DESIGN OF A NOVEL MEDIUM ACCESS CONTROL PROTOCOL FOR OPTIMIZATION OF CDMA-BASED PASSIVE RFID: BITWISE CDMA

by

Ramsey Ziad Doany, B.S.

A thesis submitted to the Graduate
Council of Texas State University in
partial fulfillment of the requirements
for the degree of Master of Science with
a Major in Engineering
May 2018

Committee Members:

Harold Stern, Chair

William Stapleton

Semih Aslan

DEDICATION

This thesis is dedicated to the memory of my late grandfather Rachid Koleilat who believed in me in a way that only he could. I carry his love, his philosophy of life, and his belief in my abilities wherever I am. His memory drives me to pursue a life of learning.

I also dedicate this thesis to the memory of Monah Fakhoury who unexpectedly passed at the beginning of the writing of this thesis. Monah showed me how with dream and dedication, anything is possible. I am inspired by his creativity, perseverance, entrepreneurship, and joy for life. The antique compass he gave me as a child provides me not only with the memory of his life and the impact he had on mine but also represents a reminder to always explore.

ACKNOWLEDGEMENTS

First and foremost, I must thank Dr. Harold Stern for not only being my advisor but for being my mentor and friend. Dr. Stern inspired me as an undergrad to not only engineer but to create. He believed in me as an undergraduate student, took me under his wing, and provided me with the mentorship and guidance I needed to succeed as both an undergraduate and graduate student. I will be forever grateful for Dr. Stern and his dedication to his work and his students.

I must also thank Dr. Semih Aslan who provided me with countless hours of discussion as both an undergraduate student and graduate student. Dr. Aslan went above and beyond and provided me with not only opportunities to learn but also to create, write, and teach others.

I thank Dr. William Stapleton for aiding in my understanding of the specific engineering constraints to achieve the goals of this thesis. Dr. Stapleton provided me with extremely important insights, without which this thesis would not have been possible. I also thank Dr. Stapleton for his dedication as a professor.

I thank Dr. Karl Stephan for his guidance in my undergraduate and graduate career, particularly his support during my undergraduate senior design project, without which the conception of this thesis would not have been possible. Dr. Stephan’s dedication to his students is immense and I am grateful to be among those students.

I thank Dr. Michael Casey for providing me with the framework with which to build my electrical engineering career upon. Dr. Casey cares not only about his students' success in academics but their success as human beings. The knowledge that he has imparted upon me and the care that he shows his students is something that I will carry with me for the rest of my life.

Finally, I must thank my family. My father, Ziyad Doany, an electrical engineer himself, believed in me at a young age, even when very few others did. He spent uncountable hours, days, and years ensuring that regardless of my grades, I was learning. His support throughout my life has defined my abilities today and I am glad to finally prove him right. My mother, Dr. Nada Koleilat Doany, provided me with an environment to create both academically and artistically, always without judgement. I must also thank my girlfriend Emma Randles who gave me the emotional support to not only complete this thesis but to complete it in less than 16 months. I would not have been able to succeed without their patience and support.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

ABSTRACT

UHF passive RFID is a wide and continuously growing technology. As such, it is becoming increasingly important to develop highly efficient and inexpensive RFID systems. Presently available RFID tags utilize MAC protocols that are highly susceptible to noise, inefficiently utilize interrogator power, and in the case of CDMA-based RFID, require internal power sources. There exists a need to develop passive RFID systems that efficiently use interrogator power, are robust in high noise environments, and can simultaneously read multiple RFID tags. This thesis develops the background of currently available RFID protocols with a focus on CDMA-based RFID protocols, analyzes their trade-offs, and provides a novel protocol (Bitwise CDMA) to overcome several of the disadvantages of CDMA-based systems while maintaining the benefits of such systems. The framework for inexpensive UHF RFID tag research and development is also discussed as currently available tools are limited in functionality and/or prohibitively expensive.

I. INTRODUCTION

A. Problem Statement

The problem statement and goal of this thesis is twofold: to analyze and improve upon implementations of CDMA-based passive RFID tags and to begin the development of the necessary tools to do so. RFID is an ever-growing technology that is finding more use every day. As such, high efficiency cannot be achieved using a one-size-fits-all tag, with some environments and implementations requiring highly specialized tags. High noise environments or environments in which EM radiation from the RFID system must be minimized can benefit from the advantages of a CDMA-based passive RFID system, some of those advantages being inexpensive tags and a robustness in low SNR environments. Thus, it is valuable to find solutions to the problems that are preventing adoption of CDMA into the current passive RFID standard. Much of the research in UHF RFID is impeded by the lack of research tools. As is discussed later in this thesis, there is a large market for extremely modular and "hackable" LF and HF development tools but the market for passive UHF RFID development tools is sparse and has extremely limited use for the test of novel tag architecture or protocols. This thesis addresses this issue by beginning the development of an FPGA-based solution in which tags and readers can be designed in Verilog and the logic of the architecture and/or protocols can be tested in hardware on an FPGA.

B. Scope of Thesis

This thesis begins by briefly reviewing the historical and current uses of RFID and the market of this technology. Next, there is an overview of the RFID system, including some review of communication system, communication network, and wireless communication theory. A summary of the Class 1 Generation 2 EPC standard is then provided to give context of the specific technology within RFID that is targeted within this paper and a literature review follows. Next, the potential anti-collision MAC protocols are discussed and are analyzed mathematically and in simulation to determine performance. A new implementation is then discussed, and its performance is analyzed alongside the other implementations. The goal of this new implementation is to improve upon the previously developed CDMA RFID protocols while correcting for the near-far problem with minimal change to the Class 1 Generation 2 specifications and tag structure and addressing the specific constraints of passive CDMA. A description of the Verilog code for a Slotted ALOHA tag and interrogator (testbench) and the Verilog code for the CDMA tag and interrogator implementing the novel anti-collision MAC are then given. MATLAB and Vivado simulations are used to compare performances of the Slotted ALOHA and novel implementations. Finally, an overview of the entire thesis is given along with recommendations for future work.

C. Organization of this Thesis

This thesis is organized as follows:

Chapter 1 provides a background and introduction to this thesis. Chapter 2 gives a brief history of RFID, an overview of the RFID system, the components of the RFID system, the RFID market, a review of communication system and wireless communication theory, and explores previous work in CDMA-based RFID and gives a summary of the current standard Class 1 Generation 2 RFID protocol. Chapter 3 develops qualitative and quantitative analysis of TDMA, CDMA, and combination TDMA-CDMA medium access control protocols in RFID. Chapter 4 describes a novel medium access control protocol for CDMA-based RFID, referred to as Bitwise CDMA and compares its tradeoffs to those of Slotted ALOHA, CDMA, and combination TDMA-CDMA medium access control protocols. Chapter 5 explains the Verilog implementation of the Slotted ALOHA system and a tag utilizing Bitwise CDMA. Chapter 6 provides a conclusion to the thesis and exists as a review of the topics covered in the thesis and a summary of Bitwise CDMA. Chapter 7 discusses future work that can potentially add to and expand upon the work of this thesis.

II. RFID and MAC

In this chapter, the background for this thesis is developed. This chapter begins by providing a history of radio frequency identification as well as its market standing and use today. This chapter also provides a review of information theory, communication system theory, and wireless communication theory. Previous work in CDMA-based RFID including journal articles and conference papers are also reviewed. Finally, a summary of the current Class 1 Generation 2 standard is provided.

A. History and Current Use of RFID

The first radio frequency identification system was used during WWII by British military to distinguish the country of origin of fighter planes [1]. As communications engineering improved in the following decades and RF research continued, RFID was used in antitheft systems with the first RFID patents being granted in the 1970s. 125kHz systems (Low-frequency or LF) were the first to be widely used and by the 1990s, MHz range and 100 MHz (High-frequency or HF) range tags were developed by IBM [1]. Currently, there are two main types of RFID systems: active and passive. Active tags have an internal battery, generally operate in the 100 MHz to the GHz range (Ultra high-frequency or UHF), and can have a read range of up to 100 meters. Passive tags operate from the kHz range up to 1 GHz, do not contain any internal power, have a read range of up to 30 feet (or more) and are a fraction of the price of active tags [2]. Both categories, as well as the third category, semipassive tags, are discussed further in later sections.

The cost of a tag depends on several factors including operating frequency, storage capacity, read range, antenna design, battery power and lifespan (in the case of active tags), as well as the packaging. The specific application determines the packaging requirements and can increase the cost of the tag (i.e. tags designed for high-temperature environments). Additionally, the manufacturing failure rate of passive RFID tags has historically been very high, and the costs associated with those losses must be accounted for in the list price of the tag. Active tags generally have a lower failure rate. Ultra-high frequency (UHF) passive RFID systems are the most common systems, the interrogators of which cost several thousand dollars. RFID systems also require a computer (handheld or otherwise) and software [3].

B. Basics of the RFID System

Radio-frequency identification (RFID) is a wireless communication technology that employs the use of electromagnetic waves to detect and identify tags that are generally embedded within an object (such as a credit card or a mobile device) or embedded within a sticker that is affixed to an object [4].

RFID systems consist of two main components: an interrogator (also referred to as a “reader”) and a tag. The interrogator, often connected to a host computer or a mobile device, contains within it an antenna, an RF transceiver, and a processing unit. The RFID tag consists of an antenna, a low-power IC, memory, power harvesting and rectification circuitry, an analog RF front end, and signal modulation circuitry. RFID tags can be either active, semi-passive, or passive. These classifications indicate whether a tag has an internal battery that is used for all its functions, a battery that is only used to provide power to the IC (but not for communication), or if the tag has no battery at all (e.g. the tag receives all power from the interrogator), respectively [4].

Types of RFID Tags

Passive and semi-passive RFID tags employ a communication technique called backscattering. This means that rather than generating their own transmission signal, they reflect the interrogator's signal back to the interrogator, modulating it in the process to send information back to the reader. Passive RFID systems make up over 80% of the RFID market with active and semi-passive RFID systems making up the other 20% [5]. The current size of the RFID market is approximately $16 billion and is expected to increase to over $24 billion by 2020 [6]. Passive RFID tags are a key component of this growth due to their potential for a wide range of uses including inventory tracking, subcutaneous implants, anti-theft systems, low-power embedded systems, and location systems.

There are 6 classes of RFID tag, referred to as Class 0-5. Class 0 and Class 1 tags are passive, write-once-read-multiple (WORM) tags that do not have reprogrammable memory fields and employ backscattering. The simplest of this type of tag is the so-called 1-bit tag. The 1-bit tag can be detected but contains no other memory and is often used in anti-theft systems [4]. Class 1 tags generally have some additional functionality that Class 0 tags do not [7]. As Class 1 and Class 2 tags have become more established, Class 0 have been phased out [8]. Class 2 tags are passive and employ the same backscatter technique as Class 0 and 1 tags but contain at least 65 KB of re-programmable memory with many of the Class 2 tags on the market today having larger memory banks. Class 3 tags are like Class 2 tags but contain an internal power supply for the IC, making them semi-passive. Class 4 tags are active tags that contain an internal power supply that is used by the IC and in transmission. Class 5 tags are like Class 4 tags but have the added capability of being able to communicate between each other and with other devices [9].

Due to their relatively low cost, passive tags (i.e. Class 1 and Class 2 tags) are expected to achieve ubiquity in inventory systems, tracking systems, medical systems, and similar fields. However, they have had some difficulty gaining complete acceptance due to a problem known as collision [10].

Types of Collision

When an RFID system is used for an application such as inventory, the interrogator powers up all the tags within range, instructs them to generate a random number which is used to determine their wait time before responding, and the reader then acknowledges and reads each tag as their turn (or slot) is called. This protocol is referred to as Slotted ALOHA and has 2 potential types of collision: interrogator collisions and tag collisions. Interrogator collisions occur when a tag is in the read-range of two or more interrogators simultaneously (known as interrogator-interrogator collision) or when a tag is read by the incorrect interrogator in a situation where multiple interrogators are present (known as interrogator-tag collision). Tag collisions occur when multiple tag reflections interfere with one another, preventing proper reading of the colliding tags. Interrogator collisions are only a problem for multi-interrogator systems, which are less common and in the case of active or semi-passive tags, tag collisions are fairly easy to resolve [10]. Passive tag collisions are more difficult to solve but due to passive tags' potentially ubiquitous use, low power solutions to collision in passive RFID systems are extremely valuable [4].

C. Medium Access Control and Multiplexing

Anti-collision is an issue that is in the realm of medium access control (MAC) and multiplexing (e.g. the methods used to allocate a single medium among many users). There are three general types of medium sharing in wireless communication: Time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA). These systems are not mutually exclusive and a combination of TDMA, FDMA, and/or CDMA is often used within a single communication system.

In TDMA systems, each user is given a time slot that is occupied by no more than one user at a time within a frequency band in the communication system. An example of TDMA is the T1 transmission system that allows for 24 telephone calls to share a single line [11]. TDMA systems can force a single user to complete their transmission before any other user is allowed to transmit. In some implementations, multiple users' transmission bits can be interleaved as part of the MAC protocol. A diagram of a general TDMA system can be seen in Figure 1.

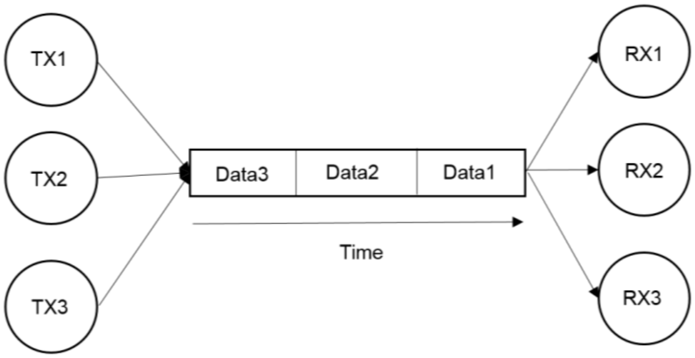


Figure 1. Diagram of general TDMA System.

FDMA employs the use of bandpass filters to divide up the medium by frequency. Each user in an FDMA system is only allowed to access a small frequency range. It is important that sharp filters are used in FDMA systems so that neighboring users do not interfere with one another. FDMA is used to divide up the electromagnetic spectrum between technologies (i.e. WiFi, LTE, AM/FM Radio, etc.). FDMA is used very heavily in cellular systems to further divide the electromagnetic spectrum between service providers, cells, users, and uplink/downlink channels. A general FDMA system can be seen in Figure 2.

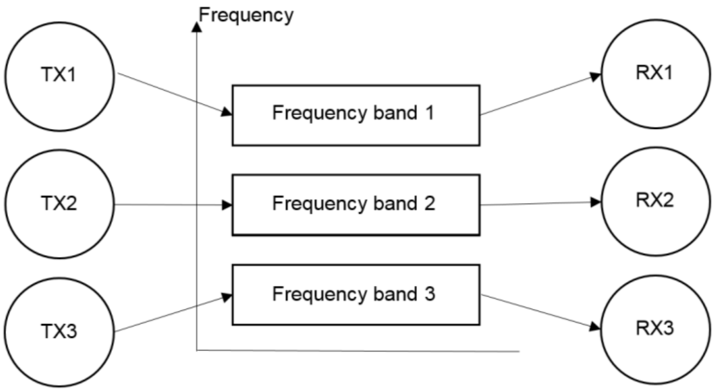


Figure 2. Diagram of general FDMA System.

In DS-CDMA, each user’s data is mixed with a high frequency unique spreading code and is transmitted at the same bitrate that would have been used without applying a spreading code, thus effectively spreading the power of the signal across a larger bandwidth. A rectangular pulse in the time domain yields a sinc function in the frequency domain. The width of the pulse in the time domain is inversely related to the bandwidth of the main lobe of the sinc function in the frequency domain [12]. A visualization of this can be seen in Figure 3.

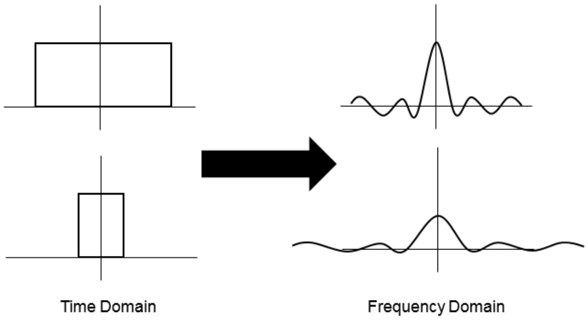


Figure 3. Rectangular pulse in the time domain and in the frequency domain.

For CDMA to be possible, unique and orthogonal (or nearly orthogonal) spreading codes are required. Codes with zero cross-correlation are considered truly orthogonal [13]. Since truly orthogonal codes are limited, pseudorandom (PN) codes are often used. Good PN codes are not truly orthogonal but still have very low cross-correlation [14]. Desirable characteristics of PN sequences exhibit the following properties [12]:

1. There should be a balance of "1"'s and "0"'s such that the difference in number of "1"'s and number of "0"'s should be no more than 1.
2. Sequences of "1"'s or "0"'s in a code should follow the following distribution: P(n sequential "1"'s or "0"'s) = 1/(2^n). This indicates randomness and independence.
3. If an n-bit code is cyclically shifted j places to the right, and XORed with the original code, the result should exhibit the first property for all $0 < j < n$

The quality of the spreading codes used in CDMA affects the number of users that can occupy a channel simultaneously, and the susceptibility of the system to the near-far problem, the effects of which are discussed in later sections in this thesis. There are several algorithms used to develop both truly orthogonal codes as well as PN codes, both of which are outside the scope of this thesis. Well known algorithms include Walsh codes, Gold codes, Kasami codes, and PN sequences can also be generated using linear feedback shift registers [12-14].

