# RTL Design and Testing Methodology for UHF RFID Passive Tag Baseband-Processor

## Enabling Internet-of-Things (IoT) Technology

Syifaul Fuada[1]
Program Studi Sistem Telekomunikasi
Universitas Pendidikan Indonesia
Bandung, Indonesia

Trio Adiono[3]
Electrical Engineering Department
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Bandung, Indonesia

Aris Agung Pribadi[2]
University Center of Excellence on Microelectronics Insitutut
Teknologi Bandung, Bandung, Indonesia

Tengku Ahmad Madya[4]
Xirka Silicon Technology Ltd,
Bandung, Indonesia

***Abstract*—With the rapid growth and widespread implementation of Internet-of-Things (IoT) technology, Radio Frequency Identification (RFID) has become a vital supporting technology to enable it. Various researchers have studied the design of digital or analog blocks for RFID readers. However, most of these works did not provide a comprehensive design methodology. Hence, the motivation of this study is to full fill the research gap. This paper proposes a comprehensive design and testing methodology for the Ultrahigh Frequency (UHF) RFID passive tag baseband processor at the register transfer (RTL). A complete design procedure of each block from state diagram to schematic level is presented; it comprises several blocks, *i.e.*, transmitter, receiver, Cyclic Redundancy Check (CRC), command processing, and Pseudorandom Number Generator (PRNG). Each block produces low latency (<400 ns). Two CRCs were applied to this system for different purpose: CRC-5 and CRC-16. To perform multi-parameter combinations of as many as 1344 combinations (including timing parameter, query respond, state transition, and BLF), a Universal Verification Methodology (UVM)-based test is conducted. The simulation results reveal that the proposed RFID baseband processor passes all the testing scenarios using UVM (version 1.1d). Moreover, we also implemented the proposed design on the FPGA board (ALTERA DE2-115). The system consumes 976 logic elements and 173.14 mW of total power dissipation (*i.e.*, 0.13 mW of dynamic power dissipation, 98.6 mW of static power dissipation, and 74.34 mW of I/O dissipation), which is reasonably low. This demonstrates that our design is synthesizable and ready to be processed further. All system design and test criteria were conducted following the EPC Gen-2 standard. The developed chip can be a solution for various kinds of RFID chip-based IoT applications.**

***Keywords*—UHF RFID passive tag; baseband processor; register transfer level; universal verification methodology; Internet-of-things enabler; FPGA**

## I. Introduction

Within the last few decades, automatic identification system has become an essential part of industrial applications, such as logistics, retails, and manufacturing [1]–[4]. Now, with the rise of the Internet of Things (IoT), the use of Radio Frequency Identification (RFID) technology has a crucial role [5]. The RFID has been applied in various fields, *e.g.*, smart agriculture, smart homes, health care, medicines, transportation sector, payment, environment monitoring, disaster warning, or even telepresence for distant objects monitoring [6]–[13]. RFID is a communications technology that uses electromagnetic fields to detect tags attached to objects wirelessly. RFID technology is preferred over other identification or wireless communication technologies due to its various advantages, including bigger information storage capacity, more comprehensive coverage, better security system, more affordable price, and a universal standard. RFID is usually designed under the EPC Gen-2 standard, then adopted into the ISO 18000-6C standard. Even though the typical reading distance of an RFID tag is relatively long, about 6 meters, most of the tags are passive, which means that they do not have a self-powered source [14]. Instead, it solely relies on a converted power obtained from the RF signal sent by the reader. The analog circuit part is responsible for this power conversion or generation. The tag will then use the generated power, especially the baseband processor, to run the whole system. The RFID tag design has been finalized and matures in the industry for many years. With the detailed technical description in the Gen-2 tag standard, it is effortless to come up with a Register-Transfer Level (RTL) design. However, further observation and exploration are still an exciting topic by many researchers worldwide to fulfill numerous requirements of the recent issues and use cases [15]–[18], such as IoT applications based on-chip technology.

Flourishing along with the emergence of IoT technology, various research projects have been conducted related to RFID tags or readers, including the analog and digital circuit building blocks. The baseband processor is part of the digital block. Most research on UHF RFID baseband processors explores low power/energy-efficient design. For example, Wei et al. [19] proposed a low-power baseband processor using several techniques, including low operating frequency clock, clock gating, and asynchronous design. Using these

techniques, they can achieve a low power consumption baseband processor as low as 2.7 µW. Similarly, Lee et al. [20] also proposed a low-power baseband processor. To achieve a low power consumption, they use numerous techniques, including the lowest possible clock frequency, latch-based clock gating, variable clock frequencies, and resource combining and sharing between blocks. Their design achieves a rather high-power consumption, 29.2 µW. However, this is due to their circuit also including the analog part. Despite their advanced circuit design, they did not provide a comprehensive design and testing methodology for the baseband processor. Ismail et al. [21] presented a comprehensive baseband processor design methodology, including the Finite State Machine (FSM). However, they did not describe the command processor design and the system test. Moreover, the testing only showed the signal encoding and decoding test results. There is no other parameter for the tests was shown. Indeed, to perform many test parameters, an additional test technique or tool is needed so that a large number of test combinations can be efficiently performed. Su et al. [22] proposed a novel automatic verification strategy for a UHF RFID baseband processor. Using their verification, they could perform a set of 2700 commands testing in total. In detail, the verification or test included several command responses tests, e.g., Ready, Arbitrate, Reply, Acknowledge, and random command test. Li et al. proposed an RFID tag IC that is fabricated on 0.13 µm CMOS technology. They achieved low power consumption: 4.8 µW for reading operations and 11.5 µW for writing operations [23]. Bhanushali, et al., successfully realized a digital UHF RFID Tag IC compatible with EPC Gen2 standard applied on 55 nm CMOS technology [16]. However, in their work, no comprehensive design methodology was provided. Design, implementation, verification methodology, and testing techniques is one set discussion that essential to be reported [24]; this will gain benefit, such as it can provide insights to early the designers/engineers who want to follow the proposed design.

This work presents a comprehensive and systematic RTL design and testing methodology specifically for UHF RFID passive tag baseband processors to fill the research gap. The baseband processor design includes receiver, transmitter, and command processing block. Before designing the schematic, the state diagram of each block is investigated to ensure all possible states are included. Then, each block and the design of the integrated block is tested based on the EPC Gen-2 standard to verify and ensure the functionality of the design. The latency of each block is also calculated. For the testing, numerous tests are performed, including query command, timing parameter, state transition, and Backscattering Link Frequency (BLF) test. To conduct the tests efficiently, a Universal Verification Methodology (UVM) is used as the test methodology. Using this UVM, a set of tests, as many as 1344 test combinations, was conducted, and thus, a rigorous parameter test was able to be performed. Additionally, the RTL design is also implemented on the FPGA development board to show the practicability of the design for future real chip implementation. Having this comprehensive RTL design and methodology, a more reliable baseband processor design can be realized. The developed chip can further be a strong candidate for IoT applications.

Our main contribution is providing a comprehensive RTL design and testing methodology. It includes each block's state diagram and schematic design, comprising a receiver, transmitter, command processing, Pseudorandom Number Generator (PRNG), Cyclic Redundancy Check (CRC), and their corresponding functional tests. In addition, the integrated system test is performed using UVM to perform the tests efficiently. The rest of this paper is organized as follows. To provide a short technical background and point out the distinct contributions, a description of related works is elaborated in Section I. The details of system design, starting from each block to its integration, are provided in Section II. Section III shows the testing results of the design, including an integrated system test using UVM, the implementation design, and the test of the baseband processor RTL design on the FPGA board. Lastly, Section IV provides the conclusion of this work.

