design effort in an early stage of the development. New technologies are introduced in all fields of wireless communication, and thus also in RFID, which potentially hide unforeseen challenges due to a lack of experience on implementing these new algorithms. Hence, a rapid implementation is mandatory to tackle the demands for an early and flexible system exploration on several abstraction layers, from abstract simulation down to an implementation on hardware. Moreover such a rapid prototyping system assists in obtaining several design alternatives, thus guiding to an optimal realisation of new ideas.

In order to meet these requirements, we set up our rapid prototyping testbed for RFID designs. It allows for exploring several standards within two frequency ranges, namely the HF (13.56 MHz) and the UHF (868 MHz) domain, thus being capable of troubleshooting compatibility issues. With our highly automated design flow a very fast exploration of new systems in simulations as well as on hardware is achieved. The presented testbed is very flexible with respect to different implementations, meaning that running systems can be rapidly adapted to additional functionality

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

TRIDENTCOM 2008, 17th – 20th Mar 2008, Innsbruck, Austria.

DOI 10.4108/icst.tridentcom.2008.3138

or different standards.

Many different groups are engaged in both fields, in RFID systems implementation as well as in testbed design. N. Roy et al. [8] present an implementation of a UHF RFID reader. Their focus is on a very basic implementation, using only off the shelf components and keeping the overall cost and design effort low. Hence, their design is neither reconfigurable and thus suitable for a rapid prototyping approach, nor extendable to additional standards and future design aspects. Related to this work, Choi et al. [2] present a PSpice and VHDL simulation model as well as experimental results. Their focus lies on the investigation of RFID core problems like collision avoidance and multi-tag detection, but they do not address rapid and flexible implementations of future systems either. Han et al. [6] present a 13.56 MHz domain RFID simulation model. They accurately model analogue components and effects like I/Q imbalance, TX/RX coupling, and oscillator phase noise. However, this work incorporates no implementation in hardware and hence no experimental measurements are provided. V. Derbek et al. [3] present a methodology and platform for UHF RFID system optimisations and verifications. They present simulation models and their real-time verification in application specific conditions, with a focus on the tag implementations in the RFID system.

The remainder of this paper is organised as follows. Section 2 introduces our design flow and simulation models. Section 3 presents our rapid prototyping board and the testbed setup. Section 4 discusses an example implementation of an HF RFID reader architecture in detail, while Section 5 shows some measurement results for two different, implemented standards. The last section concludes the paper.

2. RAPID PROTOTYPING APPROACH

This section will briefly introduce our proposed design flow and rapid prototyping approach [1]. The design flow of our RFID testbed can be subdivided into three layers: a link layer model, a physical layer model and the implementation on the rapid prototyping board (Figure 1). The link layer model serves as a simulation model on a very high level of abstraction. An RFID system consisting of a reader and one or several tags is modeled in `C/C++` and `SystemC`. The generation of commands and the protocol state machine are verified on this layer. The communication between reader and tags operates on a digital level (bits), meaning that no channel influence or specific receiver architecture is taken into account. Several complex test scenarios, like anticollision protocols, can be applied to the simulation. Additionally, the performance of the protocols regarding several parameters like detection performance of several tags, can be tested on this layer. The reader part of the model is coded in `C/C++`.

The link layer model is further refined by introducing a second layer, which we call the physical layer model. The physical layer model is coded in Matlab/Simulink and additionally takes into account the effects of channels, different modulations and codings, as well as the specific receiver architecture. The utilised data types of the Simulink model are already constrained to certain bitwidths. Hence, we verify the correct signal processing of the received signals on this layer. Either bitstreams from the link layer model or generated testbench data can serve as an input to this simulation layer. Additionally captured receive signals can be

