International Journal on Applications in Information and Communication Engineering
Volume 1: Issue 12:December 2015, pp 51-59. www.aetsjournal.com ISSN (Online) : 2394-6237

Efficient Implementation Of Ecc Based Digital Baseband And Schnorr Protocol Rfid

E.Sangeetha , A.Elakkiya

Abstract— The extremely constrained resource has hin- dered the efforts to implement the elliptic curve cryptography (ECC) into a radio frequency identification (RFID) tag chip. In this paper, an ECC-based RFID digital baseband controller (DBC), which is compatible with ISO/IEC 14443 and Schnorr authentication protocol, is presented. In order to achieve low resources consumption and fast ECC computation speed, some techniques, such as the register reuse, clock multiplexer, and asynchronous counter, are adopted in this design. In addition, a linear feedback shift register-based stream encryption scenario is proposed for the data security. According to the synthesis result, the gate area and power consumption of DBC are 25.7 K and 14.7 μW, respectively, in UMC 0.13 μm CMOS technology. All of those characteristics make the realization of ECC-based DBC for RFID tag chip promising.

Index Terms — Digital baseband controller (DBC), ellip- tic curve cryptography (ECC), linear feedback shift register (LFSR), radio frequency identification (RFID) tag chip, stream encryption.

I. INTRODUCTION

RADIO-FREQUENCY identification (RFID) is a contact Less technology that enables transmitting information between readers and tags. The rapid development of Internet of Things has stimulated the growth of RFID technology [1]–[5]. For example, some papers have been published to improve the performance of anticollision algorithm, i.e., the basic method to identify a RFID tag. [6], [7].

Nowadays, with the wide use of Internet and the wireless payment technology, the new challenge for designing a RFID tag is how to embed a low-cost and high-performance security algorithm into a compact RFID tag. The demand on the security and privacy of RFID tag is also becoming more and more important recently [8]–[10]. Public key cryptography (PKC) has been proved to be one of the most excellent encryption schemes for its long keys and algorithm [11], [12]. The elliptic curve discrete logarithm problem, belonging to the catalog of the PKC, obtains the public attention because of its excellent encryption performance and its lower requirement on resource. To achieve a good tradeoff between security and constrained resource, the elliptic curve (EC)-based cryptosystem would be one of the best candidates for RFID systems [13]–[16].

The International Standard Organization (ISO) has published a series of protocols for RFID communication, such as the ISO/IEC 14443 and ISO/IEC 18 000 series protocols. However, those protocols hardly demonstrate an approach to solve the problem of secure authentication. Although a sufficient quantity of work has been done on the field of secure authentication, particularly based on EC cryptography (ECC) [17]–[21], the ECC-based RFID secure authentication has not been applied into a practical circumstance yet. Those papers demonstrated several authentication schemes with complex computation units, which might conflict with the characteristics of limited resource on the RFID tag chip.

Based on the decades of design experience in RFID tag chip, we propose a new ECC-based digital baseband controller (DBC) compatible with ISO/IEC 14 443 protocol. Considering the constrained resources on RFID tag chip, a relatively simplified ECC-based Schnorr authentication protocol is adopted. In addition, the selective EC is over $GF(2^m)$, where the m is equal to 163. Moreover, in order to achieve security in further communication, a linear feedback shift register (LFSR)-based stream encryption scenario is proposed as well. Moreover, since the EC processor (ECP) and large integer operation module consume the most significant resource, several techniques such as the register array, register reuse technique is applied into this design. On the other hand, clock multiplexer and asynchronous clock scheme are utilized for meeting the demand of real-time performance and low power consumption in RFID system.

The rest of this paper is organized as following: Section II presents the structure of ECC-based RFID DBC. In Section III, Schnorr protocol is analyzed and implemented. In Section IV, the LFSR-based stream encryption scheme is proposed. Several lightweight schemes are demonstrated in Section V. Finally, the result and comparison are summarized in Section VI and VII, followed by the conclusion in Section VII.

E.Sangeetha , M.E.Communication Systems In The Ponnaiyah Ramajayam Engineering College In Thanjavur (Email : Sangeethasri68@Gmail.Com)

A.Elakkiya,(M.E) Asst.Professor In The Ponnaiyah Ramajayam Engineering College In Thanjavur

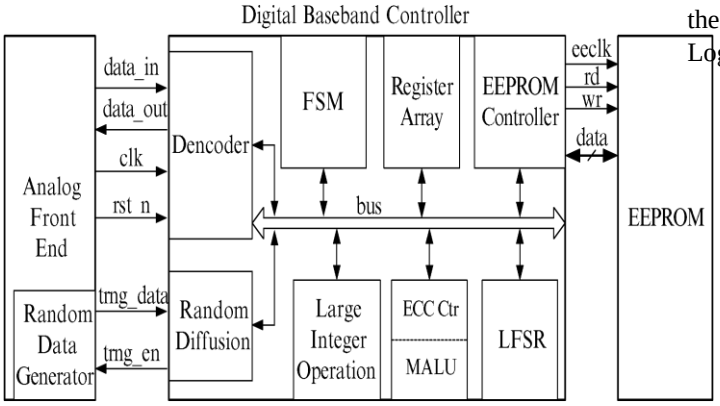

Fig. 1. Structure of the ECC-based RFID tag chip.

II. STRUCTURE OF ECC-BASED DIGITAL BASE BAND CONTROLLER

The proposed ECC-based DBC contains eight parts, as illustrated in Fig. 1. In Fig. 1, we also present the other two parts, i.e., analog front end (AFE) and electrically erasable programmable read-only memory (EEPROM), composing of an entire RFID tag chip.

