Data security for EPC Gen-2: VLSI Design and its FPGA Implementation

Joyashree Bag,Rajanna K.M. Subir Kumar Sarkar,
Jadavpur University, Jadavpur, Kolkata-700032;
e-mail: Joyashree_bag@yahoo.co.in;
e-mail:rajanna_km@ieee.org; e-mail: subirsarkar@ieee.org

Abstract —**Electronic Product Code or EPC Global class-I Generation-I and Generation-II are popular and globally accepted standard specification which enables the use of password protection for accessing the data stored in the memory of a RFID tag. Use of password gives one of the ultimate securities for the RFID system but these are not immune to 'hacking'. Hackers are adopting advance technologies like side attack, power analysis to detect data password. But in many cases it is very much important that data must be secured enough. In this paper, we propose a data security system integrated with this protocol based on programmable cellular automata. We have implemented the protocol in VHDL. A synthesizable module is presented here in this article. System efficiency is evaluated according to the implementation.**

Keywords — EPC Gen-2, RFID, VHDL, RTL schematic, Simulation, Synthesis report.

I. INTRODUCTION

The global standard EPC Gen-2 protocol is the advanced version of EPC Gen-1 protocol. Some disadvantages and shortcomings of EPC Class 1 Gen1 overcome in the revised EPC Gen-2 protocol, which uses ALOHA based random anti-collision protocol called Q protocol (Q algorithm) [1,2,3]. It is of frame slotted ALOHA based probabilistic algorithm for tag anti-collision protocol with dynamic frame length. [1--2], [8--11].

VLSI implementations of algorithms are very important to achieve desired reliability, performance, low cost and high speed production. There are three possible types of slots; empty slot, when there is no tag to reply. Collided slot, when more than one tag to respond and the successful slot, when there is only one tag to respond and exchange information. The three conditions are depicted in Figure 1 with Q value 2. Tags are asked to choose slot number SN within the range $0 \leq SN \leq (2^2-1)$. So, the tags can choose 0, 1, 2 and 3 as their SN. But, if no tag is chosen, SN = 0, then the slot will be an empty and no tag will reply. When more than one tag choose SN = 0, it results a collided slot and reader modified its Q value and ask the tags to choose new SN. Successful slot, when only one tag responds to the Reader with SN = 0 and generates the 16 bit number RN16 and transmit [2--3].

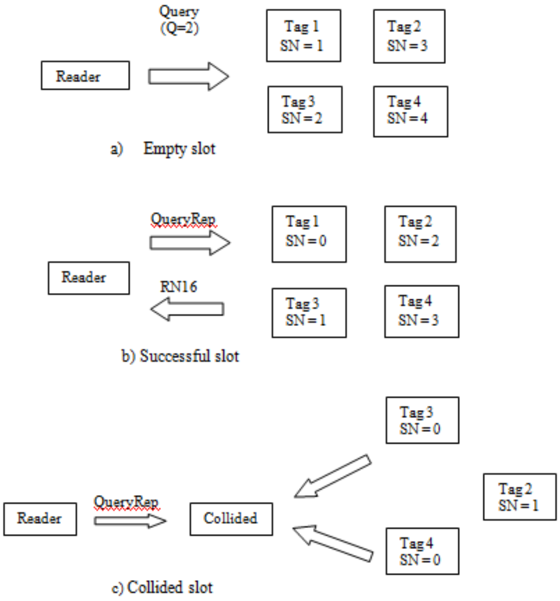


Fig 1: Empty, successful and collided slots

EPC Class 1 Gen1 and EPC Class 1 Gen2 protocols are ratified as the International standard anti-collision protocol [1--3]. But in EPC code data are not encrypted, rather covered by more pseudo-random number or data. Hence attackers or hackers can decode the data in several ways. So, password security does not provide the ultimate security for important data. It is possible to discover the details personal data if anybody has the knowledge of EPC references. Side channel attacks are executed by watching the power consumption or variations of the energy output of the devices. Important data or personal information especially sensitive personal data will need an adequate level of encryption to safeguard the data. Cryptography may be the suggested solution to give security to the information, but it has a high level of computational over head and needs a very high level processor to compute the process, may be with the cost of speed. We have used cellular automation rules to generate a secret code using a secret key within the EPC tag. We have designed the secret code generator and the processor both for the reader and the tag in VHDL code. Hence an unauthorized Reader will not be able to read the data unless the key is available providing high level data security for EPC tags.