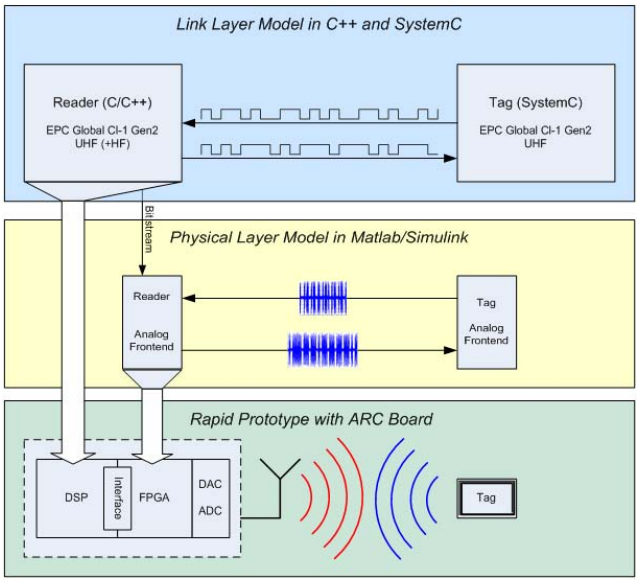


Figure 1: Design flow of an RFID prototyping system.

input as the receive data stream.

Finally, the implementation on the rapid prototyping hardware establishes the third layer of the proposed design flow. This allows for real time measurements of both protocol- and signal processing. Furthermore, verification of the system functionality is supported by capturing receive sequences of samples from the tag after distinct units in the receiver. At a certain trigger event, these samples are captured into the available memory of the receiver and are transferred to the PC by usage of a JTAG interface. These captured samples comply with example receive sequences and can then be reimported into the physical layer model as a simulation input. The signal processing algorithms of the receiver architecture can then be tuned on this higher level of abstraction which speeds up the design and verification time. As soon as the physical layer model delivers the expected results, we can step further down to the implementation on the rapid prototyping board again, for verifying the adaptations with measurements.

In order to speed up the entire design process on the one hand, and to ensure consistency of the simulation models and the implementation on the board on the other hand, we highly automated the code generation process for the rapid prototyping board. By only adjusting some global define statements in the link layer model `C/C++` code we directly reuse this code on the DSP on the rapid prototyping board. Furthermore, certain functions of the physical layer model are transformed to VHDL and embedded into the design, which is then directly synthesised to the FPGA (Figure 1). This can be achieved by means of two different tools, namely the Xilinx System Generator or the Matlab HDL Coder toolbox.

By using the described design flow, any current implementation can be tuned and modified very rapidly. The implemented architecture is examined on different levels of abstraction, allowing for a fast verification and the detection of possible defects in an early design stage. The interconnected layers of the design flow allow the replacement

of certain modules on a very high layer of abstraction and track its consequences down to the final implementation on the rapid prototyping board. For example filters or demodulators of the signal processing parts can be generated using the high level tools provided by Matlab/Simulink and fully automatically implemented on the board, exhibiting the possibility of verification of its effects on all the available layers in between. Similarly the link layer simulation can easily be adapted to other standards or further parameterisation within a standard, or even enriched with new or different features like various anticollision protocols. This can thereupon directly converted to the DSP code running on the prototyping board. Section 4 shows an example implementation of an HF RFID reader in detail, and section 5 proves the concept with some measurement results.

3. TESTBED SETUP

As a platform for our rapid prototyping system we use a prototyping board from Austrian Research Centers. Figure 2 depicts the basic structure of the prototyping board. The main reconfigurable components are a fixed-point DSP from Texas Instruments (TMS320C6416) and a Xilinx Virtex II FPGA. Additionally, it exhibits two digital to analogue converters (16 bits) and two analogue to digital converters (14 bits). The DSP is clocked with 600 MHz while the FPGA, DACs, and ADCs are sourced by a single clock of 40 MHz. An ethernet interface connects the DSP to a PC thus allowing for an external communication to an application running on a PC.