In a mutual communication between reader and tag, the AFE sends demodulated data to the DBC [22]. The Dencoder module checks the validation of the frame and extracts useful information, such as the command data. As long as the frame is valid, the finite-state machine (FSM) module will control the exchange of information in different modules through 8-bit bus architecture. When the command is authentication related, the random diffusion unit will acquire random data from AFE and delicately treated the random data by specific diffusion algorithm. ECC module is composed of ECC Ctr and modular arithmetic logic unit (MALU). In addition, since the ECC module and large integer operation module consume the most resource, both of these modules will reuse the register array during computation. Moreover, the LFSR, applied to stream encryption, reuse the register array as well. All of these three modules have the same interfaces and accesses to register array. When the DBC requires several operations on EEPROM, the EEPROM controller will generate signals, obeying the timing diagrams of EEPROM, to have the approaches to EEPROM internal data. After DBC completes all of the operation, the responding data will be encoded in Dencoder module and transferred to AFE.

A. FSM

The FSM module is the heart of this proposed DBC, controlling the data path and executing basic arithmetic operations. The simple example of an FSM is listed in the following.

Fig. 2 is the basic FSM state transition diagram. As shown in Fig. 2, there are four states in the FSM with three basic arithmetic operations, i.e., addition, right shift, and exclusive-OR. In order to implement the FSM, designers may use four arithmetic operators described by Verilog HDL. However, since each arithmetic operator will result in area consumption, the better way to design FSM is using a uniform Arithmetic Logic Unit (ALU) to solve all basic arithmetic operations.

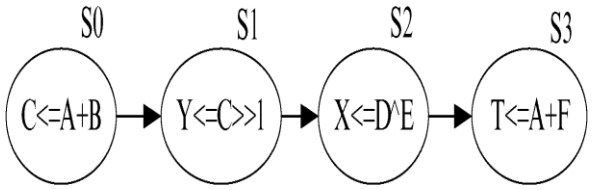

Fig. 2. FSM state transition diagram.

In this paper, the proposed FSM has the architecture, as illustrated in Fig. 3.

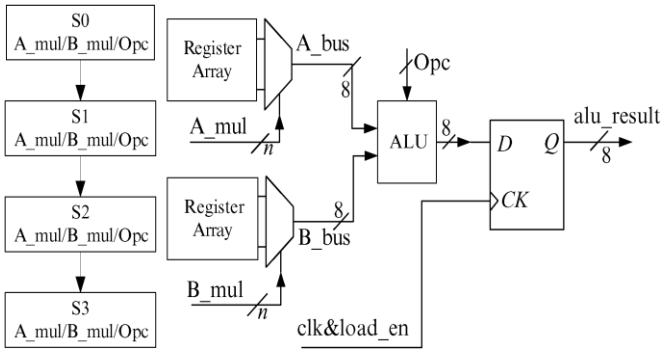

Fig. 3. FSM structure.

As shown in Fig. 3, in each of the states, the FSM will specify different values of *A_mul*, *B_mul* and *Opc* to control the operands and operators in ALU. For example, in state S0, the *A_mul* is set to a certain index value. Then, the multiplexer will transfer the data *A* stored in register array through bus structure *A_bus*. The same process will be repeated until the ALU gets the correct data *B*. In state S0, an addition should be executed, and the *Opc* is employed to tell the ALU what kind of arithmetic operation is preceded. When the results of the ALU need to be stored in registers, the FSM will set the signal *load_en* and latch the results in flip-flops. For example, the FSM will latch the result of *C* in state S0 and use it for right shift in state S1. In our proposed FSM, we reduce a large number of redundant operators and add a lowcost index list. However, it is clear that if the FSM has more states and arithmetic operations, the proposed FSM will have better performance in area reduction. According to the report of Design Compiler, this ALU-based FSM can save about 20% area overhead.

B. ECP

A typical ECP contains three parts, namely, register files, controller, and MALU, as shown in Figs. 1 and 4. The ECP is used to solve the problem of scalar multiplication on EC.

The basic algorithm for scalar multiplication is the so-called binary method [23]. However, the Montgomery ladder [24]

algorithm is utilized for its security against simple side channel attacks. In addition, The López–Dahab projective coordinate[25] is adopted to accelerate the Montgomery ladder algorithm.

The coordinate exchange from affine coordinate to projective coordinate is shown as

$$X = XZ \ , y = YZ2 \ . \ (1)$$

Due to the López–Dahab projective coordinate and themontgomery Algorithm, we reduce the number of field inversion over *GF*(2163) to only one.

Fig. 4. Structure of ECP module.

TABLE I

FOUR OPERATIONS OF MALU ON *GF*(2163)

Operation	Action
MOV[$addr_A$] [$addr_C$]	Move [*A*] to [*C*]
SQR[$addr_A$] [$addr_C$]	Proceed squaring on [*A*] and stores results in [C]
ADD[$addr_A$][$addr_B$] [$addr_C$]	Proceed addition on [A],[B] and stores results in [C]
MUL[$addr_A$][$addr_B$] [$addr_C$]	Proceed multiplication on [A],[B] and stores results in [C]

In Fig. 4, Register Files are composed of 163 6-bit register arrays for storing temporary calculation results during scalar multiplication. MALU is responsible for three basic arithmetic operations in *GF*(2163), i.e., addition, square, and multiplication.The Inversion on EC can be replaced by square and multiplication according to the Fermat's Little Theorem [26].Controller provides *operand_mul* to select specific register sets as MALU operands. Similarly, the *operator_mul* is used to select a kind of operator for MALU. In this proposed ECP, four operations are employed, as listed in Table I.

C. Large Integer Operation Module

This module is employed for solving the problem of big integer arithmetic operation, including addition, multiplication, and modular reduction. In fact, the modular reduction operation needs more computation than the addition and multiplication themselves in the PKC system. Although we can use efficient reduction such as Montgomery's reduction [27] and Barrett's algorithm [28], these methods require a large area overhead on data transformation and temporary memory. Thus, we adopt the algorithm described in reference [13], which only needs 44 bytes to store the contemporary results and two precalculated data. Considering the practical situation, the two precalculated data with a length of 163 bits will be stored in EEPROM and extracted by this module.