## II. METHODOLOGY

The structure of the proposed RFID tag system is depicted in Fig. 1. The digital block has three inputs (i.e., data, clock signal, and voltage source) and one output (backscattering data). The digital block is the baseband processor itself; it comprises five blocks: receiver (PIE decoder and data buffer), command processor, transmitter (Miller encoder, FM0 encoder, and frame generator), CRC (CRC-5 and CRC-16), BLF generator, and PRNG block. Each block will be described in detail as follows.

### A. Receiver Block

The receiver block comprises three main components, i.e., Pulse Interval Encoding (PIE) decoder, data buffer or command parser, and BLF generator. In Fig. 2, the PIE decoder and data buffer are combined and represented as a “logic block.” When a data signal comes from an RF demodulator, the data will be processed and interpreted by the PIE decoder by using Counter and Logic blocks. Those interpreted data are then stored in the data buffer and transmitted to command processing. The command parser will then separate the reader commands stored in Memory. Besides storing the data, the data buffer will also separate the query command to obtain Miller index (M) and Divide Ratio (DR) parameters. These two parameters are essential for the RFID system since M is used to determine which encoder to be used; either FM0 or DR is used to determine the value of BLF [25].

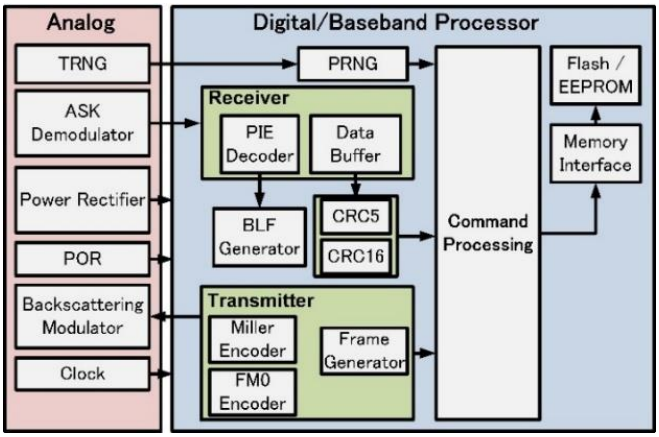



Fig. 1. Block Diagram of RFID Tag System.

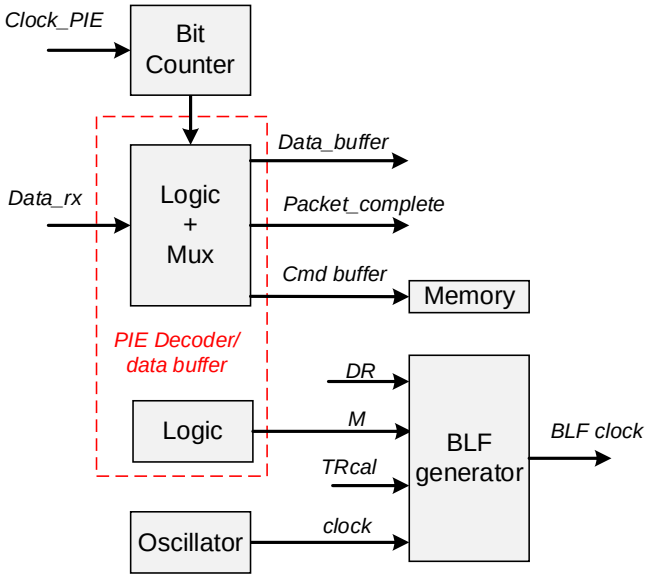



Fig. 2. Schematic of the Receiver Block.

Before designing the PIE decoder, the FSM should be determined first. PIE decoder consists of two state machines: PIE counter and data packet structure states. The PIE counter state machine detects the symbol of the bit logic state, either high or low, and then gives a corresponding instruction to the counter. As shown in Fig. 3(a), after the transition from logic low to logic high, the counter counts a full symbol until the end. Afterward, it begins to reset the counter when a new symbol comes. Meanwhile, the data packet structure state machine extracts the query parameters. The data packet structure sent by the reader includes delimiter, data-0, Reader-to-tag calibration (RTcal), Tag-to-reader calibration (TRcal), and data. In detail, the delimiter is a timing parameter with a constant value of 12.5 µs, ensuring that the tag can receive data from the reader even at the longest possible data-0. Data-0 is data that contains all binary 0. RTcal is a sum of the duration of symbols one and zero, used to determine the bit rate of the reader's transmitter. TRcal is a timing parameter used along with the DR parameter to determine the value of BLF. However, not every command has a TRcal. The TRcal will exit only if the symbol value is less than RTcal. This data packet structure state machine is illustrated in Fig. 3(b).

To decode the PIE symbol correctly, a 16-bit counter is used; this counter calculates the length of the PIE symbol. Afterward, the result is then compared to the calibration value in the preamble. According to the EPC Gen-2 standard, the counter requires at least 324 kHz of sampling clock since the minimum interval length between "0," and "1" symbols is 3.1 µs. In our design, the counter accommodates a clock input up to 16 MHz without causing the overflow to the longest possible TRcal symbol. The schematic structure of the PIE decoder is shown in Fig. 4. Data buffer and BLF generator are represented as "logic blocks." The logic and mux block are responsible for separating a command from its accompanying parameters, whereas another logic block is to extract DR and M values. These DR, M, and TRcal values are then used to determine the BLF value (Eq. 1),

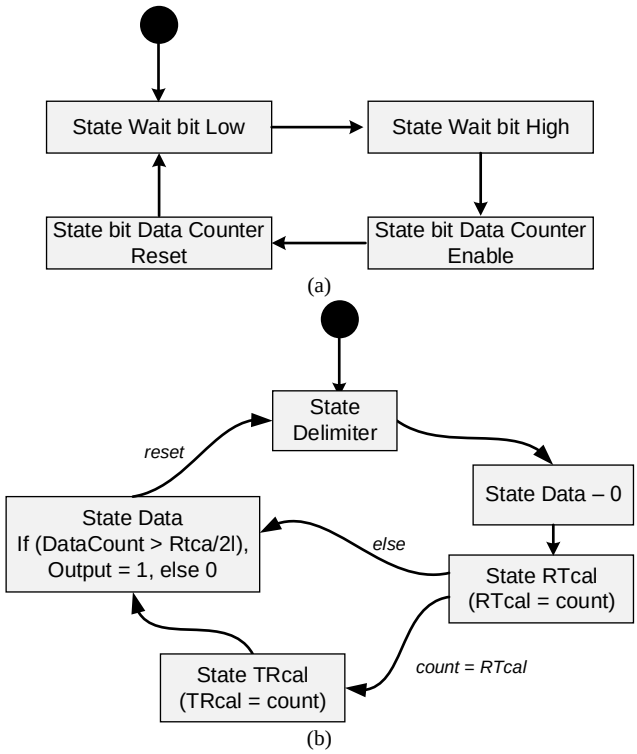



Fig. 3. (a) State Diagram of PIE Counter; (b) State Diagram of PIE Data Packet Structure.

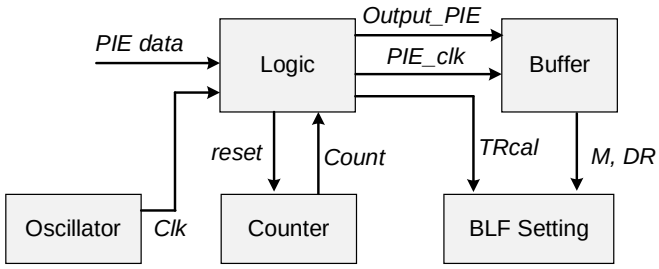



Fig. 4. Schematic Design of PIE Decoder.

$$BLF = \frac{DR}{TRcal} \quad (1)$$

The DR parameter has a value of either 8 or 64/3, depending on the parameter set in the query command. TRcal is calculated by the tag using oscillator frequency clock (f0) as a counter, and thus written as in Eq. 2, where TRcount defines the TRcal counts obtained from the receiver counter,

$$TRcount = TRcal \times f0 \quad (2)$$

### B. Transmitter Block

The transmitter block (Fig. 5) comprises five inputs from the command processing block: (1) data input (PC_EPC); (2) M carrier signal, which is used to determine the encoding type; (3) TRext for deciding the use of pilot tone in the preamble; (4) CRC; and (5) Tx_enable signal. These inputs are processed in the frame generator, which is responsible for choosing the encoder type and inserting preamble and termination signals. In this study, the design of the transmitter block is focused on FM0 and Miller Encoder only.