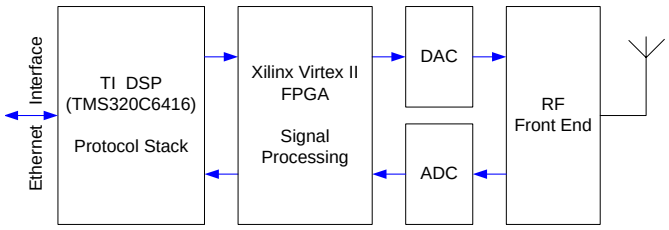


Figure 2: Block diagram of the rapid prototyping board.

As already mentioned, the RF frontend is exchangeable and either supports the 13.56 MHz domain (HF) or the 868 MHz domain (UHF). It exhibits a carrier suppression module and the option to use an analogue envelope demodulator. The 13.56 MHz carrier is directly synthesised in the FPGA and the modulated transmit sequence is output to the DAC. The ADC input signal is a bandpass signal centered at 13.56 MHz as well. In case of an UHF RFID implementation the 13.56 MHz signal is additionally upconverted to 868 MHz using a heterodyne conversion, and the 868 MHz receive signal is downconverted to 13.56 MHz using the same principle. A further extension of the frontend to 2.4 GHz is foreseen.

The FPGA design is decomposed into two parts. One part consists of a fixed, handcoded framework that controls all its external interfaces and provides a wrapper module for embedding the RFID design. The second part is established by the automatically generated code from the physical layer model. This part is embedded within the wrapper part thus providing the flexibility within the RFID reader design.

Note that not the entire RFID reader design is transformed from the physical layer model, but only certain functionality with predominant data flow (filters, integrators, up- and downconverters, modulators etc.) rather than control flow oriented functionality. All state machines and control flow oriented parts of the design are directly coded in VHDL, as this is better suited for modeling these tasks than Simulink. However, all these blocks are implemented very flexibly, ensuring that the design can be adapted to different standards and augmented to additional functionality very easily. All the common parameters of many different RFID standards are either accessible for adaption via VHDL packages or registers. The parameters defined in the packages are configurable at synthesis time while the registers can be accessed by software during runtime. Hence, we move all parameters that are fixed for a specific implementation to the packages (e.g. link timings) and provide all modifiable parameters (e.g. modulation depths, modulation formats, etc.) of a certain standard accessible via registers.

Similarly we embed the DSP RFID reader software obtained from the link layer model into a fixed DSP software framework. This software framework provides interfaces to the surrounding components of the DSP, like the FPGA, external SRAM, ethernet interface, flash etc. and handles the board initialisation processes. The RFID reader DSP software differs from the link layer software only in these interfaces, that can be adapted using some global defines. While the link layer model communicates with the `SystemC` testbench model of the tag the DSP software needs to connect to the various interfaces.

In the following section we will present an example realisation of an HF reader following the draft of the EPC global HF version 2 standard [4].

4. IMPLEMENTATION OF AN EPC GLOBAL HF READER

An example architecture of an RFID HF reader is shown in Figure 3. The DSP software processes the protocols and generates the transmit bit streams. The FPGA transmit path (TX) encodes these bits, switches back to continuous carrier transmission after the entire message is encoded, modulates and shapes the sequence and finally upconverts it to 13.56 MHz. This 13.56 MHz signal is directly forwarded to the DAC and applied to the transmit antenna after the RF frontend. In receive direction (RX) the carrier of the signal is first suppressed at the frontend, then the RX signal is sampled at the ADC. In the FPGA RX path the signal is first bandpass filtered, then envelope demodulated, integrated, and finally the digital levels are decided in the slicer. Furthermore, a synchronisation and control unit and a symbol decoder are used to extract the single bits out of this signal, which are finally forwarded to the DSP. In the following further details of the various units are provided.

