Technical University of Crete, Greece
School of Electronic and Computer Engineering

Synchronization and Detection for Gen2 RFID Signals

Michael Bamiedakis-Pananos

Thesis Committee

Associate Professor Aggelos Bletsas (ECE)
Associate Professor George Karystinos (ECE)
Associate Professor Antonios Deligiannakis (ECE)

Chania, January 2015

Abstract

This thesis studies EPC Global Class 1 Gen 2 (C1G2) protocol, which specifies the physical, link and medium access control (MAC) layer, for radio frequency identification (RFID) systems, consisting of interrogators and passive (i.e. battery-less) tags. Over the past decade, EPC Gen2 has established itself as the standard for UHF implementations and is at the heart of more and more RFID implementations.

This thesis examines interrogator queries to the tags and response from the tag, when the latter adheres to one out of two C1G2 supported line codes: FM0 and Miller. Signal processing at the interrogator is presented assuming a bistatic topology, i.e. interrogator emitter and receiver antennas are different, with carrier frequency offset (CFO) at the interrogator. This thesis presents packet synchronization and non-coherent detection for each line code. An algorithm for detecting tag collisions, i.e. simultaneous transmission from two or more tags, is also presented.

Acknowledgements

First, I would like to thank my Professor, Aggelos Bletsas, for his valuable guidance and encouragement during the course of this thesis.
Then, I would like to thank M.D. student N. Kargas and PhD student P.Alevizos who supported me and shared some excellent ideas with me.
Also, I would like to thank my friends, Mitsos, Panos (Panagiotakis), Babis, Jim, Dimitris(Kourtis), Elena, Eleni and Fotini, for their continued friendship and the great times we had together.
Last but not least, I would like to thank my parents and family for their love, support and constant encouragement.
To hope, this great thing that makes people evolve.

Contents

List of Figures

List of Tables

LIST OF TABLES

Chapter 1

Introduction

Backscatter communication is a wireless technology, comprised of Interrogators and Tags. An Interrogator is a wireless communication device, that also contains the 'logic' of the system and acts as a transceiver, while the Tag is a relatively small circuit without a battery or any other power source, other than the signal of the Interrogator. The Tag operates as a transceiver and responds to the Interrogator's commands, while the Interrogator receives the Tag's responses. The Tag achieves this passive type of telecommunication, by reflecting the Interrogator's signal right back to it, modulating the signal to transmit data.

The protocol presented in this thesis is the EPC Radio-Frequency Identity Protocol Class-1 Generation-2 UHF RFID Protocol, for communications at 860 MHz-960 MHz, which was originally developed as a replacement for bar-code identification systems. This type of telecommunication is being and will continue to be extensively developed in the near future, because of the key-advantages it provides, such as read-ranges more than 10 meters, non-line-of-site operation, high inventory rates and rewritable Tag IDs. It could be used in many applications, such as monitoring a warehouse's supplies, tracking items in a supply chain, having the Tags connected to moisture sensors and the Interrogator taking measurements in a field, checking for theft in a store, automatically checking a passenger's card at the tolls or at the borders to reduce delay times and more.

Therefore, it is important to fully comprehend how the protocol operates in practice, in order to deploy an efficient, low-cost backscatter network. The deployment of a low-cost network can be achieved, by establishing successful communication between the Interrogator and the Tags, while having as many Tags, in the vicinity of the Interrogator,

as possible. At the same time, the vicinity of the Interrogator must be as wide as possible, without consuming excessive energy.

However, RFID Readers, currently in the market are black boxes, meaning that physical layer configuration, Mac layer design, error rates or timing information is not exposed, while they only return a list of the Tags available in their vicinity. This makes diagnosing problems or possibly improving their behaviour, difficult.

1.1 Thesis Contribution

This thesis presents the aforementioned Gen2 protocol and examines this type of communication, between a Interrogator and a Tag in its vicinity, assuming a bistatic topology, with carrier frequency offset (CFO) at the Interrogator. This thesis addresses problems, such as the packet synchronization, the coherent detection of the packet and the Tag collision detection, if there is one, for the FM0 and the Miller encodings at the Interrogator.

1.2 Thesis Outline

In the second chapter, the signal model is presented. In the third chapter, the EPC-global Protocol class 1 Generation 2 protocol is introduced to the reader and several aspects of the protocol, regarding the physical layer and the Tag Identification layer are presented. Also, two encodings FM0 and Miller, used by the Tag, are explained. In the fourth chapter, we present three synchronization schemes and noncoherent detection, for both encodings. Finally, in the fifth chapter, a Tag collision detection method, is presented.

Chapter 2

Backscatter Radio Links

In order to achieve an Interrogator-to-Tag communication, initially, an Interrogator has to transmit a valid command. After issuing the command to the Tag, the Interrogator transmits a carrier, that is used by the Tag to backscatter its data. There are two possible topologies that can be implemented, monostatic and bistatic. In a monostatic topology, the Interrogator uses one antenna, for both transmitting and receiving the packets, while in a bistatic topology, the emitter and the receiver of the Interrogator use different antennas. Bistatic backscatter communication is preferred over monostatic backscatter communication, because it provides advantages [1], such as increased coverage for a large number of Tags, in a relatively large vicinity.

As specified at the protocol section in chapter 3, a Tag replies a command (Query), issued by the Interrogator, with a 16 bit random number (RN16). If the Interrogator receives the RN16, it acknowledges the Tag by transmitting the same RN16, along with a preamble and a acknowledgement(ACK)command code. If the Tag receives erroneous RN16, it does not transmit the EPC word back to the Interrogator. Therefore, the Interrogator has to be a transceiver, in order to be able to both transmit the commands and have knowledge of the Tag's RN16.

In this section, the signal model is defined, for the bistatic topology. This topology is comprised of two Software Defined Radios (SDRs), one acting as an emitter and the other as a receiver. The SDRs are assumed to be connected with a laptop, which is responsible for the signal processing and the communication, between the emitter and the receiver. The topology introduced a problem, due to the carrier frequency offset (CFO), discussed later in this chapter.

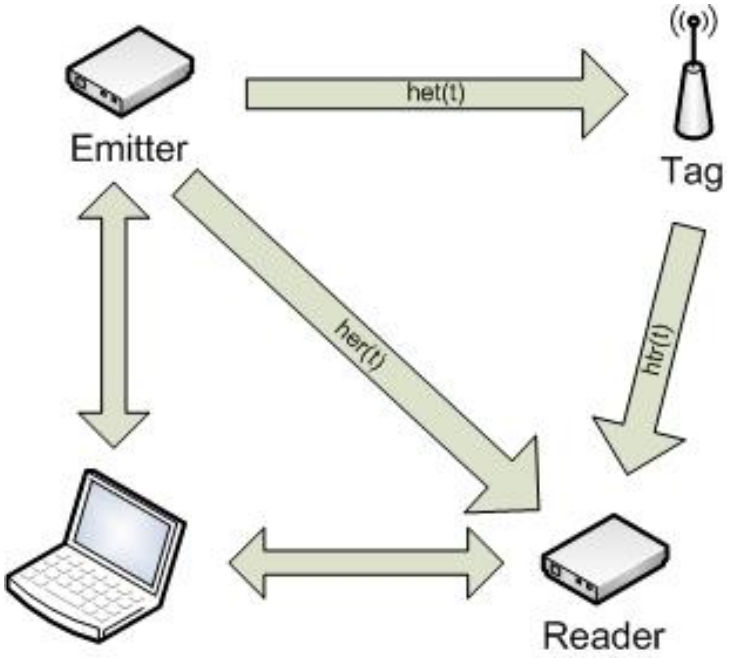

Figure 2.1: RFID bistatic topology: two SDRs per Interrogator.

2.1 Signal model

As stated before, there are two possible topologies for RFID, bistatic and monostatic, however, because of the Gen2 restrictions, regardless of the RFID topology the transmitter and the receiver must communicate with each other.

In our approach, a bistatic topology is used. An Interrogator is comprised of a pair of Software Defined Radios(SDRs) connected to a laptop, which processes the signal. In case of two SDRs per Interrogator(Figure: 2.1), there is a frequency offset at the receiver because of the differences in the hardware, they are implemented upon. After the command is transmitted, the Interrogator sends a continuous wave (CW), which is single tone, at the ultra high frequency (UHF) band. The Tag switches on and off its RF transistor, to modulate its data and scatters the signal back to the Interrogator. We assume flat-fading, because the bandwidth of the channel is much larger, than the bandwidth of the signal. Also, the bit duration is, relatively, small and the distance between the operating devices is small, meaning that the time between the emitter transmitting a signal and the Tag receiving it and vice-versa is, also, small. Therefore, all the bits of the packet, sent by the Tag receive the same attenuation, from the channel.

The passband channels are defined:

$$h_{et}(t) = \alpha_{et}\delta(t - \tau_{et}) \tag{2.1}$$

$$h_{tr}(t) = \alpha_{tr}\delta(t - \tau_{tr}) \tag{2.2}$$

$$h_{er}(t) = \alpha_{er}\delta(t - \tau_{er}) \tag{2.3}$$

Where:

- $\alpha \in R$.
- $h_{er}(t)$ denotes the channel between the emitter and the receiver.
- $h_{tr}(t)$ is the channel between the Tag and the receiver.
- $h_{et}(t)$ is the channel between the emitter and the Tag.