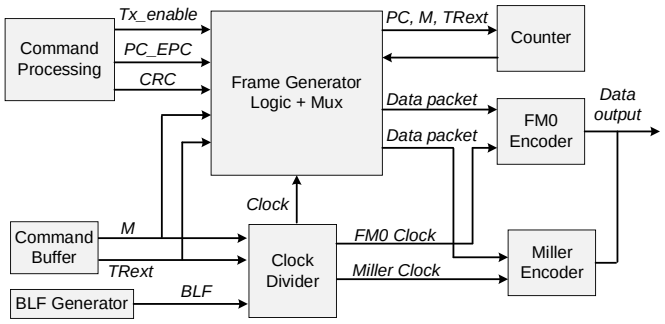
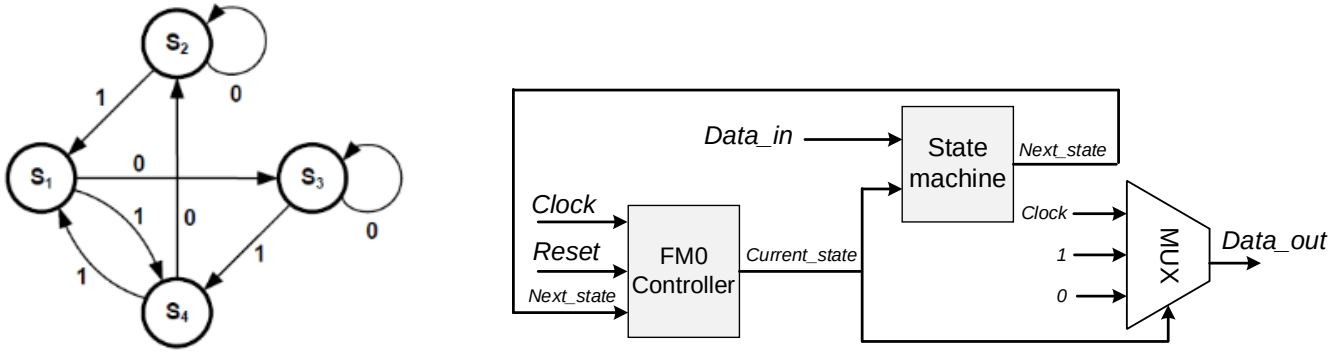



Fig. 5. Schematic of the Transmitter Block.

The default operation mode for data transmission from the tag uses FM0 Encoding. In transmitting the data, the tag changes the backscatter status in each symbol edge. Based on EPC Gen-2 standard, data transmission must be terminated with a dummy binary symbol "1" followed by a "low" state. The change of FM0 binary value determines the next sequential binary value, and therefore there are four states in total, as depicted in Fig. 6(a). The schematic design of this FM0 Encoding using the state machine model in Verilog is shown in Fig. 6(b). A Miller Modulated Signal (MMS) encoding is also added to the transmitted signal to avoid interference noise and be more flexible in managing the data rate. Any FM0 encoded signal will be encoded with an MMS subcarrier signal with a particular M value. The M value represents how many clock cycles MMS encodes the FM0 signal. The higher the M value, the slower the data rate. In the Miller encoder, the binary symbol "1" (a state transition) is inserted in the middle of the symbol. In contrast, the binary symbol "0" (no state transition) is in the middle of the symbol.

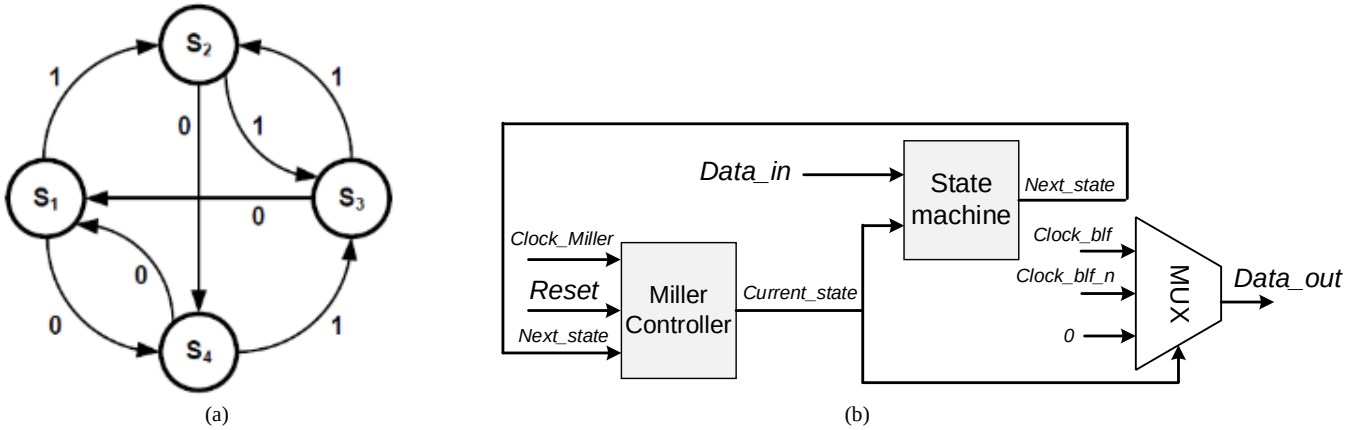
Moreover, there is no state transition in symbol transition, except when the binary symbols "0" meet consecutively. Furthermore, the encoding results are multiplied by the *M* value, where *M* is either 2, 4, or 8. The state diagram for this Miller encoding is represented in Fig. 7(a), while the Verilog schematic design is shown in Fig. 7(b).

The command processing is a block responsible for controlling the transition between EPC Gen-2 states and determining the tag response according to the reader commands. The command processor has two crucial logic functions to manage: slot logic as the anti-collision protocol and session logic. The schematic of the command processor block is depicted in Fig. 8(a). The state diagram for the command processing block is depicted in Fig. 8(b). The RFID transponder tag consists of seven states: Ready, Arbitrate, Reply, Acknowledged, Open, Secured, and Killed. When the tag is powered, it stands by in the Ready state until the Query command is received. The tag will then enter one of the two following states: 1) The tag changes to the slot state if a suitable session exists, or 2) The tag changes to the Arbitrate state if the slot value does not equal zero.




(a) (b)

Fig. 6. (a) State diagram of FM0 encoding [26]; (b) Schematic design of FM0 encoder




(a) (b)

Fig. 7. (a) State Diagram of Miller Encoding [26]; (b) Schematic Design of Miller Encoder.

The anti-collision protocol is applied within the Arbitrate state, and the tag's state will be held until the slot value equals zero. At this hold state, there are three command query options for the reader: (1) reduce slot number using QueryRep command; (2) change slot masking using QueryAdj, or (3) repeat the Query command. Once the slot has zero value, the tag will enter the Reply state and send a 16-bit random number (RN16). Afterward, the tag will wait for a response from the reader and enter one of the following states: 1) return to the Ready state if the received command query is an invalid session, or 2) return to the Arbitrate state if the received command query matches the valid session. If the reader correctly receives RN16, the reader will send an acknowledgment signal along with the previously received RN16. If the number matches the sent number, the tag will send the EPC value to the reader and change the session state from A to B or vice versa. Otherwise, the tag will return to the Arbitrate state. Additional blocks, i.e., CRC and PRNG, are required to make the system works correctly and adequately. For the CRC block, we employed two CRCs as suggested by the EPC Gen-2 standard: CRC-16 and CRC-5.