978-0-7695-4941-5/13 $26.00 © 2013 IEEE
DOI 10.1109/ACCT.2013.77

Key points of EPC Gen-2 Tags

EPC Gen-2 tags have the ability to generate random numbers. The Reader will inform the tags the range in which it should generate a random number by issuing a query command and a Q value. The range will be 0 to (2^Q-1). If it often gets back no response to its queries, it will automatically reduce the Q value. If it gets more than one tag responding, it will increase the Q value, thereby increasing the range of numbers that can be generated by the tags [1, 4]. EPC Gen-2 Tags must be able to communicate in the frequency between 860 MHz to 960 MHz and are able to understand various modulation schemes such as: DSB-ASK, SSB-ASK and PR-ASK. Reader will determine which modulation scheme to be used. In this protocol, Tags must be able to transmit at several speeds or data rates because Reader will determine what speed to use according to application. EPC Gen-1 tags supports EPC code up to 96 bits long whereas Gen-2 tags supports EPC's up to 256 bits long. Gen-2 includes a method to support 'dense-interrogator channelized signaling' which is an attempt to reduce interference between EPC readers.

Basics of Q-Algorithm:

EPC Gen2 uses the Q-algorithm which represents the important transmission control strategy. It keeps the current frame size in track with the tag responses as well as collision and empty slots. The performance of Q algorithm can be significantly improved when changes of Q value are restricted. In Q algorithm the value of Q changes according to the collided slot and empty slot to improve the system efficiency, i.e. the success rate of an inventory round to detect a tag [8, 9], [11, 12]. The Q-algorithm started with a standard value of Q = 4; it is denoted as Q_{fp} as it has floating value rather than fixed at 4. The Reader considers the current Q_{fp} as the Q value and transmits. If there is zero response, it is an idle case and Q is set its value as Q = Q-1(value of Q remain unchanged if Q = 0); on the other hand, if more than one tag responds, it is a case of collision; Q is modified its value as Q=Q+1(value of Q remain unchanged if Q = 15). Finally, for the successful slot, when only one tag responds to the Reader's query, Q remain unchanged and another inventory round started. For each case Reader transmits different commands to the tags like 'Query', 'Queryrep' and 'Queryadjust' to instruct them for next operation according to the algorithm for EPC Gen2
Protocol. This paper is organized as, in Section II narrates the introduction to data security, in Section III the basics of Cellular automata are described. Section IV describes the proposed algorithm and operational flow chart of the protocol. The implementation and test bench simulation Results as well as synthesis report of the implemented processor are introduced in Section V. Finally, we conclude in Section VI.

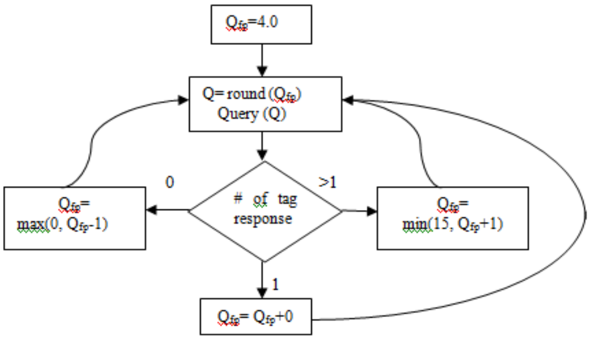


Fig 2. Q-Algorithm

II. Data Security

There are three qualities that define data security in an RFID system, firstly, controlled access to the data, i.e. only authorized reader can read and write information. Secondly, the control over access to the system, i.e. only authorized entities can configure and add to the system and all devices on the system are authentic and trustworthy. Finally, confidence and trust in the system, i.e. users share a general perception that the system is safe and secure. This is a more subjective criteria, but important. To trust a system these qualities should be fulfilled. Now, every communication system has its own level of data security—from wireless devices to the Internet. Every type of data does not merit the highest level of security because enhancing levels of security leads to introduce extra cost and technological complexity, and RFID is not an exception. It's critical and challenging to balance security threats against security costs. For widespread acceptance, RFID technology must achieve an appropriate level of confidence, security and trust.

