Baseband-Processor for a Passive UHF RFID Transponder

José A. Rodríguez-Rodríguez*, Manuel Delgado-Restituto and Ángel Rodríguez-Vázquez

Institute of Microelectronics of Seville (IMSE-CNM-CSIC) - University of Seville
Parque Tecnológico de la Cartuja, Avda. Américo Vespucio s/n, 41092-Seville, SPAIN

Abstract – This paper describes the design of a digital processor targeting the Class-1 Generation-2 EPC Protocol for UHF RFID transponders, and proposes different techniques for reducing its power consumption. The processor has been implemented in a 0.35µm CMOS technology process using automatic tools for both the logic synthesis and layout. Post-layout simulations confirm the fully functionality of the prototype and predict a worst-case power consumption of only 2.9µA at 1.2V supply.

I. INTRODUCTION

Nowadays, Radio Frequency IDentification (RFID) devices find many applications in fields such as manufacturing, product distribution and sales, automotive, transportation and customer services, and building access control [1, 2]. RFID communications use a master-slave configuration formed by a reader and a set of transponders (tags, in short) [3, 4]. Each tag has a unique identification number stored in a non-volatile memory, which is addressed by the reader to establish the communication link. Upon the commands sent by the reader, the selected tag delivers the requested information. In the so-called sensory tags, such information might not only consist on identification data but also contain environmental readouts (e.g., temperature, pressure, optical or chemical variables) obtained from an embedded sensor interface. The ability of sensory tags to monitor, record and even react to ambient conditions are expected to promote a new world of applications for RFIDs.

Tags are classified into active or passive depending on how energy is supplied to the device. Passive tags have no internal power source available, as in the case of active transponders, but they are remotely biased by the reader by means of an on-chip RF-to-DC conversion stage [5, 6]. Because of the scarce supplying conditions, power consumption minimization is a priority for passive tags.

This paper focuses on the design of the digital section a passive UHF RFID sensory tag for half-duplex communications in the 860-960 MHz range. The baseband processor implements the EPC™ Class-1 Generation-2 (Gen2) protocol [7], which is briefly reviewed in Section II. Given the complexity of the protocol, the power consumption of the baseband processor is comparable to that of the analog section of the tag [4]. Hence, it is necessary to apply low-power design strategies at its implementation. These power saving techniques and the architecture of the baseband processor are presented in Section III. Next, Section IV shows the layout of the processor and presents extracted simulations which confirm the system functionality and predict a worst-case power consumption of only 2.9µA at 1.2V supply. Finally, Section V concludes the paper.

II. EPC GEN 2 REVIEW

The EPC™ Class-1 Generation-2 (Gen2) protocol [7] is a highly flexible protocol which allows a wide variety of air interface and encoding possibilities:

- Reader to tag communications (forward link) can be done with three types of ASK modulation using Pulse-Interval Encoding (PIE) format.
- Tags communicate information to the reader (backward link) by backstretch modulating the amplitude and/or phase of the RF carrier using either I-phase space (FM0) or Miller-Modulated Subchaser (MMSC) encoding formats.
- The standard supports different data rates both at the forward (from 26.7 to 128 kbps) and backward (from 5 to 640 kbps) links.

Forward link communications is always preceded by a preamble. Fig.1(a) shows an example, corresponding to a *Query* instruction. The preamble comprises a fixed-length start *delimiter*, a *data*-0 symbol, a reader-to-tag calibration symbol (*RTcal*), and tag-to-reader calibration symbol (*TRcal*). These two latter symbols are used to define the forward and backward data rates, respectively. In other instructions, only the *RTcal* symbol is transmitted. The duration of *RTcal* is equal to the length of a *data*-0 symbol plus the length

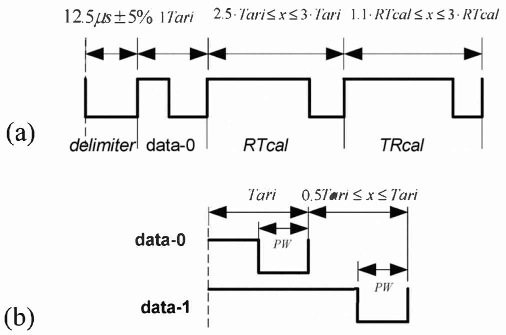

Fig. 1: (a) Preamble used in reader to tag signalling. (b) Data encoding in PIE format.