4.1 Interfaces

As the interface between DSP and FPGA is operated at 100 MHz, while the FPGA only runs at 40 MHz a data rate conversion is required. In both directions dual clocked FIFOs are used to achieve this task. Furthermore, several registers that are accessible from the DSP are used to exchange control information (modulation parameters, message lengths etc.). In transmit direction the DSP writes the control parameters and the entire message to the reg-

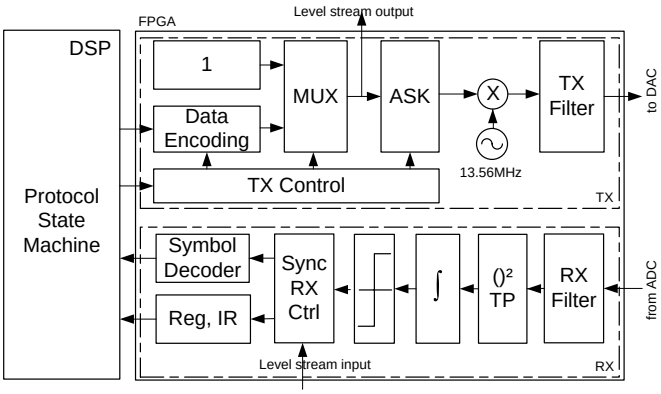


Figure 3: Structure of the FPGA transmit and receive paths.

isters and the FIFO in the FPGA and sets a certain flag for signaling that all required information is available and transmission is to start. In receive direction the FPGA issues an interrupt when 16 bits are ready to be decoded and validated by the DSP software. In order to signal the end of the transmission, padded zeros are transmitted to the DSP.

Within the FPGA wrapper module the DAC and ADC interfaces are very simple 16 and 14 bit synchronous interfaces respectively, that are accessed with every clock cycle from the FPGA.

4.2 Transmitter

In the TX path the transmit signal is synthesised. Most RFID standards use pulse interval encoding (PIE). Each transmitted command sequence is framed with a preamble and an end of frame signalisation. This sequence is generated in the block that we call data encoding by a finite state machine out of the input bits and the corresponding register information. All sequences are multiplexed with a continuous carrier that is needed to supply the tags with power. We call the transmit sequence at this point the *level stream*: it consists of two signaling levels with its appropriate timings. Most RFID standards, as well as the implemented draft of the EPC global HF, use amplitude shift keying as a modulation format. Finally, the pulses of the signals are shaped in the transmit filter and the signal is upconverted to 13.56 MHz and forwarded to the digital to analogue converter. All of the components of the transmitter except of the interfaces and the FSM are converted from the physical layer model.

4.3 Receiver

In order to reduce the noise, the sampled signal is first bandpass filtered in the receiver. The bandwidth is set to let the third harmonic of both sidebands of the data signal pass. After that the signal is envelope demodulated using a square operation and a low pass filter. An integrator averages the samples over the time of one half data rate period. All of these blocks are easily reconfigurable in the physical layer simulation model, using our previously described automated design flow, and automatically mapped to the FPGA implementation. The remaining units of the receiver will be discussed separately in more detail.

4.4 Slicer

The slicer has to cope with very different levels of receive signal power. We implemented two modes of operation in order to tackle that problem: the decision threshold can either be set from the software running on the DSP via a register, or an automatic decision threshold detection method can be used (Figure 4). In the second case the receive signal level of the idle channel is measured (*idle_level*). A certain idle time of the channel is ensured in all RFID standards between the interrogator request and the tag answer (time a in Figure 4). From this observed signal level (*idle_level*) we compute a minimum threshold (*min_threshold*), e.g $1.25 * idle\_level$. While expecting the tag answer (time interval b in Figure 4) the maximum receive value within one data rate period (*max_level*) is determined, and the value $(max\_level + idle\_level)/2$ is computed. If this value is greater than the minimum threshold, it is set as the new threshold, otherwise the minimum threshold is applied. Setting the threshold always to a greater value as the minimum value guarantees that the threshold is not set too low and noise corrupts the decisions. Additionally, a hysteresis is implemented to ensure a smooth output of the slicer.