Consumers want to ensure that their personal information isn't misused, and that RFID tags are used responsibly. Corporations also want to use RFID technology to increase efficiency, serve consumers better, and gain a competitive advantage. So, RFID system needs to ensure the reliability and security of their systems, as well as their usefulness and competitiveness.

Shortcomings of EPC Generation 2 protocol:

In the EPC Generation 2 protocol, there are several issues that serve as blockage to deployments at the consumer level:

Data in an EPC code is not encrypted, but cover coded by means of a pseudo-random number transmitted by the tag. This code can be recovered very easily by a side-channel attack.

Like data, passwords are not encrypted, but cover coded - which is less robust than a strong cipher.

Lack of authentication introduces the risk of rogue/clone tags or rogue/unauthorized readers to an RFID implementation.

So, it is clear that the level of security in EPC Gen 2 is not sufficient to meet the original criteria of data security. Access to the data is not secured. Access to the RFID system is similarly open to manipulation and attack. And most importantly, security levels are not high enough to generate the high levels of consumer trust that will enable widespread acceptance of RFID at the item level. Rogue/clone tags, rogue/unauthorized readers, and side-channel attacks all threaten data security.

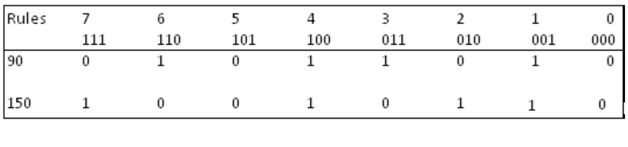
III. The Basics of Cellular Automata

In their article, 'Theory and application of non-group cellular automata for message authentication' Mr. Prabir Dasgupta et.al. have described the programmable automata rule. [5,6]A cellular automaton consists of several identical cells or nodes which are a finite state automaton (FSA) governed by simple rules, like 90's rule, 60's rule, 150's rule etc. Rules 90 and 150 are important. Rule 90 is the sum modulo 2 of the states of the nearest two neighbors. Rule 150 is the sum modulo 2 of the states of the nearest two neighbors and the state of the cell itself. Both rules 90 and 150 are linear.[7] The cellular automaton is synchronized i.e, at each time step each cell evolve a global update function applied uniformly over all the cells and updates its state according to some set of automation rules. The next state of each cell depends on the present state of the neighbor cells. This update function takes the cell's present state and the states of the cells in its interaction neighborhood. Moreover, PCAs are suitable for hardware implementation since they are very simple, regular, locally interconnected, and modular. [6,7]. Structure of Programmable cellular Automata is very simple, where we find 4-bit PCA resembling a 4-bit parallel in parallel out 4-bit register consists of four D-Flip-Flops and four XOR gates connected in such a way that each state is determined by the two neighboring states.[5,6]

Generation of secret code 'Sc' using automation rules:

We have used rule 90 and rule 150 due to their advantages to implement in VHDL code. So here we will discuss briefly about the required rules. In automation rule the next state of cell can be determined if the present state and previous state is known. Table 1 shows the chart of these states.

Table 1. Rules that update the next state of the cells



Rules	7 111	6 110	5 101	4 100	3 011	2 010	1 001	0 000
90	0	1	0	1	1	0	1	0
150	1	0	0	1	0	1	1	0

The binary number $(01011010)_2$ represents the decimal number 90 and the binary number $(10010110)_2$ represents the decimal number 150. The proposed encryption system it is realized using a combination of two CA. We use a first CA as a key stream generator, a CA pseudorandom number generator (PSRG)[7] that combines in some way two rules (the rules 90 and 150), to provide the key sequence.

The rules 90 and 150 can be expressed as follows:

$Sc_i\ (t + 1) = Sc_{i-1}(t)$ XOR $Sc_{i+1}\ (t)$…………… Rule. 90

$Sc_i\ (t + 1) = Sc_{-1}(t)$ XOR $Sc_i(t)$ XOR $Sc_{i+1}(t)$… Rule. 150

This CA is used to provide real-time keys for the block cipher in this paper. The operation of CA can be represented by a state-transition graph.