CRC-5 is used when inventory mode is started or when the reader gives the Query command. CRC-16 is used when the reader sends command select, Req-RN, NAK, Read, Kill, and Lock. The CRC-5 has a specification in the form of polynomial $x^5 + x^3 + 1$ [27], residue 000002, Preset 010012, and length 5 bits. The CRC-5 is implemented using a register flip-flop and XOR circuit in our design.

In this design, the value of preset 010012 is inserted into the register flip flop. Input data will then enter one by one according to the clock signal. CRC-5 check will return pass/successful if the value in the register output is 000002. On the other hand, CRC-16 has a specification in polynomial $x^{16} + x^{12} + x^5 + 1$ [28], residue 1D0Fh, Preset FFFFh, and length 16 bits. The same with the CRC-5 implementation, the CRC-16 uses a register flip flop and XOR circuit. The difference is that CRC-16 has more flip-flops compared to CRC-5. In this design, in the beginning, the FFFFh preset value is inserted into the register flip flop. Input data will then enter one by one according to the clock signal. The CRC-16 check will return pass/successful if the value in the register output is 1D0Fh.

### C. Pseudo Random Generator Number (PRNG)

The PRNG block commonly generates a 16-bit random value. This random value will determine the slot value in the tag according to the Q value received by the tag. The most recent advancement of a hardware implementation for a security system employs the physical variation in an integrated circuit; it can be called Physically Unclonable Functions (PUF), as presented in [29], [30]. However, the system becomes complicated, so it is not preferred for RFID applications. Instead of PUF, this study employs the Linear Feedback Shift Register (LFSR) for the PRNG block to obtain a uniform output distribution. LFSR uses registers or flip-flops [31] whose input is based on linear functions from the previous state, which is implemented using the XOR function. The LFSR uses polynomial 1 + x11 + x13 + x14 + x16. This polynomial selection is based on the study by [32].

### D. Baseband Processor

The overview of the baseband processor is depicted in Fig. 9. It works through the following steps: (1) the receiver sends input to command processing in the form of decoded data buffer along with a package complete flag as an indicator that input data has been received; (2) at the same time, the receiver also sends BLF and query parameters to the transmitter; (3) command processing then sends the reply data to the transmitter, transmitting the data back to the reader.

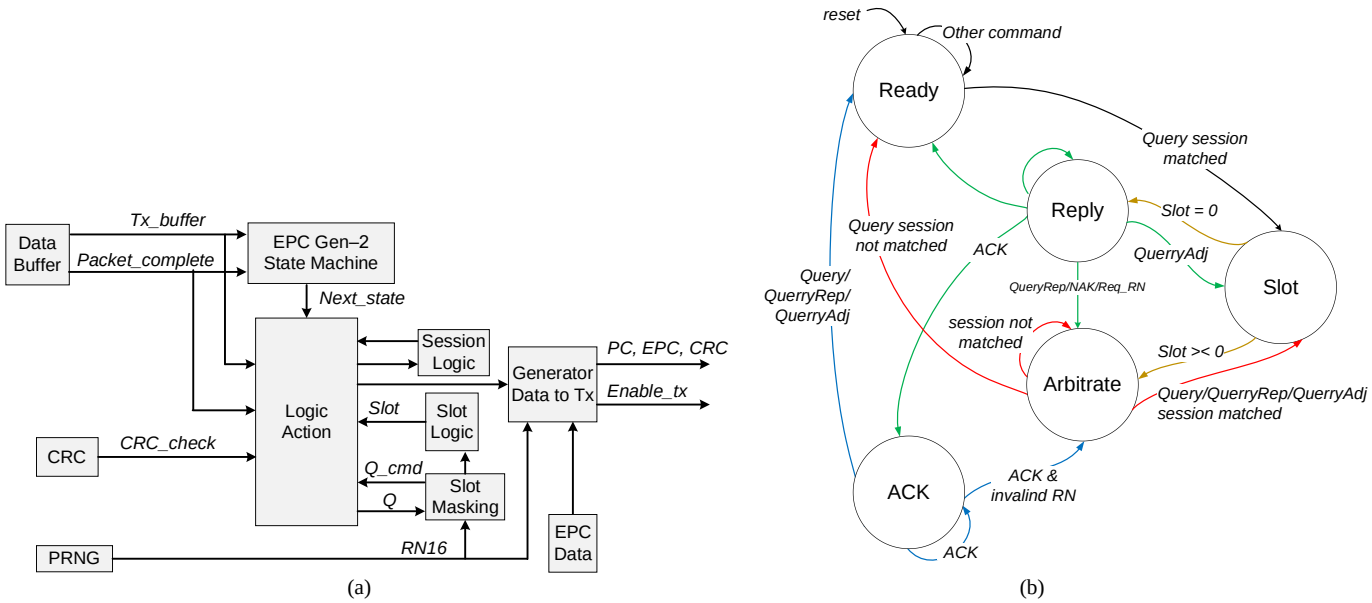



Fig. 8. (a) Schematic of Command Processing; (b) State Diagram of Command Processing Block.

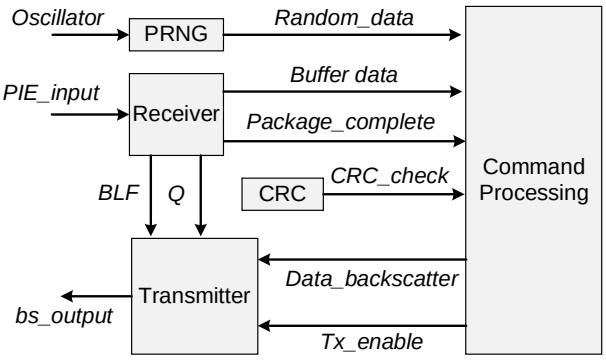



Fig. 9. Schematic of UHF RFID Tag baseband Processor.

## III. Result and Discussion

In this section, we evaluate our design by testing each block that contains the receiver, transmitter, and command processing block. Also, we verify the entire system by implementing the UVM. In this test, we use a reader that fulfills the ISO8000-6C standard and evaluates our proposed design by confirming the RFID reader and tag data. The test was conducted on an RTL simulation.

### A. Receiver Block

We first tested the PIE decoder block on the receiver side. The central controller is the PIE state machine. When the enable counter is active at the delimiter state, the state machine will check whether the delimiter value is 12.5 µs or not. Then, there are two possible states: data-0 and RTcal. Both are used as a comparison to determine the logic value "1" or "0". Next is TRcal state, which is used to obtain the TRcal value. Finally, this TRcal will be used as a parameter to produce the desired BLF value. After the PIE block has been tested, the whole receiver block (including the PIE encoding, command buffer, and BLF generator) is evaluated. The value of PIE_output, originating from the PIE decoder, is stored in the data buffer module. It is then transmitted to command processing and the packet_complete signal, indicating the packet's end. The data buffer module will also perform a command filter/parser. The command filer/parser will extract the query parameter used by the BLF generator module. The BLF module will only work to reduce dynamic power consumption if it receives a signal from a transmitter. The optimum throughput speed can be obtained using this receiver architecture, which only produces the latency of two-oscillator clock cycles. In our design, an internal 8 MHz oscillator is used, which causes a latency of 250 ns. However, this latency is insignificant compared to the whole system, and it can be neglected.