III. SCHNORR PROTOCOL ANALYSIS

Recently, many works in the literature have shown great interest in the field of mutual authentication based on ECC. However, these proposed authentication protocols require tags to do complex large integer operation, such as inversion, which requires complex logic control units and computation resource. As illustrated in Fig. 5, the Schnorr protocol is presented [13], [17]. Compared with some mutual authentication schemes, the Schnorr protocol is a single direction authentication protocol. In addition, the Schnorr protocol only need tags do the work of scalar multiplication and simple modular

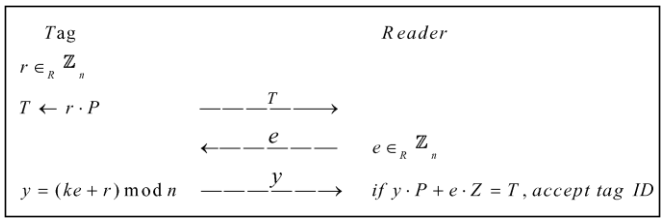

Fig. 5. Schnorr authentication protocol.

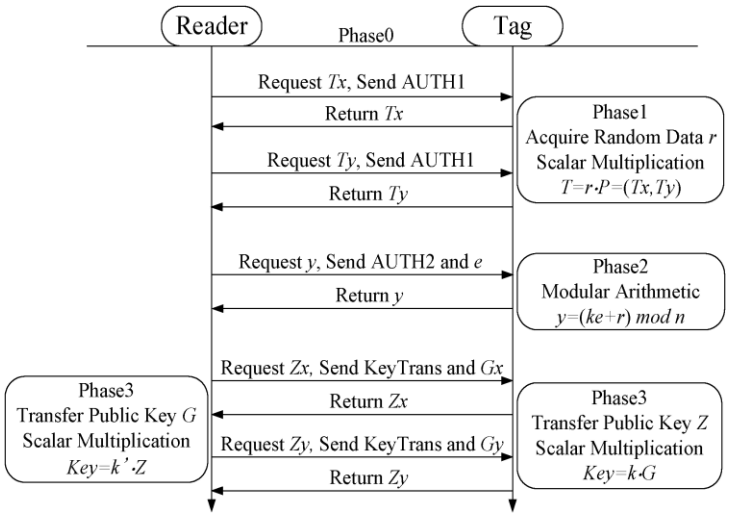

Fig. 6. Process of Schnorr protocol in command level.

arithmetic. In Schnorr protocol, tag should send its public key Z without any risks of revealing tag's private key k. Similarly, the reader will not reveal its private key k by sending its public key G. In Fig. 5, the r and e is random data with the length of m bits and the n is the order of the base point P on EC. The chosen EC has the form of (2); thus, the validation of the Schnorr protocol can be proved by (3)

$$y^2 + xy = x^3 + a_2 x + a_6 \quad (2)$$

$$y \cdot P + e \cdot Z = (ke + r) \cdot P + e \cdot (-k \cdot P) = T. \quad (3)$$

Note that, in practical application, there might be a large number of tags in the same electromagnetic (EM) fields created by a reader; thus, the reader will send commands to choose a specific tag before authentication. Some basic commands of ISO/IEC 14 443 will be adopted to select a specific tag, such as command REQA and SELECT [29]. In this paper, the process

of the Schnorr protocol can be interpreted by three commands in command level, as shown in Fig. 6. The AUTH1 and AUTH2 are commands designed for completing the Schnorr protocol, whereas the KeyTrans is the public key exchange command. Phase 0 is a preauthentication phase used for selecting a single tag for authentication. In this phase, the reader will send anti-collision commands to distinguish a specific tag from dozen of them.

In Phase 1, reader should send command AUTH1 to acquire parameter T. As the T consists of two coordinates with the length of 42 bytes, we separate the T into two parts, i.e., Tx and Ty, for reducing transmission time in one response, which means the reader ought to send twice AUTH1 to acquire the entire parameter T.

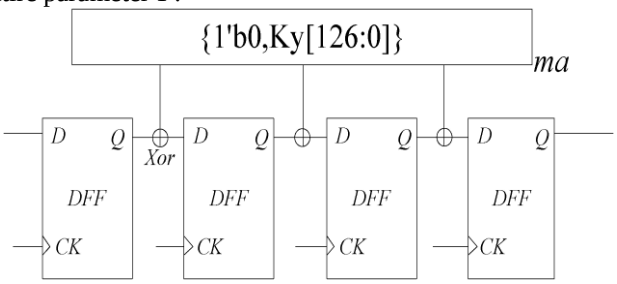

Fig. 7. LFSR-based stream encryption.

In Phase 2, the reader would send command AUTH2 and random data e. Meanwhile, the tag should launch the large integer arithmetic (LIA) calculation. After computation, the tag would return the parameter y.

Phase 3 is used for exchanging public key between reader and tag after Schnorr authentication. In Phase 3, reader and tag will obtain other's public key and execute scalar multiplication to calculate the communication key *Key*. Moreover, the public keys are transferred by two separated parts as the process of parameter T.

As previously presented, the tag must be capable of conducting scalar multiplication and large integer operation executed by ECP and LIA module, respectively.

IV. STREAM ENCRYPTION SCHEME

The LFSR, which has been proved to be one of the most effective methods for digital logic circuit implementation, has been applied into many fields. We propose a LFSR-based stream encryption scenario for low-cost requirement of RFID tag.