Suppose we choose key Matrix as $K=\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$

Now if the K_{i0}='0' then the entire row will follow the rule 150 and if K_{i0}='1' then it will follow rule 90. From the chosen key Matrix, 1st and 3rd row will follow rule 90 whereas 2nd and 4th row will follow rule 150.

For rule 90:

i. Data of 1st column of secret code Matrix Sc will be as $Sc_i = K_i \oplus K_{i+1}$

ii. Data of 4th column of secret code Matrix Sc will be as $Sc_i = K_{i-1} \oplus K_i$

iii. Data of other column of secret code Matrix Sc will be as $Sc_i = K_{i-1} \oplus K_i \oplus K_{i+1}$

For rule 150:

i. Data of 1st column of secret code Matrix Sc will be as $Sc_i = K_{i+1}$

ii. Data of 4th column of secret code Matrix Sc will be as $Sc_i = K_{i-1}$

iii. Data of other column of secret code Matrix Sc will be as $Sc_i = K_{i-1} \oplus K_{i+1}$

Following these rules the code,

Matrix for Sc will be as $Sc=\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Now if we integrate this code Matrix with the tag ID, only the Reader, who knows the exact Key Matrix, will be able to decode the ID and the data or information stored within the tag. Thus we can prevent data hacking by unauthorized Reader.

As cryptography needs a high level processor and computational overhead, we have proposed these programmable cellular automata based data security for RFID system. If we are thinking about EPC Gen 3 protocol for higher security with fewer complexes, more reliable and minimized cost, this proposed algorithm may introduce a novel idea.

IV. DESIGN OF THE READER AND TAG PROCESSOR

For design simplicity and development, the operational flow chart of both the Reader and Tag are specified. The detailed operation of Reader and Tag end is as shown in Fig 3. We have designed the processor for both the reader and tag module to perform the operation efficiently within a fraction of seconds. The performances are tested using suitable test bench simulations following a successful synthesis process. Flow charts of the Reader and Tag module are described in Fig 4 and Fig 5 respectively.

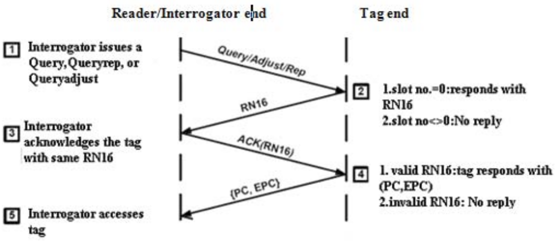

Reader/Interrogator end
Tag end
1 Interrogator issues a Query,Queryrep, or Queryadjust
Query/Adjust/Rep
2 1.slot no.=0:responds with RN16
2.slot no<>0:No reply
RN16
3 Interrogator acknowledges the tag with same RN16
ACK(RN16)
4 1. valid RN16:tag responds with (PC,EPC)
2.invalid RN16: No reply
{PC, EPC}
5 Interrogator accesses tag

Fig 3. Reader and Tag Operation

Reader operation:

Interrogator or reader, in this protocol manages the tag population using the four basic operations:

- *Select*- The process by which an interrogator selects a tag population for inventory and access.
- *Inventory*- The process by which an interrogator identifies a tag. An interrogator begins an inventory round by transmitting a 'Query' command with Q='4' and ask the tags to activate their slot counter and choose a slot number 'SN';[$0 \leq SN \leq 15$];if any tag choose SN ='0' it request the 'RN16' and send an acknowledgement signal 'ACK' containing same 'RN16';otherwise if SN≠'0' then transmit 'Query rep' command which ask the tag to reduce SN by '1' and modifies Q decreasing by '1'; if more than one tag replies, reader transmit 'Query adjust' command and modifies Q increasing by '1';Request the identified tag for information, like PC, EPC, CRC-16 etc.
- *Security Check*:- The process by which a reader decodes the EPC code like password and Tag ID using same programmable secret code generator and cellular automata rules.
- *Access*- The process by which an interrogator transacts with the individual tags.