### B. Transmitter Block

The FM0 and Miller encoder blocks are tested well on an RTL simulation. When the value of enable_FM0 is "1", the FM0 encoder module becomes active. The input data from the frame generator will enter the state machine, and the next state will be determined. The FM0 controller module will provide input to the multiplexer in the form of data_selector to select the output of fm0_data. This FM0 encoder architecture design has a latency of one cycle fm0_clock that depends on the BLF value. Hence, the latency in the FM0 module is defined as Eq. 3,

$$FM0\_latency = \frac{TRcal}{DR} \quad (3)$$

In testing the Miller encoder block, the subcarrier encoding with M = 2 is conducted. The test result shows that the working principle of the circuit is similar to the FM0 encoder module, where the data_selector output from the controller determines the Miller encoder data output. The main difference is in the clock that it uses. In the FM0 encoder module, the FM0 clock has the same value as the BLF clock, whereas, in the Miller encoder, the Miller clock value depends on the M value. For instance, the Miller encoder latency depends on the BLF and M values, as defined in Eq. 4.

$$Miller\_latency = \frac{TRcal}{DR \times M} \quad (4)$$

We also tested the whole transmitter block. Input data from command processing in PC/EPC/RN16 data and tx_enable signals are processed by the frame generator and sent serially to the encoder module. Depending on the PC's value, counters, preamble, data, and CRC will activate. In this test, the FM0 and TRext are set to "0". Thus, the value of fm0_enable becomes "1", activating the FM0 encoder module. Furthermore, clock for BLF and M value determine the clock value of each encoder module. Simulation reveals the latency of the transmitter block depends on the encoder type. The FM0 encoder has low latency. Nevertheless, when miller encoding with M = 8 is used, the latency becomes significant, as inferred in Eq. 5, where Tx_latency denotes the latency of the transmitter block and clk_osc defines the oscillator clock.

$$Tx\_latency = \left(2 \times clk\_osc + \frac{TRcal}{DR}\right) + \left(\frac{TRcal}{DR \times M}\right) \quad (5)$$

### C. Command Processing Block

The complete testing sequence of the command processing module has been verified carefully. It is tested from the query command is received until the tag sends the EPC code. When the receiver module receives a packet, it will send a signal to indicate that the packet is complete, and thus the command processing will make a response accordingly. Finally, the CRC module checks whether the command query has a valid CRC value or not. The latency of this command processing is three oscillator clock cycles. 16 MHz of the internal oscillator is used in this design, which produces a latency of 375 ns; this latency is not significant.

### D. Full System Test

The entire system is simulated and tested using UVM, specifically the Coverage Driven Verification (CDV) approach. The UVM is version 1.1d and implemented using Synopsys VCS 2014.10 on the EDA playground application. The testing diagram using UVM is illustrated in Fig 10(a). We used a reader that already fulfills the ISO 18000-6C standard requirement for the whole system test as a master. The reader has simple instructions and is intended only to identify the EPC value of the tag.

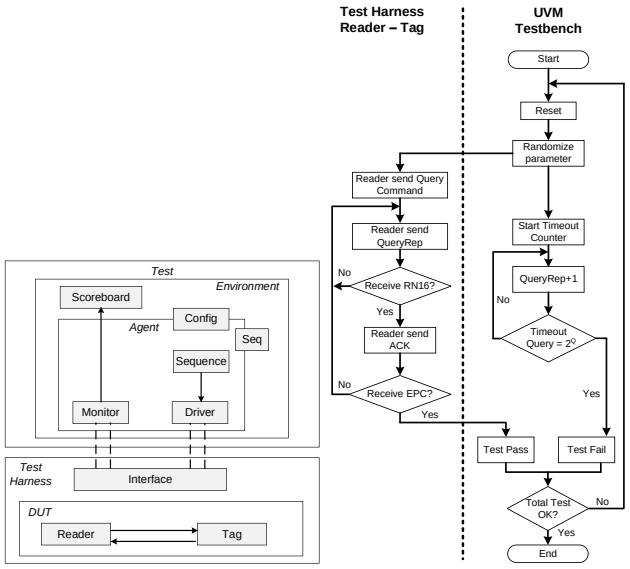



Fig. 10. (a) Full System Testing Diagram using UVM; (b) Flowchart of the System Testing using UVM.

In general, the test is conducted based on the following sequences: (1) after a reset signal is given to the reader, it sends the Query command to the tag; (2) the reader will continuously send QueryRep in order to have the tag reduces the slot value until zero and until the tag sends 16-bit random number (RN16) to the reader; (3) using "ACK" command, RN16 will be transmitted back to the tag, after which the reader will wait for EPC value from the tag. The test session is considered to be finished once the reader receives the EPC value from the tag. The flowchart of the testing sequences is illustrated in Fig. 10(b). The functional system test is intended to check both timing and query parameters to be used by the reader and the command response of the tag. Four available techniques are chosen depending on the dataset size, testing type, and specification. The techniques can obtain those parameters, i.e., constrained random test, direct test, coverage test, and edge corner test. A timeout program is also set up in the test to anticipate a situation when the tag does not send any response that makes the reader unable to obtain the EPC value. The timeout value depends on the Q value, where Q denotes the query parameter used to determine the tag's random slot, and 2Q can find the total timeout of UVM (UVM_timeout).

*1) Query parameter test:* A query command is an inventory command that the reader first sends. The tag should be able to accept and process all possible combinations of query parameters (including Q, TRext, DR, and encoding). The query parameter test is conducted using constrained random and coverage tests. The coverage tests all possible combinations to generate the parameters, which amounts to 512 combinations. While a random test is conducted to identify possible errors caused by a state transition. All these possible combinations are summarized in Table I. Based on the tests results shown in Table II, the design100% passes all the test combinations.

*2) Timing parameter test:* The timing parameter is conducted using the edge corner and constrained random tests. The edge corner test is conducted only on the smallest and biggest possible value of Tari, PW, RTcal, and TRcal parameters, resulting in 16 combinations. Meanwhile, the constrained random test is conducted by setting the parameters within certain limits, i.e., 6.25 μs to 25 μs for Tari, then 2.5Tari to 3Tari for RTcal, and 1.1RTcal to 3RTcal for TRcal. The test results are shown in Table II. It shows that the design passes all the testing combinations of the timing parameter test.

*3) State transition test:* An RFID tag design must be able to respond according to specifications based on the command it receives from a reader. The RFID tag has four states in our design: ready, reply, arbitrate, and acknowledge. On the other hand, there are six commands from the reader that the tag must recognize: query, queryrep, queryadj, select, ACK, and NAK. The command response test is conducted using direct testing. The result is summarized in Table III.

*4) BLF Test:* The BLF test is conducted using 16 MHz of frequency sampling. Based on the specification regulated in ISO 18000-6C, the deviations in each parameter must be kept below 2.5%. Based on the test results shown in Table II and Table IV, the deviations of the parameters are all below 2.5%; thus, all tests are 100% passed. Further test will follow a procedure from [33].