When the authentication process is over, the reader will send its public key G to tag for the further data encryption. After the tag receives the public key G, the tag will start another scalar multiplication, as shown in (4), to calculate the communication encryption and decryption key *Key*. Meanwhile, the reader will get the same Key by point multiplying its private key k with tag's public key Z. In this paper, we propose a LFSR based stream encryption scheme, which is suitable for RFID constrained resource, as shown in Fig. 7

$$Key = k \cdot G = k \cdot Z = k \cdot k \cdot P = (Kx, Ky). \quad (4)$$

We utilize the Galois LFSR model while setting the LFSR initial value with the low 128 bits of Kx. The next LFSR register value is defined by (5). In (5), *ma* is composed of the low 127 bits of Ky and the highest bit of *ma*is fixed at logic 0 so that Q_0 is equal to Q_{127} when LFSR is running

$$Qi = \begin{cases} Qi - 1^{mana\ i-1}, & 1 \le i \le 126 \\ Q127, & i = 0 \end{cases} \quad (5)$$

After running *mb* counts of clock cycles, where the *mb* is defined by (6), the LFSR completes the shifting process and the output value of LFSR register is used for data encryption and decryption

$$mb = Ky[159 : 152]. \quad (6)$$

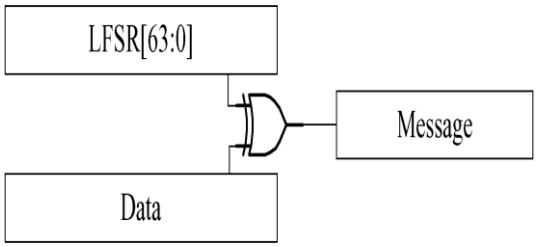

Fig. 8. Stream encryption scheme.

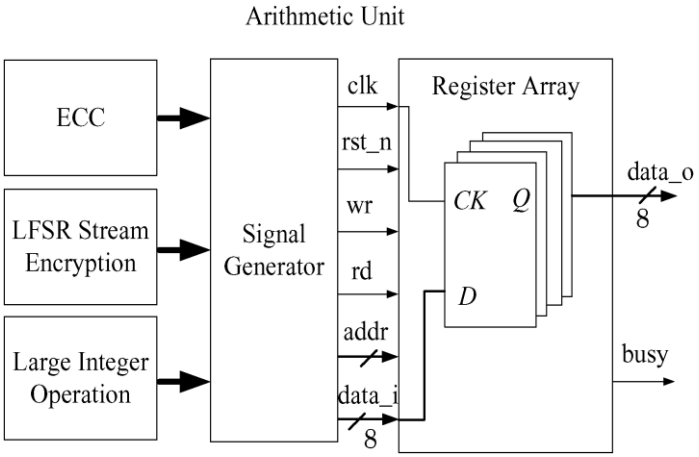

Fig. 9. Structure of register reuse.

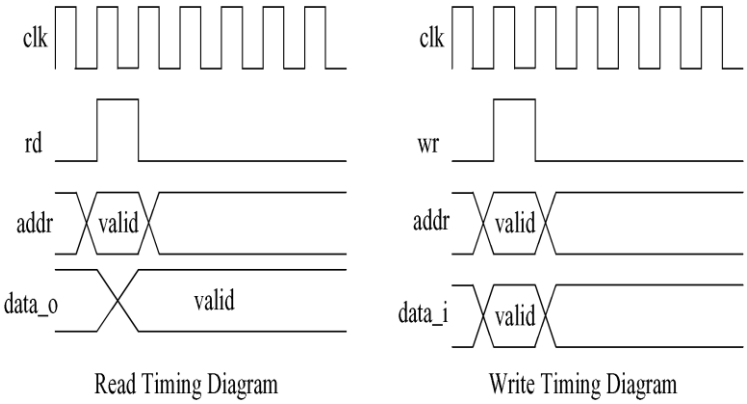

Fig. 10. Timing diagrams.

Moreover, the tag will only utilize the low 64 bits of LFSR value for data encryption and decryption so that the attacker would not obtain the entire LFSR stream out value and LFSR

coefficients, not mentioned the Key. Considering the constrained resource on RFID tag, the encryption and decryption scheme is based on exclusive-OR gate, as shown in Fig. 8. When every single round of LFSR running is over, the LFSR low 64-bit result will be kept in separated space within register array so that the system controller will utilize the LFSR stream out bit, whereas the LFSR can continue generating next new 128-bit LFSR data. Noting that we only utilize one XOR gate in encryption and decryption, the stream key scheme will not cost extra area consumption.

V. LIGHT WEIGHT SCHEMES

A. Register Reuse

Since the register array dominates the area of design, a

$$mb = Ky[159 : 152]. \quad (6)$$

scenario of register reuse for three modules is proposed. As illustrated in Fig. 9, the interfaces to register array is the same for the three different modules. When one of the three modules requires data from register array, the signal generator will manage the value of *addr* to have the accesses to the different 8-bit space of the register array. Fig. 10 shows the LIU *et al.*: DESIGN AND IMPLEMENTATION OF A DBC FOR RFID TAG CHIP 4369

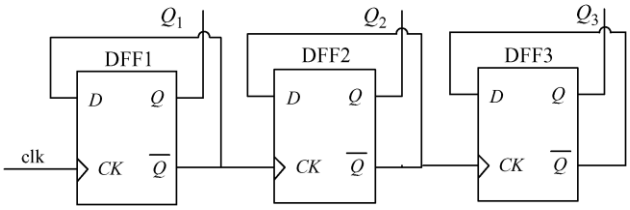

Fig. 11. Asynchronous counter architecture.