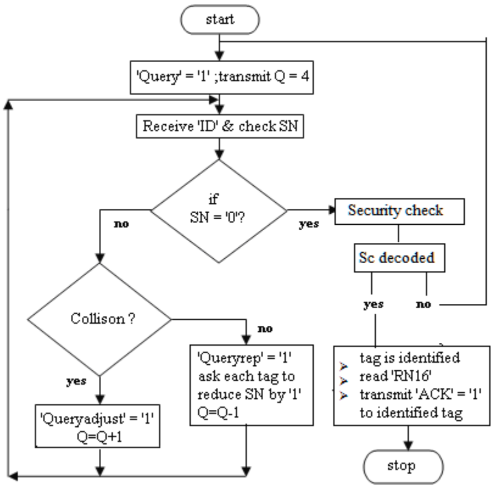

start
'Query' = '1' ;transmit Q = 4
Receive 'ID' & check SN
if SN = '0'?
no
yes
Security check
Sc decoded
yes
no
Collison ?
no
yes
'Queryrep' = '1' ask each tag to reduce SN by '1' Q=Q-1
tag is identified
read 'RN16'
transmit 'ACK' = '1' to identified tag
'Queryadjust' = '1' Q=Q+1
stop

Fig 4. Reader Operation

Tag Operation:

- Tag shall implement a Slot counter. Upon receiving a 'Query' or 'Query adjust' command tag shall preload into its slot counter a value 'SN' between 0 and (2^Q-1). When Q=4, the range is 0-15; a 'Query' specifies Q=4; a 'Query adjust' may modify Q from the prior Q value.
- If a tag select its slot number SN='0', it is in the reply state. Otherwise it decreases SN by '1';
- Tag shall implement a random number generator. The 16-bit number 'RN16' is generated.
- The tag backscatter the number 'RN16';
- If the interrogator acknowledges the tag with an 'ACK' signal containing same 'RN16', the identified tag transmit the important information, i.e. EPC.
- The tag turns itself into sleep mode and ceases to respond.

Tags of Gen2 protocol have random bit generator which generates slot number 'SN' and the 15-bit random number 'RN-16'.For experimental purpose we have designed a format of initial tag ID which will communicate with the reader first containing the SN (slot number, RN16 and tag); if and only if SN='0' then the RN16 will be generated and transmitted. After receiving 'ACK' the 2-bit tag will be '01' which in turn gives a signal to the tag to transmit the information stored in it. The standard tag data frame contains kill password, access password, EPC, TID and extra data space for user. A Tag EPC is 64 bits in length including 16 bit PC, 16-bit EPC and CRC-16. Frame size may be up to 496 bits. Manufacturer can choose the frame size as per the requirement of the product to be coded. In our design when the 'tag' bit of data frame will be '01', it indicates that primary task is over and activates the

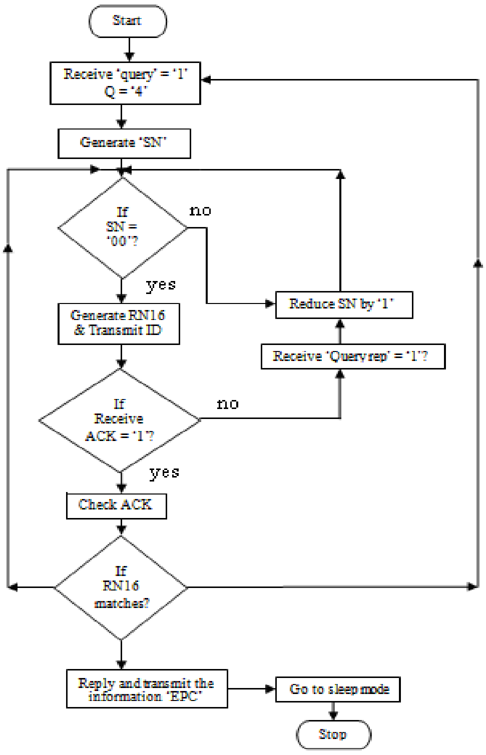


Fig.5: Tag operation

standard data frame so that the information can be transmitted to the reader more secured and accurate form. Because secured data transmission is the basic importance of RFID system. If we transmit the tag ID and password generated from a secret key generator module using programmable cellular automata rules, reader or hacker will not be able to read the ID or other information unless they have been provided with the secret key. The authentic tag and authentic reader will be provided with a secret key. Both will have the same module for secret code generation, thus will use same automata rules. Hence the tag ID and the password is now protected and secured. Unless the reader be able to detect the tag ID and the password it is quite impossible to access the memory of the tag in the system.