TABLE I. QUERY COMMAND COMBINATIONS

| | **Command** | ***DR*** | ***M*** | ***TRext*** | **Sel** | **Session** | **Target** | ***Q*** | **CRC-5** |
|---|---|---|---|---|---|---|---|---|---|
| **#of bits** | 4 | - | 2 | 1 | 2 | - | 1 | 4 | 5 |
| **Description** | 1000 | 0 :DR=8<br>1:DR=64/3 | 00:M=1<br>01:M=2<br>10:M=4<br>11:M=8 | 00: No. Pilot<br>01:Use Pilot | 00:All<br>01:All<br>10:~SL<br>11:SL | 00:M=1<br>01:M=2<br>10:M=4<br>11:M=8 | 0:A<br>1:B | 0-15 | - |

TABLE II. SUMMARY OF SYSTEM TESTING RESULTS USING UVM

| **Testing Parameter** | **Testing technique** | **#of Combination** | **Result** |
|---|---|---|---|
| Query Parameter | Coverage Test | 512 | 100% pass |
| | Random Test | 500 | 100% pass |
| Timing Parameter | Edge Corner Test | 16 | 100% pass |
| | Random Test | 300 | 100% pass |
| BLF | Direct Test | 16 | 100% pass |

TABLE III. TEST RESULT OF COMMAND RESPONSE TEST

| Command | Condition | Transition State | | | | Result |
|---|---|---|---|---|---|---|
| | | Ready | Arbitrate | Reply | Acknowledge | |
| *Query* | Slot=0 | Reply | Reply | Reply | Reply | Pass |
| | Slot><0 | Arbitrate | Arbitrate | Arbitrate | Arbitrate | Pass |
| *QueryRep* | Slot=0 | Ready | Reply | Arbitrate | Ready | Pass |
| | Slot><0 | Ready | Arbitrate | Arbitrate | Ready | Pass |
| *QueryAdj* | Slot=0 | Ready | Reply | Reply | Ready | Pass |
| | Slot><0 | Ready | Arbitrate | Arbitrate | Ready | Pass |
| *ACK* | - | Ready | Arbitrate | Acknowledge | Acknowledge | Pass |
| *select* | - | Ready | Arbitrate | Arbitrate | Ready | Pass |
| *NAK* | - | Ready | Arbitrate | Arbitrate | Arbitrate | Pass |

TABLE IV. BLF PARAMETER TEST RESULTS

| *DR* | *TRcal* (µs) | Expected BLF (kHz) | BLF Result (kHz) | Variance % |
|---|---|---|---|---|
| 64/3 | 33.3 | 640 | 640 | 0 |
| | 44.4 | 480 | 484.84 | 1.008 |
| | 66.7 | 320 | 320 | 0 |
| | 74.1 | 288 | 290.9 | 1.007 |
| | 83.3 | 256 | 258.06 | 0.805 |
| | 102.6 | 208 | 210.52 | 1.212 |
| | 159.8 | 133.5 | 134.45 | 0.712 |
| | 211.2 | 101 | 101.26 | 0.257 |
| 8 | 17.2 | 465 | 457.14 | -1.690 |
| | 25 | 320 | 320 | 0 |
| | 27.8 | 288 | 285.71 | -0.795 |
| | 31.25 | 256 | 253.96 | -0.797 |
| | 38.46 | 208 | 207.79 | -0.1009 |
| | 50 | 160 | 160 | 0 |
| | 59.9 | 133.5 | 133 | -0.374 |
| | 200 | 40 | 40 | 0 |

### *A. Implementation*

The baseband processor is implemented on the Altera DE2-115 FPGA development board. The board uses Cyclone EP4CE115 chip, which has 114,480 logic elements, 3,9 Mbits RAM, 266 multipliers [34], and a Nios II soft-processor. Moreover, the board is also equipped with several interfaces such as Ethernet, RS232, PS2, and USB. The implementation diagram of the baseband processor and the reader in the DE2-115 board is depicted in Fig. 11. The implemented system comprises two blocks: the processing system and programmable logic. The processing system comprises a Nios II soft-processor and memory-on-chip, while the programmable logic comprises the RFID tag baseband processor and the reader. The programmable logic is connected to the Avalon interface through the PIO. The Nios II soft-processor sends input in the form of EPC data to the tag and timing and query parameters to the reader. It will monitor the outputs in the form of protocol status, EPC data, and *TRcal*. The result is displayed on a host computer connected to Nios II through the UART JTAG interface. The RTL design synthesis is performed using Altera Quartus® software. Firstly, the implemented design is tested for its system communication. Then, the test process and results are monitored and displayed on the Nios II console. The DE2 communicates with the host PC via the JTAG interface.

As the test program is run, Nios II will initialize the parameters by giving the corresponding input to the reader. Afterward, Nios II will monitor any changes in the protocol status since it indicates the exchange data process or status between the reader and tag. Every communication from the reader is displayed on the Nios II console. There are three stages of communication between reader and tag. In the first stage, after Nios II initialization, the reader will send data in the form of the Query command, then followed by the QueryRep command. In the second stage, the reader receives a reply from the tag in the form of RN16. Then, the reader will return the RN16 along with the ACK command. Finally, the reader receives the EPC value from the tag in the last stage.

These three stages of communication and their testing result are shown in Fig. 12. It can be seen that the implemented design is successful in performing all stages of communication, and the reader can also successfully receive the EPC value from the tag. The synthesis result is presented in Table V and Table VI. The design uses 976 logic elements and 939 combination functions, consisting of 573 (4-input functions), 213 (3-input functions), and 153 (2-or-less-input functions). From 976 logic inputs available, 119 are used in arithmetic mode, and the other 373 are registers. As for the submodules, the command processor uses 346 logic elements, while the transmitter and receiver use 328 and 265 logic elements, respectively.

The power consumption is calculated using Powerplay analysis provided by Quartus, as shown in Fig. 13. A Powerplay is a standard tool used by various scholars to analyze the power consumption of the designed chip [35]–[37]. Based on the analysis result, the proposed chip consumes 173.14 mW with the following details: 0.13 mW from dynamic power dissipation, 98.6 mW from static power dissipation, and 74.34 mW from I/O dissipation. The power consumption is relatively low and it can also answer one of various challenges in RFID tag chip design, which is low-power allowing with low-cost [38]. This data is expected to transfer the design into a specific chip dedicated to IoT application, specially targeted for a low-power application case that requires RFID to operate.

The RFID tag design made is a digital block from a complete transponder tag. The phase for developing RFID tags is to integrate digital blocks with analog parts such as antennas, modulators, and voltage regulators. The design of the RFID tag made is an initial study that focuses on a digital block architecture that can accommodate the ISO 18000-6C standard. The development space for digital blocks is still vast, including design optimization so that a tag design can be obtained that has a smaller size and more efficient in consuming the power.

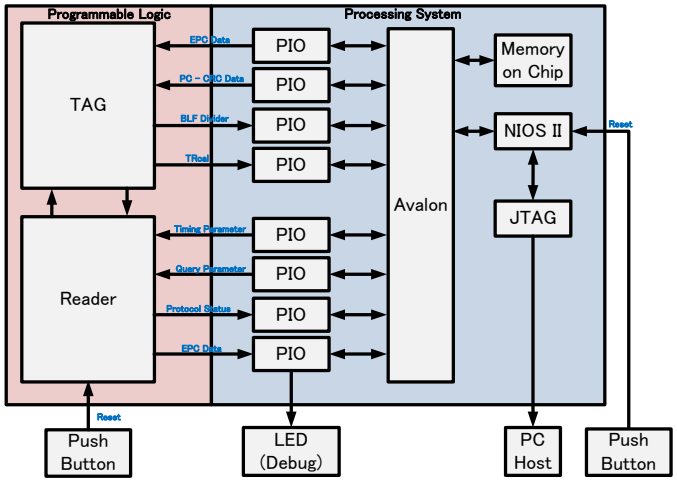



Fig. 11. Block Diagram of the Full RFID System as Implemented in FPGA Altera DE2-115 Board.