* Contact author: rodriguez@imse-cnm.csic.es; phone +34954466666; http://www.imse-cnm.csic.es

978-1-4244-6878-2/10/$26.00 ©2010 IEEE

of a *data*-1 symbol. They are both represented in Fig.1(b) and define the PIE encoding used for reader-to-tag signalling. The duration of a logical '0', called *Tari*, amounts 6.25 to 25μs. The length of the logical '1' can range between 1.5**Tari* to 2**Tari*. Rise time, fall time and pulse width (*PW*) are identical for *data*-0 and *data*-1 symbols and their valid ranges are defined in the protocol.

Signal decoding at the tag is simply accomplished by a time-to-digital conversion using a master clock signal. The number of clock cycles comprised during the symbol *RTCal* are computed and divided by 2 to define a *pivot*. If a symbol has less number of cycles than *pivot* then it is a *data*-0 symbol, otherwise is a *data*-1 symbol.

Data rates of the backward link are obtained by dividing the master clock frequency by integer values. The number of clock cycles per bit in the backward link, N_{BLF}, is, therefore, computed as

$$N_{BLF} = round\left\{\frac{int(TRcal \cdot f_m)}{DR}\right\} \quad (1)$$

where the divide ratio, DR, specified in the *Query* command, can be 8 or 64/3; f_m is the master clock frequency; and $int(\cdot)$ is an operator whose output may take on two possible integer values which are obtained by either rounding up or down its argument. In (1), such integer number depends on the a priori unknown phase relation between the local oscillator and the demodulated RF signal. The Gen 2 protocol defines tolerance margins for the different backward frequencies which can be synthesized from (1). Taking into account these tolerances and the timing resolution requirements of the forward link, as well as, the need for reducing the dynamic power consumption of the processor, it can be found that the minimum master clock frequency imposed by Gen2 requirements is 1.92 MHz [8, 9].

Besides the already mentioned *Query* command, the EPC Gen 2 protocol defined many others commands/actions both for the forward and backward links. Their description is beyond the scope of this paper, nonetheless, it is worth mentioning that they all have been fully implemented in the proposed baseband processor.

III. BASEBAND PROCESSOR

III.1. Low-Power Design Strategies

Besides selecting the lowest clock frequency able to comply with the Gen2 specifications, two hardware-level techniques have been considered for power saving. They are referred to as Clock-Gating (CG) and Clock-Management (CM) [6].

A. Clock Gating

Power consumption can be reduced by activating the minimum number of blocks [10]. For instance, if the system has not completely interpreted a received command, there is no need to activate those blocks required for backward link communications. Clock gating builds on this idea, i.e., disabling blocks when they are dispensable [11]. This can be simply done by and-combining the clock or trigger pulses which activate the block with an *enable* flag, in accordance with the command that the processor is currently handling.

B. Clock Managing

Depending on the processor state, some of their blocks can operate at frequencies below the master clock. As will be shown next, depending on the command that is being handled, only four blocks need to run at full speed: *Pie_Decoder*, *FSM_Core*, *FSM_Tx* and *Tx*. The rest of the blocks can be clocked at a fraction of the master frequency to save power. The lower limit of the dedicated clock frequencies is determined by the time interval between two rising edges of the demodulated input signal.

III.2. Architecture

Fig.2 shows the block diagram of the processor. It is a system controlled by a *Timing Unit* which generates the clock and trigger signals required for the decoding, encoding and processing operations. This block is responsible for the implementation of the low-power design strategies described above.

In the decoding section, a falling-edge triggered flip-flop is used to synchronize the demodulated signal coming from the analog front-end of the RFID, *data_dem,* to the master clock signal. Reader to tag communications use Pulse-Interval Encoding (PIE) format and, therefore, the resulting digitized forward link, *data_in*, must be converted into binary format. This is accomplished in the *PIE Decoder* block whose output is sequentially stored in a 16-bit *Shift Register* block at a rate defined by the trigger pulses, *en_pulse_shift*. Once the preamble parameters are read, the *PIE Decoder* sets on the *end_prea* flag.