The phases that are introduced to the signals are:

$$\varphi_{et}(t) = 2\pi F_c \tau_{et} \tag{2.4}$$

$$\varphi_{tr}(t) = 2\pi F_c \tau_{tr} \tag{2.5}$$

$$\varphi_{er}(t) = 2\pi F_c \tau_{er} \tag{2.6}$$

The CW transmitted by the Interrogator is of frequency F_c and amplitude A.

$$c_m = A\cos(2\pi F_c t) \tag{2.7}$$

Therefore, the signal received by the Tag is:

$$c_m * h_{et} = \alpha_{et} A\cos(2\pi F_c(t - \tau_{et})) = \alpha_{et} A\cos(2\pi F_c t - \varphi_{et}(t)) \tag{2.8}$$

The Tag modulation is achieved, by switching the antenna load to different values, that correspond to different reflection coefficients Γ_i, with $i = 0, 1 \dots M-1$. The reflection coefficients can be expressed, as a function of time, as:

$$\Gamma(t) = \{\Gamma_i\}_0^{M-1} \tag{2.9}$$

Every Tag has a load-independent term A_s, that is related to its structural mode[2],[3], also the reflection coefficients are, in this case, only two Γ_0 and Γ_1 and both A_s and Γ_i are complex. Therefore, the Tag's baseband signal, as a function of time, is:

$$x(t) = A_s - \Gamma(t) = a_x(t)e^{j\varphi_x(t)} \tag{2.10}$$

In the above equation $a_x(t)$ and $\varphi_x(t)$ are the amplitude and the phase of $A_s - \Gamma(t)$. Therefore:

$$a_x(t) = |A_s - \Gamma(t)| \tag{2.11}$$

$$\varphi_x(t) = \angle(A_s - \Gamma(t)) \tag{2.12}$$

The Tag backscatters the signal:

$$x_m(t) = A\alpha_{et}s(t)a_x(t)\cos(2\pi F_c t - \varphi_{et}(t) + \varphi_x(t)) \tag{2.13}$$

with $s(t)$ being a time-variant variable, that represents the backscattering efficiency of a Tag and, for a small number of transmitted bits, can be assumed to be constant. Therefore, the Interrogator receiver, receives the superposition of the backscattered signal and the carrier transmitted, from the transmitter, through channels $h_{tr}(t)$ and $h_{er}(t)$. Thus, the received signal will be (s(t) is considered constant):

$$y_m(t) = A\alpha_{er}\cos(2\pi F_c t - \varphi_{er}) + A\alpha_{tr}\alpha_{et}sa_x(t-\tau_{tr})\cos(2\pi F_c t - \varphi_{et}(t) - \varphi_{tr}(t) + \varphi_x(t-\tau_{tr})) + w(t) \tag{2.14}$$

Where $w(t)$ is band-limited, additive Gaussian noise with power spectral density (PSD):

$$SW(F) = \begin{cases} \frac{N_0}{2} & |F \pm F_c| < W \\ 0 & \text{elsewere} \end{cases} \tag{2.15}$$

Where $2W$ is the passband receiver bandwidth and $F_c \gg W$. The incoming signal is demodulated by the Interrogator, with $F_c + \Delta F$ local oscillator carrier and receiver phase φ_r, then the high frequency components are filtered. ΔF is the carrier frequency offset, between the transmitter and the receiver. The lowpass Inphase and Quadrature components are:

$$I(t) = LPF[y_m(t)\cos(2\pi(F_c + \Delta F)t)] \tag{2.16}$$

$$Q(t) = LPF[-y_m(t)\sin(2\pi(F_c + \Delta F)t)] \tag{2.17}$$

After lowpass filtering, the Inphase and the Quadrature components are:

$$I(t) = \frac{A\alpha_{er}}{2}\cos(2\pi\Delta Ft + \hat{\varphi_{er}}) + \frac{A\alpha_{tr}\alpha_{et}s\alpha_x(t-\tau_{tr})}{2}\cos(2\pi\Delta Ft + \hat{\varphi_{etr}} - \varphi_x(t-\tau_{tr})) + n_I(t) \tag{2.18}$$

$$Q(t) = -\frac{A\alpha_{er}}{2}\sin(2\pi\Delta Ft + \hat{\varphi_{er}}) - \frac{A\alpha_{tr}\alpha_{et}s\alpha_x(t-\tau_{tr})}{2}\sin(2\pi\Delta Ft + \hat{\varphi_{etr}} - \varphi_x(t-\tau_{tr})) + n_Q(t) \tag{2.19}$$

with

$$\hat{\varphi_{er}} = \varphi_{er} + \varphi_r \tag{2.20}$$

$$\hat{\varphi_{etr}} = \varphi_{tr} + \varphi_{et} + \varphi_r \tag{2.21}$$

while $n_I(t)$ and $n_Q(t)$ are lowpass Gaussian noise components with PSD:

$$S_{n_I}(F) = S_{n_Q}(F) = \begin{cases} \frac{N_0}{4} & |F| < W \\ 0 & \text{elsewere} \end{cases}$$

and their variance:

$$\sigma_n^2 = E[n_I^2] = E[n_Q^2] = \frac{2N_0W}{4} = \frac{N_0W}{2} \tag{2.22}$$

proof can be found in [1]. Therefore, the complex baseband received signal is:

$$y(t) = I(t) + Q(t)j = \frac{A}{2}[\alpha_{er}e^{-j\hat{\varphi_{er}}} + \alpha_{et}\alpha_{tr}sx(t-\tau_{tr})e^{-j\hat{\varphi_{etr}}}]e^{-j2\pi\Delta Ft} + n(t) \tag{2.23}$$

while:

$$n(t) = n_I(t) + n_Q(t)j \tag{2.24}$$

and $E[n^2(t)] = E[n_I^2(t)] + E[n_Q^2(t)] = 2\sigma_n^2$.

2.2 Modulation

Tags use a combination of the Amplitude-shift keying (ASK) modulation, to transmit their data. This modulation can be viewed, as on-off keying (OOK) modulation at the receiver. Using this modulation, they either use FM0 or Miller encodings to encode their data. If a Tag changes its antenna load coefficients, between states Γ_0 and Γ_1 and Γ_0 is used for bit '0', while Γ_1 is used for bit '1', the baseband signal $x(t)$, shown in Equation 2.25, is expressed as:

$$x(t) = (As - \frac{\Gamma_0 + \Gamma_1}{2}) + \frac{\Gamma_0 - \Gamma_1}{2} \sum_{n=0}^{N-1} s_{x_n}(t - nT) \tag{2.25}$$

With As being a complex variable dependent on the Tag antenna's structural mode, [2],[3], $x_n \in \{1, 2, 3, 4\}$ and $s_{x_n}(t)$ being the supported waveforms of the FM0 encoding, shown in Figure 3.4 or of the supported waveforms of the Miller encoding, shown in Figure 3.9. The Miller supported waveforms $s_n(t)$ that are sent, are multiplied with a square wave, at M times the symbol rate.

The backscattered signal in Equation 2.25 can be expressed as:

$$x(t) = m_{dc}e^{j\theta_{dc}} + m_{mod}e^{j\theta_{mod}} \sum_{n=0}^{N-1} s_{x_n}(t - nT) \tag{2.26}$$

where:

$$m_{dc} = |A_s - \frac{\Gamma_0 + \Gamma_1}{2}|, \theta_{dc} = \angle(A_s - \frac{\Gamma_0 + \Gamma_1}{2}) \tag{2.27}$$

$$m_{mod} = \frac{|\Gamma_0 - \Gamma_1|}{2}, \theta_{mod} = \angle(\Gamma_0 - \Gamma_1) \tag{2.28}$$

The received signal at the SDR, as in [1], [4], is:

$$y(t) = y_{nl}(t) + n(t) = [\hat{m_{dc}}e^{j\hat{\varphi}_{dc}} + \hat{m}_{mod}e^{j\hat{\varphi}_{mod}} \sum_{n=0}^{N-1} s_{x_n}(t - \tau_{tr} - nT)]e^{-j2\pi\Delta Ft} + n(t) \tag{2.29}$$

with

$$\hat{m}_{dc}e^{j\hat{\varphi}_{dc}} = \frac{A}{2}[\alpha_{er}e^{-j\hat{\varphi}_{er}} + s\alpha_{et}\alpha_{tr}m_{dc}e^{j(\theta_{dc} - \hat{\varphi}_{etr})}] \tag{2.30}$$

symbol	first bit	followed by
0	data-0	data-1
	data-1	data-0
1	data-0	data-0
	data-1	data-1

Table 2.1: Symbol to data map (FM0 encoding).

$$\hat{m}_{mod}e^{j\hat{\varphi}_{mod}} = \frac{As\alpha_{et}\alpha_{tr}m_{mod}}{2}e^{j(\theta mod - \hat{\varphi}_{etr})} \tag{2.31}$$

Finally, after sampling the baseband signal the digitalized signal is

$$y[k] = y(kT_s + \tau_{tr}) = y_{nl}[k] + n[k] \tag{2.32}$$

The term y_{nl} is the noiseless signal and is consisted of a DC and a MOD term. The DC term is the carrier signal sent and received by the Interrogator, while the MOD term is the modulated data, sent by the Tag. In case of the two SDRs bistatic scenario, there will be a carrier frequency offset(CFO) term, that is a difference between the modulation and the demodulation of the carrier, because of the differences between the two SDRs oscillators. CFO makes harder the decoding of a signal, therefore its removal is desired. Figure 2.2 shows an RN16 signal's samples with CFO. The Tag does not affect the CFO. It is denoted, that a number of samples A correspond to one bit, data-0 and data-1 and that two bits correspond to one symbol. For example the sequences allowed by the FM0 encoding are shown in Table 2.1.