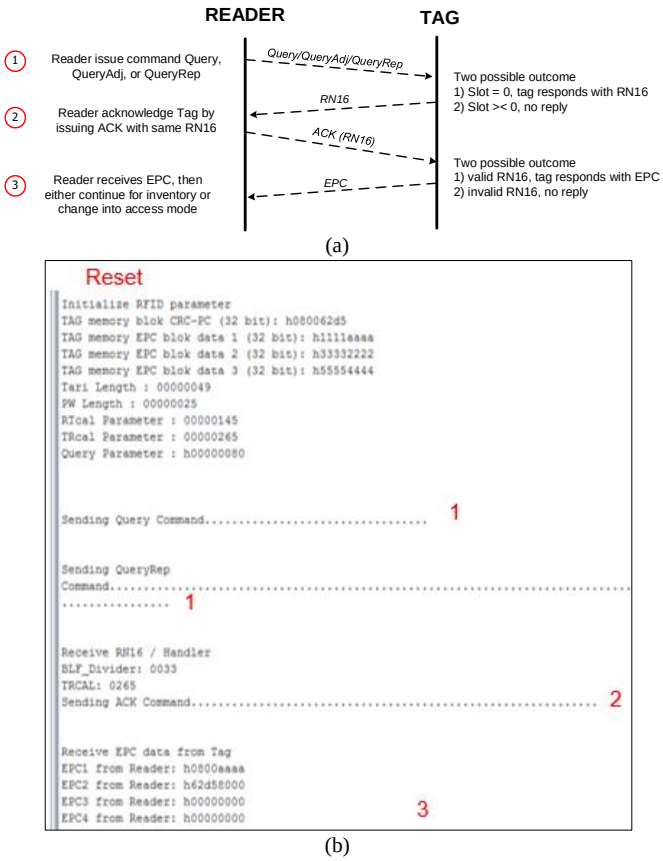



Fig. 12. (a) Communication Stages between Reader and Tag; (b) Its Corresponding Implemented System Result.

TABLE V. SYNTHESIS RESULT OF THE FULL RFID TAG BASEBAND PROCESSOR (MAIN MODULES)

| Resource | Usage |
|---|---|
| Estimated total logic elements | 976 |
| Total combinational functions | 939 |
| Logic element usage by number of LUT inputs | |
| • 4 input functions | 573 |
| • 3 input functions | 213 |
| • ≤ 2 input functions | 153 |
| Logic elements by mode | |
| • Normal mode | 820 |
| • Arithmetic mode | 119 |
| Total registers | 373 |
| • Dedicated logic registers | 373 |
| • I/O registers | 0 |
| I/O pins | 168 |
| Embedded multiplier 9-bit elements | 0 |
| Maximum fan-out node | Clk-input |
| Maximum fan-out | 268 |
| Total fan-out | 4641 |
| Average fan-out | 2.82 |

TABLE VI. SYNTHESIS RESULT OF THE FULL RFID TAG BASEBAND PROCESSOR (SUBMODULES)

| Entity name | LC Combinations | LC Registers |
|---|---|---|
| *Debug* | 939 (0) | 373 (0) |
| *Tag_transmitter* | 328 (138) | 196 (145) |
| *Miller_enc* | 25 (25) | 13 (13) |
| *Frame_generator* | 140 (140) | 24 (24) |
| *Fm0_enc* | 18 (18) | 11 (11) |
| *clock* | 7 (7) | 3 (3) |
| *Tag_receiver* | 265 (2) | 126 (1) |
| *Pnrg_module* | 11 (11) | 16 (16) |
| *Pie_demod* | 70 (55) | 46 (35) |
| *Counter_rx* | 15 (15) | 11 (11) |
| *Cmd_buffer* | 114 (114) | 54 (54) |
| *Blf_generator* | 68 (68) | 9 (9) |
| *Command processor* | 346 (346) | 51 (51) |

| PowerPlay Power Analyzer Summary | |
|---|---|
| PowerPlay Power Analyzer Status | Successful - Fri Oct 27 15:02:47 2017 |
| Quartus Prime Version | 16.0.2 Build 222 07/20/2016 SJ Lite Edition |
| Revision Name | RFIDTag |
| Top-level Entity Name | tag_RFID |
| Family | Cyclone IV E |
| Device | EP4CE115F29C7 |
| Power Models | Final |
| Total Thermal Power Dissipation | 173.14 mW |
| Core Dynamic Thermal Power Dissipation | 0.13 mW |
| Core Static Thermal Power Dissipation | 98.67 mW |
| I/O Thermal Power Dissipation | 74.34 mW |

Fig. 13. Power Consumption Analysis Result from Power Play.

## IV. CONCLUSION

This study has designed a baseband processor design for UHF RFID passive tag. The baseband processor consists of several blocks, *i.e.*, receiver, transmitter, command processing, CRC (CRC-5 and CRC-16), and PRNG. A detailed RTL design and testing methodology of the baseband processor was also presented. All the design and testing criteria were constructed compliant with the EPC Gen-2 standard. Each block showed a low latency (below 400 ns), which is negligible to the whole system. Using the UVM testing method, 1344 parameter combinations were conducted on the whole system design. The results reveal that it passed all the testing scenarios by the UVM, including query parameter, timing parameter, state transition, and BLF test. Moreover, the design was also successfully implemented on an FPGA board (Altera DE2-115). The implemented system used 976 logic elements and consumed 173.14 mW of power dissipation. This comprehensive RFID design and testing methodology allow a more reliable RFID baseband processor design to be realized, leading to a higher yield of other real chip fabrication processes. The results show that it can be used to enable a chip-based, low-power IoT Application. Future work will focus on the chipset design and tap-out the design.

## ACKNOWLEDGMENT

The publication fee for this work is fully handled and sponsored by program peningkatan *Global Competitiveness* Perguruan Tinggi Indonesia Universitas Pendidikan Indonesia 2021 Batch II with No SK 1370/UN40/PT.01.02/2021.

## REFERENCES

[1] Condea, F. Thiesse, and E. Fleisch, "RFID-enabled shelf replenishment with backroom monitoring in retail stores," Decision Support Systems, vol. 52, no. 4, pp. 839–849, Mar. 2012, doi: 10.1016/j.dss.2011.11.018.

[2] M. Liukkonen and T.-N. Tsai, "Toward decentralized intelligence in manufacturing: recent trends in automatic identification of things," Int J Adv Manuf Technol, vol. 87, no. 9, pp. 2509–2531, Dec. 2016, doi: 10.1007/s00170-016-8628-y.

[3] M. Kirch, O. Poenicke, and K. Richter, "RFID in Logistics and Production –Applications, Research and Visions for Smart Logistics Zones," Procedia Engineering, vol. 178, pp. 526–533, Jan. 2017, doi: 10.1016/j.proeng.2017.01.101.

[4] N. M. Sahar, M. T. Islam, and N. Misran, "Design of dualband antenna for RFID applications," International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 4, Aug. 2019, doi: 10.11591/ijece.v9i4.pp3146-3152.

[5] E. E. Abel, A. L. M. Shafie, and W. H. Chan, "Deployment of internet of things-based cloudlet-cloud for surveillance operations," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 10, no. 1, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp24-34.

[6] R. Abdulla, A. Abdillahi, and M. K. Abbas, "Electronic Toll Collection System based on Radio Frequency Identification System," International Journal of Electrical and Computer Engineering (IJECE), vol. 8, no. 3, Jun. 2018, doi: 10.11591/ijece.v8i3.pp1602-1610.

[7] A. Abdulkareem, "Development and implementation of a miniature RFID system in a shopping mall environment," International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 2, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1374-1378.

[8] A. J. Samuel and S. Sebastian, "An algorithm for IoT based vehicle verification system using RFID," International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 5, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3751-3758.

[9] H. A. Khan, R. Abdulla, S. K. Selvaperumal, and A. Bathich, "IoT based on secure personal healthcare using RFID technology and steganography," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 4, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3300-3309.

[10] M. O. Adebiyi, R. O. Ogundokun, A. I. Nathus, and E. A. Adeniyi, "Smart transit payment for university campus transportation using RFID card system," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 5, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4353-4360.

[11] D. He, G. Mujica, G. Liang, J. Portilla, and T. Riesgo, "Radio propagation modeling and real test of ZigBee based indoor wireless sensor networks," Journal of Systems Architecture, vol. 60, no. 9, pp. 711–725, Oct. 2014, doi: 10.1016/j.sysarc.2014.08.002.

[12] D. Lautner, X. Hua, S. DeBates, and S. Ren, "WaaS (Wireless-as-a-Sensor): Conception, design and implementation on mobile platforms," Journal of Systems Architecture, vol. 88, pp. 65–73, Aug. 2018, doi: 10.1016/j.sysarc.2018.05.009.

