International Journal of Engineering & Technology, 7 (2.28) (2018) 276-281

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Behavioral Simulation of ISO 18000-6 Type-C Class 1 Gen2 Protocol for RFID UHF Transponder and its Application as Anti-collision Protocol in Interference Case

Hadjer Saadi[1]*, Ahmed Rnnane[1], Rachida Touhami[1], Mustapha C.E. Yagoub[2]

[1] LNS, FEI, USTHB, Bab Ezzouar Algiers, Algeria
[2] EECS, University of Ottawa, Ottawa ON, Canada
**Corresponding author E-mail: hadjer_saadi@yahoo.fr*

Abstract

In RFID systems, the Transponder Protocol usually uses the standard ISO 18000-6 Type-C Class 1 Generation 2, originally developed to communicate with the reader. Since a typical RFID system could be used in a myriad of tasks from product identification to environmental sensing, behavioral software functionality and hardware cost constraints are extremely constricted, principally due to their standard's requirements. Thus, in this paper, an advanced behavioral simulation of the Tag ID layer of ISO 18000-6 Type-C protocol is proposed with all its states, commands and functionality, a crucial step toward effective design and test. The approach was then successfully applied to collision issues in interference case.

Keywords: *ISO 18000-6 Type-C, RFID, VHDL, Tag identification-layer, Simulation.*

1. Introduction

In a RFID system (using Identification by Radio-Frequency signals), many sequences of signals can be exchanged between a reader and the tags in its reading area until each tag will be "singulated" by the reader [1], [2], i.e., uniquely identified. This is regulated by standards through specific commands, known as tag identification protocol [2], which purpose is to determine when tags can transmit their signals to the reader as well as what they should transmit.

Currently, the ISO 18000-6 Type-C RFID Protocol, famous again as EPC Class-1 Generation-2, can be used in various applications from simple product identification to more complex tasks such as environmental sensing. Below the Auto-ID Center, completed by EPCglobal [3], [4], the RFID system originally developed to communicate with the reader in the UHF band (860 - 960 MHz), used a GEN2 accommodating RFID transponder that is composed of main constituents: Antenna, RF front end (transceiver), Physical layer (that constitute principally encoder and decoder) and the Tag Identification layer [4]. In these two layers of operation, the energy of the passive RFID tag is taken from the radiofrequency (RF) signal from the reader, demodulates and decodes the commands and then backscatters the response back to the reader.

Numerous parameters can define/affect the operation of an RFID system, such as the reading distance, the characteristics of the tags and the reader … [5]. Also, the baseband system plays the role of a central controller which controls the reader-tags communication and avoids the persistence of possible issues such as collisions during the tag identification process. It is in charge of encoding the PIE data, decoding the FM0 or Miller data, managing the memory access, setting the clock generator, monitoring the sending of commands to the tags, and processing information reflected on the label [6].

Different approaches can be retained to design a microcontroller-based RFID reader [7] such as Field Programmable Gate Array (FPGA) [8], [9] or DSP digital signal processor. FPGA technology has many advantages; it is cost-effective, consumes less power, can program all functions, allows adding custom features …. The use of FPGA technology in RFID has already been discussed [10]-[13] and several FPGA-based works have been presented to design the baseband processor of the RFID reader [8], [14]-[16].

On the other side, if architectural exploration needs can be met using simple functional models, an entirely different modeling solution must be found for performance evaluation and optimization needs. This can be achieved with behavioral models. Indeed, a behavioral modeling of the reader and the tag is a suitable solution that allows a complete simulation of the system with sufficient precision and acceptable CPU time. In fact, a behavioral description, which often takes the form of an algorithm, focuses on explaining the behavior of a model, without worrying about the structure itself.

We then focused on the development of an efficient behavioral model for the tag-identification protocol layer, including all the states, commands and their functionality, as a basis for future standard conformity tests. This was principally focused on the operating State Machine Functionality (Finite State Machine, FSM), again other blocs such as the Memory Banks, the Random Number Generator, the Slot Counter, etc. The purpose in this paper is therefore to first, fully define the Tag ID layer architecture according to the specifications of the ISO 18000-6 Type-C Protocol, and then to successfully apply the EPC Class1Gen2 as anti-collision protocol.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The article is ordered as follows. Section II reviews the EPC Class-1 GEN2 protocol, followed by the Basic Emulator Digital Identification Layer Tag architecture in section III. The results are introduced in Section IV while Section V shows the application of the proposed architecture to collision issues, followed by the conclusion.