The set of samples of number A, that belong to a cluster, represent a data-0 or a data-1 and the absolute value of the samples is used. This is a way to eliminate the existing CFO. Specifically, the function that corresponds the noiseless bits to the absolute values is:

$$s_k = |y_{nl}[k]| = \begin{cases} a & \text{if bit} =' 0' \\ b & \text{if bit} =' 1' \end{cases}$$

Figure 2.4 shows the absolute value, of a received signal with FM0 encoding. This leads to non-coherent detection, as the channel is not subtracted from the signal. We can easily extract two levels from the waveform, one for bit '0' and one for bit '1'. This is a

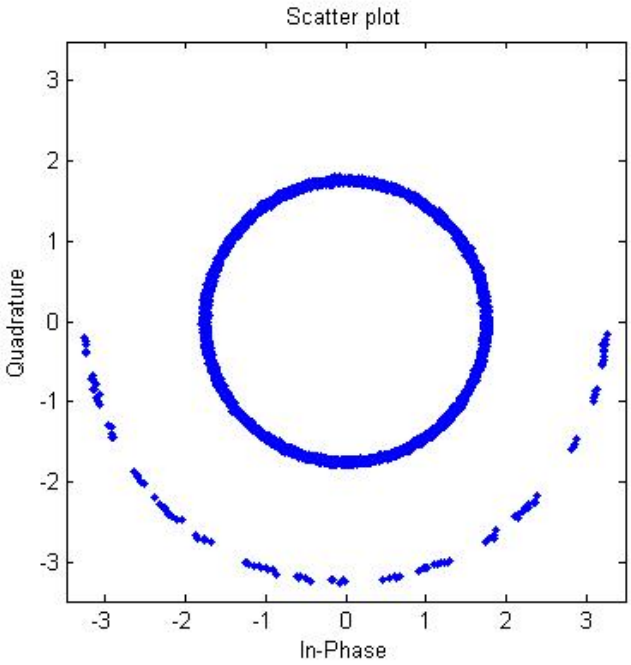

Figure 2.2: Samples extracted from a carrier and an RN16 with CFO.

good way to process the signal, because the imaginary and the real parts are combined and we obtain one absolute number that contains information for the sample that is sent by the Tag, also it is easy computationally, because the computation of the CFO is computationally "expensive".

A simple way of eliminating the CFO is by using a discrete Fourier transform (DFT) on the received carrier before the RN16, while having a sufficient number of samples, we can determine the ΔF, meaning the difference between the actual carrier frequency F_c of the transmitter and the carrier frequency of the receiver, [5]. Then, we use the equation below to eliminate the CFO:

$$y[k] = y[k]e^{2\pi\Delta FkT_s} \tag{2.33}$$

An example of extracted samples with CFO. After the DFT and the CFO correction the samples are shown in Figure 2.6.

An effective algorithm for CFO estimation could be used for CFO elimination at the receiver. However, if the CFO is not estimated correctly, a signal with CFO will

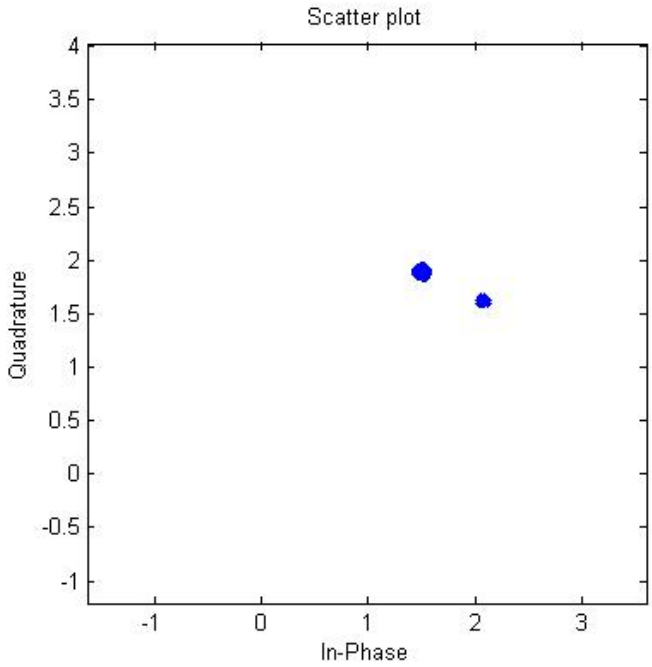

Figure 2.3: Samples extracted from a RN16 without CFO.

be obtained. There are effective algorithms for CFO estimation, however we must take careful consideration before their use, as their complexity must not add much time to the processing of the signal, because the Interrogator has to reply to the Tag in a specific time, as is explained in chapter 3. Otherwise, the Tag will not scatter its EPC back to the Interrogator, even if the Interrogator answers correctly to the Tag, its RN16. For all the above reasons, the absolute value of the samples was computed and used at the next steps of the signal processing, as it eliminates the problem of CFO, thus saves precious time of signal processing.

Finally, this analysis is valid whether an RN16 or a EPC is sent by the Tag. The EPC mentioned in chapter 3 is the answer sent by the Tag after the Interrogator sends a valid ACK command, also mentioned in chapter 3. Also, the analysis is valid in both FM0 and Miller encodings, as their differences is the mapping of the symbols to the sequence of the bits sent, while the same set of pulses is transmitted by the Tag.

One more case to be taken into consideration is when the absolute value, that corresponds to a bit '1', is less than the absolute value for a bit '1'. This case is shown

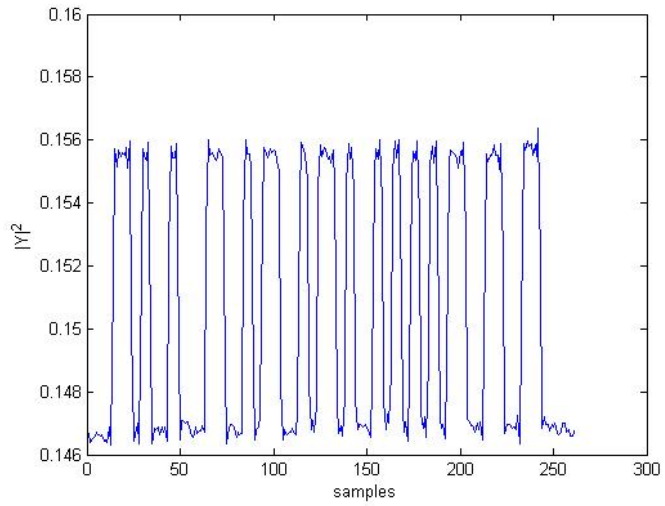

Figure 2.4: Tag response to a Query command with FM0 encoding.

in Figure: 2.7 and introduces a problem to the synchronization and detection schemes discussed, in later chapters.

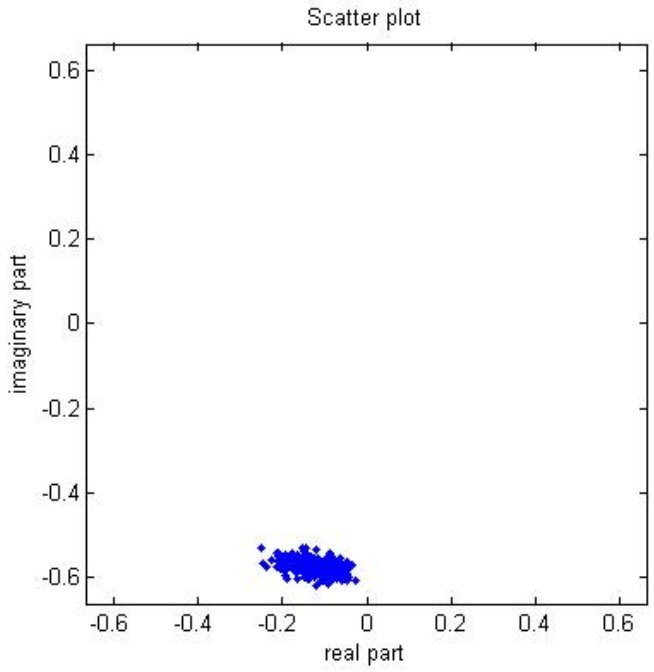

Figure 2.5: RN16 scatterplot of samples with erroneous CFO correction.

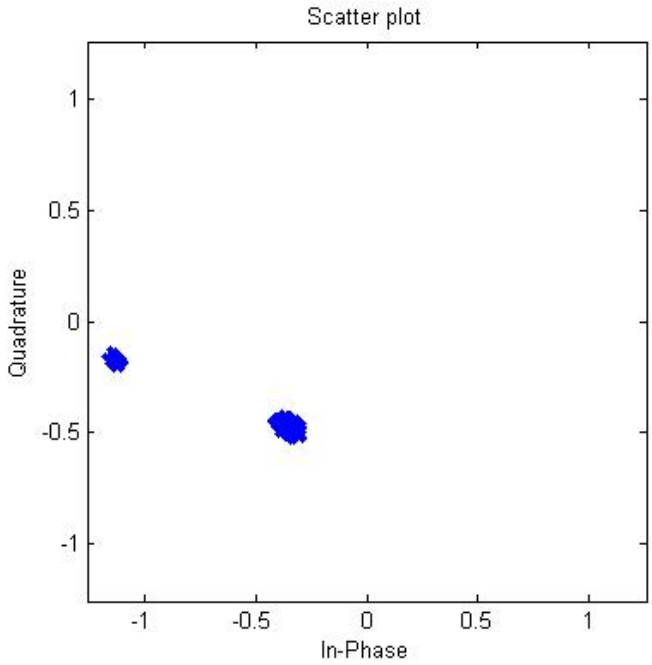

Figure 2.6: RN16 scatterplot of samples with CFO corrected.

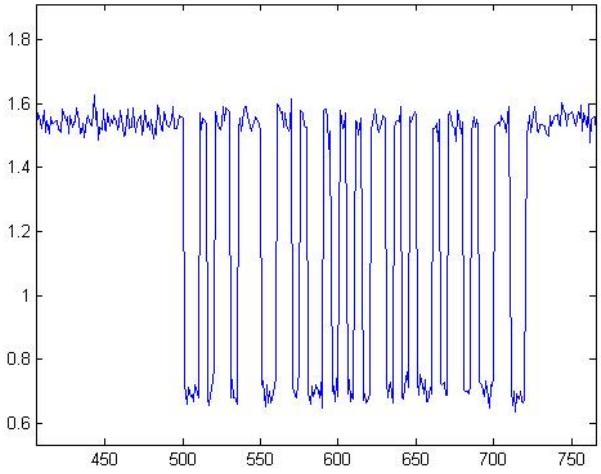

Figure 2.7: Inversed RN16 absolute value.