SN(4-bit)	RN-16(16-bit)	Tag (2-bit)

Fig.6: Proposed data frame of tag

Kill password	Access password	EPC	TID	User memory(optional)

Fig.7: Standard data frame of tag

V. Implementation of the Reader and Tag processor

We use VHDL to describe a system at the behavioral level so that we can simulate the system to check the algorithm used and to make sure that the sequence of operations is correct. VHDL simulation provides the verification of the hardware organization and operation at structural level whether the selected bit widths for internal and external signals are sufficient for achieving a low precision to speed up the system efficiency and reduce the implementation complexity. The step by step process by the test bench simulation results are described here. We have designed the processor in VHDL code and tested using Xilinx 9.2 ISE simulator to get the desired output. The test bench timings are in 'ns' range and clock frequency of RF range. Test bench simulation of designed module is followed by the synthesis process which results the RTL schematic diagram of the processor along with a study of device utilization, circuit delay, components requirements and static power consumed by the circuit during simulation. We can modify the design as per requirements with no time. The Synthesizable module may be easily implemented on the reconfigurable FPGA board.[16-19]

First we consider the Reader module, which transmit commands and query according to the tag response. Once a Tag interrogates with the Reader, the reader will receive data encrypted with automata rules using a definite code and rules without which reader will not be able to decode the tag identity and password to steal or capture the information stored within the Tag.

According to the design, initially Reader starts inventory round with Q = 4. The modified Q is represented as Q_f. after successfully identification of a tag, reader broadcast an ACK signal and decrease its Q_f by '1' automatically up to Q = '1'. But, if no tag responds or more than one tag responds then Q_f increases by '1' automatically until Q = '15'.

When the tag is identified by reader it sends ACK signal with secret code 'Sc' generated by the 'Sc_Generator' module. ACK_{16}='1' and ACK_{15-0} as 'RN16' read from I_{17-2}. When tag is identified Q_f = Q-1=3 and otherwise Q_f = Q+1=5. From Fig. 9, we can study the output waveform. We have verified the result for different cases. Fig. 8 shows

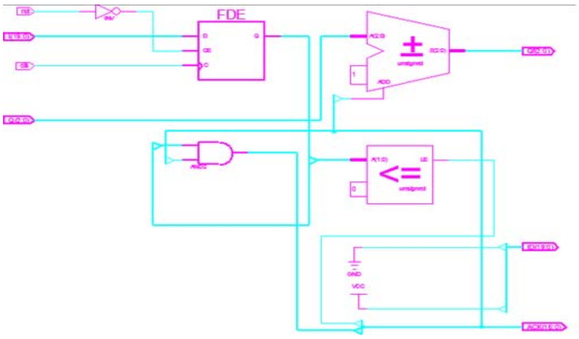


Fig. 8: RTL View of Reader Processor

RTL view of the Reader Processor. Now, at the Tag end we will observe the operational performance of our design. As it receives ‘Query’ signal with Q=’4’ it activates its Random bit Generator unit and generate the Slot number ‘SN’ and the 15 bit number ‘RN-16’. Upon receiving response from the reader end it decides whether it will transmit ‘RN-16’ or reduce the slot number. We have used this module to design the tag ID generator i.e, the module to integrate the original tag ID with the generated secret code Sc from the Sc_generator module for the Tag. The dataframe for tag will include the 8-bit unique ID and this code. So to retrieve data from this dataframe, Reader must know the key Matrix to decode the data frame.

As Fig.5 describes the operational flow chart of the tag, we design the tag processor to meet the requirements. The RTL schematic diagram of the tag is shown in Fig. 11. From Fig. 12, we observe that the output of the tag processor after the test bench simulation. If Q=’4’ and ‘query’ signal is high (‘1’) then if ‘ACK’ signal is there (ACK is received only if SN=’0’), the tag will transmit its ID, otherwise it do not transmit RN-16.