Next, the *Command Decoder* block evaluates the data stored in the register to identify which instruction has been sent by the reader. This is a simple task because commands in the EPC Gen 2 protocol include a field which unequivocally addresses the instruction received by the tag. Operation of the *Command Decoder* block is controlled by the trigger signal *en_pulse_cmd*, a delayed version of *en_pulse_shift,* to allow a more uniform distribution of current consumption over time. When the command received is identified, the *Command Decoder* sets on the *end_cmd* flag and codifies the instruction in a 4-bit vector, *cmd_ID.*

Besides filling *Shift Register*, the output of the *PIE Decoder* block is also transferred to a Cyclic Redundancy Check (CRC) unit for transmission error detection. The EPC Gen 2 protocol uses two types of CRC;

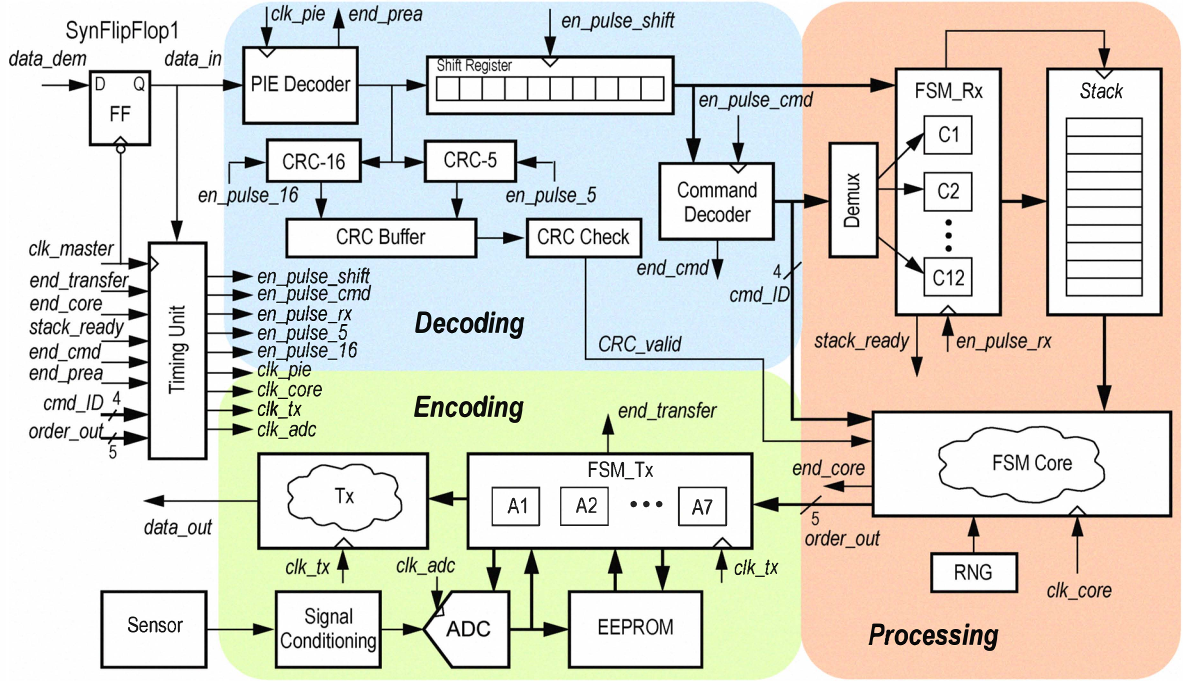

Fig. 2: Architecture of the proposed baseband-processor

CRC-5 and CRC-16. The former is used by *Query* commands, whereas the latter is used by *Select* and *Access* commands. *Inventory* commands are unprotected. Once the *Command Decoder* identifies the type of instruction that it is being received, it disables the useless CRC block(s) for power saving. The results of the CRC computations are stored in buffers and these values are used by a *Check CRC* block to assess their validity. The CRC blocks, CRC-5 or CRC-16 are enabled each rising edge of *data_in*, employing pulses *en_pulse_5* and *en_pulse_16*, respectively.