Just as in every engineering pursuit, each one of these methods comes with its own tradeoffs. For example, if the number of users is too great, a TDMA system could become unusably slow when compared to FDMA and CDMA. On the other hand, TDMA can be preferable to FDMA and CDMA because FDMA and CDMA require a larger bandwidth than TDMA for the same number of users, all other factors kept equal. CDMA is less suitable in systems that require minimal computational complexity but is significantly more robust in the presence of noise and has inherent encryption.

The MAC scheme used in the Class 1 Generation 2 RFID system is known as the Slotted ALOHA protocol. Slotted ALOHA is a TDMA system that requires that a user completes its transmission before any other user begins their transmission. In this system, each user waits until their time slot is called before transmitting. This differs from pure ALOHA in that in pure ALOHA, any user can transmit at any time and users are only instructed to wait if a collision occurs (i.e. if multiple users happen to transmit at the same time). Slotted ALOHA has double the rate of successful transmission that pure ALOHA does and more than double the throughput [15].

D. Previous Work

Literature Review

The previous work on CDMA-based RFID is expansive but has focused on the types of spreading codes best suited for RFID [14], descriptions of the system as a whole [10], or has not adequately addressed the specific challenges of applying CDMA to passive RFID tags and has been focused on the concept of CDMA-based (or CDMA-TDMA combination) RFID and/or its application [16-19]. Much of the work that describes complete systems involve semi-passive or active RFID tags but do not provide a solution for passive CDMA-based RFID [20].

In previous work, we have demonstrated that power efficiency can be increased from the interrogator’s perspective but did not fully address the issue of the near-far problem that occurs in CDMA-based RFID in the form of shadowing. The issue is discussed further in future sections but fundamentally, there exists a problem in which lower power reflections from tags can be large enough that they negatively affect the SNR of higher reflected power tags but are not high enough in power to be accurately decoded by the interrogator. The previously proposed solution was the use of adaptive interference cancellation [21]. Adaptive interference cancellation (or AIC) is an analog solution that records the tag reflections, reads the highest power tag reflection, subtracts it from the conglomerate signal then continues to read the lower powered tags. AIC does not effectively increase SNR during the communication itself and is less optimized for such an application than the newly proposed protocol.

Some of the existing literature rightfully points out that the sheer number of bits transmitted in a CDMA system is potentially higher than that of a modified Slotted ALOHA system but incorrectly asserts that this necessarily yields an increase in time to read all the tags within the read-range. The paper also does not address the other benefits of using CDMA, such as the increase in accuracy in low SNR environments, the fact that CDMA is close to truly simultaneous tag reading, and the fact that CDMA contains inherent security [22].

RFID Research Tools

The current market of RFID research tools is quite varied when it comes to LF and HF RFID readers and tags. Electronics educational supply companies such as Sparkfun and Adafruit Industries (as well as general electronics suppliers such as Digi-Key and Mouser) have a wide range of LF and HF development products including interrogator transponder chips with breakout boards as well as modular Arduino-compatible RFID interrogator shields [23-25]. The online community of Instructables.com and independent engineering and electronic hobbyist blogs have generated a wide variety of LF and HF RFID tag projects that range from very simple tags that can power an LED to universal RFID keys and functional passive tags with an embedded microcontroller [26-29]. These tools and projects are modular enough that they are relatively simple to modify if one were so inclined (i.e. if one desired to modify and test new protocols or system architecture that requires a partial or total change in reader and/or tag design). This statement is simply untrue when it comes to UHF RFID systems. Just as in the LF and HF RFID development systems, there are essentially three categories of development tools: interrogators, tags, and whole RFID systems. The vast majority of the UHF RFID development kits/tools are whole RFID systems that are focused on the development of RFID system application in new environments or for potential customers to try-out an RFID system before converting their warehouse/store/etc. into a fully RFID integrated environment. None of the kits available on the market today allow for any modification to the RFID communication protocol as they are designed to function with off-the-shelf tags [30]. There are some online community and educational projects (most available on GitHub) that employ software defined radio devices so that one can design and build their own UHF RFID interrogator, but these projects are also designed to work with off-the-shelf tags, of which none are usefully reprogrammable to handle a protocol change in the reader [31]. The Alien ALR 9900+ RFID development system was utilized in this thesis to examine the standard protocol but was not used in the design process. The only reprogrammable UHF RFID tag available on the market today is the Wireless Identification and Sensing Platform (WISP) that was developed by a research team at the University of Washington [32]. At its core, the WISP is an open-source device that utilizes an ultra-low power MCU connected to a UHF antenna that is compatible with off-the-shelf and SDR-based EPC Class 1 Generation 2 UHF RFID readers. While this system is versatile, it is quite expensive (several hundred dollars for a single tag), rarely available for purchase, and still does not allow for gate level development of RFID tags. For these reasons, WISP tags were not ideal for use in this thesis. In the work leading up to this thesis, the HackRF One SDR along with GNURadio were used to emulate the receiver side of the RFID reader, the Agilent N5182A MXG Vector Signal Generator was used to emulate CDMA-encoded tag signals, and MATLAB and Simulink were used to simulate the whole system [21]. As the scope of this thesis is focused on developing a new CDMA-based RFID protocol and designing the logic for such a tag as well as assessing the performance of this protocol compared to the standard passive RFID protocol (Slotted ALOHA), the HackRF One, a Lime SDR, vector signal generator, and RFID development kit were only used in the initial research phases but not in the final design or test of the system. The protocols were developed and analyzed using MATLAB as well as pen-and-paper mathematics and the design and test of the tag utilizing this novel CDMA-based protocol were done using Verilog, Icarus Verilog, Vivado, Xilinx, and the Digilent Basys 3 FPGA board.

Verilog is a hardware description language (HDL) that is used to design, test, and implement digital circuitry into FPGAs and application specific integrated circuits (ASIC) [33]. The use of Verilog on an FPGA, as opposed to using C or assembly language on an MCU, allows for highly optimized code that can be directly used to design and manufacture an IC capable of the novel CDMA-based protocol. It also allows for the assessment of a wider range of IC designs as it does not have to complete tasks sequentially, as an MCU does.

Vivado and Xilinx are Integrated Synthesis Environments (ISE) that aid in the design and testing of Verilog code. Icarus Verilog (or iVerilog) is a terminal-based Verilog synthesizer that allows for relatively simple and rapid test of Verilog code.

The HackRF One and the Lime SDR are software defined radios (SDR). SDRs allow for inexpensive wireless communication research and development. This is achieved with a hardware component made up of an analog front end that is antenna enabled, contains analog-to-digital converters (and digital-to-analog converters) and can also incorporate an FPGA. SDRs are controlled by software, usually connected to a computer via USB or serial port. SDRs have become an inexpensive and simple tool for wireless research because many of the traditionally hardware components, including mixers, amplifiers, filters, etc. are implemented in software [34].

E. Current Protocol for Class 1 Generation 2 RFID Tags

Summary of ALOHA

The ALOHA (or ALOHAnet) is a MAC protocol that was first used in the 1970s for ground radio broadcasting but has since been used in satellite and RFID systems [35]. There are two main types of ALOHA systems: Pure ALOHA and Slotted ALOHA. In Pure ALOHA, users transmit to a receiver at-will, expecting an acknowledgement of their transmission. If packets are received from 2 or more users (e.g. a collision occurs), the communication link is terminated, and the collided users wait a random amount of time before re-attempting their transmission. Slotted ALOHA was developed to decrease the probability of collision by dividing the medium into time slots and requiring each user to begin and end its transmission during one slot. Slots are chosen randomly by the transmitters and are randomly re-selected if a collision occurs [36]. Slotted ALOHA systems assume that the arrival of users' transmissions follow a poisson distribution, data transmitted during a collision is lost, and that only 3 states exist: empty slot, successful data reception, or collision [37]. Markov chains are used to model Slotted ALOHA systems [38-40].

ALOHA systems differ from user-assigned TDMA systems in that each user randomly selects when they will transmit their information as opposed to dedicated time slots in user-assigned TDMA. The use of this kind of random TDMA system was developed to decrease delay in standard TDMA systems in which users are assigned specific slots to transmit whole packets or partial packets. Acknowledgements are used in ALOHA systems to ensure that a user's data has been received at the receiver and does not need to be re-transmitted. Some of the inefficiencies in ALOHA arise from the channel not being fully utilized (i.e. when it is idle), the number of times a user must re-transmit its information, the amount of time the user waits before re-transmission, or the amount of time between transmissions [38].

There are several variants of the ALOHA protocol, but their details are outside the scope of this thesis. This section is provided to give background for the following section on the current UHF passive RFID standard.

Summary of Class 1 Generation 2 Standard

In this section, a summary of the Class 1 Generation 2 standard for passive UHF RFID tags [41] is provided to show the framework in which the new MAC protocol could reside. This section is not given as a comprehensive analysis of the standard but exists to provide the reader with knowledge on the current MAC standard as a comparison to the new protocol and to show how the current standard and the specifications of currently available tags indicate that passive UHF RFID tags have the capability to handle the new protocol.

The general conformance requirements for the interrogator are to implement a list of mandatory commands, conform to radio regulations for the country/region of use and to be able to communicate with conforming tags. The general requirements for tag conformance are to operate over the 860-960MHz frequency range, to be able to handle the mandatory instructions listed in the standard, to only create a backscatter (reflection) after receiving the associated command from the interrogator, and to conform to all radio regulations for the country/region of use. There is a list of mandatory commands and a list of optional commands. Proprietary or custom commands are allowed but must not duplicate or replace any of the mandatory commands and must have the ability to be disabled. The specification also outlines commands that are reserved for future use that cannot be replaced by proprietary or custom commands. The allowed interrogator modulation techniques are DSB-ASK, SSB-ASK, or PR-ASK and tags must be able to demodulate all three modulation types. The data transmitted by the interrogator must be encoded using PIE. In

PIE, “high” is transmitted for time length of t and then “low” is transmitted for a time length of t to represent a “0”. To represent a “1”, the signal is “high” for 3t and “low” for t. A “1” is twice as long as a “0” in PIE.

All interrogator-to-tag communication must begin with a preamble or a frame-sync. The preamble is transmitted before an inventory round begins and contains a start delimiter, a “0” symbol, an interrogator-to-tag calibration symbol, and a tag-to-interrogator calibration symbol. The frame-sync contains the same information but does not include the tag-to-interrogator calibration symbol. The tag uses a cyclic redundancy check to ensure that the instruction is valid, and the tag and interrogator must meet a link timing requirement. The tag has reserved memory, EPC memory, and TID memory that must meet memory size minimums, but the standard does not list a maximum amount of memory that conforming tags may contain [41].

As will be seen in future sections, the new protocol does not violate any of the above requirements given that none of the reserved commands are replaced and that the tags have the capability to switch between the current protocol and the proposed protocol.

Slotted ALOHA in Class 1 Generation 2 RFID

As previously mentioned, Class 1 Generation 2 RFID tags utilize the Slotted ALOHA protocol as their medium access control technique. In this protocol, tags are powered up and instructed to load a random number generator to generate their slot number. The interrogator then instructs all of the tags in the read-range to decrement their slot number by 1 until the randomly generated number reaches 0, indicating that the tag’s time slot has been reached. At this point, the tag creates a temporary ID (known as its handle) and reflects it to the interrogator. The interrogator then acknowledges the tag and instructs the tag to reflect its EPC data. Upon completion of the communication, the interrogator instructs the tag to power down. The interrogator continues to issue decrement commands to the remaining tags. If a collision occurs (e.g. two or more tags generate the same slot number), the unread tags are instructed to regenerate their slot numbers and the process continues [21]. An inventory round in completed when all possible slot numbers have been attempted to be read by the interrogator. This system suffers from several inefficiencies that are explored in the following sections. A diagram of this protocol can be seen in Figure 4 below.

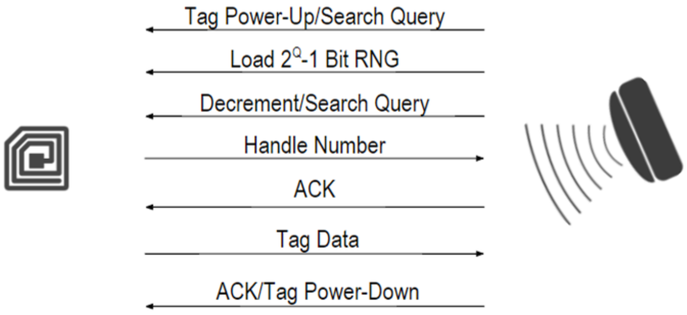


Figure 4. Diagram of Slotted ALOHA MAC protocol in Class 1 Generation 2 RFID.

III. CHARACTERISTICS OF TDMA AND CDMA IMPLEMENTATIONS

This chapter analyzes Slotted ALOHA systems, CDMA systems, and combination TDMA-CDMA systems from both a qualitative and quantitative perspective. As it is the standard, Slotted ALOHA systems are discussed first, then a basic CDMA-based system is discussed, and its variants are explored. Next, an analysis of combination TDMA-CDMA is done. Finally, a qualitative and quantitative comparison of the implementations is performed. To begin the analysis, a definition of the calculation terms is necessary and can be seen below.

Calculation terms:

SCC - Same Code Collision

SSC - Same Slot Collision

TC - Total Collision (SSC and SCC)

N = Number of Tags

S = Number of possible slots

E = Number of possible codes

B = Number of unique Code Banks

P(RNG) = probability of random number generator collision

P(ACC) = Probability of accurately reading a tag

P(EMPTY) = Probability of an empty slot

SGSCC = Same-group-same-code collision

G = Number of groups

A. Slotted ALOHA (TDMA)

To adequately assess the potential value of using CDMA in passive RFID systems, one must compare the most appealing possible implementations of CDMA to each other as well as to the current Slotted ALOHA system.

As it is the current standard, the collision probability of the Slotted ALOHA protocol will be discussed first. There are 3 possible events in each slot: An empty slot, a slot with an accurately read tag, and a slot with a collision. The probability of an empty slot is defined as the probability of no tags selecting one or more slots in an inventory round. At the beginning of each inventory round, the probability of an empty slot can be represented as,

$$P(EMPTY) = (\frac{S-1}{S})^{N} \tag{1}$$

A slot with an accurately decoded tag occurs when only one tag selects the slot in question and is read successfully. With at least one tag in the system, the probability of a slot with an accurately decoded tag is,

$$P(ACC) = (\frac{S-1}{S})^{N-1} \tag{2}$$

Thus the probability of accurately reading every tag in the read-range is,

$$if N \leq S(at\ the\ beginning\ of\ each\ inventory\ round),$$

$$\begin{aligned} P(ACC_{ALL}) &= (\frac{S-1}{S})(\frac{S-2}{S})\ldots(\frac{S-(N-1)}{S}) \\ &= (\frac{1}{S^{N-1}})(\frac{(S-1)!}{(S-N)!}) \end{aligned} \tag{3}$$

$$if N > S, \qquad P(ACC_{ALL}) = 0 \qquad 4$$

This can also be described as the probability of no collisions occurring.

A collision in the Slotted ALOHA protocol occurs when two or more tags generate the same response delay (e.g. two or more tags respond to the reader at the same time). The probability of collision of this system is based on the number of slots (defined by the interrogator, RFID system designer, and/or user) and the number of tags within read-range. For this analysis of the probability of collision, adequate SNR will be assumed such that the bit error rate for an uncollided tag is very low. The number of predefined slots presents a significant trade-off: Many slots will yield a low probability of collision but will also yield a larger probability of empty slots, thus wasting time and energy, few slots will yield fewer empty slots but will also yield more collision, thus wasting time and energy. The probability of collision is the probability that the random number generators of two or more tags in the read region will generate the same number. Assuming a high quality random number generator whose limit is equal to the number of slots, the probability of an inventory round having at least one same-slot collision (SSC) can be described as,

$$P(SSC) = 1 - P(ACC_{ALL}) = 1 - (\frac{1}{S^{N-1}})(\frac{(S-1)!}{(S-N)!}) \qquad 5$$

As can be seen in the equations above, the probability of collision in this TDMA system is defined by the number of potential slots and the number of tags in the read-range. With many tags relative to the number of slots, the probability of collision will be very high, and will continue to be high until the number of colliding tags is reduced. In the current standard protocol, non-collided tags are read and instructed to shut off, and the probability of colliding tags decreases with each inventory round. However, the number of empty slots increases, thus decreasing the speed and power efficiency of the system. Dynamic Frame

Slotted ALOHA has previously been presented as a solution to this problem [22]. In Dynamic Frame Slotted ALOHA, the number of slots changes dynamically, based on whether a tag was accurately read, a tag collision occurred, or an empty slot occurred. This system has been shown to outperform some CDMA systems in terms of number of transmitted bits but still does not outperform CDMA in terms of robustness in the presence of noise and in security [22]. As Slotted ALOHA, not Dynamic Frame Slotted AOHA, is the standard TDMA protocol for this kind of system, the following CDMA implementations will be evaluated against the standard Slotted ALOHA protocol.