2. EPC GEN2 Principal

The EPCglobal Class-1 Gen-2 RFID Protocol for Communications is an agreement between reader / tags on how information is exchanged. For passive RFID systems, careful attention needs to be paid to reduce the requests on restricted power and computational capability of the tag. This protocol determines the physical and logical necessities for a passive backscatter, Reader-Talk-First (RTF) RFID system [1].

In the physical layer, the forward link is achieved by ASK modulation using Pulse-Interval Encoding (PIE) while in the backward link, the tags broadcast its ID to the interrogator by backscattering the amplitude and/or phase of the Radio Frequency carrier modulation by means of FM0 or Miller-Modulated Subcarrier (MMSC) encoding scheme. The protocol can include several data rate ranges, i.e., [26.7-128 kbps] for the forward link and [5-640 kbps] for the backward link [18]. Also, the communication link is half-duplex. However, the forward link communication starts every time by a preamble, which comprises a delimiter of fixed-length as well as data-0, tag-to-reader calibration (TRcal) and reader-to-tag calibration (RTcal) symbols. By the mean of these signals we can fix the data rates of forward and backward link. In certain commands, the RTcal signal is lonely communicated.

Fig.1-a illustrates a Query instruction [4], [17], and Fig.1-b defines the PIE used for reader-to-tag signaling. The number of clock cycles during the RTCal signal is calculated and then divided by two to fix a *pivot* threshold. If the number of cycles is less than this threshold, the signal is a data-0 symbol, else, it is a data-1 symbol [18].

In the tag Identification Layer, the reader accomplishes the tag population by three elementary operations.

- Select: used to choose a specific tag group founded on user defined criteria.
- Inventory: a reader initiates an inventory cycle by communicating a Query command in one among four sessions. One or several tags can respond. The interrogator chooses only one tag and demands the PC, UII and CRC from the Tag. The Inventory process includes more than one command and operates in only one session.
- Access: the procedure of reading from or writing on to a tag. Access also consists of several commands [3], [4].

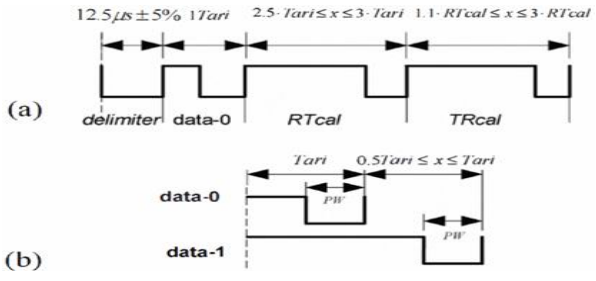


Fig.1. (a) Preamble used in reader to tag signaling. **(b)** Data encoding in PIE format (from [19]).

3. Proposed digital identification layer tag architecture

Fig. 2 illustrates the suggested architecture of the proposed digital Identification Layer tag, which emulates the RFID tag. This design is conform to the EPC C1 G2 UHF RFID protocol. The system of Identification Layer is composed of numerous well-designed modules that manage communication, command computation, and data storage. Once the signal received, the RF block will pass it to the physical layer demodulator and decoder, which extracts the data rate and related timings communication parameters. After decoding, the digitized and stable command sequences bits are sent as inputs to the tag identification layer. Furthermore, the FSM controls the operation of the whole system components of the identification layer, which performs the EPC C1 G2 protocol (e.g., the data exchange with the memory). The FSM uses the CRC decoder for CRC5 and CRC16 computing in order to assure the communication with the reader integrity.

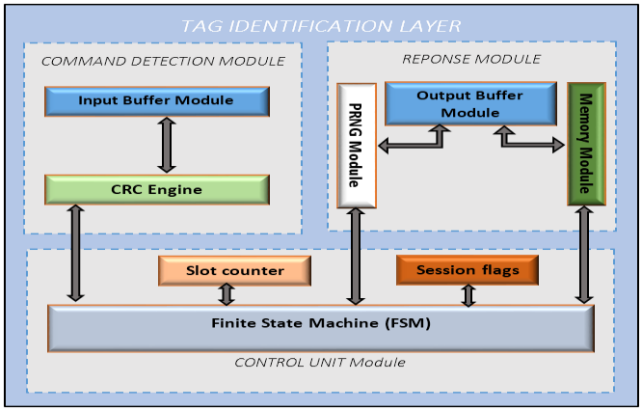


Fig. 2. Basic block diagram Identification Layer

Generally, the number of states of the FSM and the other included features determine the complexity and functionality of the whole tag [3], [19], [20].