Chapter 3

EPC Global class 1 Generation 2 Protocol Overview

3.1 RFID Protocols

A communications protocol defines the way, in which communication devices manage to successfully communicate. In case of RFID protocols, the protocol determines the way, in which Interrogators and Tags communicate. The protocol defines:

- The air interface, which includes information, such as which modulation is used by the reader to define a binary one, how fast does the information transfer, the way that packets are handled or what kind of signal is sent by the tag.
- Medium Access Control which defines when a device is obliged to transmit or the manner, that the collision are resolved.
- Data definitions, which is the meaning and the kind of data associated with the Tags and the Readers.

Passive RFID Tags' construction is cheap. Also, their transmit power depends on the Reader's transmit signal. Therefore, the communication between the Tags and the Readers faces problems not seen in other digital communications. Modulations such as Quadrature Amplitude Keying (QAM) or phase-shift keying (PSK) are not available [6], because of the assumption of bad channel conditions. Furthermore, modulations that

turn-off the power of the Reader are also rejected, because of the connection between the Reader's transmit power and the Tags transmit power. Consequently, in order to achieve successful communication between the Reader and the Tags, all the above problems must be addressed and solved.

3.2 EPCglobal Protocol class 1 Generation 2

3.2.1 EPCglobal RDFID Protocols

EPCglobal released a series of protocols (EPCglobal class 0,EPCglobal class 1), that are considered generation 1. However, first-generation standards have significant disadvantages, such as the difficulty of addressing a specific Tag or problems with late arrivals, i.e. Tags that enter the read zone, when a tag inventory has started. Moreover, the two standards are incompatible, whereas the world needs one global standard, so that every Interrogator is compatible with every Tag.

To address the above problems, EPCglobal released a new generation standard that provides sufficient performance at low cost. EPCglobal Class 1 Generation 2 protocol was ratified in 2005 and was also ratified by International Organization for Standardization (ISO), as an ISO18000-6C. Class 1 Gen2 Protocol has gradually replaced older UHF protocols and will continue to be used in a large portion of the market, because of the low-cost efficiency it provides.

3.2.2 Physical Layer

This chapter presents some basic concepts of the EPC Global RFID Class-1 Gen2 Protocol, as defined in [7]. An Interrogator sends information to one or more Tags by modulating an RF carrier using double-sideband amplitude shift keying (DSB-ASK), single-sideband amplitude shift keying (SSB-ASK) or phase-reversal amplitude shift (PR-ASK) using a pulse-interval encoding (PIE) format. Tags receive their operating energy from this same modulated RF carrier.

An Interrogator receives information from a Tag, by transmitting an unmodulated RF carrier and listening for a backscattered reply. Tags communicate information by modulating the amplitude and/or phase of the RF carrier. The encoding format, selected

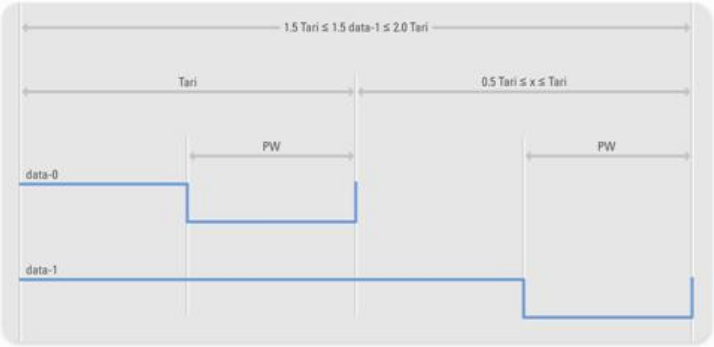

Figure 3.1: PIE symbols, extracted from [7].

in response to Interrogator commands, is either FM) or Miller-modulated subcarrier. The communications link between Interrogators and Tags is half-duplex, meaning that Tags shall not be required to demodulate Interrogator commands, while backscattering. A Tag shall not respond to a mandatory or optional command using full-duplex communications.

3.2.2.1 Signalling

The signalling interface between Interrogators and Tags may be viewed as the physical layer of the network system. The signal interface defines the operating frequencies, the modulations that may be used, data encoding, RF envelope, data rates and other parameters.

Tags shall receive power from and communicate with Interrogators within the frequency range from 860 MHz to 960 MHz. Interrogators shall use DSB-ASK, SSB-ASK or PR-ASK to modulate the signal and PIE encoding. Tari, that can be seen in Figure 3.1, is the duration of a data-0. High values represent transmitted CW, while low values represent attenuated CW. Tari values will range from 6.25μs to 25μs with a tolerance of +/-1%.

3.2.2.2 Reader's preamble and frame-sync

An Interrogator begins an R⇒T signalling by adding a preamble or a frame-sync to the packet transmitted. A preamble precedes a Query command and denotes the start of an inventory round. Everything else transmitted by the Interrogator begins with a frame-sync. The tolerance specified in units of Tari shall be +/-1%. A preamble is comprised

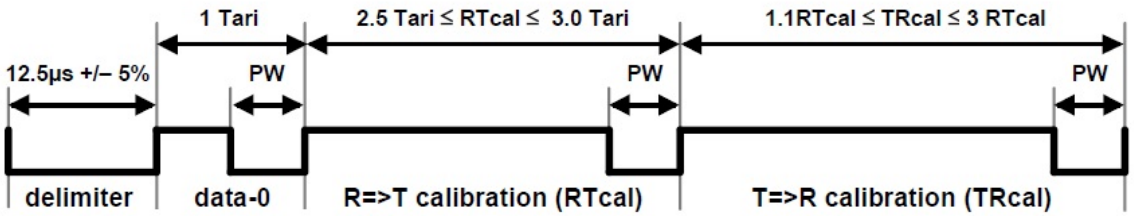

Figure 3.2: Preamble, extracted from [7].

of a fixed-length start delimeter, a data-0 symbol, an R⇒T calibration (RTcal) symbol and a T⇒R calibration (TRcal) symbol.

- An Interrogator sets the RTcal symbol equal to the length of a data-0 symbol, plus the length of a data-1 symbol. The Tag measures the RTcal length and computes $pivot = \frac{RTcal}{2}$. Then, the Tag decodes symbols longer than pivot as data-1s and shorter than pivot as data-0s. Symbols longer than 4 RTcals are interpreted by the Tag as invalid. Before changing RTcal, an Interrogator transmits CW for a minimum of 8 RTcals.

- An Interrogator specifies a Tag's backscatter link frequency (FM0 datarate or the frequency of its Miller subcarrier) using the TRcal and divide ratio, located in the preamble and payload, respectively, of a Query command, that initiates an inventory round. A Tag adjusts its backscatter link frequency according to equation:

$$BLF = \frac{DR}{TRcal} \tag{3.1}$$

$$1.1 \times RTcal \leq TRcal \leq 3 \times RTcal \tag{3.2}$$

An Interrogator, for the duration of an inventory round, shall use the same length RTcal in a frame-sync, as it used in the preamble that initiated the round. Frame-syncs are identical to preambles, minus the TRcal symbol. A preamble and a frame-sync are shown in Figure 3.2 and Figure 3.3, respectively.

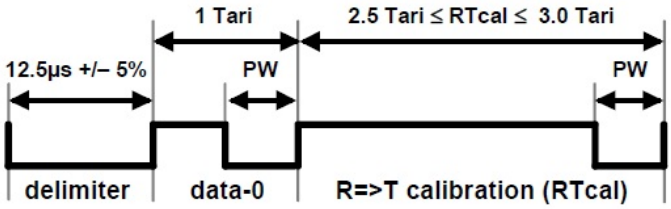

Figure 3.3: Frame-sync, extracted from [7].

3.2.2.3 Tag to Interrogator Communications

A Tag communicates, with an Interrogator, using backscatter modulation. Practically, it switches the reflection coefficient of its antenna between two states, depending on the data being sent. A Tag shall backscatter using fixed modulation format, data encoding and data rate for the duration of an inventory round. Tags select the modulation format, while Interrogators select encoding and data rate using a Query command. Tags use ASK and/or PSK modulation and use either FM0 baseband or Miller modulation for a subcarrier at the data rate.

3.2.2.4 FM0 baseband

FM0 encoding inverts the baseband phase at every symbol boundary, while a data-0 has an additional mid-symbol phase inversion. The left side of Figure 3.4 shows the FM0 basic functions transmitted. Data-0 is transmitted either with a $s_2(t)$ or a $s_3(t)$ function, depending on the data bit that was previously transmitted. The same applies for data-1, it is either transmitted with a $s_1(t)$ or a $s_4(t)$ function, depending on the previously transmitted bit, meaning that the FM0 encoding has memory. All the transitions between the functions of the FM0 encoding are shown in Figure 3.4. For example, a transition from state S_2 to state S_3 is disallowed, because the resulting transmission would not have a phase inversion on a symbol boundary. Examples of FM0 sequences are shown in Figure 3.5. FM0 encoding always "ends" with a dummy data-1 bit, as shown in Figure 3.6.

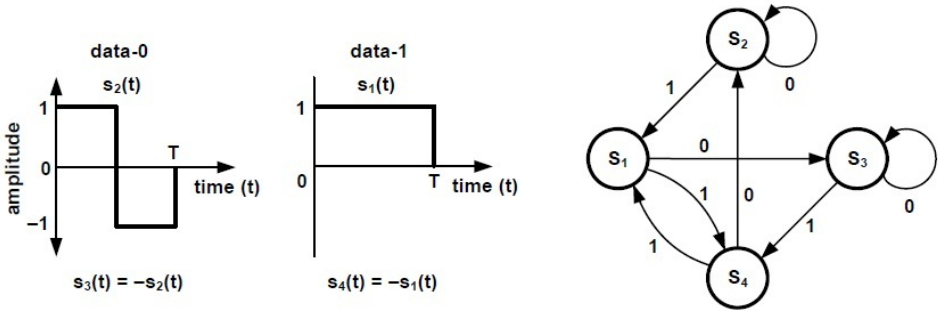

Figure 3.4: FM0 functions and state diagram, extracted from [7].