Once the command is identified, the *Command Decoder* passes *cmd_ID* to the *FSM Rx* block, inside the processing section. This block is controlled by the trigger pulses *en_pulse_rx,* a delayed version of signal *en_pulse_cmd,* and it is formed by a set of Finite State Machines (FSM), one per Gen2 command. Only that FSM addressed by the *Command Decoder* is active; the others are disabled. The active FSM sequentially stores the command parameters in the registers of the *Stack* block. Only that register which is being addressed by the *FSM Rx* block is active, the others remain off. The *FSM Rx* block notifies the *Timing Unit* by means of the *stack_ready* flag when the reception is finished and, then, the trigger pulses of the CRC modules and the *PIE Decoder* block are disabled.

The *FSM Core* block decides the tag's state, performs the required state transitions, read the parameters stored in the *Stack* by the *FSM Rx,* and triggers the *FSM Tx* block according to the command that has been received. When operations at *FSM Core* are concluded a non-zero 5-bit vector, *order_out*, is transmitted to the *FSM Tx* block and *FSM Core* is disabled by the *Timing Unit.*

At the encoding section, the *FSM Tx* block performs the actions requested by the reader such as write/read the EEPROM, gather parameters or information to send, and control the transmitter for the backward link. The actions are grouped into 7 types, according to the data format to be transmmited or the operation to be executed by the *Encoding* section of the baseband-processor. There is one FSM for each possible action type, and, as before, only one FSM is enabled at a time. The *Tx* block encodes the data in FM0 or MMSC at the bit rate requested by the reader. When the requested action has been finished or the transmission is completed, the *FSM Tx* block sets on the flag *end_transfer*, *FSM Tx* and *Tx* (if required) are disabled, and *FSM Core* is activated again to check if the processor must change the state or remain in the same configuration. After this evaluation, *FSM Core* sets on the flag *end_core.*

Other important blocks of the RFID tag are a Random Number Generator (RNG), for securing communications, and a mixed-signal circuitry for generic sensor signal acquisition. This latter consists of a *Signal Conditioning* block and a Successive Approximation Register (SAR) ADC. The former is used to adapt the sensory information to the signal range of the ADC. The ADC is clocked by the *Timing Unit* block through the signal *clk_adc*, and its operation is controlled by the *FSM Tx* block. Additionally, a flip-flop, not shown in Fig.2, has been included to synchronize a general baseband reset signal to the master clock.

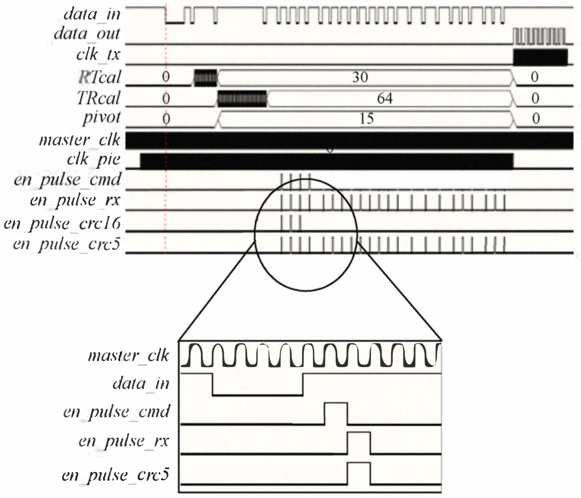

Fig. 3: Experimental Results.

IV. SIMULATION AND EXPERIMENTAL RESULTS

Prior to silicon integration, the baseband processor was synthesized on a Xilinx XC3S1000 FPGA for debugging purposes. After an exhaustive test, the VHDL code was found fully functional. As an illustration, Fig.3 shows the experimental verification of a *Query* command transmission at 128 kbps, and a response transmission at 640 kbps, both the maximum frequency allowed in the forward and backward link, respectively. The scopes were captured using the logic analyser Agilent 16902B. The figure shows that the processor successfully calculates the timing constants implicated in the RFID communication.

Fig.3 also illustrates the clock gating technique employed in the processor. The *PIE Decoder* is only enabled when the tag is receiving data and it is disabled otherwise. The *FSM Tx* works at maximum frequency in this particular example (a *Query* command) but it is only enabled during data transmission. The *CRC-16* and *CRC-5* modules are enabled when the tag is decoding the information sent by the reader and, when the processor detects that the received command do not use *CRC-16*, corresponding block is disabled. Finally, it can be seen in the inset that the different clocks are delayed among themselves to lower the peaks of dynamic power.