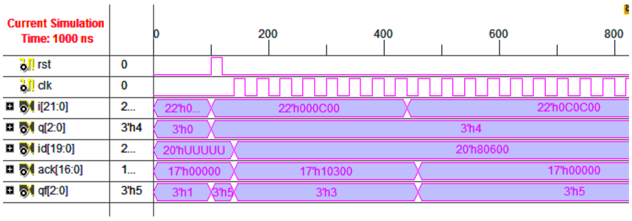


Fig. 9: Test bench simulation of reader processor (inventory part) module

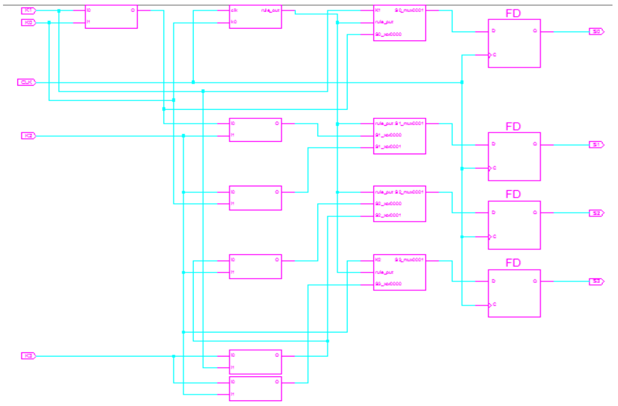


Fig 10: RTL schematic view of Sc_Generator (secret code generator)

Synthesis report

For synthesis, we have used Xilinx 9.2 ISE simulator for the simulation process of our processor. The static power consumption for the simulation is only 25mW, but in case of practical design we may choose different suitable low voltages to reduce power consumption.

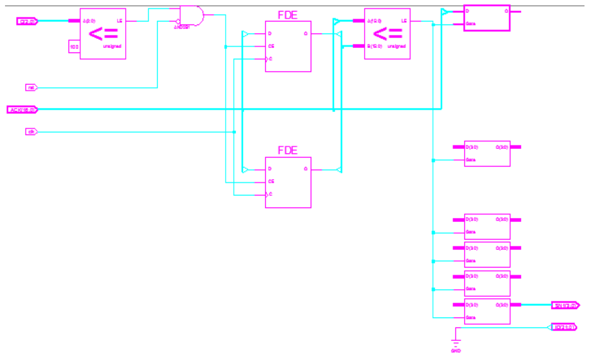


Fig 11 RTL schematic of Tag

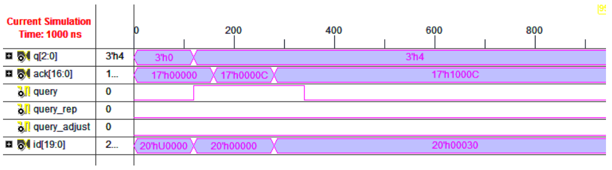


Fig. 12: Simulated output of tag

The timing scale is set in ‘ns’ range and simulation is performed for 1000 ns duration. We can modify our processor for any type of design constraints and requirements and implement on easily reconfigurable FPGA development kit.

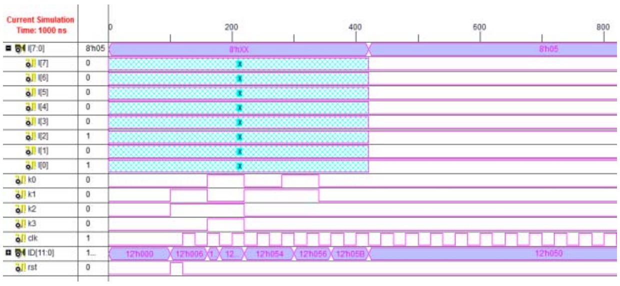


Fig.13: Test bench simulation result for the reader processor (security part).