The architecture has eight major sub modules classified into three principal sections [21], [22]:

- Command Detection Module: Input Buffer and CRC Engine.
- Control Unit module: Finite State Machine, Slot Counter, and Session Flags.
- Response Module: Pseudo Random number generator (PRNG), Memory Module, and Output Buffer.

The command detection module is in charge of receiving, decoding and validating the command transmitted by the interrogator. This module incorporates the logic blocks namely, the Input Buffer Sub Module (for receiving and collecting the incoming bits from the RF block [3], [4]) and the CRC Engine Sub Module (as parts of the Gen2 are 5-bit and 16-bit cyclic redundancy checks, also named CRC5 and CRC16, in that order). The two circuits were built from a linear feedback shift register (LFSR), and used to safe information against bit errors during transmission [3].

The control unit module is the most important part of the digital logic. It decides what to reply upon receiving a complete data packet from the decoder. This module is in charge to perform the required functions as specified by the functioning of the FSM illustrated in the GEN2 [3], [19], [21], [22]. It contains three main sub modules, (i) Finite State Machine (FSM) to supervise the state transitions of the digital Identification Layer tag architecture, (iii) Session flags to allow more than one reader to independently manage the same population of tags at the same time, and (iii) the slot counter : to copute the slot number at which the Tag will answer. The slot number is calculated using the Q-value sent by the interrogator in the Query command and the random number of 16 bits RN16 generated by the PRNG module as

$$Slot_{number} = (RN16) \bmod (2^{Q}) \quad (1)$$

4. Response module

This module depends on the command sent by the interrogator. Its validation is prepared in the FSM module that backscatters the response from the tag [3], [4].

4.1. PRNG Sub module

The Pseudo Random number generator (PRNG) submodule generates 16-bit random numbers in which a linear feedback shift register (LFSR) is used with clock enable and parallel load. The PRNG source is fed to the transmitter in response to Query, QueryRep, and Req_RN commands as specified in C1Gen2 Protocol.

4.2. Tag memory Sub module

The tag memory submodule can be classified as four distinct banks:

- Bank 00 that contains the RESERVED of the kill and access passwords.
- Bank 01 that contains the Object Identification (OID), a CRC-16, and a Protocol-Control (PC).
- Bank 10 that contains the Tag-Identification (TID), the 8-bits class identifier, the 12-bit task mask-designer, and higher functionality of tag vendor and other specific data.
- Bank 11 that contains the User Memory that allows specific data storage.

Actually, no more than one module can access a memory block simultaneously. To manage this, the signal of *rd-en* and *wr-en* flags can be used to read from or write into the data memory.

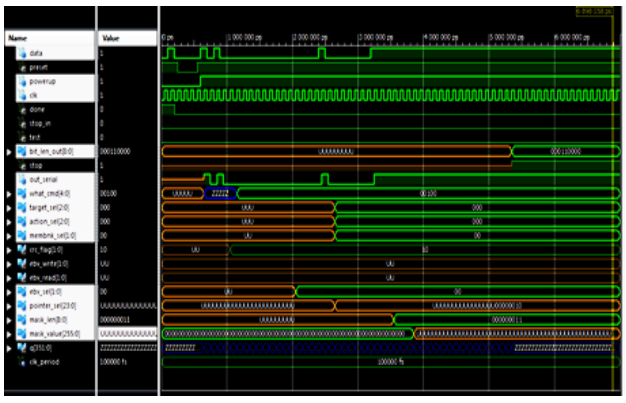
4.3. Output buffer Sub module

This Sub Module cares of the ordering the departing bits so that to send bit by bit at each rising edge of the clock to the physical layer encoder.