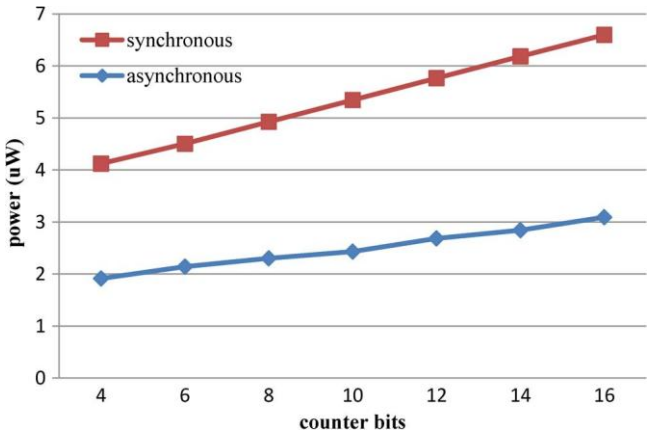

Fig. 12. Power consumption of different counter bits.

timing diagrams of reading and writing process to the register array. Therefore, in practical application, when the ECC module starts the process of scalar multiplication, it will load related data to the register array through *data_i*. After the calculation is over, the ECC module will read the calculated data through *data_o*. The large integer operation module and LFSR stream encryption have the similar approach to the register array.

B. Asynchronous Counter

In the proposed DBC, there are two divided clocks generated by one source clock. Using synchronous counter might bring in large quantity of power consumption, since every bit of registers would be trigged on each of clock edges. In order to reduce the power of counter, we adopt the asynchronous counter to replace the synchronous counter in the proposed DBC. According to a typical structure of the asynchronous counter in Fig. 11, only the first flip-flop would be trigged by *clk*. The subsequent flip-flops are trigged by the former flip-flops. Thus, the unnecessary switches in registers can be minimized as well as reducing the power. Fig. 12 shows the dynamic power consumption of different counter bits at the frequency of 13.56 MHz in UMC 0.13-μm technology. Compared with synchronous counters, asynchronous counters with more than 4-bit flip-flops structure can reduce power consumption by 50% at least. According to the synthesis results, the power consumption of DBC with synchronous counters is 17.8 μW at the frequency of 423 kHz. Meanwhile, the power consumption of DBC with asynchronous counters is 14.7 μW, as depicted in Section VII. Thus, the asynchronous counters can reduce power consumption of DBC by 17%.

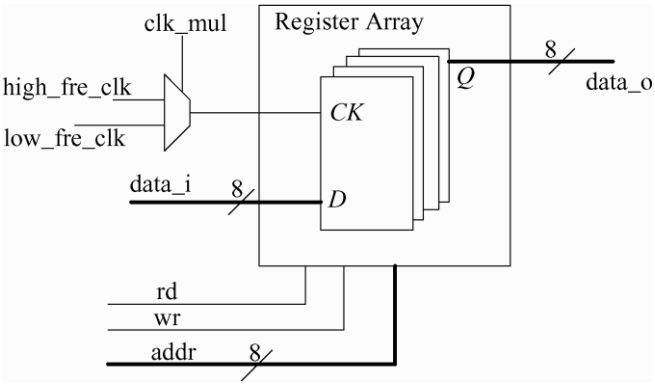

Fig. 13. Clock multiplexer.

C. Clock Multiplexer

As illustrated in Fig. 13, the *clk_mul* is controlled by FSM in DBC. When the system controller writes data or reads calculated results through bus architecture of register array, the clock frequency of register array would be low. However, when the calculation is undergoing in register array, the clock frequency should be set high by FSM. After calculating, the system controller will reset the *clk_mul* to keep the low frequency of the register array in order to control the power consumption. According to the simulation results, scalar multiplication costs about 170-K clock cycles, whereas the large integer operation costs nearly 19-K clock cycles. In this paper, the designed DBC is compatible with ISO/IEC 14443 protocol; thus, the highest frequency of the clock is 13.56 MHz, which means the time spent on scalar multiplication is only 12.52 and 1.42 ms on large integer operation. Thus, the time overhead for Schnorr protocol and data transformation between reader and tag is only

40 ms. The high computation speed makes this design meet the requirement of real time in RFID system.

D. General Clock Management

There are 3 clock domains in the proposed DBC. The source clock is 13.56 MHz coming from AFE. Among the 3 clocks, 2 clocks are generated by source clock. The main dominating clock is a 32-fractional clock generated by source clock for reducing the power of register array and the proposed DBC. In addition, we manage these three different clocks as asynchronous clocks in register transfer level design and back-end implementation. Since all clocks in this design are asynchronous, we reduce the power consumption on clock trees generated by clock tree synthesis (CTS) in placement and routing process.

VI. RESULTS AND VERIFICATION

A. Simulation and Layout Results

Fig. 14 shows a whole command proceed when tags enter into an EM field generated by the reader. The *pro_fsm* is the tag's state indicators. After the SELECT command, the tag was selected and ready for Schnorr protocol. After Schnorr protocol, the tag can respond any command encrypted by LFSR-stream

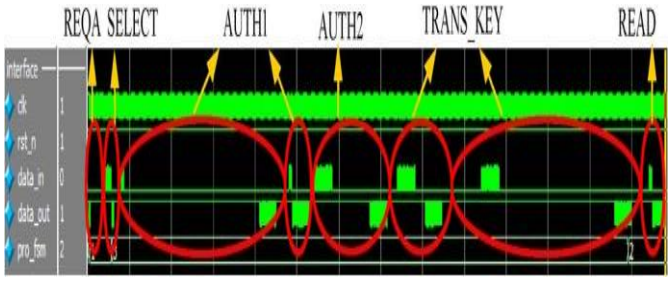

Fig. 14. Tag command precede.

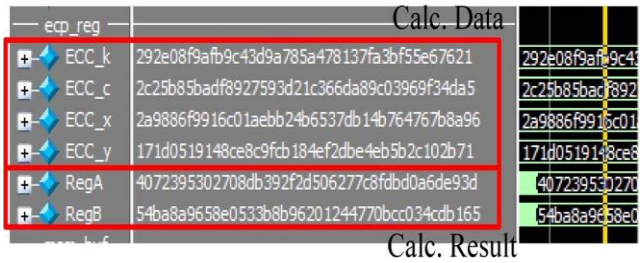

Fig. 15. Scalar multiplication simulation result.