B. Straight CDMA and Grouped CDMA

To illustrate the challenges as well as to introduce the factors that determine the feasibility and potential benefits from each CDMA implementation, the simplest version will be discussed first. This implementation assumes synchronization between the tags in the system and the interrogator, no near-far problem, and that all tags in the read-range can be powered up and have truly simultaneous reflection. In addition, initially, we will assume that only truly orthogonal codes (as opposed to PN sequences) are used. This system will henceforth be referred to as Simple Straight CDMA Implementation (SSCDMA). In this system, a collision occurs when two tags use the same spreading code. If the number of available codes exceeds the number of tags in the read region, the probability of collision is zero, assuming an adequate SIR and that each tag is encoded with a unique spreading code. The mathematics will show that the collision probability is only a function of the number of tags in the read-region and the number of codes used, regardless of where the codes are generated, assuming that they are randomly generated.

SSCDMA Implementation: Precoded EPC

In the precoded EPC implementation of SSCDMA, each tag has its product code (low frequency data) encoded by the spreading code (or chipping sequence) before it is written into the tag's EPC memory bank. In this implementation, each tag's RNG-determined wait time is forced to be equal. Using E to represent the number of possible codes (unique and truly orthogonal in this case) and N to represent the number of possible codes, we can say,

$$if E \geq N, \qquad P(SCC) = 0 \qquad 6$$

$$if E < N, \qquad P(SCC) = 1 \qquad 7$$

Where SCC represents Same Code Collision and with the assumption that when E ≥ N, spreading codes are not reused. In general, E is determined by the methodology with which the spreading codes are generated, the maximum acceptable bit error rate (BER), and in this case, the size of the EPC field and the length of the product code itself (because the processing gain will be limited by the ratio of the EPC field size to the length of the product code). The main benefits of this system, other than the ones inherent to CDMA multiplexing, are that since no hardware changes need to be made, it should be easily integratable to the current generation of UHF RFID tags. However, due to the relatively small EPC size of most general-purpose UHF RFID tags, the number of tags that can be read while maintaining useful BER is relatively small. The number of tags that can be accurately decoded is also limited by the ratio of the product code length to the EPC bank size. Furthermore, PN sequences require a larger processing gain to maintain an equal BER to that of a system using truly orthogonal codes. PN sequences will have the following probabilistic characteristics,

$$if E \geq N, \qquad P(SCC) \geq 0 \tag{8}$$

$$if E < N, \qquad P(SCC) = 1 \tag{9}$$

This is due to the imperfect orthogonality of PN sequences. Although it is not referred to as SSCDMA in the paper, it is discussed at length in [21].

Another possible implementation is for the tag to store its code in its deeper memory so that the product code can use the full EPC memory bank. This system would allow for longer spreading sequences, providing a lower BER or a larger number of tags that can be read without SCC and would require a minimal redesign of the tag. This system has some benefits over the previously discussed SSCDMA implementations, but the processing gain is still limited by the size of the tags' deeper memory and the EPC memory bank size and this system does not provide a superior solution to the problems that will be introduced later in this section (i.e. the near-far problem, synchronization, robustness in the presence of noise, etc.). The collision probability is the same as in Equations 8 and 9.

SSCDMA Implementation: Code Bank

Other implementations are possible, such as SSCDMA with each tag storing a bank of codes that it randomly selects from and applying the selected code to its EPC. This has the benefits of the implementation presented in the section above but suffers from the same problems. There are two possible implementations of this system: all tags containing every possible code in their code bank, or groups of tags contain a non-overlapping set of the full code bank. In the first implementation, a collision occurs when the random number generators of two or more tags generate the same number, thus selecting the same code. This probability is related to both the number of codes and the quality of the RNG.

Assuming a high quality RNG, to estimate that the RNG is truly random, the probability of collision of this implementation can be represented as,

$$if E > N \, and \, N = 1, \qquad P(SCC) = 0 \tag{10}$$

$$if E \geq N \, and \, N > 1, \qquad P(SCC) = 1 - (\frac{1}{E^{N-1}})(\frac{(E-1)!}{(E-N)!}) \tag{11}$$

$$if E < N, \qquad P(SCC) = 1 \tag{12}$$

Equation 12 produces the same performance as is obtained in Equation 5 for Slotted ALOHA when E = S.

Grouped CDMA

Another implementation is for each tag to only contain a subset of the entire code bank. In this implementation, tags are divided into groups with the number of codes stored within the code bank of each tag within each group being the total number of available tags divided by the number of desired groups. The groups in this implementation do not have any overlap in their code banks. In this implementation, collision occurs when two tags from the same group select the same spreading code. Using the same assumptions from the implementation in the section above, the probability of collision can be stated as,

$$if N \leq G \leq E \, and \, assuming \, that \, each \, group \, is \, represented \, once \, at \, most,$$

$$P(SCC) = 0 \tag{13}$$

$$\begin{aligned} &if G < N \qquad P(SGSCC) = P(Same \, Group)P(SameCode \, within \, group) \\ &\leq E, \qquad and \, assuming \, the \, tags \, are \, uniformly \, distributed \, among \\ &\qquad\qquad\qquad the \, groups, \end{aligned}$$

$$= 1 - (\frac{1}{E^{N-1}})(\frac{(E-1)!}{(E-N)!}) \qquad 14$$

$$if\ N > E, \qquad P(SGSCC) = 1 \qquad 15$$

Thus, storing all of the codes within each tag or dividing the tags into groups, each with a subset of codes, yields the same probability of collision, assuming that the tags are equally distributed among the groups. The benefit of dividing the codes between groups of tags, when compared to storing all of the codes in the memory of all of the tags is a reduction in tag complexity and cost due to the smaller minimum storage requirement. The probability of SCC for tags that generate their own spreading codes (as opposed to selecting from a bank of spreading codes) is defined by the same equation.

C. Combination TDMA-CDMA

The next implementation is a combination TDMA-CDMA system in which CDMA is used as an anti-collision protocol that comes into play when two or more tags have a same-slot collision (SSC). In this system, a total collision (TC) occurs when two or more tags have both a same-slot collision and a same-code collision (e.g. both tags select the same time slot and use the same anti-collision CDMA code). This probability can be described as,

$$ifN \leq S\ and\ N \leq E,$$

$$and\ assuming\ that\ each\ code\ (or\ group\ of\ codes)\ is\ represented\ at\ most\ once,$$

$$P(TC) = 0 \qquad 16$$

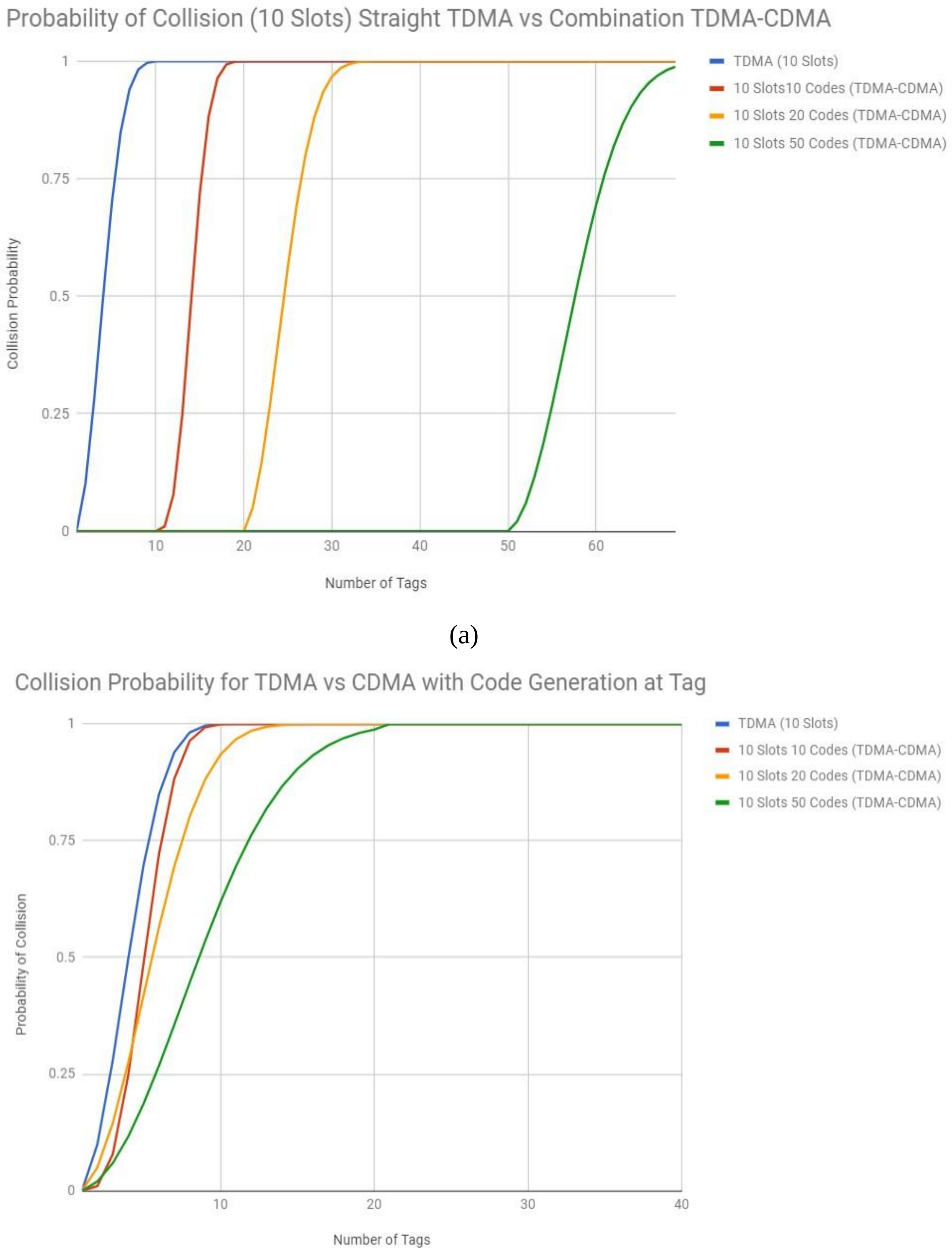
$$Otherwise, the\ probability\ of\ total\ collision\ is,$$

$$P(TC) = 1 - (\frac{1}{E^{N-1}})(\frac{(E-1)!}{(E-N)!})(\frac{1}{S^{N-1}})(\frac{(S-1)!}{(S-N)!}) \qquad 17$$

The probability of an empty slot in the combination TDMA-CDMA system is the same as that of the standard Slotted ALOHA system but combination TDMA-CDMA could potentially require fewer slots than the Slotted ALOHA system due to its ability to read tags simultaneously in case of same slot collision (assuming the tags in question do not have a same code collision). Assuming a relatively large number of slots and (high quality) codes, a simple comparison of the Slotted ALOHA and combination TDMA-CDMA systems can be seen below in Figure 5a.

Figure 5a shows the probability of collision (e.g. P(SSC)) when using a Slotted ALOHA-based RFID system versus the probability of collision (e.g. P(TC)) for a combination TDMA-CDMA system with each tag hardcoded with a specific spreading code and the ability to ensure that no more than one tag is using any single spreading code when the number of tags is less than or equal to the number of possible codes. The probability is analyzed for the use of 10 spreading codes, 20 spreading codes, and 50 spreading codes. Both systems use 10 slots. The figure shows the characteristic curve of the collision probabilities and upon observation, it is clear that in terms of probability of collision, combination TDMA-CDMA is superior to Slotted ALOHA, given the ability to control which codes are being used by the tags in the read-range when the number of tags is equal to or less than the number of possible codes.

Figure 5b shows the comparison of collision probability for a Slotted ALOHA system (e.g. P(SSC)) and the collision probability for a combination TDMA-CDMA system (e.g. P(TC)) where the tag either generates its own spreading code or the tag randomly selects a spreading code. This is a more realistic scenario than the one shown in Figure 5a. The curve of the probability of collision for combination TDMA-CDMA systems in this scenario will also be that of precoded EPC if there is no control over which tags are used (i.e. the tags are randomly selected “off the shelf”). It is evident that in terms of collision probability, a relatively large number of spreading codes must be used to have a significant advantage over Slotted ALOHA in this instance. For CDMA-based passive RFID to gain wide acceptance, a highly efficient collision handling protocol must be developed to increase the performance benefit of CDMA-based RFID over TDMA-based RFID.



(a)

(b)

Figure 5. Collision Probabilities. (a) Collision Probability for straight TDMA vs combination TDMA- CDMA (Precoded EPC). (b) Comparison of collision probability for straight TDMA vs combination TDMA-CDMA (code generation or "off the shelf" precoded EPC)

D. COMPARISON OF TDMA, CDMA, AND TDMA-CDMA

All of the above analyses use the assumption that with a given maximum BER, adequate SIR, and true orthogonality of the spreading codes used, the receiver will be able to accurately decode all of the simultaneously responding tags, even if all of the possible spreading codes are being used simultaneously (e.g. N = E). In reality, the receiver will not be able to accomplish this feat, as most sets of spreading codes (larger than two codes) will not be truly orthogonal. To compensate for the lack of true orthogonality, minimum SIR is increased. This means that at a certain number of tag responses, the receiver will be overwhelmed and will not be able to accurately read any of the responding tags, holding all other factors constant. This can be mitigated by increasing SIR. If we refer to the number of simultaneously responding tags (e.g. same-slot colliding tags) as X and we refer to the maximum number of simultaneously responding tags as N_{th}, we can refer to the probability of this kind of collision as $P(X > N_{th})$. Considering this phenomenon, all the collision probabilities of the CDMA systems mentioned above are as follows,

$$for\ SSCDMA\ and\ SSCDMA\ with\ Groups,$$

$$ifE \geq N, \qquad P(Collision) = P(X > N_{th}) \tag{18}$$

$$ifN > N_{th}, \qquad P(X > N_{th}) = 1, \qquad thus, \tag{19}$$

$$P(Collision) = 1$$

$$for\ Combination\ CDMA - TDMA, \qquad ifN \leq S\ and\ N \leq E,$$

$$and\ assuming\ that\ each\ code\ (or\ group\ of\ codes)\ is\ represented\ at\ most\ once,$$

$$P(Collision) = P(X > N_{th}) \tag{20}$$

$$Otherwise,$$

$$P(Collision) = P(X > N_{th})$$

$$+\left(1 - \left(\frac{1}{E^{N-1}}\right)\left(\frac{(E-1)!}{(E-N)!}\right)\left(\frac{1}{S^{N-1}}\right)\left(\frac{(S-1)!}{(S-N)!}\right)\right)$$

$$-\left[P(X > N_{th})\left(1 - \left(\frac{1}{E^{N-1}}\right)\left(\frac{(E-1)!}{(E-N)!}\right)\left(\frac{1}{S^{N-1}}\right)\left(\frac{(S-1)!}{(S-N)!}\right)\right)\right] \qquad 21$$

$$= \ P(X > N_{th}) + \ [1 - P(X$$

$$> N_{th})][1 - (\frac{1}{E^{N-1}})(\frac{(E-1)!}{(E-N)!})(\frac{1}{S^{N-1}})(\frac{(S-1)!}{(S-N)!})]$$

IV. NEW PROTOCOL: BITWISE CDMA (B-CDMA)

This chapter describes a novel medium access control protocol referred to as Bitwise CDMA. A general description of the protocol is provided, an analysis of the tradeoffs of the system is given, a comparison of the system to Slotted ALOHA, straight CDMA, CDMA variants, and TDMA-CDMA is given, two example systems are discussed and analyzed, and a MATLAB simulation to compare Bitwise CDMA and Slotted ALOHA is provided.

A. Description of Protocol

Before the new implementation, referred to as Bitwise CDMA (or B-CDMA) is discussed, it is valuable to assess the shortcomings of the previously mentioned implementations that need to be addressed.

In the Slotted ALOHA system, the obvious disadvantage is that simultaneous tag reading is not possible. However, Slotted ALOHA can potentially outperform CDMA in total number of transmitted bits per read tag. CDMA has the advantage of having far more robustness in high noise environments due to the redundancy used in its encoding.

In the straight CDMA system, tags can be read simultaneously but if a particular spreading code is used by more than one tag at the same time, the tags using that spreading code cannot be accurately read without additional collision handling protocols for same code collisions.

In combination TDMA-CDMA, it is less likely that tags will collide since a collision will only occur when two or more tags share both a slot and a spreading code but this system still suffers from the fact that only a fraction of the slots will read more than one tag at a time, even if all tags could potentially be read simultaneously. For example, if there are N tags and S codes in a TDMA-CDMA system, with N < S and no tags share a spreading code, all the tags could potentially be read within a single slot if the processing gain is sufficiently high. However, all the tags are unlikely to randomly generate the same slot and the interrogator will attempt to read tags at each slot regardless of their presence, thus leading to an inefficiency of time and power.

B-CDMA solves this problem by generating the necessary number of slots on-the-fly while reading the maximum number of tags during each slot and removing colliding tags with the transmission of each encoded data bit. It should be noted that in B-CDMA, “slot” does not refer to an assigned or randomly-generated time slot as it would in a TDMA system. A B-CDMA “slot” simply indicates the time from the beginning of the reflection of the first EPC bit to the conclusion of the reflection of the final EPC bit. The B-CDMA algorithm is as follows:

1. Interrogator powers up tags in read-range and sets bit rate, frequency, etc.
2. Tags reflect X-bits corresponding to their first encoded EPC bit.
3. Interrogator decodes conglomerate signal using each of the possible spreading codes stored within its memory, yielding a vector that is E bits long, where E is the number of spreading codes used in the system. If an interrogator is unable to decode a bit, either due to a collision or a lack of a tag using a spreading code, the bit is assumed to be a “0”. This vector is then placed in its corresponding row in a matrix representing the data of the decoded tags, with each column representing one bit of the EPC of the tag represented by each spreading code.