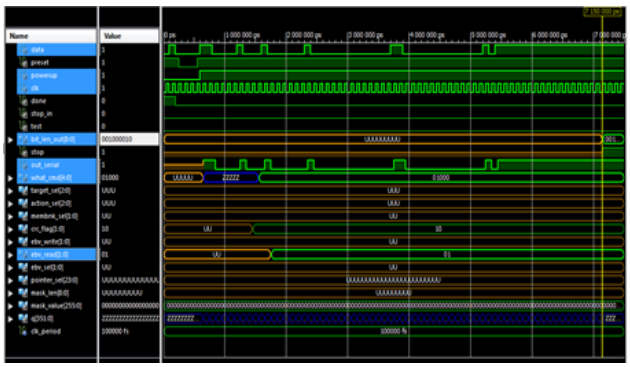
5. Simulation Results

5.1. Command detection Sub module

This module detects any command sequence transmitted by the reader that consults the first sequences bits from the MSB (the code of a command) and extracts the necessary parameters values from its corresponding fields as shown in Figure 3 and summarized in Table 1 [3].



(a)



(b)

Fig.3. Behavioral simulation (a) *SELECT* command (b) *READ* command

5.2. Cyclic-Redundancy Check Sub module

According to the command code bits detected, the CRC engine submodule transmits the totality of the command bits and checks the validity of the received sequences command bits. A CRC_VALID signal is sent to the FSM module when the contents of Q [15:0] registers = 1D0Fh for CRC-16 or Q [4:0] = 00000_2 for CRC-5. As illustration, we present the successful behavioral simulation tests of the CRC-16 module with no error during transmission (Fig. 4-a) and when an error is intentionally created (Fig. 4-b). Thus, the CRC_VALID signal switches to high state for one clock signal period.

5.3. Slot counter Sub module

The slot counter calculates the number of the slot in which the tag informs the FSM submodule that the operation is completed by making a change on the value of its slot (Fig.5).

Table 1. Select command

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: Inventoried (S0) 001: Inventoried (S1) 010: Inventoried (S2) 011: Inventoried (S3) 100: SL 101: RFU 110: RFU 111: RFU	See Table 21	00: RFU 01: UII 10: TID 11: User	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	



(a)



(b)

Fig. 4. Behavioral simulation of CRC-16 engine (a) with no error during transmission (b) with error during transmission.



Fig.5. Behavioral simulation of the Slot Counter Engine

The random number shown in Fig.6 is produced by means of a Linear Feedback Shift Register (LFSR), in which a clock is enabled and a parallel load assigned.

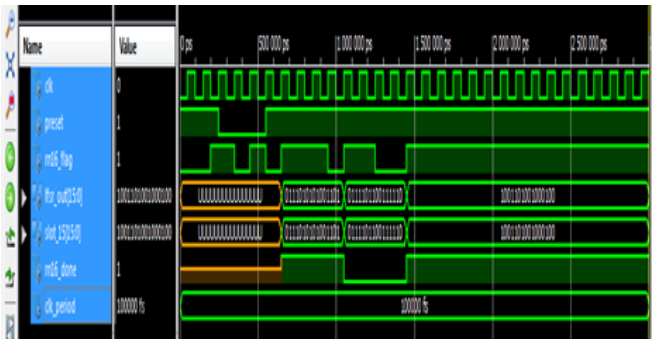


Fig. 6. Behavioral simulation of the PRNG module

6. Application to anti-collision

The EPC GLOBAL Class1 Gen2 is usually used to avoid collisions in RFID systems between tags and reader. It defines the control and operating procedures to setup the physical and logical communication between the tags and the interrogator, including an anti-collision algorithm [3]. In this protocol, the identification process comprises several commands and responses between the interrogator and the tags. The interrogator commands are Select, Query/Query_Rep, and ACK. The label responses contain the unique RN16 key as well as the EPC data. The corresponding workflow can be defined as follows:

- The reader sends the 22-bit Query command to the passive tags in its interrogation area. Each command has a Q field and is also utilized to indicate the size of the frame.
- Each tag arbitrarily chooses a value between 0 et 2^{Q-1}.
- The tag that chooses 0, responds arbitrarily in a particular slot with its own RN16.
- Every remaining tags decrement the counter of their slots to "1" and delay their response to the next communication cycle.
- If there is no collision, a tag is positively detected and the reader will send it back an 18-bit ACK command.
- After reception of the ACK command, the tag transmits its own EPC of 96-256 bits and 16 bits of CRC. Once identified, this tag must be disabled/deactivated.
- The next communication cycle begins when the interrogator sends a 4-bit Query Rep command to all remaining tags.
- Tags with 0 begin to respond, and the procedure is repeated until all tags are recognized “identified”.

The flowchart of the process, shown in Fig. 7 [3], is relatively simple to implement, but its major disadvantage is that the size of the frame cannot be adjusted linearly when the number of colliding slots or empty slots increases.