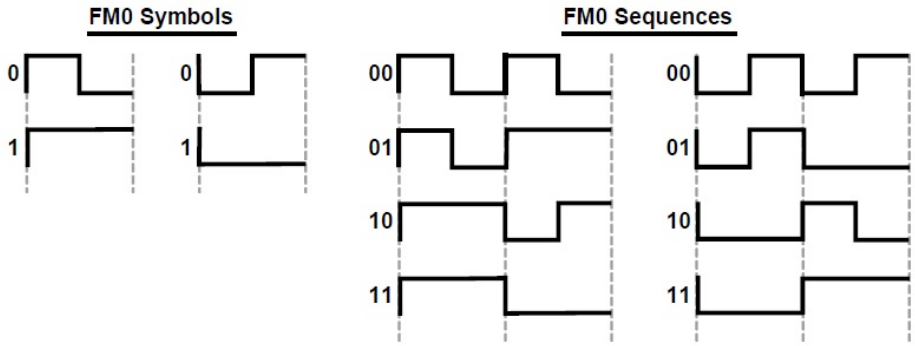

Figure 3.5: FM0 symbols and sequences, extracted from [7].

3.2.2.5 FM0 preamble

FM0 signalling begins with one of the preambles, shown in Figures 3.7 and 3.7, respectively. The choice depends on the value of TRext bit located in the Query command, that initiated the inventory round, unless a Tag is replying to a command that writes to memory, in which case the Tag shall use the extended preamble, regardless of TRext. The "v", shown in both these figures. indicate an FM0 violation (i.e. a phase inversion should have occurred but did not).

3.2.2.6 Miller-modulated subcarrier

Baseband Miller inverts its phase between two data-0s in sequence, also it places a phase inversion in the middle of a data-1 symbol. Figure 3.9 represents the basic functions of Miller encoding as well as its state diagram. The state labels $S_1 - S_4$ indicate all the possible functions allowed in this encoding, represented by the two faces

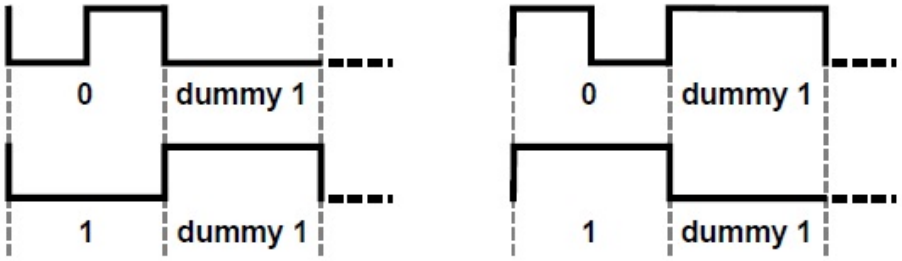

Figure 3.6: FM0 end of signalling, extracted from [7].

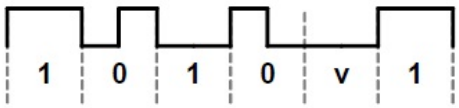

Figure 3.7: FM0 preamble (Trext=0), extracted from [7].

of each of the Miller basis functions, while the arrows indicate the transitions between them, i.e. transition from state S_3 to state S_4 is disallowed, because there is no phase shift between a data-0 and a data-1. On the other hand, a transition between S_2 and S_3 is possible, if the next symbol in the sequence is a data-1. The state labels, also, represent the baseband Miller waveform, that is generated upon entering the state. The state transmitted waveform is the baseband waveform, multiplied by a square-wave at M times the symbol rate.

The Miller sequence shall contain exactly two, four or eight subcarrier cycles per bit, depending on the M value, located in the Query command, that initiated the inventory round, as shown in Figures 3.10 and 3.11. The duty cycle of a 0 or 1 symbol shall be a minimum of 45% and a maximum of 55%, with a nominal value of 50%. Miller signalling, always, ends with a "dummy" data-1 at the end of the sequence, as shown in Figure 3.12.

3.2.2.7 Miller Preamble

Miller encoding is, always, preceded by one of the preambles, shown in Figure 3.13. The choice depends on the value of TRext bit ,which is located in the Query command, that initiates the current inventory round. However, if the Tag replies to a command

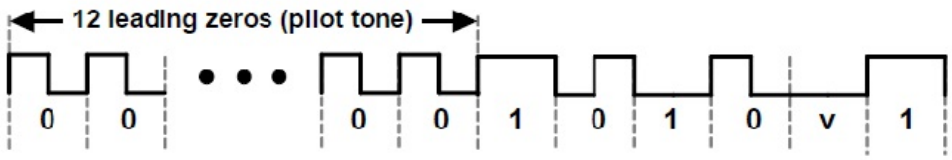

Figure 3.8: FM0 preamble (Trext=1), extracted from [7].

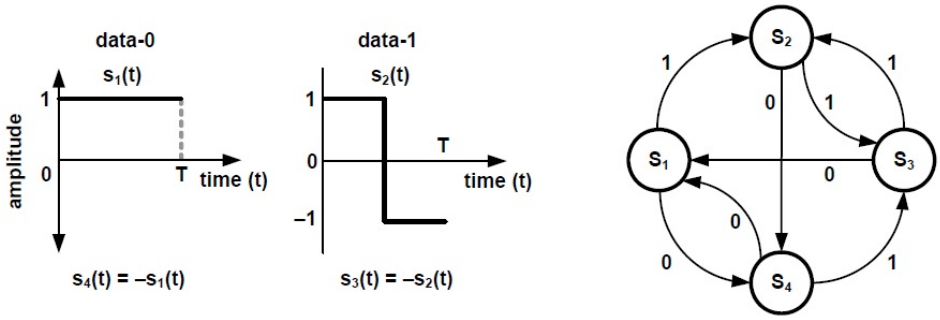

Figure 3.9: Miller functions and state diagram, extracted from [7].

that replies to memory, it will use the extended preamble, regardless of the TRext bit's value.

Tags support all R⇒T Tari value, from 6.25μs to 256.25μs.

3.2.2.8 Tag Parameters

The parameters that define the backscatter frequency, modulation type(FM0 or Miller) and T⇒ data rate. for the round are DR, M, TRcal and BLF. BLF is computed using 3.1, while the other parameters are located in the Query command. Transmission order, for both R⇒T and T⇒ communications, shall be the most-significant bit (MSB) first. Within each word and each message the MSB will be transmitted, first. Also, a certain CRC (cyclic redundancy check) will be included, in order to ensure the validity of certain R⇒T and the Interrogator uses it, to ensure the validity of certain backscattered T⇒ replies. The protocol uses two CRC types a CRC-16 and a CRC-5. Figure 3.14 shows the link-timing between commands sent by the Reader and replies backscattered by the Tag. Where:

- T_1 is time from Interrogator transmission to Tag response measured at the Tag's antenna Terminals. T_1's value is set according to Equation 3.3.

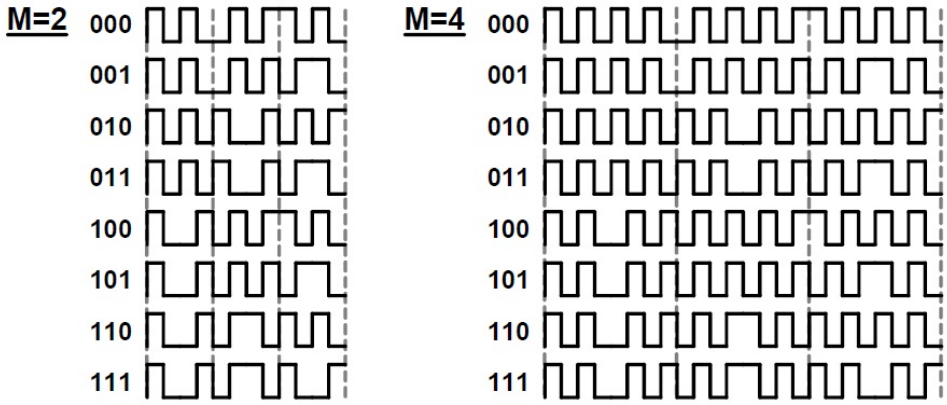

Figure 3.10: Miller sequence (for M=2, M=4), extracted from [7].

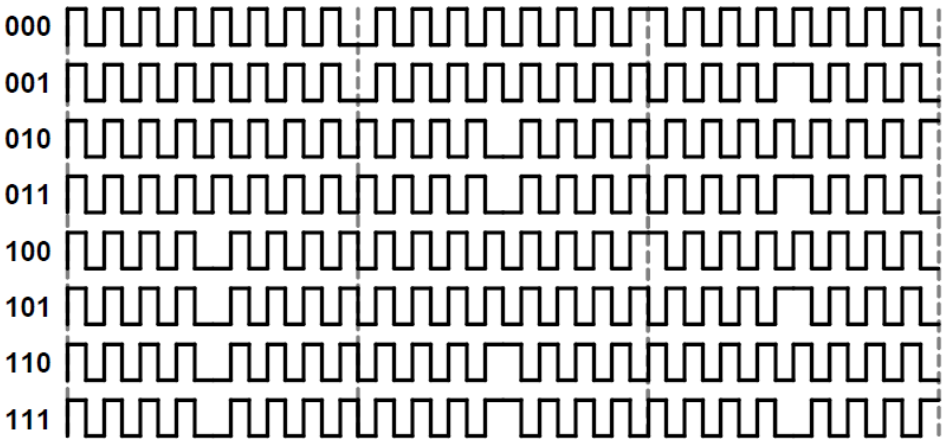

Figure 3.11: Miller sequence (for M=8), extracted from [7].