4. The interrogator transmits the vector to the tags. This vector is referred to as the check vector.
5. The tags check the data bit they just reflected with the bit within the vector corresponding to their spreading code. If the bits disagree, the tag enters no-transmit mode until a new round begins.
6. The tags that have their reflected bit confirmed then reflect the next encoded bit and steps 3-6 are repeated until the E by Y matrix is full, where Y is the length in bits of the EPC used in the system.
7. At this point, the tags that have completed their EPC reflection and have confirmed the accuracy of their reflection with the check vector shut off and the process repeats from step 1 with the tags that entered no-transmit mode in Step 5.
8. Steps 1-7 are repeated until an empty round occurs, indicating that all tags within the read-range have been read. An empty round in this case is when the first one or two decoded bits are indeterminate for all spreading codes.

B. Analysis of B-CDMA

This implementation allows for the interrogator to read the maximum number of tags at each slot and/or read the tag(s) with the highest received power, mitigating the issues resulting from the near-far problem (shadowing). Each subsequent slot reads the next highest number of tags that can be simultaneously read and/or the tag(s) with the next highest received power. The robustness of this implementation is achieved through both the redundancy created by the spreading codes and the ability to eliminate colliding tags before they complete their entire EPC transmission. This error control aspect of B-CDMA differs from the Adaptive Interference Cancellation (AIC) method outlined in previous work [21] in that AIC does not eliminate or even identify potentially colliding tags or very "loud" tags until the end of an inventory round. The error control procedure in B-CDMA is able to totally eliminate interference from colliding tags as soon as they are detected, thus creating an even more robust CDMA-based MAC protocol.

T = Total number of tags in population

N = Number of tags being read in current inventory round

R = Number of tags that have been successfully read and shut off

r = Number of tags found to be colliding and have been removed from current round

n = Number of bits in EPC

At the beginning of the first inventory round,

$$N = T \qquad 22$$

After the first EPC bit is read and tags that were found to collide were instructed to either continue or to wait,

$$N = T - r_0 \qquad 23$$

$$where\ r_0 = number\ of\ collided\ tags\ that\ were\ removed\ (instructed\ to\ enter\ no - transmit\ mode)$$
$$after\ reading\ the\ first\ EPC\ bit, b_0$$

After the second EPC bit is read,

$$N = T - (r_0 + r_1) \qquad 24$$

After the final EPC bit, $b_n$is read,

$$N = T - R_0 + \sum_{i=0}^{n} r_i \tag{25}$$

$$where\ R_0 = number\ of\ tags\ that\ were\ successfully\ read\ and\ instructed\ to\ shut\ off\ at\ the\ end\ of\ the\ first\ round$$

At the end of the second inventory round,

$$N = T - (R_0 + R_1) + \sum_{i=n+1}^{2n} r_i \tag{26}$$

$$where\ R_1 = number\ of\ tags\ that\ were\ successfully\ read\ and\ shut\ off\ at\ the\ end\ of\ the\ second\ round$$

When all of the tags are read, N will be zero and the interrogator, after attempting to decode the next group of tags will find no tags left in the read-range, at which point the interrogator will shut off and the inventory reading will be considered to be complete. This will be evident to the interrogator by the first several decoded bits of all tags being indeterminate.

Beginning with the assumption that all tags will reflect a signal at the same or very close signal power level and assuming the tags are relatively synchronized, the expected number of inventory rounds can be characterized as,

$$ifT \leq E\ and\ spreading\ codes\ are\ not\ reused,$$

$$E[K] = 1 \tag{27}$$

$$where\ K = the\ number\ of\ inventory\ rounds\ to\ read\ all\ ofthe\ tags\ in\ the\ read\ range$$

Assuming no tags use both the same code and contain the same EPC data,

$$if\ E < T \leq 2E, \qquad E[K] = 2 \tag{28}$$

As can be extrapolated from the above equations,

$$in\ general,$$

$$E[K] = \frac{T}{E}, rounded\ up\ to\ the\ nearest\ integer \qquad 29$$

Of course, without perfect synchronization and/or without highly orthogonal codes, T has an upper bound that can be increased by increasing SIR (e.g. increasing processing gain).

Since the proposed system removes colliding tags after reading each EPC data bit, the SIR of the system increases when each bit is read until all the colliding tags are removed. The caveat of the tags not using the same code and containing the same EPC data is removed by the use of short handles, which is discussed in later sections.

Assuming an equal likelihood of a tag transmitting an encoded EPC data bit “1” or “0”, and a large tag population with each tag equally likely to use any of the possible spreading codes, the number of tags removed at the end of each inventory round can be expressed by,

$$P(bit = 1) = P(bit = 0) = 0.5 \qquad 30$$

$$P(decoded\ bit = 1) = P(decoded\ bit = 0) = 0.5 \qquad 31$$

$$P(EPC\ bit = Check\ Vector\ bit) = P(EPCbit \neq Check\ Vector\ bit) \qquad 32$$

$$= 0.5$$

And since a tag removes itself if the EPC bit that is currently being read disagrees with its corresponding bit within the test vector, the number of colliding tags that enter no-transmit mode at each bit can be represented by,

$$r = \lceil 0.5N' \rceil \qquad 33$$

$$Where\ N' = number\ of\ collided\ tags$$

The total number of tags rejected at the end of each inventory round is therefore approximately,

$$r_T = (0.5)N' + (0.5)^2N' + \cdots + (0.5)^nN' \qquad 34$$

$$from\ the\ first\ bit\ until\ the\ bit\ at\ which\ all\ colliders\ are\ removed$$

Thus, the number of tags read at the end of each inventory round can be stated as,

$$R = T - r_T \qquad 35$$

The addition of the check vector after each EPC bit transmitted by the tags may increase the number of bits transmitted by the interrogator compared to the ideal scenario of Slotted ALOHA (e.g. number of tags = number of slots, no collisions) but from an information theory and interrogator power usage perspective, this system is more efficient in that it receives information for every transmission and has better performance if collisions occur. The length of the vector has some predictability and can therefore be compressed but since the check vector may be too short for compression to significantly improve performance, its investigation is suggested for future work. This predictability arises from the fact that if a bit cannot be read, due to a collision or a lack of a tag using the spreading code corresponding to one or more bits in the check vector, it is assumed to be a “0”. This assumption makes it more likely that any given bit in the check vector will be a “0” than is likely to be a “1”.

C. B-CDMA Example 1

In this implementation, the tag acts as an n+2 state finite state machine (states n, n-1, n-2, …, 0, Z) where n = number of bits in EPC. The initial state of the tag is “State n” and the final state is “State *Z*”. In State n, the tag reflects its first encoded EPC bit to the interrogator. The tag then moves to State n-1. In State n-1, the tag receives the check vector from the interrogator and uses an XNOR on its corresponding bit in the check vector and on its EPC bit n to set its “Reflection” flag. If the reflection flag is set, signifying that the first encoded EPC bit was successfully received by the interrogator, the tag reflects its n-1 bit to the interrogator and moves to State n-2. In State n-2, if the flag is not set, the tag does not reflect any data (e.g. the tag enters no-transmit mode). In State n-2, the EPC bit is only reflected if the reflect flags for both bit n and bit n-1 were set. This continues until the tag reaches its State 0. In State 0, if all previous reflect flags were set, the tag reflects its bit 0 of EPC data. Finally, after reflecting the final EPC bit, the tag receives one more check vector in State Z and it sets its reflect flag one final time for this inventory round. If the flag is set, the tag remains in State Z, which is its “off” state. If the flag is not set, the tag goes back to State n and retries reflecting bit n. An example system using 16 spreading codes and 5 3-bit tags can be seen below in Table 1.

Table 1. Example B-CDMA System Tags

Tag Number	Data	Spreading Code Number
1	[001]	3
2	[010]	6
3	[011]	7
4	[100]	12
5	[110]	12

Using the above example with 5 tags, 3 of which use unique spreading codes, and 2 of which use colliding spreading codes, we can create an example to illustrate how this protocol functions. This example can be seen in Figure 6.

	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1							...
2							...
3							...
4							...
5							...
6							...
7							...
8							...
9							...
10							...
11							...
12							...
13							...
14							...
15							...
16							...

6(a)

In the diagram above, each entry represents the bit decoded by the interrogator using each one of the spreading codes on the vertical axis, during each one of the EPC bit slots shown on the horizontal axis. A strong “1” is represented by a “**1**”, a strong “0” is represented by a “**0**”, and an indeterminate bit that is set as a “0” is represented by a “0”. A “hit” on a spreading code is represented using an “*”. After the first bit (MSB) is reflected by the tags,

	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0						...
2	0						...
*3	**0**						...
4	0						...
5	0						...
*6	**0**						...
*7	**0**						...
8	0						...
9	0						...
10	0						...
11	0						...
**12	**1**						...
13	0						...
14	0						...
15	0						...
16	0						...

6(b)

This first column is used as the first check vector that sets the reflect flags of the tags. In this first reflection, the interrogator has found hits on codes 3, 6, 7, and 12 but since tags 4 and 5 sent the same bit using the same code, the interrogator has no indication that there is a collision. After the next reflection,

	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0					...
2	0	0					...
*3	**0**	**0**					...
4	0	0					...
5	0	0					...
*6	**0**	**1**					...
*7	**0**	**1**					...
8	0	0					...
9	0	0					...
10	0	0					...
11	0	0					...
**12	**1**	0					...
13	0	0					...
14	0	0					...
15	0	0					...
16	0	0					...

6(c)

Again, the interrogator has found hits using codes 3, 6, 7, and 12 but since tags 4 and 5 reflected opposite bits using the same code, there is a collision and tag 5 is removed. The SIR for tags 1, 2, 3, and 4 have now increased due to the removal of tag 5. Now, tag 5 is in its “no transmit” state until the next round begins. After the next reflection,

	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0				...
2	0	0	0				...
*3	**0**	**0**	**1**				...
4	0	0	0				...
5	0	0	0				...
*6	**0**	**1**	**0**				...
*7	**0**	**1**	**1**				...
8	0	0	0				...
9	0	0	0				...
10	0	0	0				...
11	0	0	0				...
**12	**1**	0	**0**				...
13	0	0	0				...
14	0	0	0				...
15	0	0	0				...
16	0	0	0				...

6(d)

At this point, the third and final check vector is transmitted. Tags 1, 2, 3, and 4 have reflected their full EPCs and the interrogator has accurately read and stored their data. As such, tags 1, 2, 3, and 4 enter their “off” state. Since the reflect flag was not set for tag 5, it continues to transmit in the next round automatically.

	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0	0			...
2	0	0	0	0			...
3	**0**	**0**	**1**	0			...
4	0	0	0	0			...
5	0	0	0	0			...
6	**0**	**1**	**0**	0			...
7	**0**	**1**	**1**	0			...
8	0	0	0	0			...
9	0	0	0	0			...
10	0	0	0	0			...
11	0	0	0	0			...
*12	**1**	**0**	0	**1**			...
13	0	0	0	0			...
14	0	0	0	0			...
15	0	0	0	0			...
16	0	0	0	0			...

6(e)

Now, the only tag in the system is tag 5 and code 12 is the only code with a “hit”. The process continues until tag 5 is read. Tag 5’s SIR is significantly higher than it was in the previous round because it is now the only tag responding.

	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0	0	0	0	...
2	0	0	0	0	0	0	...
3	**0**	**0**	**1**	0	0	0	...
4	0	0	0	0	0	0	...
5	0	0	0	0	0	0	...
6	**0**	**1**	**0**	0	0	0	...
7	**0**	**1**	**1**	0	0	0	...
8	0	0	0	0	0	0	...
9	0	0	0	0	0	0	...
10	0	0	0	0	0	0	...
11	0	0	0	0	0	0	...
*12	**1**	0	**0**	**1**	**1**	**0**	...
13	0	0	0	0	0	0	...
14	0	0	0	0	0	0	...
15	0	0	0	0	0	0	...
16	0	0	0	0	0	0	...

6(f)

Figure 6. Example B-CDMA System

The interrogator then begins another inventory round but will conclude that there are no tags in range if the first several slots are all indeterminate.

The state machine of the 3-bit tag can be seen in Figure 7 below.

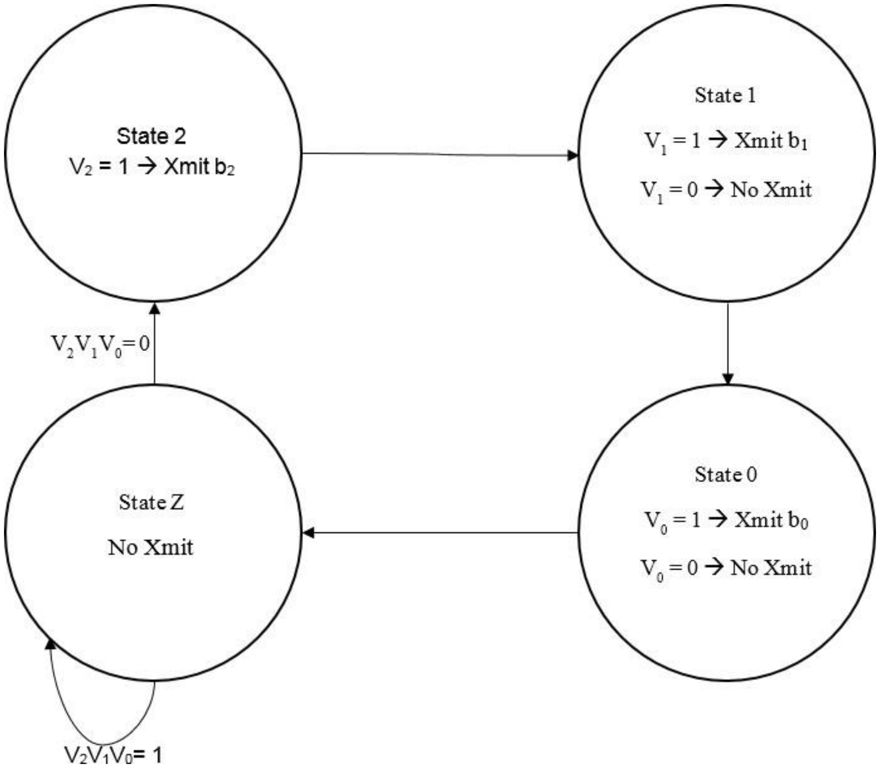

State 2
$V_2 = 1 \rightarrow$ Xmit b_2
State 1
$V_1 = 1 \rightarrow$ Xmit b_1
$V_1 = 0 \rightarrow$ No Xmit
$V_2V_1V_0=0$
State Z
No Xmit
State 0
$V_0 = 1 \rightarrow$ Xmit b_0
$V_0 = 0 \rightarrow$ No Xmit
$V_2V_1V_0= 1$

Figure 7. FSM of tag in B-CDMA System. If V = 1, the "Reflection" flag is set to be true. If V = 0, the "Reflection" flag is set to be false

D. Comparison of B-CDMA, TDMA, SSCDMA, and TDMA-CDMA

In a Slotted ALOHA system, the expected value of the number of tags read in each inventory round is equal to the number of slots in the system, as long as the number of tags is equal to or less than the number of slots. If the number of tags is fewer than the number of slots, there will be an inefficiency in that there will be empty slots in addition to the final empty inventory round that indicates that all tags within range have been read successfully. Given the probability of collision and probability of empty slots in a Slotted ALOHA system, the expected number of inventory rounds is related to the number of collisions. A larger number of tags will cause a higher probability of collision which in turn causes a larger number of slot reassignments, creating a need for more inventory rounds, and by extension a larger number of slots, causing a larger number of tag transmissions that do not result in accurately read tags, thus decreasing efficiency.

Similarly, in the straight CDMA and grouped CDMA systems, maximum efficiency is reached when the number of tags is equal to the number of codes, with no overlap in code usage. While in these CDMA implementations, there are no empty slots, there still exists the adverse effects of the near-far problem (NFP) in the form of shadowing. The NFP is caused by users in a CDMA system transmitting at significantly different enough power levels that the SNR of some users' transmission is so high that one or more of the users cannot be accurately communicated with [42]. In cellular systems, this problem is mitigated by individual cellular devices adjusting their power output, as dictated by the control protocol from the base station [43]. Adjusting the power output level is not an option for passive CDMA and is therefore not solved by either straight CDMA or grouped CDMA implementations without additional collision handling protocols. Combination TDMA-CDMA outperforms Slotted ALOHA and straight (or grouped) CDMA when it comes to collision probability but still suffers from the near far problem and suffers from the inefficiency of empty slots.

B-CDMA has a solution for the near-far problem in CDMA-based passive RFID. There are several tag population configurations in which the near-far problem can be observed. First, for simplicity, a system with only 2 tags will be discussed. In this situation, if one tag has a much higher reflected power level than the other tag, 2 possibilities exist: the tag with the lower amount of reflected power receives the instruction from the interrogator but its reflection is low enough in power that it is below the noise floor of the interrogator or the reflection is low enough in power that its SNR is too low for proper tag reading but high enough to decrease the SIR of the tag with the higher reflected power. In the first case, the check vector will only contain information about the EPC of the tag with the larger amount of reflected power and the tags with lower reflected power will thus automatically remove themselves and wait for the beginning of the next inventory round. In the second case, where the near-far problem is clearly evident (e.g. the presence of the low reflected power tag adversely affects the SIR of the higher power tag and vice versa), the negative effects resulting from the lower power tag will only be present for the first bit or the first 2 bits (see Equations 27-29 regarding expected value of rejected tags) before being automatically removed. If the low reflected power tag is high enough that it can be decoded but decreases the SIR of the higher reflected power tag, the higher reflected power tag will automatically be removed after the first bit or the first two bits are read and will wait until the next inventory round. If there are more than two tags, with a variety of reflected power levels, some will be accurately read and those that are not will be removed after the completed reading of each bit, with approximately 50% of the colliding tags being removed after each decoded bit, given a random data distribution.