After FPGA validation, a silicon prototype has been designed in a 0.35µm CMOS technology. Its layout is shown in Fig.4, where a bank of supply capacitors and an ADC for sensor signal acquisition can be identified together with the processor and the EEPROM. The chip occupies 7mm^2 including pads. The ADC is an ultra low-power 10-bit SAR converter, intended for slow-varying signals. It is clocked by a signal 128 times slower than the master clock (the ADC internally divides *clk_adc* by 2), requires 12 clock cycles to complete a conversion and only consumes 150nW.

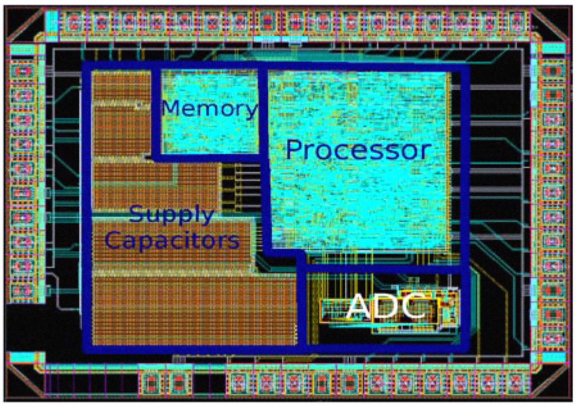

Fig. 4: Layout of the RFID baseband-processor.

Fig.5 shows an exemplary post-layout simulation of the prototype, in this case, illustrating the ADC operation when the tag receives a *Write* command. After an initial reset, the ADC makes five consecutive conversions which are transferred to the *FSM Tx* block at every *adc_data_ready* pulse. Once the last conversion is performed, the ADC enters in powerdown mode and the *FSM Tx* block averages the samples and stores the result in the EEPROM.

The power consumption of the processor was estimated using digital and analog models for the baseband-processor and the *ADC,* respectively, assuming maximum bit-rates for the forward and backward links. Fig.6 illustrates the dissipation per block during a communication flux which involves five consecutive commands, including tag selection and memory reading. In total, the processor consumes less than 2.9µA, assuming worst-case timing conditions. Note that the *Timing Unit*, *FSM Tx, TX* and *PIE Decoder* blocks are the most power-demanding elements of the processor because they are clocked at the master frequency. Anyhow, the power consumption of the latter two blocks has been considerably reduced thanks to the applied clock gating techniques.

Table I compares the achieved performance with other implementations in the literature. As can be seen, [6] achieves lower power consumption, however, it is clocked at a master frequency of 480kHz and it does not support the complete EPC Gen2 standard. References [10], [12] and fully comply with the standard but at a higher power consumption than in the proposed processor. Finally, [13] adds an AES cryptographic

TABLE I. Comparison with previously published designs.

Reference	Master Frequency	Current Consumption	CMOS Technology
[6]	480 kHz	1.47 µA	0.35 µm
[10]	2.56 MHz	6.40 µA	0.18 µm
[12]	1.28 MHz	5.1 µA	0.35 µm
[13]	2MHz	4.17 µA	0.18 µm
This work	1.92 MHz	2.9 µA	0.35 µm