- T_2 Interrogator response time required, if a Tag is to demodulate the Interrogator signal, measured from the end of the last bit of the Tag response to the first falling edge of the Interrogators transmission(Equation: 3.4).
- T_3 Time an Interrogator waits, after T1, before it issues another command(Equation: 3.5).
- T_4 Minimum time between Interrogator commands(Equation: 3.6).

$$MAX(RTcal, 10{\times}T_{pri}){\times}(1-|FFT|){-}2\mu s \leq T_1 \leq MAX(RTcal, 10\times T_{pri}) \times (1+|FFT|)+2\mu s \tag{3.3}$$

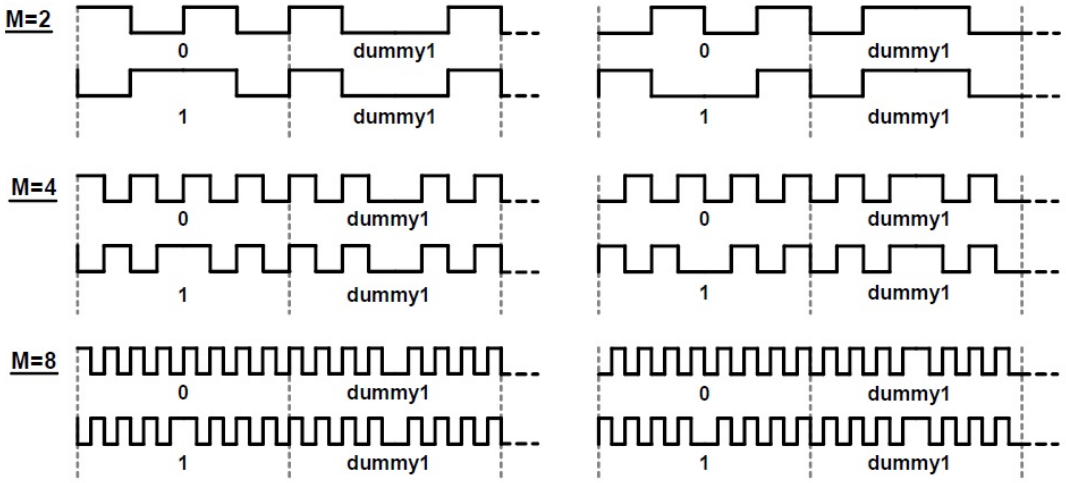

Figure 3.12: Miller end of signalling (for M=2,4,8), extracted from [7].

$$3T_{pri} \leq T_2 \leq 20T_{pri} \tag{3.4}$$

$$T_3 = 0.0T_{pri}, \tag{3.5}$$

$$T_4 = 2RTcal \tag{3.6}$$

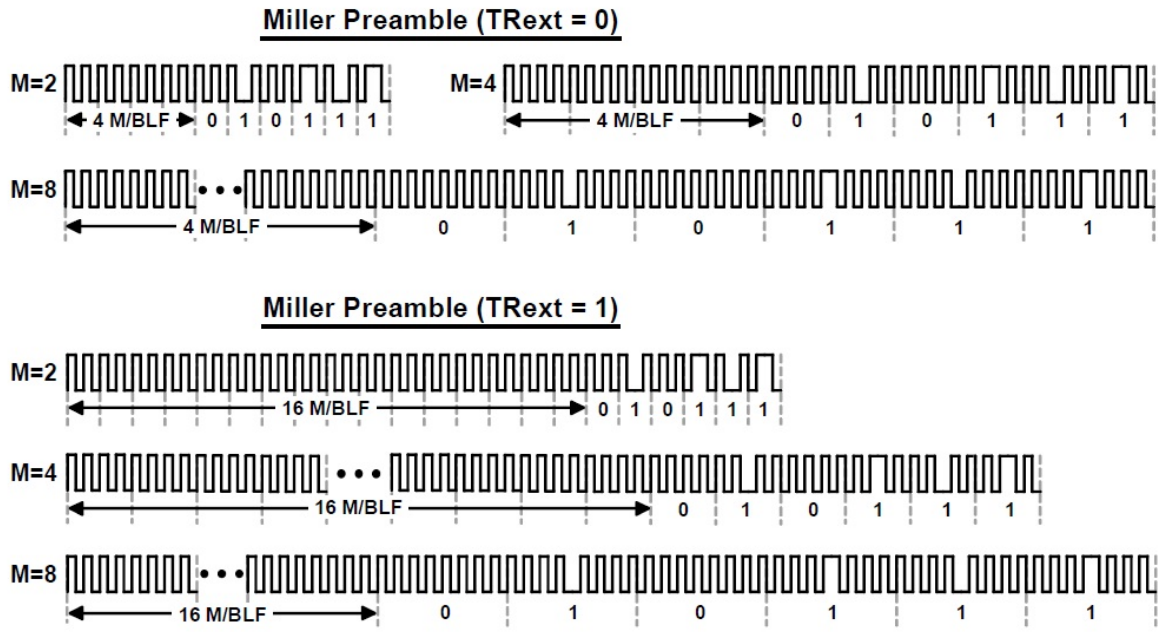

Figure 3.13: Miller Preamble for different values of M and Trext, extracted from [7].

3.2.3 Tag Identification Layer

An Interrogator manages Tag populations using three basic operations:

- **Select:** The operation of choosing a Tag population for inventory and access. A select command may be applied successively to select a particular Tag population based on user-specified criteria. This operation is analogous to selecting records from a database.

- **Inventory:** The operation of identifying Tags. An Interrogator begins an inventory round by transmitting a Query command in one of four sessions. One or more Tags may reply. The Interrogators detects a single Tag reply and requests the PC/XPC word(s), EPC and CRC from the Tag. Inventory comprises multiple commands. An inventory round operates in one and only one session at a time.

- **Access:** The operation of communicating with (reading from and/or writing to a Tag. An individual Tag must be uniquely identified prior to access. Access comprises multiple commands some of which employ one-time-pad based cover-coding of the R⇒T link.

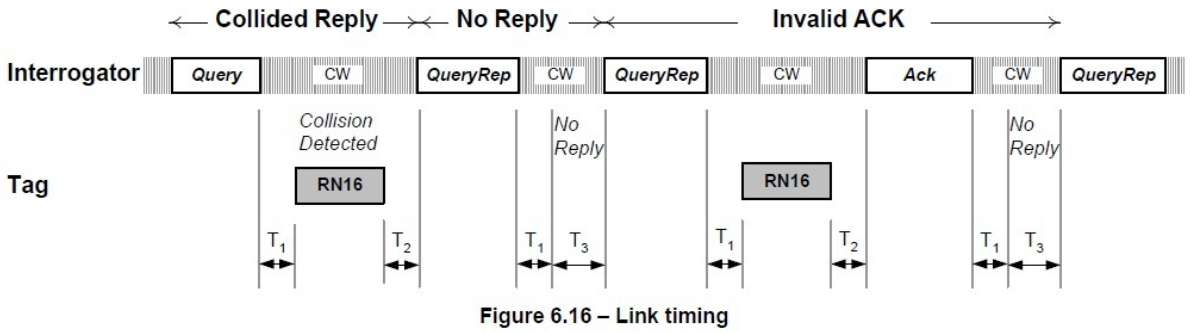

Figure 3.14: Link-timing, extracted from [7].

3.2.3.1 Inventorying Tag Populations

The Inventory command set includes Query, QueryAdjust, QueryRep, ACK and NAK. Query initiates an inventory round and decides which Tags participate in the round. An inventory round is a period initiated by a Query command and is terminated either by a new Query command(which starts a new inventory round) or a Select command. Query contains a slot-count parameter Q. Upon receiving a Query participating Tags pick a random value in the range $(0, 2^Q - 1)$, inclusive, and load this value into their slot counter. Tags that pick a zero value transition to the reply state and reply, immediately, while Tags that pick a non-zero value transition to the arbitrate state and await a QueryAdjust or a QueryRep command. For a single Tag, the algorithm proceeds as follows (Figure 3.15):

1. The Tag backscatters an RN16, as it enters reply.

2. The Interrogator acknowledges the Tag with an ACK, containing this same RN16.

3. The acknowledged Tag transitions to the acknowledged state, backscattering a reply shown in Figure 3.15.

4. The Interrogator issues a QueryAdjust or QueryRep command, causing the identified Tag to invert its inventoried flag and transition to ready, potentially causing another Tag to initiate a query-response dialogue with the Interrogator, starting in step (a), above.

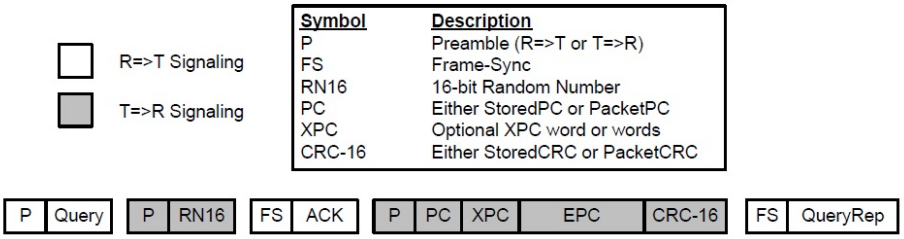

Figure 3.15: One Tag reply, extracted from [7].

If the Tag fails to receive the ACK in step (2) within time T2 (see Figure: 3.14) or receives the ACK with wrong RN16, it returns to arbitrate. If multiple Tags reply in step (1) but the Interrogator, by detecting and resolving collisions at the waveform level, can resolve an RN16 from one of the Tags, the Interrogator can ACK the resolved Tag. Unresolved Tags receive erroneous RN16, thus they return to arbitrate state, without backscattering the reply in (3). At some point the Interrogator will issue a new Qery, thereby starting a new inventory round. Tags in the arbitrate state decrement their slot counter every time they receive a QueryRep, transitioning to the reply state and backscattering an RN16, when their slot counter reaches 0000_h. Tags whose slot counter reached 0000_h, who replied and were not acknowledged (including Tags that responded to the original Query and were not acknowledged) return to arbitrate with a slot value of 0000_h and decrement this slot value from 0000_h to $7FFF_h$ at the next QueryRep, thereby effectively preventing subsequent replies, until the Tag loads a new random value into its slot value.