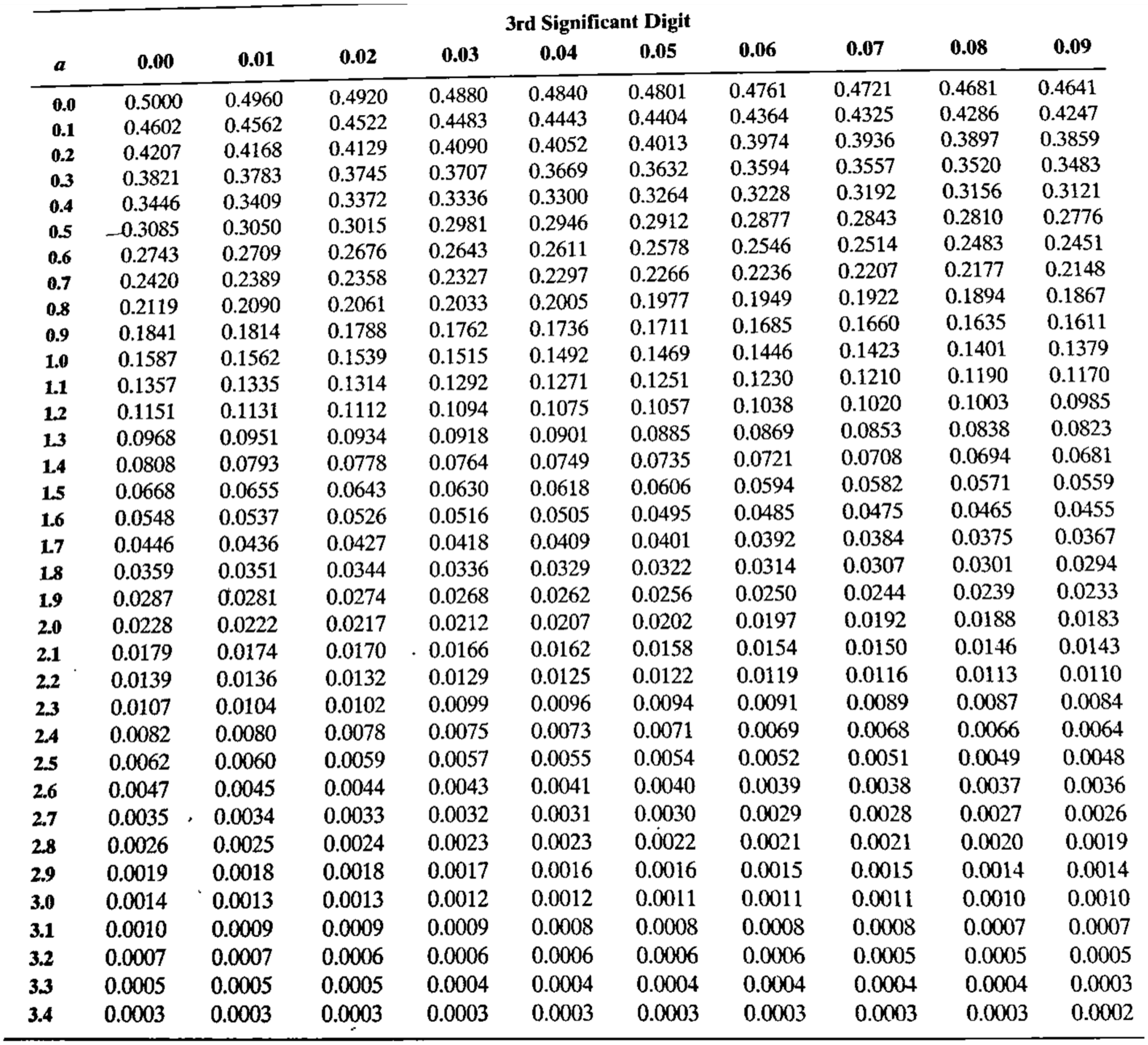
E. Comparison of B-CDMA and Slotted ALOHA

In Slotted ALOHA based RFID systems, a collision occurs when multiple tags generate the same time slot. A collision is detected because the tag handles are different. When such a collision occurs, all the tags that have yet to be read are instructed to regenerate a new slot (the number of possible slots changes at this point in some variants of Slotted ALOHA) and the process continues. In the previously discussed CDMA-based RFID systems, a collision occurs when the same code is selected in the case of straight and grouped CDMA and a collision occurs when both the same code and the same slot are selected in combination TDMA-CDMA. A collision in B-CDMA is not as straightforward and is based on several factors including the number of possible codes, the processing gain, the number of tags in the system, and the acceptable bit error rate. The relationship between SIR, processing gain, and number of tags can be seen in Equation 36 and the relationship between bit error rate and SIR can be seen in Equation 37, where the Q function is tabulated in Figure 8 below. These equations are also under the assumption that tag reflections are modulated using amplitude shift keying.

$$SNR(After\ Despreading) = \frac{G_P}{N-1} \qquad 36$$

$$BER = P(Bit\ Error) = Q(\sqrt{\mathrm{SIR}}) \qquad 37$$

$$Where\ Gp = Processing\ Gain$$



					3rd Significant Digit					
a	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
0.7	0.2420	0.2389	0.2358	0.2327	0.2297	0.2266	0.2236	0.2207	0.2177	0.2148
0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
1.2	0.1151	0.1131	0.1112	0.1094	0.1075	0.1057	0.1038	0.1020	0.1003	0.0985
1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
3.0	0.0014	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
3.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0002

Figure 8. The Q function (Gaussian Distribution) [12]

Some examples of required processing gain to achieve a desired BER for 5, 10, 20, 50, 100, and 200 tags can be seen below in Table 2.

Table 2. Required processing gain for various desired BER

		Required Processing Gain					
Bit Error Rate	**Required SNR After Despreading**	**5 Tags**	**10 Tags**	**20 Tags**	**50 Tags**	**100 Tags**	**200 Tags**
1.00E-03	3.73	14.92	33.57	70.87	182.77	369.27	742.27
1.00E-04	4.27	17.08	38.43	81.13	209.23	422.73	849.73
1.00E-05	4.76	19.04	42.84	90.44	233.24	471.24	947.24
1.00E-06	5.2	20.8	46.8	98.8	254.8	514.8	1034.8
1.00E-07	5.61	22.44	50.49	106.59	274.89	555.39	1116.39
1.00E-08	6	24	54	114	294	594	1194

The amounts of processing gain listed in Table 2 are for the desired BER when reading the listed number of tags simultaneously, each using a different spreading code. In B-CDMA, the processing gain must be large enough that all possible codes can be read simultaneously with the maximum desired BER.

F. B-CDMA Example 2

There are several possible scenarios in B-CDMA systems: Not all codes are used and each used code is used exactly once, not all codes are used and some codes are used more than once, all codes are used and each code is only used once, all codes are used and some codes are used more than once, and the scenario where all codes are used more than once. These scenarios are examined using the following assumptions:

1. E = The number of codes used in the system
2. N = Number of tags

3. P = Processing gain
4. BER_{max} = Maximum acceptable BER
5. P is sufficiently large such that requirements for BER are met when E tags are reflecting simultaneously, assuming each tag is using a different code
6. All the tags are reflecting at a similar power level

Examining a 3-bit tag system with 8 unique spreading codes with a processing gain large enough to have an acceptable BER when simultaneously using all 8 spreading codes, we can analyze the following scenarios:

Scenario 1: 8 3-bit tags, each with a unique spreading code and unique data

Given the above assumptions, it can be concluded that all tags will be read simultaneously in the first round. Unique data is assumed (e.g. Tag 0 Data = 000, Tag 1 Data = 001, Tag 2 Data = 010, ... , Tag 7 Data = 111) because otherwise, handles are required (the addition of handles are discussed later in this paper).

Scenario 2: 8 groups of 8 3-bit tags with each group using a unique spreading code and each tag within each group containing unique data.

In this scenario, the 8 groups contain the tags listed in Table 3

Table 3. Example Tag Data

Group 0		Group 1		...	Group 7	
Tag Number	**Data**	**Tag Number**	**Data**	...	**Tag Number**	**Data**
Tag 0	[000]	Tag 0	[000]		Tag 0	[000]
Tag 1	[001]	Tag 1	[001]		Tag 1	[001]
Tag 2	[010]	Tag 2	[010]		Tag 2	[010]
Tag 3	[011]	Tag 3	[011]		Tag 3	[011]
Tag 4	[100]	Tag 4	[100]		Tag 4	[100]
Tag 5	[101]	Tag 5	[101]		Tag 5	[101]
Tag 6	[110]	Tag 6	[110]		Tag 6	[110]
Tag 7	[111]	Tag 7	[111]		Tag 7	[111]

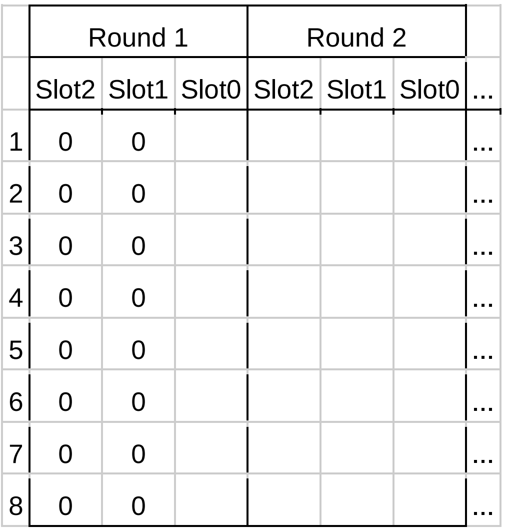
Assuming the tags in Group 0 use Spreading Code 0, the tags in Group 1 are using Spreading Code 1, and so on, it can be seen that after the tags reflect their first bit, the result will be indeterminate since half of the tags are reflecting a “0” and half are reflecting a “1”.

	Round 1			Round 2			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0						...
2	0						...
3	0						...
4	0						...
5	0						...
6	0						...
7	0						...
8	0						...

9(a)

The interrogator transmits an all-zero check vector and all tags with a “1” first bit (Tags 4-7 in each group) enter their no-transmit state until the new round begins.

The second bit of half of the remaining tags is a “0” and the second bit of the other half is a “1”. As such, the result will again be indeterminate, and the interrogator will transmit another all-zero check bit.



	Round 1			Round 2			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0					...
2	0	0					...
3	0	0					...
4	0	0					...
5	0	0					...
6	0	0					...
7	0	0					...
8	0	0					...

9(b)

Again, the remaining tags that have a “1” as their second bit (Tags 2 and 3 in each group) enter no-transmit mode, and the system continues to the next bit.

Again, the last 2 tags in each group, Tag 0 and Tag 1 have a bit disagreement so the interrogator assumes the bit to be a “0” and sends another all-zero check vector.

	Round 1			Round 2			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0				...
2	0	0	0				...
3	0	0	0				...
4	0	0	0				...
5	0	0	0				...
6	0	0	0				...
7	0	0	0				...
8	0	0	0				...

9(c)

Tag 0 of each group is now read and shuts off. The new round can now begin and Tags 1-7 re-enter transmit mode.

In the first bit of the second round, four tags are reflecting a “1” and three tags are reflecting a “0”. Assuming a 50% threshold, the interrogator will see this first bit as a “1”.

	Round 1			Round 2			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0	**1**			...
2	0	0	0	**1**			...
3	0	0	0	**1**			...
4	0	0	0	**1**			...
5	0	0	0	**1**			...
6	0	0	0	**1**			...
7	0	0	0	**1**			...
8	0	0	0	**1**			...

9(d)

At this point, Tags 1-3 enter no-transmit mode and Tags 4-7 reflect their next bit. Tags 4 and 5 are reflecting a “0” and Tags 6 and 7 are reflecting a “1” so the interrogator interprets this bit as a “0”.

	Round 1			Round 2			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0	**1**	0		...
2	0	0	0	**1**	0		...
3	0	0	0	**1**	0		...
4	0	0	0	**1**	0		...
5	0	0	0	**1**	0		...
6	0	0	0	**1**	0		...
7	0	0	0	**1**	0		...
8	0	0	0	**1**	0		...

9(e)

At this point, Tags 6 and 7 enter no-transmit mode and Tags 4 and 5 reflect their next bit which again results in an indeterminate bit and the interrogator transmits an all-zero check vector, thus indicating that Tag 4 has been read correctly and can be shut off and Tag 5 waits for the next round to begin.

	Round 1			Round 2			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	0	**1**	0	0	...
2	0	0	0	**1**	0	0	...
3	0	0	0	**1**	0	0	...
4	0	0	0	**1**	0	0	...
5	0	0	0	**1**	0	0	...
6	0	0	0	**1**	0	0	...
7	0	0	0	**1**	0	0	...
8	0	0	0	**1**	0	0	...

9(f)

The first bit of round three has an equal number of reflected “1”’s and “0”’s so Tags 5-7 enter no-transmit mode, the second bit has a majority of “1”’s so Tag 1 enters no-transmit mode, and the third bit is indeterminate so Tag 3 enters no-transmit mode, and Tag 2 is read successfully so it shuts off. In Round four, the first bit has a majority of “1”’s so Tags 1 and 3 enter no-transmit mode, the second bit has a majority of “1”’s so Tag 5 enters no-transmit mode, and the third bit is indeterminate, so Tag 7 enters no-transmit and Tag 6 is read successfully and shuts off.

	Round 3			Round 4			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	**1**	0	**1**	**1**	0	...
2	0	**1**	0	**1**	**1**	0	...
3	0	**1**	0	**1**	**1**	0	...
4	0	**1**	0	**1**	**1**	0	...
5	0	**1**	0	**1**	**1**	0	...
6	0	**1**	0	**1**	**1**	0	...
7	0	**1**	0	**1**	**1**	0	...
8	0	**1**	0	**1**	**1**	0	...

9(g)

In Round 5, the first bit is indeterminate so Tags 5 and 7 enter no-transmit mode, the second bit is indeterminate, so Tag 3 enters no-transmit mode. Tag 1 is the only reflection contributing to the third bit and is read successfully and therefore shut off.

In Round 6, the first bit is determined to be a "1" so Tag 3 enters no-transmit mode, the second bit is indeterminate so Tag 7 enters no-transmit mode, and Tag 5 is the only tag contributing to the third bit and is read successfully and shut off.

	Round 5			Round 6			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	0	**1**	**1**	0	**1**	...
2	0	0	**1**	**1**	0	**1**	...
3	0	0	**1**	**1**	0	**1**	...
4	0	0	**1**	**1**	0	**1**	...
5	0	0	**1**	**1**	0	**1**	...
6	0	0	**1**	**1**	0	**1**	...
7	0	0	**1**	**1**	0	**1**	...
8	0	0	**1**	**1**	0	**1**	...

9(h)

In Round 7, the first bit is indeterminate, so Tag 7 enters no-transmit mode and Tag 3 is the only tag contributing to the second and third bit and is read successfully and therefore shuts off. In Round 8, Tag 7 is the only tag in the system and is therefore read successfully and shuts off.

	Round 7			Round 8			
	Slot2	Slot1	Slot0	Slot2	Slot1	Slot0	...
1	0	**1**	**1**	**1**	**1**	**1**	...
2	0	**1**	**1**	**1**	**1**	**1**	...
3	0	**1**	**1**	**1**	**1**	**1**	...
4	0	**1**	**1**	**1**	**1**	**1**	...
5	0	**1**	**1**	**1**	**1**	**1**	...
6	0	**1**	**1**	**1**	**1**	**1**	...
7	0	**1**	**1**	**1**	**1**	**1**	...
8	0	**1**	**1**	**1**	**1**	**1**	...

9(i)

Figure 9. Example B-CDMA System

At this point, all 64 tags have been read successfully, with 8 tags being read simultaneously in each round.

Of course, this system is limited by the constraint that all tags using the same spreading code must have unique data. This constraint is removed by the use of handles, just as in Slotted ALOHA systems to detect multiple tags that contain the same data. Handles can be integrated into this system by having tags that, upon completion of a successful reading, reflect their handles between rounds. Unlike ALOHA, this handle usage does not stop a tag from reflecting its data if a collision has occurred. It only has tags transmit its handle after the completion of transmission, thus allowing the reading to still occur. Example 2 also highlights B-CDMA’s ability to use shorter handles than would be necessary in an ALOHA system. Using the tag data from the example, each tag with data [000] would need a unique handle, each tag with data [001] would need a unique handle, etc. in a Slotted ALOHA system. Due to the use of the spreading codes, B-CDMA does not require handles for the given example. The required length of the handles in B-CDMA systems is therefore shorter than the required length of the handles in Slotted ALOHA systems.

G. MATLAB Simulation

An investigation into the number of bits transmitted from the interrogator as well as the total number of necessary inventory rounds can give a sense of the amount of time it will take to successfully read a certain number of tags. The number of bits transmitted from the interrogator will also give an indication of how much power the interrogator consumes.