The signal shown in Figure 4 was derived by sampling the input of the slicer in the FPGA. Note the strong DC offset due to the envelope demodulation of the continuous carrier.

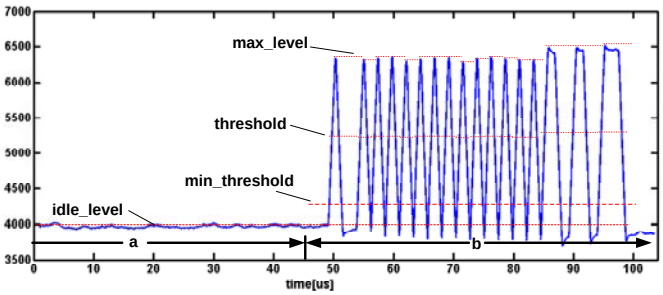


Figure 4: Adaptive setting of decision threshold.

4.5 Synchronisation Unit

Finally, a synchronisation unit corrects the duty cycles of the slicer output signal. The time that the input signal holds a level is measured and the closest multiple of data rate periods is chosen and thereby the receive bits are extracted. By means of this mechanism we can guarantee to synchronise data correctly if the receive data rate offset is less than 3% (Manchester M3 modulation selected) and up to 20% (FM0 modulation selected) in the worst case.

4.6 Reader Interface Application

In order to control and interface the rapid prototyping system, we developed a PC application that connects to the rapid prototyping board via ethernet. A screenshot of this program is depicted in Figure 5. Several system parameters, like protocol selection, modulation and encoding parameters, data rates, timings etc. as well as certain application scenarios, can be applied. Moreover this application serves as a logging client, recording the number of inventoried tags, the electronic product code, the memory read, the success of commands etc.

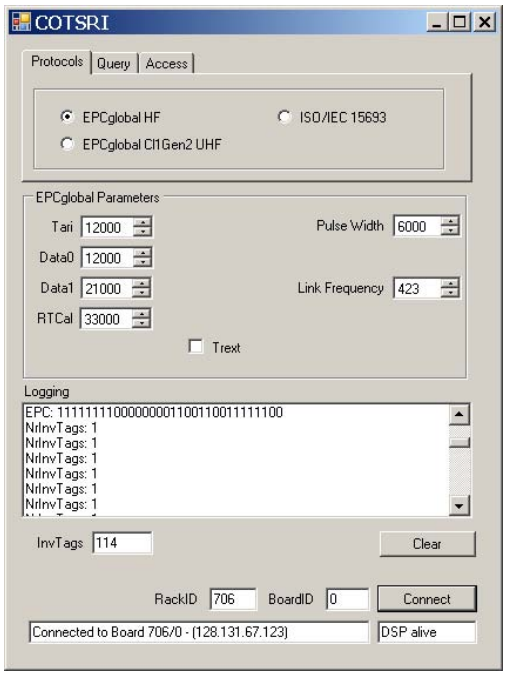


Figure 5: Interface application of the RFID reader.

4.7 Outlook

For the UHF frequency domain the receiver architecture is expected to be further tuned by replacing the envelope demodulator with an IQ demodulator. Additionally, the focus for future work is on developing a framework that supports the use of multiple transmit and receive antennas to enable MIMO RFID systems. Thereby, we expect to significantly improve data throughput and read distances.

5. MEASUREMENTS

5.1 Communication according to the EPC global standard

To demonstrate our proposed rapid prototyping concept we show some measurement results of the implementation following the draft of the EPC global HF RFID standard, presented in the previous section. Figure 6 shows a transmit sequence (query command) interrogating the tag. The

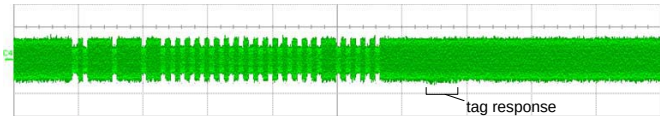


Figure 6: Transmit sequence and tag response at the air interface.