[13] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," in 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, China, Apr. 2012, pp. 1282–1285. doi: 10.1109/CECNet.2012.6201508.

[14] S.-Y. Wong and C. Chen, "Power efficient multi-stage CMOS rectifier design for UHF RFID tags," Integration, vol. 44, no. 3, pp. 242–255, Jun. 2011, doi: 10.1016/j.vlsi.2011.03.005.

[15] M. Škiljo, P. Šolić, Z. Blažević, L. D. Rodić, and T. Perković, "UHF RFID: Retail Store Performance," IEEE Journal of Radio Frequency Identification, pp. 1–1, 2021, doi: 10.1109/JRFID.2021.3129694.

[16] K. Bhanushali, W. Zhao, W. S. Pitts, and P. D. Franzon, "A 125 μm × 245 μm Mainly Digital UHF EPC Gen2 Compatible RFID Tag in 55 nm CMOS Process," IEEE Journal of Radio Frequency Identification, vol. 5, no. 3, pp. 317–323, Sep. 2021, doi: 10.1109/JRFID.2021.3087448.

[17] I. Galko, R. Kuffa, P. Magdolenová, J. Svetlík, and A. Veľas, "RFID tags at the operation of fire stations," Transportation Research Procedia, vol. 55, pp. 941–948, Jan. 2021, doi: 10.1016/j.trpro.2021.07.062.

[18] E. H. Hadj-Mihoub-Sidi-Moussa, R. Touhami, and S. Tedjini, "Design and Evaluation of an RFID Localization System based on Read Count," IETE Journal of Research, vol. 0, no. 0, pp. 1–10, Feb. 2021, doi: 10.1080/03772063.2021.1880341.

[19] D. Wei, C. Zhang, Y. Cui, H. Chen, and Z. Wang, "Design of a low-cost low-power baseband-processor for UHF RFID tag with asynchronous design technique," in 2012 IEEE International Symposium on Circuits and Systems (ISCAS), Seoul, Korea (South), May 2012, pp. 2789–2792. doi: 10.1109/ISCAS.2012.6271889.

[20] J.-W. Lee, N. D. Phan, D. H.-T. Vo, and V.-H. Duong, "A Fully Integrated EPC Gen-2 UHF-Band Passive Tag IC Using an Efficient Power Management Technique," IEEE Transactions on Industrial Electronics, vol. 61, no. 6, pp. 2922–2932, Jun. 2014, doi: 10.1109/TIE.2013.2278519.

[21] Smarani Ismail and A. Ibrahim, "Modelling and Simulation of Baseband Processor for UHF RFID Reader on FPGA," International Journal of Electrical and Electronic Systems Research, vol. 6, pp. 53–67, 2013.

[22] J.-T. Su, Z. Xie, X.-A. Wang, and Y. Cao, "Design and Automatic System Verification of Digital Baseband for UHF RFID Tag," International Journal of Electronics and Electrical Engineering, vol. 1, no. 3, pp. 130–134, 2013.

[23] S. Li et al., "A −20 dBm Passive UHF RFID Tag IC With MTP NVM in 0.13-μm Standard CMOS Process," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 12, pp. 4566–4579, Dec. 2020, doi: 10.1109/TCSI.2020.3007952.

[24] M. Yang, S. Wei, M. Liu, R. Huang, M. Yu, and F. Chen, "Design Verification and Test Techniques for UHF RFID Tag IC," in 2019 IEEE Sustainable Power and Energy Conference (iSPEC), Beijing, China, Nov. 2019, pp. 2500–2505. doi: 10.1109/iSPEC48194.2019.8975371.

[25] L. Xie, W. Nie, X. Yang, Y. Wang, and M. Zhou, "A BLF Generation Scheme with Clock Variance-Tolerance for Baseband Processor of EPC Gen2 UHF RFID Tag," in Cloud Computing and Security, Cham, 2018, pp. 543–552. doi: 10.1007/978-3-030-00015-8_47.

[26] K. Fyhn, R. M. Jacobsen, P. Popovski, A. Scaglione, and T. Larsen, "Multipacket Reception of Passive UHF RFID Tags: A Communication Theoretic Approach," IEEE Transactions on Signal Processing, vol. 59, no. 9, pp. 4225–4237, Sep. 2011, doi: 10.1109/TSP.2011.2159499.

[27] Y.-C. Hung and H.-J. Hung, "RFID design with CRC programmable capability, dual encode functions and dual modulation outputs," in 2010 International Symposium on Next Generation Electronics, Kaohsiung, Taiwan, Nov. 2010, pp. 219–222. doi: 10.1109/ISNE.2010.5669159.

[28] J. Zhang and H. Gao, "Implementation of CRC Algorithm in UHF RFID Test System Based on Labview," Jul. 2016, pp. 902–906. doi: 10.2991/iccia-17.2017.158.

[29] L. T. Clark, J. Adams, and K. E. Holbert, "Reliable techniques for integrated circuit identification and true random number generation using 1.5-transistor flash memory," Integration, vol. 65, pp. 263–272, Mar. 2019, doi: 10.1016/j.vlsi.2017.10.001.

[30] M. Soybali, B. Ors, and G. Saldamli, "Implementation of a PUF Circuit on a FPGA," in 2011 4th IFIP International Conference on New Technologies, Mobility and Security, Paris, France, Feb. 2011, pp. 1–5. doi: 10.1109/NTMS.2011.5720638.

[31] T. Khanom and F. Khanom, "Implementation of Pseudo-Random Number Generator Using LFSR," Publications and Research, Dec. 03, 2020. https://academicworks.cuny.edu/ny_pubs/642.

[32] Babitha K.P., Thushara T, and Dechakka M P, "FPGA based N-Bit LFSR to Generate Random Sequence Number," International Journal of Engineering Research and General Science, vol. 3, no. 3, pp. 6–10, 2015.

[33] D. G. Kuester, D. R. Novotny, J. R. Guerrieri, A. Ibrahim, and Z. B. Popovic, "Simple Test and Modeling of RFID Tag Backscatter," IEEE Transactions on Microwave Theory and Techniques, vol. 60, no. 7, pp. 2248–2258, Jul. 2012, doi: 10.1109/TMTT.2012.2195017.

[34] D. Tsiktsiris, D. Ziouzios, and M. Dasygenis, "A portable image processing accelerator using FPGA," in 2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, May 2018, pp. 1–4. doi: 10.1109/MOCAST.2018.8376566.

[35] A. Gidd, S. Ghasti, S. Jadhav, and K. Sivasankaran, "Performance Analysis of 32-Bit DADDA Multiplier Using 15–4 Compressor," in Microelectronic Devices, Circuits and Systems, Singapore, 2021, pp. 19–30. doi: 10.1007/978-981-16-5048-2_2.

[36] O. Drozd, G. Nowakowski, A. Sachenko, V. Antoniuk, V. Kochan, and M. Drozd, "Power-Oriented Monitoring of Clock Signals in FPGA Systems for Critical Application," Sensors, vol. 21, no. 3, Jan. 2021, doi: 10.3390/s21030792.

[37] M. Wagih and J. Shi, "Wireless Ice Detection and Monitoring Using Flexible UHF RFID Tags," IEEE Sensors Journal, vol. 21, no. 17, pp. 18715–18724, Sep. 2021, doi: 10.1109/JSEN.2021.3087326.

[38] J. Lu, D. Liu, H. Li, C. Zhang, and X. Zou, "A Fully Integrated HF RFID Tag Chip With LFSR-based Light-weight Tripling Mutual Authentication Protocol," IEEE Access, vol. 7, pp. 73285–73294, 2019, doi: 10.1109/ACCESS.2019.2920437.