Given that all implementations will require the initial power-up, synchronization, and initial query at the beginning of the initial inventory round, the time it consumes to do so does not affect the speed comparison of the systems. Each interrogator-to-tag communication is required to begin with a 12-bit frame-sync. In the ALOHA system, the amount of time to read an inventory of tags is based on the EPC length, number of tags, number of QueryReps (e.g. decrement command), the size of the RNG (dictated by the number of slots), number of ACKs (e.g. acknowledgement of tags), number of slots, and number of collisions. Assuming all systems use the same data rate, time can be measured in bits. Not including frame-sync, QueryRep is 4 bits long and ACK is 18 bits.

The simulation is done with the constraints listen in Table 4.

Table 4. MATLAB simulation constraints

EPC Size	16 Bits
RNG Size	Based on Number of Slots
Processing Gain	Based on Number of Codes
Acceptable BER	10E-4 ~ 10E-5

The B-CDMA System is tested using random code selection and is tested using the constraints outlined in Table 5.

Table 5. MATLAB B-CDMA simulation constraints

Number of Codes	Processing Gain
2	6
4	14
8	32
12	48
16	64
20	82
24	100

The ALOHA System is tested using the constraints listed in Table 6.

Table 6. MATLAB ALOHA simulation constraints

Number of Slots	RNG Size
2	1
4	2
8	3
12	4
16	5
20	6
24	6

The number of bits in the communication in B-CDMA is based on the length of the check vector (determined by the number of codes), the processing gain (determined by the number of codes and the acceptable BER), the number of bits of EPC in each tag, and the number of required rounds. Note, the total number of bits reflected from the tags does not determine the amount of time the system consumes because reflections are occurring simultaneously. For randomly selected codes, the expected value of number of tags read per round is equal to the number of codes, the check vector length is equal to the number of codes, and the number of time-consuming bits per round equals the number of bits in EPC times the processing gain.

The results of the Slotted ALOHA simulation can be seen below in Figure 10.

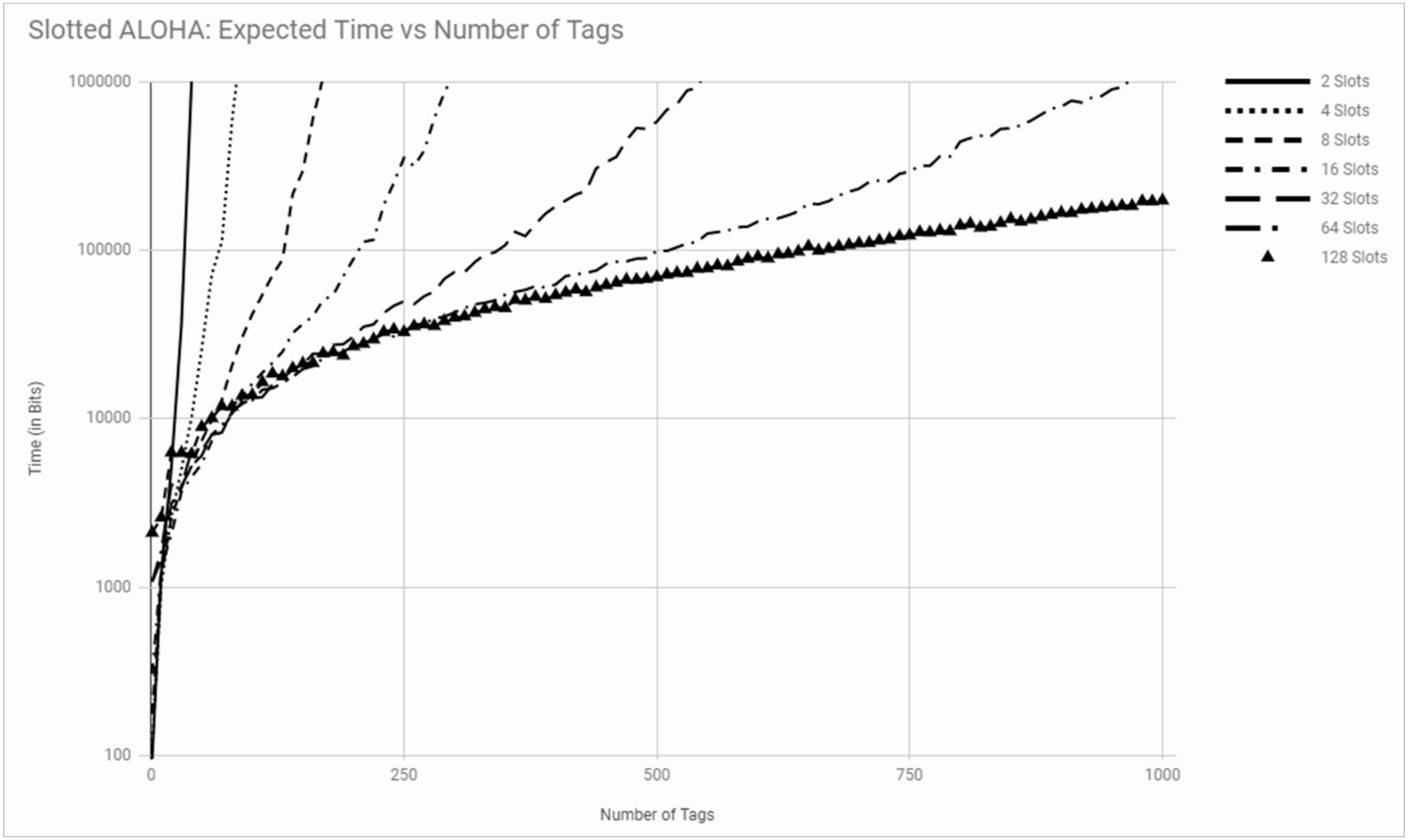


Figure 10. Slotted ALOHA protocol speed in terms of time to read 1-1000 tags (measured in bits).

The results of the B-CDMA simulation can be seen in Figure 11.

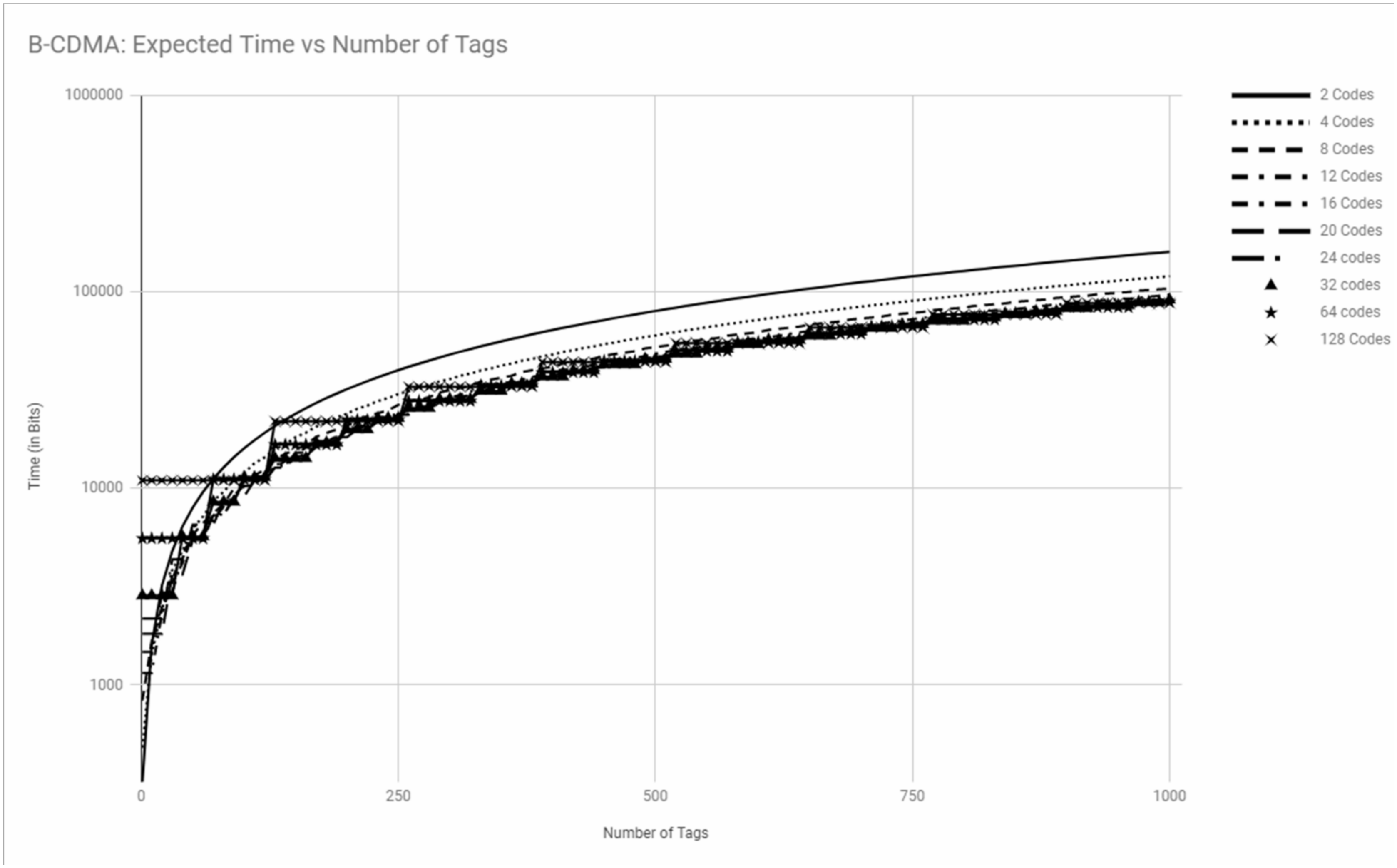


Figure 11. B-CDMA protocol speed in terms of time to read 1-1000 tags (measured in bits).

Several conclusions can be drawn from these results. An incremental change in number of tags affects the speed of Slotted ALOHA systems but does not in the B-CDMA system. Additional tags decrease the speed of the B-CDMA system only when enough tags are added to warrant an additional round. The curves indicate that some ALOHA implementations are faster than B-CDMA only for the cases of very few tags relative to the number of slots. Overall, B-CDMA is faster. Furthermore, some B-CDMA implementations are faster than others, implying that there exists an optimum number of codes (and optimum processing gain). The trend of this algorithm for increasing number of tags implies that optimization occurs with increasing the number of codes. For this kind of algorithm, optimization will be achieved by increasing the number of codes to as many as is viable for final implementation in an RFID tag.

The number of time-consuming bits correlates directly to the time as well as the interrogator power consumption.

V. VERILOG IMPLEMENTATION

This chapter discusses the Verilog implementation of both a Slotted ALOHA tag and a Bitwise CDMA tag. The code itself is discussed for each tag as well as the testbench for each implementation, which is used as both the test and verification tool as well as a simulation of the interrogator. A comparison of area and power utilization of both implementations is also explored. Finally, the development of an FPGA-based RFID research and development system is proposed.

A. TDMA-based System

To begin the design of the RFID tag IC, a simple TDMA system is developed. In this basic system, the tag has a 3-bit EPC and 16 slots. To determine its TDMA slot, the tag takes a 4 bit measurement of the ambient noise in the environment, relative to some value (4'b0000) in the simulation and uses that as its randomly generated number. This system has 4 instructions: "Power Up" and Set RNG, Decrement RNG, Return EPC, and Write EPC. Each instruction is 8 bits long and is sent from the interrogator to the tag. A Verilog testbench is used as the interrogator and also provides a random 4-bit value to be used as ambient noise. The interrogator transmits each instruction serially and the tag places each incoming bit into a shift register until it receives a full instruction.

At the beginning of an "inventory round", the interrogator instructs the tag to set its RNG and provides the 4-bit ambient noise value. Once the tag registers the instruction, the tag sets its 4-bit random number as the random noise value. The interrogator then sends the Decrement RNG command repeatedly, instructing each tag to decrement their 4-bit random number by 1. Once the tag's random number has reached 0, the tag indicates to the interrogator that it is ready to reflect its EPC by reflecting its handle, the interrogator then acknowledges the tag by transmitting the Read EPC instruction and the tag responds with its 3-bit EPC. Once the EPC is read, the tag is then instructed to shut off until a new round begins. The tag can also have its EPC written by using the Write EPC instruction. The Write EPC instruction does not require the use of the random number generator.

The behavioral simulation was done using Icarus Verilog and using Vivado.

B. B-CDMA System

To demonstrate the functionality of the proposed protocol, a Verilog description of an IC with the necessary functionality is written and tested. The tag code is a finite state machine as outlined in the B-CDMA section. It is for a 3-bit tag with 5 states: State 3, State 2, State 1, State 0 and State Z. Rather than design a separate Verilog code as the interrogator, the testbench for the tag code is used as the interrogator. The testbench provides a clock to the tag code, the period of which is half that of the tag's state machine clock. At the positive edge of the first clock cycle, the tag reflects its MSB (b2) and moves to the next state and waits to receive the check vector. The testbench then decodes the data bit, creates the check vector based on what it received, and transmits the check vector to the tag. At this point, the tag performs an XNOR on its corresponding bit in the vector and the bit that it reflected in the previous state to set the reflection flag. If the flag is set, the tag reflects its next bit (b1) to the testbench. If not, it does not have an output (1'bz). This process continues for the next and final bit (b0) in State 0 and the tag moves to State Z. If the reflection flag was set for bits b2, b1, and b0, the tag remains in State Z with no output (1'bz) until instructed otherwise. If the tag's reflection was cut short at some point (e.g. at least one of the flags was not set and the tag entered no-transmit), the tag moves from State Z back to state b2 and repeats this process until it gets to State Z and stays there.

The interrogator (testbench) despreads the tag reflections by XORing the incoming conglomerate data stream from all the tags with each possible spreading code. The interrogator then determines if the result of each XOR has a majority of “1”’s or a majority of “0”’s. If the result has a majority of “1”’s, the interrogator decides that the reflected bit was a “1”, otherwise the interrogator decides that the reflected bit was a “0”. An indeterminate is detected when there is no clear majority of “1” or “0”.

Testing of B-CDMA Verilog Implementation

B-CDMA is implemented in Verilog to test its functionality using 2 tags. This system contains 4 Verilog files: “ctag.v”, “ctag2.v”, “MAC.v”, and “ctag_tb.v”. “ctag.v” and “ctag2.v” are 3-bit tags that follow the state machine outlined in the sections above and have a processing gain of 16. “MAC.v” is used to simply simulate the signal combination that would occur if the signals were transmitted over the air. “ctag_tb.v” although technically a Verilog testbench, can be considered a simulation of a B-CDMA interrogator. The instructions, reflections, and test vector are all written to transmit on a parallel bus for simplicity but are changed to serial before the HDL is ready for ASIC implementation.

A clock is fed from the interrogator (testbench) through MAC to ctag and ctag2. At the positive edge of the clock, the tag attempts to reflect an encoded bit. Initially, the tags attempt to reflect their first encoded bit. At the negative edge of the clock, the interrogator decodes the reflection, and creates the check vector with its first entry being the bit it has decoded for ctag and the second entry being the bit it has decoded for ctag2, and transmits the vector to ctag and ctag2 through MAC. At the next positive clock edge, the tags set their Reflect flags and determine whether they are going to continue reflecting or enter no-transmit mode. If a tag enters no-transmit mode, it continues through the state machine until it reaches the final state and then returns to its initial state once the next inventory round begins. Once a tag has been read successfully, it enters the off state (State Z) where it stays until instructed to power on again. Icarus Verilog is used to test the logic of the code and Vivado is used to test the functionality. The spreading codes for this implementation were generated using Walsh-Hadamard matrices [44-45].

The Vivado generated timing simulation of the case with the conditions listed in Table 7 can be seen below in Figure 12.

Table 7. Tag data and spreading codes used in Vivado simulation

Tag	Data	Spreading Code	Code Type
ctag	[101]	[1010101010101010]	Walsh
ctag2	[110]	[1001100110011001]	Walsh

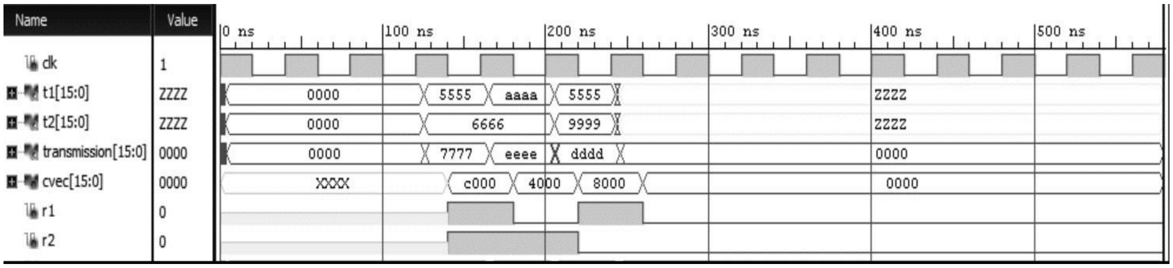


Figure 12. Vivado timing simulation (non-collided tags)

In Figure 12, clk is the clock, t1 and t2 are the reflections from ctag and ctag2 respectively, transmission is the conglomerate reflection, cvec is the check vector, and r1 and r2 are the decoded EPCs of ctag and ctag2, respectively. At the positive edge of the fourth clock cycle, the tag reflections can be seen in t1 and t2. Just after the negative clock edge, the decoded reception at the interrogator can be seen in r1 and r2 between 140 and 260 nS. At the positive edge of the seventh clock cycle, t1 and t2 indicate that both tags have confirmed their full EPC reflection with the check vector and have shut off.