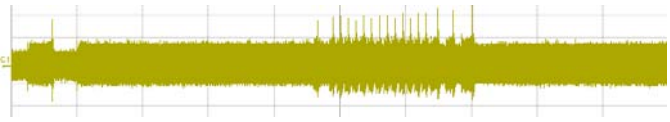
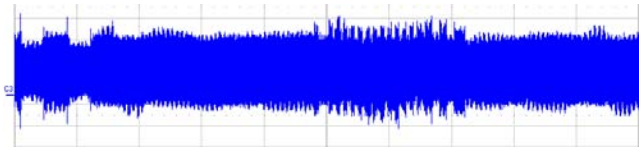


Figure 7: Receive sequence at the ADC.

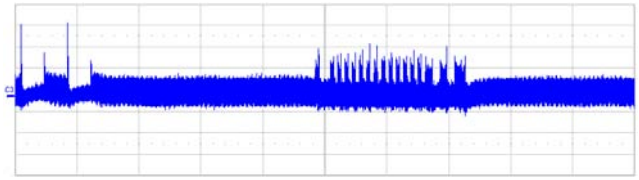
signal was measured using a sense coil at the air interface. The response of the tag is hardly visible due to the much stronger carrier signal sent by the reader in order to supply the tag with power.

Figure 7 shows the tag response of Figure 6 with zoom, after the carrier suppression in the RF frontend, at the input of the ADC.

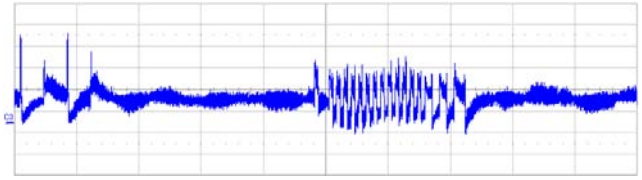
Figure 8 shows the receive sequence after the single units in the FPGA receiver (compare with figure 3). The signals have been multiplexed to the output of the second DAC on the board and measured with the oscilloscope. Note that



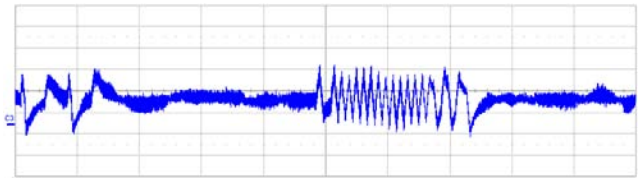
(a) RX signal after RX filter.



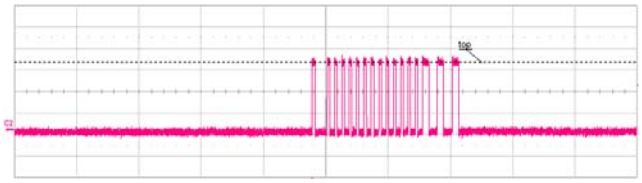
(b) RX signal after square operation.



(c) RX signal after LP filter.



(d) RX signal after integrator.



(e) RX signal at slicer output.

Figure 8: Signals after certain units of the receiver.

the analogue part of the rapid prototyping board provides a band pass filter after the DAC. Hence the DC components are suppressed.

Figure 8(a) shows the receive signal after the bandpass receive filter, Figure 8(b) shows the output of the square operation, Figure 8(c) shows the signal at the output of the lowpass filter of the envelope demodulator, while the signals in Figure 8(d) and Figure 8(e) denote the outputs of the

integrator and the slicer (with adaptive decision threshold applied), respectively.