TABLE II

SIMULATION VARIABLES IN SCALAR MULTIPLICATION

Variable	Description
ECC_k	scalar number or private key
ECC_c	EC parameter ($ECC\_c^2=a_6$)
ECC_x	x coordinate of EC base point P
ECC_y	y coordinate of EC base point P
RegA	x coordinate of point G
RegB	y coordinate of point G

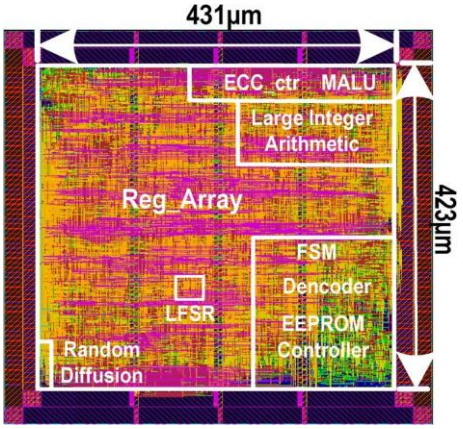

Fig. 16. Layout result of DBC.

schemes such as READ command. Fig. 15 shows a typical simulation result of scalar multiplication, $G = kP$.

Table II explains all the variables showed in Fig. 15. The calculation result of scalar multiplication are stored in *RegA* and *RegB*, which represent the x coordinate and y coordinate of point G, respectively. According to the simulation result, the time overhead of scalar multiplication is 170-K clock cycles, namely, nearly 12.5 ms at frequency of 13.56 MHz.

Our proposed DBC was implemented by UMC 0.13 μm CMOS technology. According to the synthesis result, the gate area of the DBC is 25.7 K. Fig. 16 shows the final layout. The total layout area is 0.18 mm^2 . The register array dominates the area and occupies 76% area of the DBC. Since we treat all clocks as the asynchronous clocks, the CTS will not bring in too many buffers to save area and power.

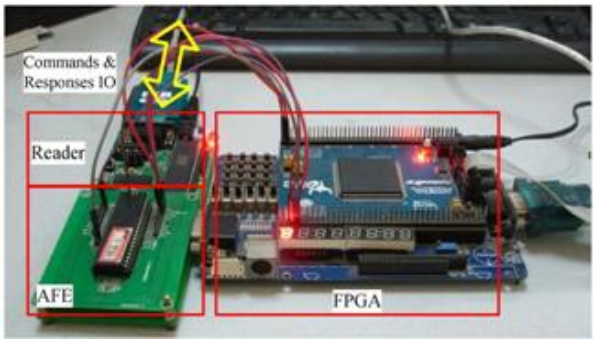

Fig. 17. FPGA test platform.

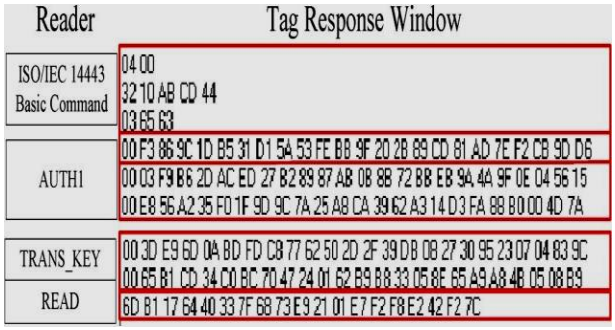

Fig. 18. FPGA command test.

TABLE III
VALUE OF LFSR AFTER THREE ROUNDS OF SHIFTING

	LFSR 64 MSBs(Hex)	LFSR 64 LSBs(Hex)
Round 0	2708db392f2d5062	77c8fdbd0a6de93d
Round 1	b55034477f78275f	f26f988b52040e2a
Round 2	8e798684b8c09bcd	35146e4df9bde207

TABLE IV
PLAINTEXT AND CORRESPONDING CIPHERTEXT IN READER AND TAG

	Plaintext and Ciphertext
Reader Req.	0730
	ee0d
Tag Resp.	ff00ff008899aabb776655448899aab00
	42e2f8f2e70121e973687f33406417b16d

B. FPGA Verification

We use Cyclone III FPGA, a product of Altera Company, to establish the test platform. As shown in Fig. 17, the reader will receive commands from PC and sends standard ISO/IEC 14 443 commands. The AFE is an independent chip providing signals for tag that replaced by FPGA. The final *data_out* signal is also transmitted to PC so that we can monitor the response data.

Fig. 18 shows several commands and responses during Schnorr protocol. The first response data of AUTH1 command is random data *r*. We make the tag response command Key- Trans with internal calculation result of *Key* so that we can monitor the LFSR-based stream encryption in better way. In addition, we add 0x00 as the first byte of tag response data for Schnorr and READ commands.

Table III shows the value of LFSR after two rounds of shift ing. The *ma* is 128'h0e0533b8b96201244770bcc034cdb 165, whereas the *mb* is 8'h4b.

Table IV shows the plaintext and ciphertext of command READ in reader and tag. Noting that the reader and tag will send 18 bytes plaintext totally, the LFSR will shift two rounds to generate the ciphertext. The results in Table IV are matched with the FPGA test results shown in Fig. 18.