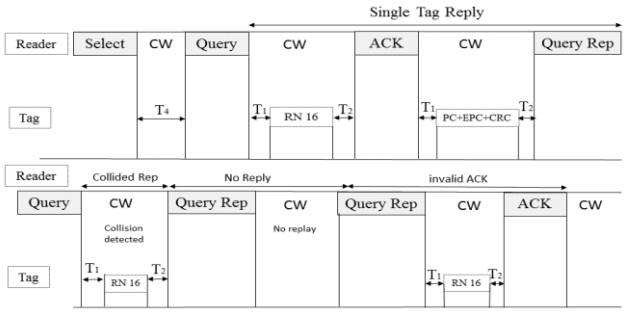


Fig.7. EPC global Class 1 Gen2 Standard (from [3]).

6.1. Simulation environment

We constructed a code in MATLAB which permits the manipulator to introduce the entire tags number to read. In every identification cycle, the RFID interrogator modifies the number of ITs according to the number of remaining unidentified RFID tags. The tags arbitrarily choose a location to answer to the request from the interrogator. If several tags reply to the identical identification time (IT), this means a collision state.
A user-defined variable was also included to state for the environmental interference: a tag read successfully will be one that was not involved in a collision and not be interfered with others.
All tags that experience collision or interference issues will remain unidentified and must participate to the next cycle (next inventory process). The simulation round will end when all the tags are successfully read and identified.

6.2. Configuration of simulation

The program was executed up to 499 tags with a step-size of 50. Each scenario was simulated 999 periods and the mean value was used as data point. The identification interval ‘IT’ was fixed to 1 ms, conforming to a bit rate of “100 kbps and 96 bits ID”.
In practice, the slot interval is determined by the communication frequency and the tag ID size. In addition, times slots are variable depending on whether the location gives rise to a unique successful response or collision, or finishes by inactive state.

6.3. Results of the EPCglobalClass1 Gen2 protocol

The time required to calculate the number of tags with constant interference is given by the proposed model:

$$T(n)=\frac{e}{1-q}n \quad (2)$$

with q the probability of interference. The graphs in Fig.8 were obtained for q values of 0.01, 0.02, 0.15, 0.30, and 0.6, relative to 'n : tags total number'. As expected, according to this figure, the time necessary to read tags rises as q increases. We then evaluated the performances of the EPCglobal Class1Gen2 protocol by comparing them to those of the S-ALOHA algorithm.

Two scenarios were used. The first supposition determines the time needed to identify the tags when the interrogator knows the entire number of tags in his area, the simulation stops when all tags are successfully identified. The second supposition accepts that the interrogator ignore the total number of tags in its reading area, here, the simulation depends on the q value which will reach zero without any other responses coming from the tags, that means a successfully identification process.

The program was executed up to 499 tags with a step-size of 50. Each simulation setting was run 999 times and the average value taken as result. For comparison purposes, we used the identical time interval of 1 ms.

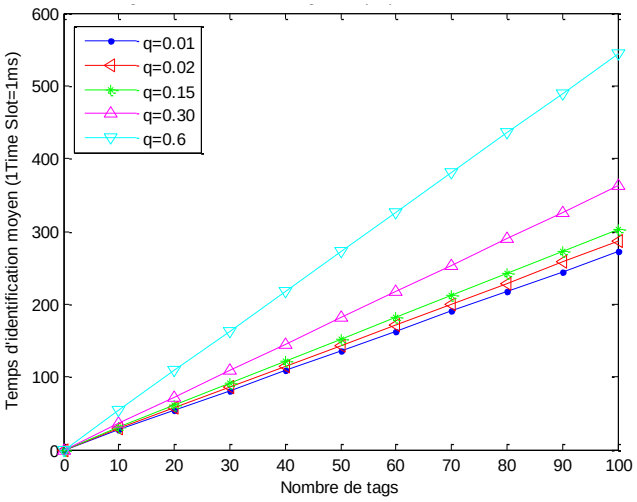


Fig.8. Reading time of RFID tags with interferences

The reading times of the RFID tags obtained with Class1Gen2 are displayed in Fig.9 and summarized in Table 2. The results illustrate a direct proportionality relation between the tag number and the identification time for the Class1Gen2 protocol and that this latter adds about $T(n)=\frac{3-e}{e}\times 100=10.7\%$ additional time compared to the S-ALOHA algorithm [23], [24]. This overload, as reported in [14], is principally caused by the limitation of the frame size, which must be a multiple of two in Class1Gen2. We also found close results (only a few times slots) between the two suppositions (number of known and unknown tags); which means that the Q algorithm works fairly well in approximating the tag number using the tag reading process. In addition, the value of the frame length adjustment constant "c" does not have a significant influence on the tag identification time.