To show the collision handling capabilities, another test was run with the conditions listed in Table 8. The timing simulation of this test can be seen in Figure 13.

Table 8. Tag data and spreading codes used in Vivado simulation

Tag	Data	Spreading Code	Code Type
ctag	[101]	[1010101010101010]	Walsh
ctag2	[110]	[1010101010101010]	Walsh

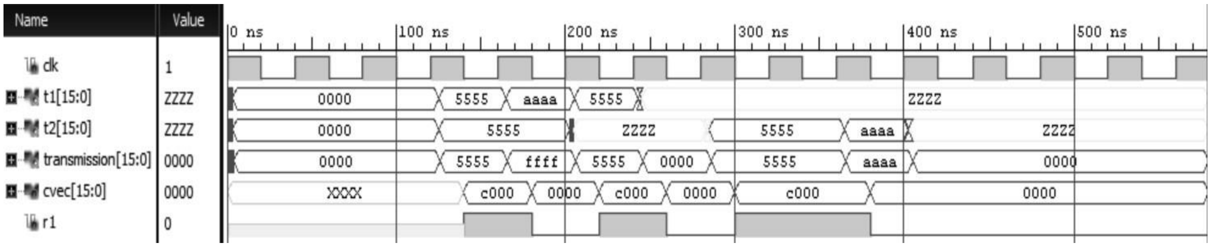


Figure 13. Vivado timing simulation (collided tags)

As can be seen in Figure 13, both tags use the same spreading code and share their first data bit. At the positive edge of the fourth clock cycle, both tags reflect their encoded MSB. At the negative edge, the check vector indicates to the tags that the interrogator received a “1” and since both tags did send an encoded “1”, they both reflect their next encoded bit at the positive edge of the fifth clock cycle. Since the second bit is indeterminate on the interrogator’s end, it assumes that the bit is a “0”, transmits its check vector to the tags, and at the positive edge of the sixth clock cycle, ctag reflects its final bit and ctag2 enters no-transmit mode. The entering of no-transmit mode can be seen in t2 just after 200 nS. The data read from ctag can be seen in r1 from approximately 140 nS to 260 nS. Just after the positive edge of the eighth clock cycle, ctag2 retries its transmission and succeeds. In this scenario, as opposed to the one shown in Figure 11, both tag responses can be seen in r1 since both tags are using the same spreading code. The data read from ctag2 can be seen on r1 from approximately 300 nS to 420 nS. Ctag is shut off at 240 nS and ctag2 is shut off at 400 nS. To obtain utilization and power reports, the targeted FPGA is the Xilinx Artix-7. This FPGA is common on entry level development boards and is readily available.

An FPGA can be used to test multiple tags by creating a module for each tag, using a medium access control module to simulate a simple communication link between the tags and the interrogator, and the testbench can be used as an interrogator. This can be used to develop and test the logic of novel control protocols for RFID systems and can be an initial step in designing ASIC for RFID tags.

C. Comparison of Slotted ALOHA and B-CDMA Verilog Implementations

Area and power utilization reports are generated using Vivado, targeting the Xilinx Artix-7 FPGA to achieve a comparison of the Slotted ALOHA implementation and the B-CDMA implementation. The comparison of area and power utilization of the two implementations can be seen in Table 9 below.

Table 9. Comparison of Slotted ALOHA and B-CDMA Verilog Implementations

Power (W)	**Slotted ALOHA**	**B-CDMA**
Signals	0.85	0.02
Logic	0.53	0.01
Total	1.38	0.03
Area		
LUT	17	12
FF	8	26
BUFG	1	1

As can be seen from the Vivado results, the B-CDMA implementation also uses less power, which is likely due to its smaller FPGA area requirement. The automation of B-CDMA allows for a smaller need for instruction handling and therefore less area is needed for instruction handling logic. However, to integrate B-CDMA into Class 1 Generation 2 conforming RFID tags, both protocols must be included, so the total chip real estate will be the same or greater than Slotted ALOHA chips. The I/O power and the static device power were not included as the former is affected by the number of testpoints used to verify the implementation and the latter is dependant on the chipset utilized in the implementation. Further investigation into FPGA implementation is discussed in the Suggested Future Work section.

VI. CONCLUSION

The limitations of both Slotted ALOHA-based RFID and CDMA-based RFID protocols has been examined and characterized. These limitations arise from the inability of currently available TDMA, CDMA, and combination TDMA-CDMA collision handling protocols to identify colliding tags, remove them, and allow the non-collided tags to be read regardless of other tags colliding. In the case of Slotted ALOHA, the collision handling protocol proves inefficient due to its high probability of empty slots and collided slots. Slotted ALOHA also does not have an automated process by which colliding tags can temporarily stop their reflection. CDMA and combination TDMA-CDMA implementations have the largest advantage over Slotted ALOHA when the number of tags is less than or equal to the number of codes and the specific codes that are utilized by the system are controllable, which can be an unreasonable constraint for large systems. CDMA-based systems outperform Slotted ALOHA systems in high noise environments but without the ability to automatically remove colliding tags, the system can be easily overwhelmed, require a very large processing gain, or require significantly more transmissions from the interrogator. Bitwise CDMA is introduced as a solution to these issues. B-CDMA maintains the advantages of CDMA but can also automatically remove colliding tags before their EPC reflection is complete. B-CDMA ensures that at least one tag will be read successfully in each inventory round, even with a much larger tag population than number of codes, so long as the processing gain is sufficiently high such that all codes can be used simultaneously. B-CDMA also does not have the inefficiency of Slotted ALOHA and combination TDMA-CDMA systems caused by empty slots. B-CDMA can also simultaneously read the highest possible number of tags in each inventory round. If tags are reflecting at a similar power level, the highest number of tags that can be possibly read simultaneously is the number of uncollided tags. If the tags have disparate power levels, the highest number of tags that can be possibly read simultaneously is the number of uncollided tags with the highest reflected power. Conventional implementations of CDMA can handle this problem when the user (e.g. tag) has internal power. Since UHF Class 1 Generation 2 tags are passive, conventional solutions to the near-far problem (shadowing) are not viable. CDMA and TDMA-CDMA implementations are highly susceptible to the near-far problem whereas B-CDMA is not. B-CDMA has similar trade-offs to those of straight CDMA but has higher robustness in the presence of noise due to its ability to increase SIR during each inventory round, is automated, and contains a solution to the near-far problem. The framework for Verilog-based RFID system design and testing has also been developed, including the use of the testbench as an interrogator and the use of an FPGA to develop and evaluate novel RFID protocols.

VII. SUGGESTIONS FOR FUTURE WORK

For B-CDMA to be optimized, the optimal number of tags, spreading code type, number of codes, processing gain, and BER trade-off must be determined. The code type and processing gain can increase the robustness of B-CDMA. It is also valuable to determine the modulation technique best suited for B-CDMA (e.g. ASK or PSK, as outlined in the Class 1 Gen 2 Standard). An investigation into the use of data compression techniques to decrease the size of the check vector can potentially reduce the power consumption of the interrogator and could also improve the speed of the system. An investigation to the value of compression to the data that is as small as the check vector may be necessary. A comparison of B-CDMA to other ALOHA variants (such as Dynamic Frame Slotted ALOHA) could be valuable. The exact changes to the basic Class 1 Generation 2 conforming RFID IC itself must also be determined to potentially include B-CDMA in the future standards. To achieve a totally comprehensive assessment of B-CDMA in all applications, it will also be valuable to test B-CDMA using tags with EPC lengths of up to several kilobits and compare its performance to Slotted ALOHA, especially in systems with several thousand tags to determine the limits of B-CDMA. Variants of B-CDMA could also be developed by reflecting bits in reverse order, reflecting bits in a random order, reflecting bits in any order other than sequentially or randomly, and even by sending chunks of bits (2 bits, 4 bits, 8 bits, etc.). ASIC specific design tools to achieve a total assessment of B-CDMA from an IC power consumption and IC real estate perspective can aid in the integration of B-CDMA into the future of RFID. An additional investigation into the use of FPGA-based RFID IC development and testing using analog to digital converters and digital to analog converters to include channel simulation could also prove to be valuable as a development tool for general RFID research.

APPENDIX

//CTAG

```
module ctag(clk,cvec1,transmission, statr,cbito,rcbit);
input clk;
reg [3:0] tagnum = 4'd15;
input [15:0] cvec1;
output [15:0] transmission;
output [2:0] statr;
output rcbit;
reg [15:0] cvec;
assign statr = state;
assign rcbit = cvec[15];
parameter
  b3 = 3'b100,
  b2 = 3'b011,
  b1 = 3'b010,
  b0 = 3'b001,
  bZ = 3'b000;
reg [2:0] state = b3;
reg [2:0] nextstate;
reg cbit2 = 1'bz;
reg cbit1 = 1'bz;
reg cbit0 = 1'bz;
reg [15:0] scode = 16'b1010101010101010;
reg [2:0] epc = 3'b101;
reg [15:0] epco = 16'bzzzzzzzzzzzzzzzz;
  always @ (posedge clk) begin
    nextstate = state;
    cvec = cvec1;
    case (state)
      b3: begin
       epco = {epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],
              epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2]} ^ scode;
          state = b2;
      end
      b2: begin
       cbit2 = cvec[15] ~^ epc[2];
      if (cbit2) begin
      epco = {epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],
              epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1]}^scode;
      end
      else epco = 16'bzzzzzzzzzzzzzzzz;
          state = b1;
      end
      b1: begin
       cbit1 = cvec[15] ~^ epc[1];
      if (cbit2 && cbit1) begin
      epco = {epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],
              epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0]}^scode;
      end
      else epco = 16'bzzzzzzzzzzzzzzzz;
          state = b0;
      end
      b0: begin
       cbit0 = cvec[15] ~^ epc[0];
        epco = 16'bzzzzzzzzzzzzzzzz;
        if (cbit2 && cbit1 && cbit0) begin
        state = bZ;
        end
        else state = b3;
      end
      bZ: begin
      epco = 16'bzzzzzzzzzzzzzzzz;
      state = bZ;
      end
```

```
    endcase
  end
assign transmission = epco;
output [2:0] cbito;
assign cbito = {cbit2,cbit1,cbit0};
endmodule
```

//CTAG2

```
module ctag2(clk,cvec1,transmission, statr,cbito,rcbit);
input clk;
reg [3:0] tagnum = 4'd14;
input [15:0] cvec1;
output [15:0] transmission;
output [2:0] statr;
output rcbit;
reg [15:0] cvec;
assign statr = state;
assign rcbit = cvec[14];
parameter
  b3 = 3'b100,
  b2 = 3'b011,
  b1 = 3'b010,
  b0 = 3'b001,
  bZ = 3'b000;
reg [2:0] state = b3;
reg [2:0] nextstate;
reg cbit2 = 1'bz;
reg cbit1 = 1'bz;
reg cbit0 = 1'bz;
reg [15:0] scode = 16'b1001100110011001;
reg [2:0] epc = 3'b110;
reg [15:0] epco = 16'bzzzzzzzzzzzzzzzz;
  always @ (posedge clk) begin
    nextstate = state;
    case (state)
      b3: begin
       epco = {epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],
              epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2],epc[2]} ^ scode;
          state = b2;
```

```
end
b2: begin
cvec = cvec1;
 cbit2 = cvec[14] ~^ epc[2];
if (cbit2) begin
epco = {epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],
        epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1],epc[1]}^scode;
end
else epco = 16'bzzzzzzzzzzzzzzzz;
    state = b1;
end
b1: begin
cvec = cvec1;
 cbit1 = cvec[14] ~^ epc[1];
if (cbit2 && cbit1) begin
epco = {epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],
        epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0],epc[0]}^scode;
end
else epco = 16'bzzzzzzzzzzzzzzzz;
    state = b0;
end
b0: begin
cvec = cvec1;
 cbit0 = cvec[14] ~^ epc[0];
  epco = 16'bzzzzzzzzzzzzzzzz;
  if (cbit2 && cbit1 && cbit0) begin
  state = bZ;
  end
  else state = b3;
end
bZ: begin
epco = 16'bzzzzzzzzzzzzzzzz;
state = bZ;
```

```
      end
    endcase
  end
assign transmission = epco;
output [2:0] cbito;
assign cbito = {cbit2,cbit1,cbit0};
endmodule
```

//CTAG TESTBENCH

```
module ctag_tb;
reg clk;
initial begin
clk = 1;
   forever begin
      #20  clk = ~clk;
   end
end
reg [15:0] cvec;
wire [2:0] statr1,statr2;
wire [15:0] transmission;
reg [15:0] c1 = 16'b1010101010101010;
reg [15:0] c2 = 16'b1001100110011001;
reg received;
wire [2:0] cbito1,cbito2;
wire rcbit1,rcbit2;
reg [15:0] decode1,decode2;
integer a1,a2;
reg r1,r2;
wire [15:0] t1,t2;
  initial
    begin
      $dumpfile("MAC_test.vcd");
      $dumpvars(0,MAC);
      $monitor("%t \n cvec = %b \n transmission = %b \n tag1 = %b cbito1: %b state1: b%d rcbit1 = %b \n
tag2 = %b cbito2: %b  state2: b%d rcbit2 = %b   \n t1 = %b \n t2 = %b",
           $time,cvec,     transmission,     r1,     cbito1,   statr1,   rcbit1,      r2,     cbito2,    statr2,
rcbit2,         t1,        t2);
  #0
  #140  received = transmission;
        decode1 = transmission ^ c1;
        a1 = 0;
        a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
            decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
            decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
            if (a1 > 8) begin
            r1 = 1'b1;
            end
            else begin r1 = 1'b0; end
        decode2 = transmission ^ c2;
        a2 = 0;
        a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
             decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
             decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
        if (a2 > 8) begin
          r2 = 1'b1;
        end
       else begin r2 = 1'b0; end

    cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
    #40  received = transmission;
          decode1 = transmission ^ c1;
          a1 = 0;
          a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
              decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
              decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
              if (a1 > 8) begin
              r1 = 1'b1;
              end
              else begin r1 = 1'b0; end
          decode2 = transmission ^ c2;
          a2 = 0;
          a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
               decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
               decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
          if (a2 > 8) begin
       r2 = 1'b1;
      end
     else begin r2 = 1'b0; end
    cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
    #40  received = transmission;
       decode1 = transmission ^ c1;
       a1 = 0;
       a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
         decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
         decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
         if (a1 > 8) begin
         r1 = 1'b1;
         end
         else begin r1 = 1'b0; end
       decode2 = transmission ^ c2;
       a2 = 0;
       a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
          decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
          decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
       if (a2 > 8) begin
        r2 = 1'b1;
       end
      else begin r2 = 1'b0; end
     cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
     #40  received = transmission;
        decode1 = transmission ^ c1;
        a1 = 0;
        a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
          decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
          decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
          if (a1 > 8) begin
          r1 = 1'b1;
          end
```

```
        else begin r1 = 1'b0; end
    decode2 = transmission ^ c2;
    a2 = 0;
    a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
         decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
         decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
    if (a2 > 8) begin
      r2 = 1'b1;
    end
   else begin r2 = 1'b0; end
cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
#40  received = transmission;
      decode1 = transmission ^ c1;
      a1 = 0;
      a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
          decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
          decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
          if (a1 > 8) begin
          r1 = 1'b1;
          end
          else begin r1 = 1'b0; end
      decode2 = transmission ^ c2;
      a2 = 0;
      a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
           decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
           decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
      if (a2 > 8) begin
        r2 = 1'b1;
      end
     else begin r2 = 1'b0; end
  cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
  #40  received = transmission;
        decode1 = transmission ^ c1;
    a1 = 0;
    a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
        decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
        decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
        if (a1 > 8) begin
        r1 = 1'b1;
        end
        else begin r1 = 1'b0; end
    decode2 = transmission ^ c2;
    a2 = 0;
    a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
         decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
         decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
    if (a2 > 8) begin
      r2 = 1'b1;
    end
   else begin r2 = 1'b0; end
cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
#40  received = transmission;
      decode1 = transmission ^ c1;
      a1 = 0;
      a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
          decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
          decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
          if (a1 > 8) begin
          r1 = 1'b1;
          end
          else begin r1 = 1'b0; end
      decode2 = transmission ^ c2;
      a2 = 0;
      a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
           decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
           decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
    if (a2 > 8) begin
      r2 = 1'b1;
    end
   else begin r2 = 1'b0; end
cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
#40  received = transmission;
      decode1 = transmission ^ c1;
      a1 = 0;
      a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
          decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
          decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
          if (a1 > 8) begin
          r1 = 1'b1;
          end
          else begin r1 = 1'b0; end
      decode2 = transmission ^ c2;
      a2 = 0;
      a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
           decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
           decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
      if (a2 > 8) begin
        r2 = 1'b1;
      end
     else begin r2 = 1'b0; end
  cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
  #40  received = transmission;
        decode1 = transmission ^ c1;
        a1 = 0;
        a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
            decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
            decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
            if (a1 > 8) begin
            r1 = 1'b1;
    end
    else begin r1 = 1'b0; end
  decode2 = transmission ^ c2;
  a2 = 0;
  a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
     decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
     decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
  if (a2 > 8) begin
   r2 = 1'b1;
  end
 else begin r2 = 1'b0; end
cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
#40  received = transmission;
   decode1 = transmission ^ c1;
   a1 = 0;
   a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
     decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
     decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
     if (a1 > 8) begin
     r1 = 1'b1;
     end
     else begin r1 = 1'b0; end
   decode2 = transmission ^ c2;
   a2 = 0;
   a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
      decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
      decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
   if (a2 > 8) begin
    r2 = 1'b1;
   end
  else begin r2 = 1'b0; end
 cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
 #40  received = transmission;
```

```
                         decode1 = transmission ^ c1;
                         a1 = 0;
                         a1 = decode1[15]+decode1[14]+decode1[13]+decode1[12]+decode1[11]+
                             decode1[10]+decode1[9]+decode1[8]+decode1[7]+decode1[6]+
                             decode1[5]+decode1[4]+decode1[3]+decode1[2]+decode1[1]+decode1[0];
                             if (a1 > 8) begin
                             r1 = 1'b1;
                             end
                             else begin r1 = 1'b0; end
                         decode2 = transmission ^ c2;
                         a2 = 0;
                         a2 = decode2[15]+decode2[14]+decode2[13]+decode2[12]+decode2[11]+
                              decode2[10]+decode2[9]+ decode2[8]+ decode2[7]+ decode2[6]+
                              decode2[5]+ decode2[4]+ decode2[3]+ decode2[2]+ decode2[1]+decode2[0];
                         if (a2 > 8) begin
                           r2 = 1'b1;
                         end
                        else begin r2 = 1'b0; end
                      cvec = {r1,r2,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0,1'b0};
#40 $finish;
end
  MAC MAC(clk,cvec,transmission, statr1,statr2,cbito1,cbito2,rcbit1,rcbit2,t1,t2);
Endmodule
```

```
//MAC
module MAC(clk, cvec, transmission, statr1,statr2,cbito1,cbito2,rcbit1,rcbit2,t1,t2);
input clk;
input [15:0] cvec;
output [15:0] transmission;
output [2:0] statr1,statr2;
output [2:0] cbito1,cbito2;
output rcbit1,rcbit2;
wire [15:0] tx1,tx2;
output [15:0] t1,t2;
reg [15:0] t;
integer i;
ctag ct(clk,cvec,tx1,statr1,cbito1,rcbit1);
ctag2 ct2(clk,cvec,tx2,statr2,cbito2,rcbit2);
always @ (tx1 or tx2) begin
for (i = 15; i >= 0; i = i - 1) begin
if (tx1[i] || tx2[i]) begin
t[i] = 1'b1;
end
else begin t[i] = 1'b0; end
end
end
assign transmission = t;
assign t1 = tx1;
assign t2 = tx2;
endmodule
```