TABLE V
PERFORMANCE OF ECP AT DIFFERENT FREQUENCIES

Freq.(MHz)	13.56	6.78	3.39	1.695	0.847	0.423	0.212
Power(μW)	208.4	107.1	54.7	28.3	15.2	8.75	5.51
Times(ms)	12.5	25.1	50.3	100.6	201.3	403.5	808.6
Energy(μJ)	2.61	2.69	2.75	2.85	3.06	3.53	4.45

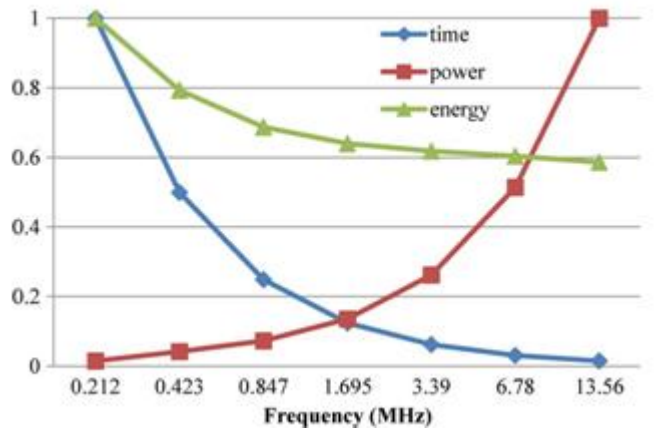

Fig. 19. Normalized time, power, and energy for one scalar multiplica- tion under different work frequencies.

VII. PERFORMANCE AND COMPARISON

Table V lists the performance of ECP at different frequencies. According to Table V, the power consumption is depended on the frequency of the clock. Although the low frequency can benefit the system power, the time spent on awaiting EC scalar multiplication should be avoided in some high-speed computing occasions. Thus, we choose the 13.56 MHz as the high_fre_clk, as shown in Fig. 13. Thus, the EC scalar multiplication and large integer operation can be done in 12.5 and 1.42 ms, respectively. The Schnorr protocol can be completed in less than 15 ms. The capability of high-speed computation enhances the throughput and real time of RFID system. In order to compare the frequency power benefits, Fig. 19 shows the normalized time, power and energy for one scalar multiplication under different work frequency.

Table VI shows the comparison of the proposed ECP with related work. As shown in Table VI, the proposed ECP only needs 170 K cycles to complete one scalar multiplication. Compared with all references, the proposed ECP has an advantage in term of clock cycles. For reasonable comparison, the energy is much closer to practical circumstance, since it considers the time spent on one scalar multiplication. Many designers try to reduce the clock frequency in order to achieve low power consumption in RFID tag chips. However, our self-designed AFE chip shows that an AFE chip can supply at least 1.25 mW power. Therefore, 208.4 μW is affordable for an AFE chip. It is not necessary to reduce the frequency, as well as the calculation speed.

Although the area of the proposed ECP is larger than any other literatures, in fact, those literatures barely consider the

parameters of EC as the resource consumption and store the parameters into a memory, such as RAM or ROM. However, in this proposed ECP, we believe four extra register groups are needed to store the EC parameters, i.e., EC base point P, parameter $a6$ and private key k. ALL of them should be taken into account as a part of area consumption. That is the reason why the proposed ECP is larger than other literatures. When the base point $P = (x, y)$ and parameter $a6$ are hardwired, the gate area of ECP can be reduced to 14.6 K.

The synthesis results show the power consumption of DBC, including ECP and other units in Fig. 1, is only 14.7 μW at low frequency of 423KHz. Considering the power consumption of the proposed DBC, we have achieved the balance between speed and power.

TABLE VI

COMPARISON WITH RELATED WORK

Ref.	Freq.(MHz)	Power(μW)	Cycles	Time(ms)	Area(GE)	Energy[2](μJ)	Technology
This Work	13.56	208.4	170K	12.5	21.8K[1]	2.6	UMC 0.13μm
[13]	1.13	36.62	276K	244.43	12.5K	8.9	UMC 0.13μm
[14]	13.56	N.A.	430K	31.8	15.2K	N.A.	AMI 0.35μm
[15]	0.106	8.57	296K	2793	13.3K	23.9	UMC 0.18μm
[16]	0.4	7.3	220K	547.9	11.7K	3.9	UMC 0.13μm

1. When the base point $P = (x,y)$ and parameter a_6 are hardwired, the gate area can be reduced to 14.6K.

2. Energy for one scalar multiplication.

VIII. CONCLUSION

A compact ECC-based RFID digital baseband controller, which is compatible with the ISO/IEC 14 443 protocol, has been proposed in this paper. For adapting to the constrained resource on RFID tag chip, we utilized and proposed several techniques or architectures based on practical application, such as the register reuse, clock multiplexer, and asynchronous counters. By reusing register array, the tag can utilize the register array for scalar multiplication, receiving data, large integer operation, and LFSR operations with the same interfaces of register array. The clock multiplexer leads to a balance between real-time requirement on RFID system and low power consumption. The asynchronous counters reduce the dynamic power of registers. Finally, the proposed LFSR-based stream key scenario provides the security for data transformation in the air.

We synthesized the presented ECC-based DBC in UMC 0.13 μm CMOS technology and give the final layout result. The core unit in this work is the ECP unit with 21.8 K gates area. Compared with previous works, the proposed ECP has great advantage in energy consumption and calculation speed. According to Section VII, the proposed ECP only needs 8.75 μW at frequency of 423 kHz. Even if the ECP works at the frequency of 13.56 MHz, the ECP unit only needs 208.4 μW, which is affordable for an AFE chip. The synthesis result shows the total gate area of DBC in Fig. 1 is 25.7 K, and the final layout area is 0.18 mm^2 . Clock multiplexer in Fig. 13 switches the frequency of flip-flops. At low frequency of 423 kHz, the whole units in DBC is only 14.7 μW. When the design needs the high- speed calculation such as the process of Schnorr protocol, the high frequency of 13.56 MHz is available. In addition, since time overhead for an entire Schnorr authentication process is less than 15 ms at high frequency, the real-time requirement of RFID system is satisfied. Above all, there is no problem of applying the ECC-based DBC into a practical RFID tag chip.