According to [23], [24], the reading time of the tag with Class1Gen2 was set to approximately 8 ms, the slot duration to identify a colliding tag response to 1.9 ms and the slot duration for empty slot response (with zero tag in time slot) to 0.6 ms. Fig. 10 displays the illustration of tag read times with Class1Gen2 using calculated slot times.

Table 2. Read time (ms) for Class1Gen2 with a slot of 1 ms (Number of tags known and unknown)

N tags	With a known number of tags			With an unknown number of tags		
	c=0.15	c=0.25	c=0.35	c=0.15	c=0.25	c=0.35
0	0	0	0	19	13	10
250	735.0	737.7	745.4	741.3	740.4	747.6
300	877.4	882.4	892.8	884.3	886.3	895.7
350	1024.7	1028.7	1037.3	1029.2	1033.5	1042.4
400	1169.5	1177.2	1190.1	1178.2	1182.1	1192.4
450	1317.6	1323.9	1338.2	1324.7	1327.4	1341.3
500	1458.4	1471.8	1489.4	1471.9	1476.3	1491.3

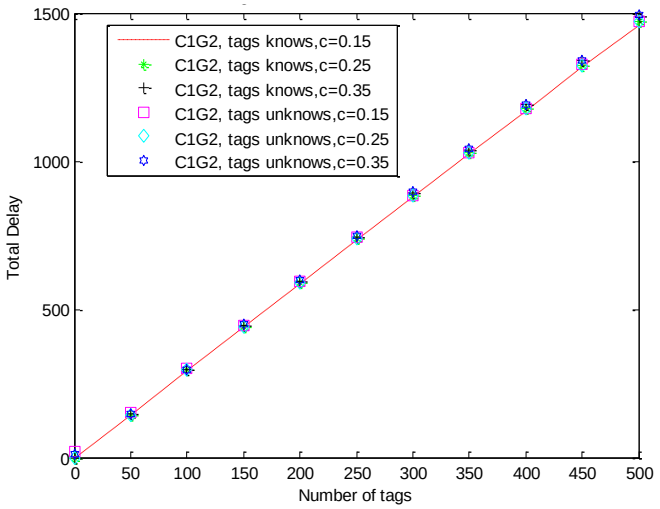


Fig.9. Reading time of 500 RFID tags using the Class1Gen2 protocol with 1 ms of slot time

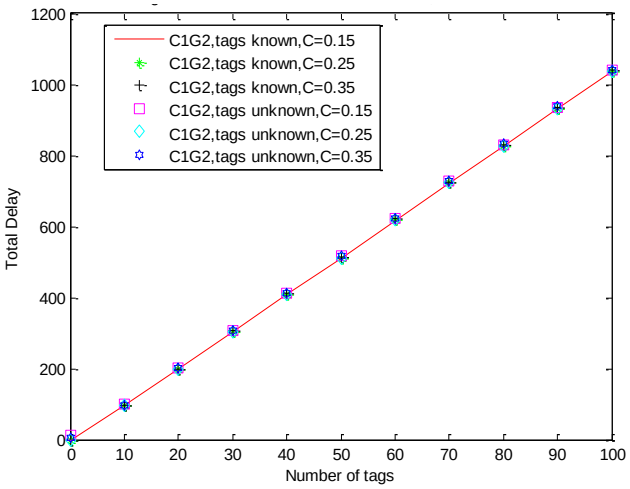


Fig.10. Time to identify 100 RFID tags using Class1Gen2 protocol with calculated frame times

6. Conclusion

In this paper, a behavioral simulation model of the EPC Gen2 protocol is discussed. Based on the various constitutional blocks of the identification layer by using the VHDL code, the proposed approach was demonstrated through improved performance for both speed and implementation of communication. Thus, the simulation results of the description modules based on the identification layers of EPC Gen2 protocol showed the expected behavior.