3.2.3.2 Tag states

Some of the states, used in this thesis, are described in this section in order to grasp a basic knowledge of the Tags' functionality. The Tags' states are:

- Ready: In the Ready state the Tag awaits either a Select or a Query command. After one of these commands is transmitted by the Interrogator, the Tag remains in this state.

- Arbitrate: While in this state, the Tag awaits for either a Select, a Query or a QueryRep command. If the Tag transmits a RN16, it moves to the Reply state.

- Reply: In the Ready state, the Tag waits for a Query, QueryAdjust or an ACK command. In case of an ACK command, it moves to the Acknowledged state. If a Query is transmitted, it initiates a new round, therefore the Tag moves to state Arbitrate. If a QueryAdjust is transmitted, the Tag backscatters a new RN16.

- Acknowledged: In this state the Tag backscatters an EPC word, then it stays silent for the remainder of the Inventory round.

3.2.3.3 Query Command

Query command initiates and specifies an inventory round. Query includes the following fields:

- DR (TRcal divide ratio) sets the T⇒R link frequency.

- M (cycles per symbol) sets the T⇒ data rate and modulation format.

- Trext chooses whether the T⇒ preamble is pretended with a pilot tone, unless a Tag replies to a command that writes to memory. In this case it, always, uses a pilot tone, regardless of TRext.

- Sel chooses which Tags respond to the Query.

- Session chooses a session for the Inventory round.

- Target selects whether Tags whose inventoried flag is A or B participate in the inventory round.

- Q sets the number of slots in the round.

The Query command's field are shown in Table 3.1. Interrogators precede a Query command with a preamble (Figure 3.2). The Query is protected by a CRC-5, if a Tag receives erroneous CRC-5 it will ignore the command. Upon receiving a Query, Tags with matching Sel and Target shall pick a random value in the range $(0, 2^Q - 1)$ and it shall load this value in their slot counter. If a Tag, in response to the Query, loads its slot counter with zero, then its reply is shown in Table 3.2, otherwise it will remain silent.

	Command	DR	M	TRext	Sel	Session	Target	Q	CRC-5
# of bits	4	1	2	1	2	2	1	4	5
description	1000	$DR = 8$ $DR = 64/3$	$00 : M = 1$ $01 : M = 2$ $10 : M = 4$ $11 : M = 8$	0:No pilot tone 1:Use pilot tone	00:All 01:All 10: SL 11:SL	00:S0 01:S1 10:S2 11:S3	0:A 1:B	0-15	

Table 3.1: Query Command.

	Response
# of bits	16
Description	RN16

Table 3.2: Tag reply to a Query command.

3.2.3.4 QueryRep

Interrogators and Tags implement the QueryRep command, as shown in Table 3.3. QueryRep instructs the Tags to decrement their slot counters and if slot=0 after decrementing, to backscatter an RN16 to the Interrogator. It is includes the following fields:

- Session, which indicates the session number for this round. If a Tag receives a QueryRep, whose session number is different from the session number of the Query that initiated the round, it will ignore the command.

A QueryRep is preceded with a frame-sync (Figure 3.3). If a Tag receives a QueryRep and after decrementing, its slot counter its slot counter value is zero, it backscatters a reply shown in table 3.4.

	Command	Session
# of bits	2	2
Description	00	00:S0 01:S1 10:S2 11:S3

Table 3.3: QueryRep.

	Response
# of bits	16
Description	RN16

Table 3.4: Tag reply to QueryRep command.

	Command	RN16
# of bits	2	16
Description	01	Echoed RN16 or handle.

Table 3.5: ACK command.

3.2.3.5 ACK

Interrogators and Tags shall implement the ACK command, shown in table 3.5. After a successful Query or QueryRep, an Interrogator sends an ACK command to acknowledge a single Tag. The Interrogator echoes back the RN16, sent previously by the Tag. If the RN16, sent by the Reader, is right and the Tag is in the right state, then the Tag will backscatter its reply, shown in table 3.6. All the process of extracting a stored EPC are seen in Figure 3.15.

	Response
# of bits	21 to 528
Description	see ACK reply from the Tag Figure 3.15

Table 3.6: Tag reply to ACK command.

Chapter 4

Synchronization and Detection

The next step of the signal processing, after the waveform extraction analysed in chapter 2, is the synchronization and the detection of the data packet, in this case the RN16 or the EPC, that is sent by the Tag. A proper synchronization detects where the packet, sent by the Tag, starts and enables proper detection of the packet's payload. However, as explained before, in chapter 2, the Interrogator has to synchronize and detect the symbols of an RN16 in a specific time, T1, otherwise even if it transmits the RN16 back to the Tag, the Tag will be already in the arbitrate state and will not backscatter an EPC word. Also, if it sends erroneous RN16 in its ACK command the Tag will not respond even the command was transmitted fast enough. Therefore, fast and reliable processing at the receiver must be achieved.

This chapter shows three synchronization schemes for both FM0 and Miller modulated encodings, the first uses the known preambles of the data packets sent, while the second uses the advantage of knowing the packets length and calculating its energy. The third method is a combination of the first two. Also, a noncoherent detection is presented for each of the encodings.

4.1 Synchronization

After the samples of the signal in absolute value are obtained, we must synchronize the received waveform, in order to determine where the packet starts, this is one step short of the detection of the payload, meaning the mapping of the data sequences to the

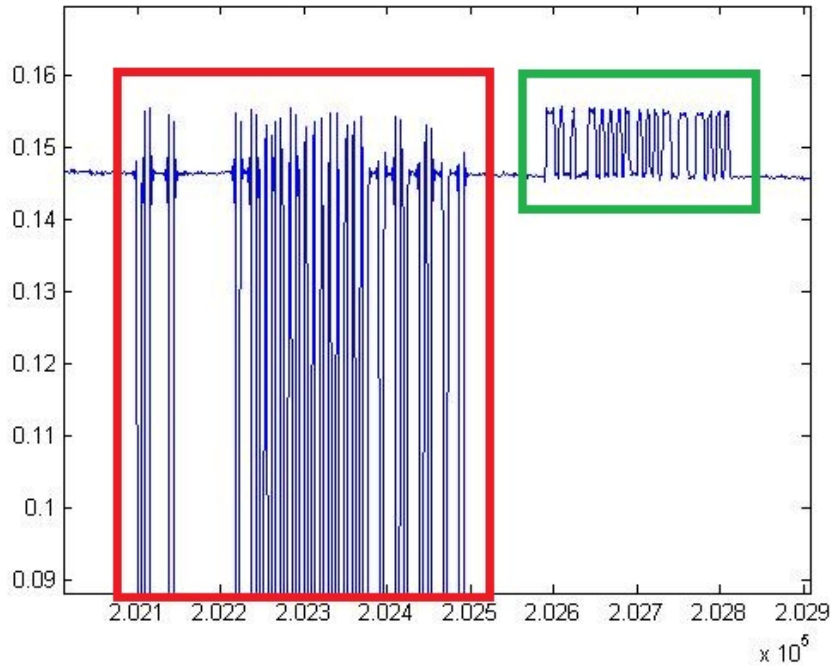

Figure 4.1: Left box Query from the Interrogator (Red box), Right box RN16 from the Tag (Green box).

Data mapping of a Preamble.											
data-1	data-1	data-0	data-1	data-0	data-0	data-1	data-0	data-0	data-0	data-1	data-1

Table 4.1: Data mapping of a FM0 Preamble.

symbols. One set of a Query and a RN16 response, with FM0 encoding, is shown in Figure 4.1, while Figure 4.2 shows the preamble of the same RN16.

As stated in chapter 2, the preamble of an FM0 encoded sequence is a set of predefined data-1s and data-0s, that map to certain symbols, according to the encoding. The sequence of the data-1s and data-0s, that compose the preamble, is shown in Table 4.1, both data-1s and data-0s have a length of $\frac{T}{2}$ where T is the duration of one pulse.

It is noted that the advantage of the preamble is located in the last three data-0s. We remind, the reader, that, in FM0 encoding, two sequential data-xs (with x being 0 or 1) correspond to one symbol. However, in a FM0 encoding the data-xs at the boundaries of the symbols must always be different, meaning that at a symbol boundary a data-1 can

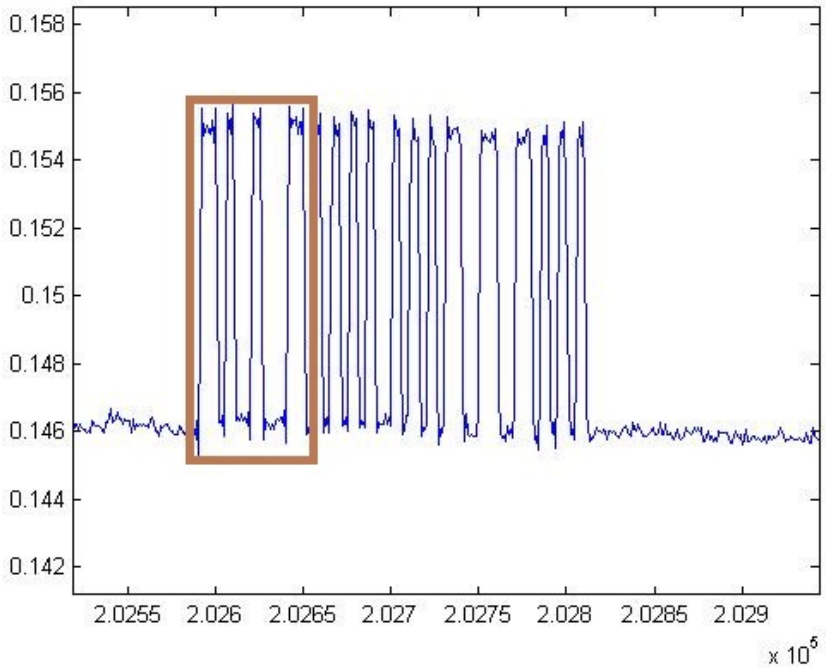

Figure 4.2: A FM0 RN16 response, in the box is the Tag's preamble.

only be followed by a data-0 and vice-versa.