REFERENCES

[1] S. K. Kuo, S. L. Chen, and C.-T. Lin, "Design and development of RFID label for steel coil," *IEEE Trans. Ind. Electron.*, vol. 57, no. 6, pp. 2180–2186, Jun. 2010.

[2] E. DiGiampaolo and F. Martinelli, "A passive UHF-RFID system for the localization of an indoor autonomous vehicle," *IEEE Trans. Ind. Electron.*, vol. 59, no. 10, pp. 3961–3970, Oct. 2012.

[3] E. DiGiampaolo and F. Martinelli, "Mobile robot localization using the phase of passive UHF-RFID signals," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 365–376, Jan. 2014.

[4] V. Fiore *et al.*, "30.4 A 13.56 MHz RFID tag with active envelope detection in an organic complementary TFT technology," in *ISSCC Dig. Tech.Papers*, Feb. 9–13, 2014, pp. 492–493.

[5] G. Cai, A. Pun, D. Kwong, and D. Kwong, "A 2.4 pJ/bit ASK demodulator with 100% modulation rate for 13.56 MHz NFC/RFID applications," in *Proc. IEEE ISCAS*, Jun. 1–5, 2014, pp. 734–737.

[6] Y. H. Chen *et al.*, "A novel anti-collision algorithm in RFID systems for identifying passive tags," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 105–121, Feb. 2010.

[7] M. He *et al.*, "A fast RFID tag identification algorithm based on counter and stack," *Expert Syst. Appl.*, vol. 38, no. 62, pp. 6829–6838, Jun. 2011.

[8] Y.-J. Huang, W.-J. Wei, and H.-L. Li, "Efficient implementation of RFID mutual authentication protocol," *IEEE Trans. Ind. Electron.*, vol. 59, no. 12, pp. 1573–1582, Dec. 2012.

[9] J. W. Lee, Q. H. Huynh, D. H. T. Vo, and S. H. Hong, "A fully integrated HF-band passive RFID tag IC using 0.18- μm CMOS technology for lowcost security applications," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2531–2540, Jun. 2011.

[10] D. Wang, J. Hu, and H. Z. Tan, "A highly stable and reliable 13.56 MHz RFID tag IC for contactless payment," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 545–554, Jan. 2015.

[11] G. D. Sutter, J. Deschamps, and J. L. Imaña, "Efficient elliptic curve point multiplication using digit-serial binary field operations," *IEEE Trans. Ind.Electron.*, vol. 60, no. 1, pp. 217–225, Jan. 2013.

[12] G. D. Sutter, J.-P. Deschamps, and J. L. Imaña, "Modular Multiplication and exponentiation architectures for fast RSA cryptosystem based on digit serial computation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 7, pp. 3101–3109, Jul. 2011.

[13] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-curvebased security processor for RFID," *IEEE Trans.Comput.*, vol. 57, no. 11, pp. 1514–1527, Nov. 2008.

[14] S. Kumar and C. Paar, "Are standards compliant elliptic curve cryptosystems feasible on RFID?" in *Proc. Workshop RFIDSec*, 2006, pp. 12–14.

[15] D. Hein, J. Wolkerstorfer, and N. Felber, "ECC is ready for RFID-A proof in silicon," in *Proc. Workshop SAC*, 2009, pp. 401–413.

[16] U. Kocabas, J. Fan, and I. Verbauwhede, "Implementation of binary Edwards curves for very-constrained devices," in *Proc. 21st Int. Conf. ASAP*, 2010, pp. 185–191.

[17] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in *Proc. Adv. Cryptology*, 1989, pp. 239–252.

[18] S. K. Islam and G. P. Biswas, "A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *J. Syst. Softw.*, vol. 84, no. 11, pp. 1892–1898, Nov. 2011.

[19] G. Gódor, N. Giczi, and S. Imre, "Elliptic curve cryptography based mutual authentication protocol for low computational capacity RFID systems-performance analysis by simulations," in *Proc. IEEE Int. Conf. WCNIS*, Jun. 25–27, 2010, pp. 650–657.

[20] M. Hutter, M. Feldhofer, and T. Plos, "An ECDSA processor for RFID authentication," in *Proc. Workshop RFID Security*, Jun. 8/9, 2010, pp. 189–202.

[21] P. Babaheidarian, M. Delavar, and J. Mohajeri, "On the security of an ECC based RFID authentication protocol," in *Proc. IEEE Int. ISCISC*, Sep. 2012, pp. 111–114.
[22] X. C. Zou *et al.*, "Design and implementation of an analog front-end circuit for semi-passive HF RFID tag," in *Proc. IEEE Radio Freq. Integr Circuits Symp.*, Jun. 1–3, 2014, pp. 389–392.
[23] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 2010.
[24] P. Montgomery, "Speeding the pollard and elliptic curve methods of factorization,"*Math. Comput.*, vol. 48, no. 177, pp. 243–264, 1987.
[25] J. López and R. Dahab, "Fast multiplication on elliptic curves over *GF*(2*m*) without precomputation," in *Proc. Workshop CHES*, 1999, vol. 1717, pp. 316–327.
[26] D. M. Burton, *Elementary Number Theory*. New York, NY, USA: McGraw-Hill, 2006.
[27] P. L. Montgomery, "Modular multiplication without trial division," *Math.Comput.*, vol. 44, no. 170, pp. 519–521, Apr. 1985.
[28] P. Barrett, "Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor," in *Proc. Adv. Cryptolog*, Aug. 1987, vol. 265, pp. 311–323.
[29] Identification Cards—Contactless Integrated Circuit(s) Cards—Proximity Cards—Part 3: Initialization and Anticollision, Jun. 1999, ISO/IEC 14443-3.