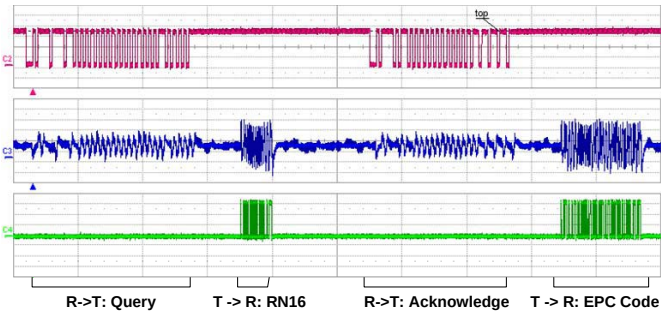


Figure 9: Readout of an EPC code.

Finally, the communication of an entire inventory round is shown in Figure 9. The first line depicts the digital *level stream* of the reader to tag (R→T) sequences, the second line denotes the receive signal after the integrator and the last line shows the decided bits of the tag to reader (T→R) sequences. The communication starts with an inventory command. Afterwards the tag responds with a 16 bit random number (RN16). The interrogator acknowledges this random number, and hereupon the tag returns its electronic product code (EPC).

5.2 Communication according to the ISO 15693 standard

Additionally the design has been adapted to comply with the ISO 15693 standard [7]. The measurement in Figure 10 demonstrates the communication between reader and tag on the example of an inventory command. The response of the tag is manchester encoded. The signal in the upper line was

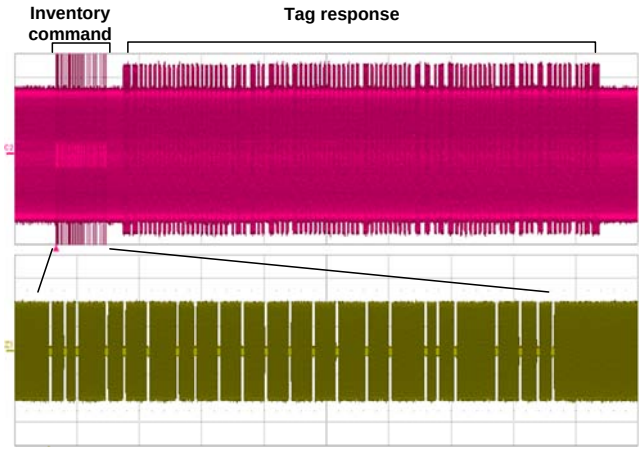


Figure 10: Communication according to ISO 15693 standard.

measured after the carrier suppression in the RF frontend. The signal in the second line is a zoom of the interrogating transmit sequence, measured at the air interface.

6. CONCLUSION

In this work we present a testbed for RFID designs. It supports two frequency domains, namely the HF (13.56MHz) and the UHF (868MHz) domain, and is very flexible for supporting several standards within these two frequency bands. By means of our automated design flow, both existing standards as well as new ideas and proposals for future RFID systems, can be evaluated and explored within minimal time and effort on several different layers of abstraction. This assists in obtaining a feeling on how to implement these new ideas most efficiently. In this paper we introduce our testbed setup and present an example implementation of an HF RFID reader in detail. Furthermore, some measurement results regarding this implementation are shown and prove our rapid prototyping concept.

Acknowledgment

We would especially like to thank our industrial partner Infineon Technologies for enabling this work and supporting us with many advices and helpful discussions. They also provided a `SystemC` testbench for our simulation model. Moreover, we would like to thank Austrian Research Centers who supported our work with a *SmartSim* rapid prototyping board.

7. REFERENCES

[1] C. Angerer, B. Knerr, M. Holzer, A. Adalan, and M.Rupp. Flexible Simulation and Prototyping for RFID Designs. In *Procceedings of the First International EURASIP Workshop on RFID Technology*, September 2007.

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

[3] V. Derbek, C. Steger, and R. Weiss. A Model-Based Methodology for Real-Time Verification and Optimization of UHF RFID Systems. In *Proceedings of the First International EURASIP Workshop on RFID Technology, p.47 - 50*, September 2007.

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

[5] K. Finkenzeller. *RFID Handbook*. John Wiley and Sons LTD, 2004. second edition, ISBN 0-470-84402-7.

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

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

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