In Fig.13, ID is the 12 bit data read by the reader as tag ID, where ID(3) to ID(0) represents the ‘Sc’, if the reader has same key and ‘Sc’, the ID of the tag which is represented by rest of the bits[ID(11) to ID(4)] could be retrieved, here it is “0000 0101” (05_h). We observe in Fig.13, that until the bits are all checked the reader output ‘I’ is in high impedance state (X) and after checking is complete, reader shows the tag ID as ‘I’ as 05_h.

VI. Conclusion

In this paper we have designed and implemented the EPC Gen-2 protocol up to RTL schematic level with modified algorithm in VHDL code to get better performance, improved speed and efficiency and security. The design is targeting the Xilinx FPGA device, this leads to real time environment verification of the system. VLSI Design of protocol is performed to achieve high speed, minimum hardware and enhanced system efficiency with highly secured data transfer to authenticated reader. It has high level syntax and supports extensive optimizations.

REFERENCES

[1] A user guide EPC Gen-2, Thing Magic, version 1.0, April2005, www.thingmagic.com

[2] Yinghua Cui and Yuping Zhao, A Modified Q-parameter Anti-collision scheme for RFID systems, Proceedings of the IEEE International Conference on Ultra Modern Telecommunications & Workshops (2009), October 12-14, Beijing, China

[3] J. H. Choi, D.Lee and H. Lee, Query Tree-based reservation for efficient RFID tag anti-collision, IEEE Communications Letters, vol. 11, pp.85-87 (2007).

[4] Ching-Nung Yang and Jyun-Yan He, An Effective 16-bit Random Number Aided Query Tree Algorithm for RFID tag Anti-collision, IEEE Communications Letters,vol.15, No.5 (2011).

[5] Prabir Dasgupta, Santanu Chattopadhyay and Indranil Sengupta: 'Theory and application of non-group cellular automata for message authentication', Journal of System Architecture, Elsevier, vol.47,issue 5,May2001, pages: 383-404.

[6] Samuel Charbouillot, Annie P´ erez, and Daniele Fronte, 'A Programmable Hardware Cellular Automaton:Example of Data Flow Transformation' 'VLSI Design, Hindawi, Volume 2008, Article ID 160728, 7 pages doi:10.1155/2008/160728, Research Article

[7] S. Wolfram, "Theory and applications of cellular automata", *Wolfram Scientific*, 1986.

[8] Specification for RFID Air Interface, www.epcglobalsp.org/standards/Tags/tag_class1gen2.pdf

[9] Bo Li, Yuqing Yang and Junyu Wang, AUTO ID LABS White paper WP-Hardware-047 (2009).

[10] Hush, D. R.,Wood, C, "Analysis of Tree Algorithms for RFID Arbitration", Proceedings of the IEEE International Symposium on Information Theory, (1998), August 16-21, New Mexico Univ., Albuquerque, NM.

[11] Floerkemeier, C., "Bayesian Transmission Strategy for Framed ALOHA Based RFID Protocols", Proceedings of the IEEE International conference on RFID, (2007), Massachusetts Inst. of Technol., Cambridge.

[12] Floerkemeier, C., "Infrastructure Support for RFID Systems", Doctoral and Habilitation Theses, (2006), ETH Zurich, University of Cambridge.

[13] Vogt, H., "Efficient Object Identification with Passive RFID Tags", Proceedings of the First International Conference on Pervasive Computing,(2002), Springer-Verlag London, UK.

[14] Jae-Ryong Cha and Jae-Hyun Kim, "Novel anti-collision algorithms for fast object identification in RFID system", Proceedings of the 11th International Conference on Parallel and Distributed Systems (2005), July 22-22, Sch. of Electr. & Comput. Eng., Ajou Univ., Suwon.

[15] WT CHEN and Guan-Hung LIN, "An efficient Anti-Collision Method for Tag Identification in a RFID System", IEICE Ttrans. Commun., vol.E89–B, No.12, pp.3386-3392 (2006).

[16] J.Bhasker, 'A VHDL synthesis Primer', BS Publication

[17] Wayne Wolf, 'Modern VLSI Design' 4th edition; PHI Learning Private Limited.

[18] Stephen Brown and Zvonko Vranesic, 'Digital Logic design' Tata McGraw Hill Publication.

[19] www.vhdl.org