Then, the proposed EPC Class1 Gen2 model was used to success-

fully solve collision issues between tags and reader. This was demonstrated by considering the effect of interferences in two different cases namely, where the tag number is *a priori* known and when this number is unknown.

References

[1] B. Nath, F. Reynolds, and R. Want, "RFID technology and applications," *IEEE Pervasive Computing*, vol. 5, pp. 22–24, 2006.

[2] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, J. Wiley & Sons, New York, NY, 2003.

[3] ISO/IEC 18000-6, *Information Technology, Radio frequency identification for item management*, Part 6 Parameters for air interface communications at 860 MHz to 960 MHz.

[4] EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz, Version 1.0.9, January 2005.

[5] P.V. Nikitin and K.V.S. Rao, "Performance limitations of passive UHF RFID systems," *IEEE Int. Symp. Antennas Propag.*, pp. 1011-1014, 2006.

[6] H. Li *et al.*, "A new implementation of UHF RFID reader," in *TENCON IEEE Region 10 Conference*, pp. 1-4, 2009.

[7] P. Xingdong *et al.*, "Design and realization of a highly integrated UHF RFID reader module," *Int. Conf. Microwave and Millimeter Wave Technology*, pp. 1503-1505, 2008.

[8] Y. Chen and F.-h. Zhang, "Design on UHF RFID reader software," *Int. Colloq. in Computing, Communication, Control, and Management*, pp. 575-578, 2009.

[9] L. Joon Goo *et al.*, "Software architecture for a multiprotocol RFID reader on mobile devices," *Int. Conf. on Embedded Software and Systems*, 2005.

[10] Y. Ching-Chien *et al.*, "The design of encoding architecture for UHF RFID applications," *Asia-Pacific Microwave Conf.*, pp.1-4, 2008.

[11] S. Tung and A. K. Jones, "Physical layer design automation for RFID systems," *IEEE Int. Symp. Parallel and Distributed Processing*, pp. 1-8., 2008.

[12] M.A. Khan *et al.*, "FSM based FM0 and Miller encoder for UHF RFID Tag Emulator," *IEEE Int. Advance Computing Conf.*, pp. 1317-1322, 2009.

[13] L. Jin and T. Cheng, "Analysis and Simulation of UHF RFID System," *Int. Conf. Signal Processing*, 2006.

[14] P.B. Khannur *et al.*, "A universal UHF RFID reader IC in 0.18-μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 43, pp. 1146-1155, 2008.

[15] L. Jing *et al.*, "ASIC design of UHF RFID reader digital baseband," *Asia Pacific Conf. on Postgraduate Research in Microelectronics and Electronics*, pp. 263-266, 2010.

[16] L.T.Y. Yan Zhang and J. Chen, *RFID and Sensor Networks-Architectures, Protocols, Security, and Integrations*, Wireless Networks and Mobile Communications Series, 2010.

[17] P. Sweeney, *RFID for Dummies*, J. Wiley & Sons, 2005.

[18] V. Naja, S. Mohammadi, V. Roostaie, and A. Fotowat-Ahmady, "A dual mode UHF EPC Gen2 RFID tag in 0:18um CMOS," *Microelectronics J.*, 2010.

[19] I. Zalbide, J.F. Sevillano, and I. Vélez, "Design considerations for the digital core of a C1G2 RFID tag," in *Radio Frequency Identification Fundamentals and Applications, Design Methods and Solutions*, Ed. C. Turcu, pp. 324, 2010.

[20] J.A. Rodríguez-Rodríguez, J. Masuch, and E. Alarcón, "An ultralow-power mixed-signal back end for passive sensor UHF RFID transponders," *IEEE Trans. Industrial Electronics*, vol. 59, Feb. 2012.

[21] P. Lala, "Transient and permanent fault injection in VHDL description of digital circuits," *Circuits and Systems*, Vol. 3, pp. 192-199, 2012.

[22] O. Abdelmalek, D. Hely, and V. Beroulle, "EPC Class 1 GEN 2 UHF RFID tag emulator for robustness evaluation and improvement," *Int. Conf. Design & Technology of Integrated Systems in Nanoscale Era*, pp. 20-24, 2013.

[23] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communication*, Norwood, MA, Artech House, 2000.

[24] V. Namboodiri, M. DeSilva, K. Deegala, and S. Ramamoorthy, "An extensive study of slotted Aloha-based RFID anti-collision protocols," *J. of Computer Communications*, vol. 35, pp. 1955-1966, 2012.