//SLOTTED ALOHA

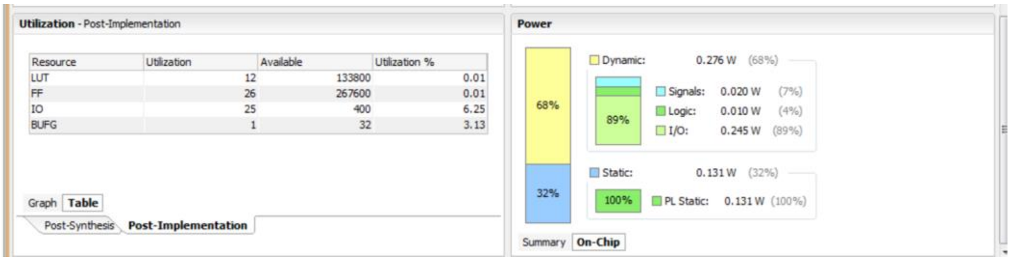
```
module ALOHATag(clk,Instruction, noise, instr, Reflection);
input clk;
input [7:0] Instruction;
input [3:0] noise;
output [7:0] instr;
output Reflection;
reg [2:0] EPC3 = 3'b101;
reg [3:0] RN4 = 4'b0000;
reg [3:0] RNT4;
integer i = 2;
reg Ref;
reg [7:0] temp;
reg [3:0] tx;
assign instr = Instruction;
reg [7:0] instruction;
always @ (posedge clk)
begin
 case(Instruction)
  8'b10101010: //Set Handle (RNG)
   begin
   RN4 = noise;
   RNT4 = RN4;
   end
  8'b11001100: //Decrement Handle
   begin
    if (RN4 == 4'b0000)
     begin
     i = 3;
     repeat(4)
      begin
       #20 Ref = RN4[i];
       i = i -1;
      end
     end
    else if (RN4 > 4'b0000)
     begin
      RN4 = RN4 - 1;
     end
   end
  8'b10011001: //Return EPC
   begin
       Ref = EPC3[i];
       i = i - 1;
   end
   8'b11110000: //WRITE EPC
    begin
      RN4 = 4'bzzzz;
      RNT4 = 4'bzzz;
      Ref = 1'bz;
     Ref <= Ref;
    end
  default: Ref = 1'bz; //Default (no_Reflection)
 endcase
end
 assign Reflection = Ref;
Endmodule
```

%SLOTTED ALOHA SIMULATION%

```
tags = 1000;
slots = 128;
r = 8;
timeMat = zeros((tags-10)/10,1);
l = 1;
for j = 10:10:1000

numberOfTags = j;
numberOfSlots = slots;
TagMat = zeros(1,numberOfTags);
currentSlot = numberOfSlots;

for i = 1:numberOfTags
    TagMat(i) = round(rand()*numberOfSlots);
end

ReadTags = 0;
Reflectors = 0;
Comm = 0;
QueryL = 22;
QueryRepL = 16; %12 for Frame-Sync, 4 for QueryRep
BitRefL = 16;
ACKL = 30; %12 for Frame-Sync, 18 for ACK
Query = 1;
QueryRep = 0;
BitRef = 0;
ACK = 0;
RNG = 0;
RNGL = r;

while ReadTags < numberOfTags
    Reflectors = find(TagMat == 0);
    if length(Reflectors) == 1
        TagMat(Reflectors) = [];
        ReadTags = ReadTags + 1;
        BitRef = BitRef + 1;
        QueryRep = QueryRep + 1
        ACK = ACK + 1;
        RNG = RNG + 1;
    elseif length(Reflectors) > 1
        for i = 1:(numberOfTags - ReadTags)
        TagMat(i) = round(rand()*numberOfSlots);
        end
        QueryRep = QueryRep + 1
        RNG = RNG + 1;
    else for i = 1:(numberOfTags - ReadTags)
            TagMat(i) = TagMat(i) - 1;
         end
        QueryRep = QueryRep + 1
    end
    Comm = QueryRep;
end

if Comm < numberOfSlots
```

```
    QueryRep = QueryRep + (numberOfSlots - Comm);
end
time = (QueryRep*QueryRepL)+(BitRef*BitRefL)+(ACK*ACKL) + (RNG * RNGL);
timeMat(l) = time
l = l + 1;
end
csvwrite('alohatime.csv',timeMat);
```

B-CDMA Vivado Generated Reports



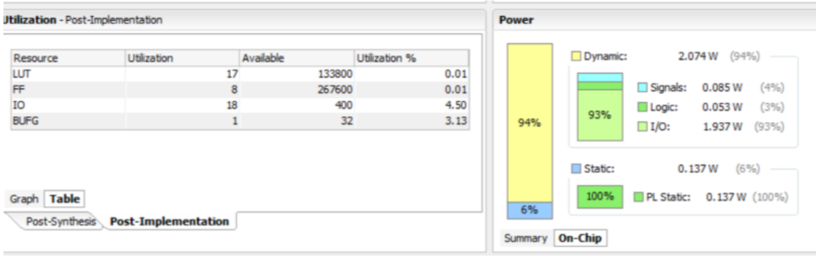
Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
LUT	12	133800	0.01
FF	26	267600	0.01
IO	25	400	6.25
BUFG	1	32	3.13

Power

Dynamic: 0.276 W (68%)
Signals: 0.020 W (7%)
Logic: 0.010 W (4%)
I/O: 0.245 W (89%)
Static: 0.131 W (32%)
PL Static: 0.131 W (100%)

Slotted ALOHA Vivado Generated Reports



Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
LUT	17	133800	0.01
FF	8	267600	0.01
IO	18	400	4.50
BUFG	1	32	3.13

Power

Dynamic: 2.074 W (94%)
Signals: 0.085 W (4%)
Logic: 0.053 W (3%)
I/O: 1.937 W (93%)
Static: 0.137 W (6%)
PL Static: 0.137 W (100%)

REFERENCES

[1] Roberti, M. (2005). The History of RFID Technology. *RFID Journal*. [online] Available at: http://www.rfidjournal.com/articles/view?1338 [Accessed 2 Nov. 2017].

[2] Violino, B. (2005). The Basics of RFID Technology. *RFID Journal*. [online] Available at: http://www.rfidjournal.com/articles/view?1337/2 [Accessed 2 Nov. 2017].

[3] Violino, B. (2005). RFID System Components and Costs. *RFID Journal*. [online] Available at: http://www.rfidjournal.com/articles/view?1336 [Accessed 2 Nov. 2017].

[4] Lehpamer, H. (2012). *RFID design principles*. Boston: Artech House.

[5] Das, R. (2017). *RFID Forecasts, Players and Opportunities 2017-2027: IDTechEx*. [online] IDTechEx. Available at: https://www.idtechex.com/research/reports/rfid-forecasts-players-and-opportunities-2017-2027-000546.asp [Accessed 2 Nov. 2017].

[6] P. (2017). *RFID market size - global forecast through 2020 | Statistic*. [online] Statista. Available at: https://www.statista.com/statistics/299966/size-of-the-global-rfid-market/ [Accessed 2 Nov. 2017].

[7] French National RFID Center. (2017). *Features of RFID tags*. [online] Available at: http://www.centrenational-rfid.com/features-of-rfid-tags-article-19-gb-ruid-202.html [Accessed 2 Nov. 2017].

[8] Roberti, M. (2004). Death of Class 0, 1 Exaggerated. *RFID Journal*. [online] Available at: http://www.rfidjournal.com/articles/view?943 [Accessed 2 Nov. 2017].

[9] Violino, B. (2005). A Summary of RFID Standards. *RFID Journal*. [online] Available at: http://www.rfidjournal.com/articles/view?1335/2 [Accessed 2 Nov. 2017].

[10] Dabas, C., Balhara, M. and Gupta, J. (2009). CDMA based Anti-Collision Deterministic Algorithm for RFID Tags. *International Journal of Recent Trends in Engineering*, 1(1).

[11] Frenzel, L. (2017). *Fundamentals of Communications Access Technologies: FDMA, TDMA, CDMA, OFDMA, AND SDMA*. [online] Electronic Design. Available at: http://www.electronicdesign.com/communications/fundamentals-communications-access-technologies-fdma-tdma-cdma-ofdma-and-sdma#%E2%80%9DTDMA%E2%80%9D [Accessed 2 Nov. 2017].

[12] Stern, H., Mahmoud, S. and Stern, L. (2004). *Communication systems*. Upper Saddle River, NJ: Pearson Prentice Hall.

[13] Choi, B. (2017). *Orthogonal Codes*. [online] University of Southampton. Available at: http://www-mobile.ecs.soton.ac.uk/bjc97r/pnseq/node5.html [Accessed 2 Nov. 2017].

[14] Mutti, C. and Floerkemeier, C. (2008). CDMA-based RFID Systems in Dense Scenarios: Concepts and Challenges. *IEEE International Conference on RFID*.

[15] *Difference Between Pure ALOHA and Slotted ALOHA (with Comparison Chart)- Tech Differences*. [online] Tech Differences. Available at: http://techdifferences.com/difference-between-pure-aloha-and-slotted-aloha.html [Accessed 2 Nov. 2017].

[16] Intel Corp (2011). *Parallel RFID System Using CDMA*. 8044773B2.

[17] Maina, J., Mickle, M., Lovell, M. and Schaefer, L. (2007). Application of CDMA for anti-collision and increased read efficiency of multiple RFID tags. *Journal of Manufacturing Systems*, 26(1).

[18] Intermec IP Corp (2010). *RFID tags with CDMA communication Capabilities*. 20100214065A1.

[19] Electronics And Telecommunications Research Institute (2011). *Method and apparatus for sharing portable terminal in CDMA system and mobile RFID system*. 7933577B2.

[20] Loeffler, A., Altmann, I. and Shuh, F. (2011). CDMA-Based UHF-RFID System with Semi-Passive UHF Transponders. *International Journal on Advances in Telecommunications*, 4(1).

[21] Doany, R., Lovejoy, C., Jones, K. and Stern, H. (2016). A CDMA-based RFID Inventory System: A CDMA Approach as a Solution for Decreased Power Consumption. *IEEE International Conference on RFID*.

[22] Vahedi, E., Ward, R. and Blake, I. (2014). Performance Analysis of RFID Protocols: CDMA Versus the Standard EPC Gen-2. *IEEE Transactions on Automation Science and Engineering*, 11(4), pp.1250-1261.

[23] Sparkfun.com. (2017). *SparkFun RFID Starter Kit - KIT-13198 - SparkFun Electronics*. [online] Available at: https://www.sparkfun.com/products/13198 [Accessed 2 Nov. 2017].

[24] Sparkfun.com. (2017). *SparkFun RFID Evaluation Shield - 13.56MHz - DEV-10406 - SparkFun Electronics*. [online] Available at: https://www.sparkfun.com/products/10406 [Accessed 2 Nov. 2017].

[25] Adafruit.com. (2017). *PN532 NFC/RFID controller breakout board [v1.6] ID: 364 - $39.95 : Adafruit Industries, Unique & fun DIY electronics and kits*. [online] Available at: https://www.adafruit.com/product/364 [Accessed 2 Nov. 2017].

[26] Instructables.com. (2017). *RFID Reader Detector and Tilt-Sensitive RFID Tag*. [online] Available at: http://www.instructables.com/id/RFID-Reader-Detector-and-Tilt-Sensitive-RFID-Tag/ [Accessed 2 Nov. 2017].

[27] Instructables.com. (2017). *A Universal RFID Key*. [online] Available at: http://www.instructables.com/id/A-Universal-RFID-Key/ [Accessed 2 Nov. 2017].

[28] Scott, M. (2017). *Using an AVR as an RFID tag | scanlime*. [online] Scanlime.org. Available at: http://scanlime.org/2008/09/using-an-avr-as-an-rfid-tag/ [Accessed 2 Nov. 2017].

[29] Scott, M. (2017). *Duct tape RFID tag #1 | scanlime*. [online] Scanlime.org. Available at: http://scanlime.org/2011/05/duct-tape-rfid-tag-1/ [Accessed 2 Nov. 2017].

[30] Atlasrfidstore.com. (2017). *RFID Development Kits: Shop for starter kits | atlasRFIDstore*. [online] Available at: https://www.atlasrfidstore.com/rfid-development-kits/ [Accessed 2 Nov. 2017].

[31] Kargas, N. (2017). *nikosl21/Gen2-UHF-RFID-Reader*. [online] GitHub. Available at: https://github.com/nikosl21/Gen2-UHF-RFID-Reader [Accessed 2 Nov. 2017].

[32] Wisp5.wikispaces.com. (2017). *WISP5 - WISP Home*. [online] Available at: http://wisp5.wikispaces.com/WISP+Home [Accessed 2 Nov. 2017].

[33] Doulos.com. (2017). *What is Verilog*. [online] Available at: https://www.doulos.com/knowhow/verilog_designers_guide/what_is_verilog/ [Accessed 2 Nov. 2017].

[34] Doany, R. (2017). *SDR Whitepaper*. [online] Ramseydoany.com. Available at: https://ramseydoanycom.files.wordpress.com/2017/04/sdrwhitepaper.pdf [Accessed 2 Nov. 2017].

[35] Abramson, N. (1970). THE ALOHA SYSTEM - Another alternative for computer communications. *Fall Joint Computer Conference*.

[36] Thakur, D. (n.d.). *ALOHA - What is ALOHA?*. [online] Ecomputernotes.com. Available at: http://ecomputernotes.com/computernetworkingnotes/communication-networks/what-is-aloha [Accessed 2 Nov. 2017].

[37] Modiano, E. (n.d.). *Packet Multiple Access: The Aloha protocol*.

[38] Carleial, A. and Hellman, M. (1975). Bistable Behavior of ALOHA-type Systems. *IEEE Transactions on Communications*, 23(4).

[39] Vahedi, E., Ward, R. and Blake, I. (2012). Analytical modeling of RFID Generation-2 protocol using absorbing Markov chain theorem. *2012 IEEE Global Communications Conference (GLOBECOM)*.

[40] Ma, R., Misra, V. and Rubenstein, D. (2009). An Analysis of Generalized Slotted-Aloha Protocols. *IEEE/ACM Transactions on Networking*, 17(3), pp.936-949.

[41] EPCglobal Inc (2008). *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.2.0*.

[42] Agee, B. (1994). Solving the Near-Far Problem: Exploitation of Spatial and Spectral Diversity in Wireless Personal Communication Networks. *Wireless Personal Communications*, pp.69-80.

[43] Khangura, S., Kaur, K. and Uppal, R. (2010). Power Control Algorithms in Wireless Communication. *International Journal of Computer Applications*, 1(12), pp.86-91.

[44] Teletopix.org. (n.d.). *How to Generate Walsh Code in CDMA*. [online] Available at: http://www.teletopix.org/cdma/how-walsh-code-generate-in-cdma [Accessed 2 Nov. 2017].

[45] Mathworks.com. (2017). *Hadamard matrix - MATLAB hadamard - MathWorks United Kingdom*. [online] Available at: https://www.mathworks.com/help/matlab/ref/hadamard.html [Accessed 2 Nov. 2017].