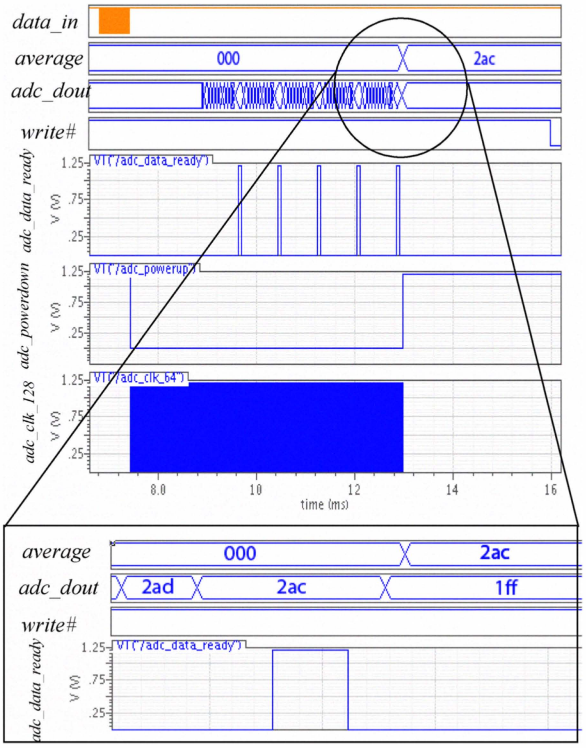

Fig. 5: Communication between the Processor and the ADC

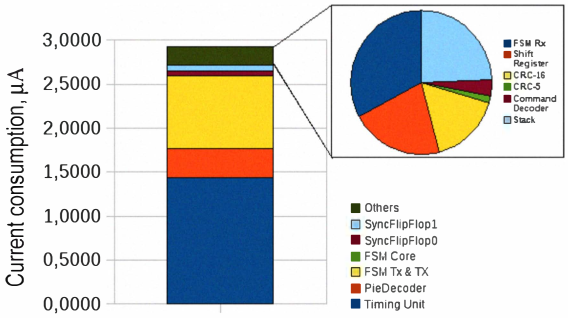

Fig. 6: Current consumption per block of the baseband-processor

module to the tag but only supports the mandatory commands defined by the standard and uses a CMOS technology with a smaller feature size.

V. CONCLUSIONS

In this work, a baseband processor for RFID applications targeting the EPC Gen2 protocol has been designed and implemented in a 0.35µm CMOS process. The design was validated in a Xilinx Spartan3 FPGA, and the post-layout simulations show that the processor consumes less than 2.9µA operating at maximum frequency allowed for the backward and forward link.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Education & Science under grants TEC2006-03022 and TEC2009-08447, and the Junta de Andalucía under grant TIC-02818.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. John Wiley & Sons Ltd, Chichester (UK), 2003.

[2] D. Dolkin, *The RF in RFID Passive-UHF - RFID in Practice*. Newnes, Amsterdam (The Netherlands), 2008.

[3] C. Ma, X. Wu, C. Zhang, and Z. Wang, "A low-power RF front-end of passive UHF RFID transponders." *IEEE Asia Pacific Conf. Circuits Systems* (APCCAS 2008), pp.73-76, Dec. 2008.

[4] J.-P. Curty, M. Declercq, C. Dehollain, and N. Joehl. *Design and Optimization of Passive UHF RFID Systems*. Springer, New York, 2007.

[5] U. Karthaus, and M. Fischer, "Fully integrated passive UHF RFID transponder IC with 16.7-µW minimum RF input power." *IEEE J. Solid-State Circ.*, vol.38, pp. 1602-1608, Oct. 2003.

[6] I. Zalbide, J. Vicario, and I. Velez, "Power and energy optimization of the digital core of a Gen2 long range full passive RFID sensor tag." 2008 *Int. Conf. on RFID*, pp.125-133, Apr. 2008.

[7] EPC^{TM} Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz. Version 1.2. May 2008.

[8] Impinj, Gen 2 Tag Clock Rate - What You Need To Know, 2005.

[9] F. Cilek, K. Seemann, G. Holweg, and R. Weigel, "Impact of the Local Oscillator on Baseband Processing in RFID Transponder." *Int. Symp. Signals, Systems and Electronics* (ISSSE 2007), pp.231-234, 2007.

[10] V. Roostaie, V. Najafi, S. Mohammadi, and A. Fotowat-Ahmady, "A low power baseband processor for a dual mode UHF EPC Gen 2 RFID tag." *Int. Conf. on Design and Technology of Integrated Systems in Nanoscale Era* (DTIS 2008), pp.1-5, Mar. 2008.

[11] M. Mueller, A. Wortmann, S. Simon, M. Kugel, and T. Schoenauer, "The impact of clock gating schemes on the power dissipation of synthesizable register files." 2004 *Int. Symp. Circuits and Systems* (ISCAS 2004), vol.2, pp. 609-612, May 2004.

[12] H. Yan, H. Jianyun, L. Qiang, and M. Hao, "Design of low-power baseband-processor for RFID tag." *Int. Symp. on Applications and the Internet* (SAINT 2006), vol.4, pp. 63-66, Jan. 2006.

[13] A. Ricci, M. Grisanti, I. De Munari, and P. Ciampolini, "Design of a 2 µW RFID baseband processor featuring an AES cryptography primitive," *15th IEEE International Conference on Electronics, Circuits and Systems* (ICECS 2008), pp. 376-379, Sep. 2008.