The FM0 preamble, however, contains a violation of the encoding in the last three data-0' which makes it easier for the receiver to locate it with a correlation computation, because the violation sequence is unique in the packet. The same applies to the Miller encoding, the preamble contains an encoding violation, that can be exploited for synchronization, at the start of the packet. Four Miller encoded zeros are sent at the start of the packet, without a phase inversion at the symbol boundaries between them. Below, two packet synchronization methods found in [8] are presented, as well as one scheme that combines them. In order to eliminate the problem introduced by having inverse RN16 as shown in Figure 2.7, we estimate the value of the samples that correspond to the state where the Tag does not backscatter data, by extracting the average value of some samples before the position, where the preamble starts. Then, we subtract this average value from the absolute value of all the signal extracted. There are two possible outcomes for this scenario, the received packet will be either positive or negative, as shown in Figure 4.3. In case of an inverted packet, i.e. the data-1s are below the data-0s, the synchronized

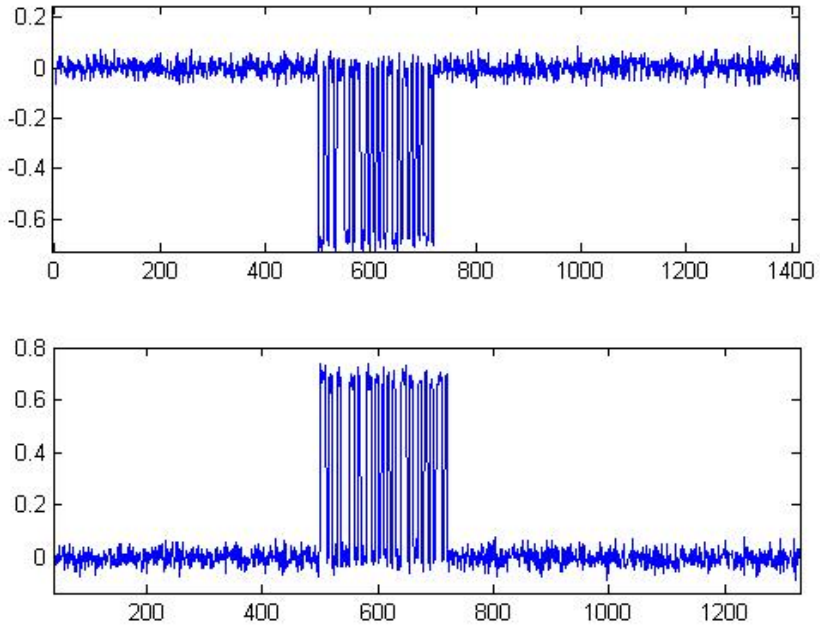

Figure 4.3: Two possible received waveforms after the sample extraction.

packet is inverted again, so that the next phase, that of the decoding, receives only one type of waveform.

4.1.1 Packet synchronization using Cross-correlation

In signal processing, cross-correlation is a measure of similarity of two waveforms as a function of time-lag applied to one of them. It is commonly used for searching a long signal for a shorter, known feature. For discrete functions the cross-correlation is defined as:

$$((f \star g)) = \sum_{m=-\infty}^{\infty} f^*[m]g[m+n] \tag{4.1}$$

In this case g and f function are the waveform and the known preamble, respectively. The f function contains the known preamble and we need to match the two functions in order to find where the packet starts in g. We can use the cross-correlation to find how much f waveform's preamble must be shifted along the x-axis to make it identical

to the g function. The above formula essentially slides the g function along the x-axis, calculating the integral of their product at each position. When the functions match, the value of $(f \star g)$ is maximized. This is because when peaks (either positive or negative) are aligned, they make a large contribution to the integral.

In our case, f waveform is consisted of real numbers, consequently $f = f^*$. This Cross-correlation synchronization technique can be used for both the FM0 and the Miller subcarrier encodings, since the Tags use a preamble, that contains an encoding violation i.e. the preamble is unique in the packet and known at the receiver, as explained in Chapter 3.

In order to decrease the computational complexity and be more accurate in our computations, we use the knowledge of the protocol specifications, meaning that if we start the processing of the packet in a little less time than T_1, we can search for the preamble in half samples of the packet, instead of searching in all the signal. The process is specified in the expressions below:

$$C_2[n] = \sum_{t=1}^{\infty} g[t]x[t+n], n \in [0, N/2] \tag{4.2}$$

Where:

- x is the received waveform.
- g is the known preamble.
- N is he number of received samples, after T_1.

$$b_{max} = \arg\max_{n} C_2 \tag{4.3}$$

Where b_{max} is the start of the packet.

As stated before, to implement this scheme, we use the known preamble and perform a correlation operation, using the known preamble and the inverse known preamble, in order to decide where the packet starts, even in case where the received packet is inverted. The correlation operation that returns the maximum result is assumed indicate, where the packet starts and if it is inverted or not. In case where the packet is inverted, we invert the signal, in order to transfer the packet to the positive side of the axis.

4.1.2 Packet synchronization using the packet's energy

Another method for packet synchronization is by using the waveform's energy. In signal processing, a signals' energy is defined as:

$$E = \int_{t=-\infty}^{\infty} |x(t)|^2 \tag{4.4}$$

where x is the received waveform. However, the Interrogator has knowledge of the packet's length in samples N, therefore it can calculate the energy of the signal using the equation below:

$$E(k) = \sum_{t=k}^{k+N-1} |x(t)|^2 \tag{4.5}$$

The above equation expresses the energy of the signal at the k_{th} position and for length N. The energy will be maximized at the start of a packet, since its the position where two signals, that of the Interrogator's carrier and that of the backscattered reply of the Tag, combine.

There is one case where the dummy bit '1' of a received packet results in a data-0, meaning that the last samples of the packet will be around the value 0. Taking into consideration, that before a packet starts the samples will also be around the value 0, then this method may be mistaken. To eliminate this case in both the Miller and the FM0 encodings after the first estimation of the start of the packet we use a window of length T, where T is the duration of one bit, around the first estimation of the start of the packet and we compute the energy of the signal for length P, where P is the length of the known preamble. We assume that the packet starts where this second computation is maximized. The preamble starts and ends with data-1s, therefore the problematic case is eliminated. The procedure is expressed as follows:

$$E_1(k) = \sum_{t=k}^{k+N-1-\frac{T}{2}} |x(t)|^2, \forall k \tag{4.6}$$

$$a_{max} = \arg\max_{k} (E_1) \tag{4.7}$$

$$E_2(k) = \sum_{t=k}^{k+P-1} |x(t)|^2, k \in (a_{max} - \frac{T}{2}, a_{max} + \frac{T}{2}) \tag{4.8}$$

$$b_{max} = \arg\max_{k} E_2 \tag{4.9}$$

Where b_{max} is the start of the packet.

Because of the absolute nature of the signal, the case, where the signal is inverted is of no consequence, however after the synchronization we have to detect if the packet is inverted, by estimating the mean value of the preamble bits. If it is inverted, we invert it again in order to decrease the computational complexity of the next steps.

4.1.3 Packet synchronization using both Energy and Cross-Correlation

Another method for packet synchronization is an algorithm, that exploits the energy and the packet synchronization, from now one we call this kind of synchronization, hybrid synchronization. This scheme, firstly, uses the energy synchronization to estimate where the packet starts, then it computes the cross-correlation of the packet and the known preamble to estimate where the packet starts, instead of using the energy of the preamble, again.

$$E_1(k) = \sum_{t=k}^{k+N-1} |x(t)|^2, \forall k \tag{4.10}$$

$$a_{max} = \arg\max_{k} (E_1) \tag{4.11}$$

After finding the start of the packet using the energy synchronization, we perform the cross-correlation:

$$C_2[n] = \sum_{t=1}^{\infty} g[t]x[t+n], n \in [a_{max} - \frac{T}{2}, a_{max} + \frac{T}{2}] \tag{4.12}$$

Where:

- x is the received waveform.
- g is the known preamble.
- N is he number of received samples.
- P is the number of the samples in the preamble.

- T is the period of one symbol.

$$b_{max} = \arg\max_{k} C_2 \tag{4.13}$$

Where b_{max} is the start of the packet.

4.1.4 Simulation Results

In this section, the simulation results are presented for each of the three synchronization schemes. In Figure 4.4, the cross-correlation, the energy and the hybrid synchronization methods are compared with a perfect synchronization for the FM0 encoding. The energy and the correlation schemes seem to give similar results with the correlation synchronization being slightly better, especially in high signal-to-noise-ratios (SNRs). However, the hybrid synchronization seems to work better due to the fact that the energy synchronization estimates, where the packet starts and then we use a window to implement the correlation synchronization, while the correlation synchronization will compute integrals from the start of the signal, to half the packet meaning that similar waveforms to the preamble may result to an erroneous synchronization. The cross-correlation synchronization works better, as more bits are added to the preamble, using the pilot tone function of the Gen2 protocol, as seen in Figure 4.5.

The Miller encoding synchronization simulation results are shown in Figure 4.6. In this encoding, the better function of the cross-correlation synchronization scheme, which works ,almost, as well as the hybrid synchronization is noticeable. While, even better results can be seen, when using a pilot tone in Figure 4.7.

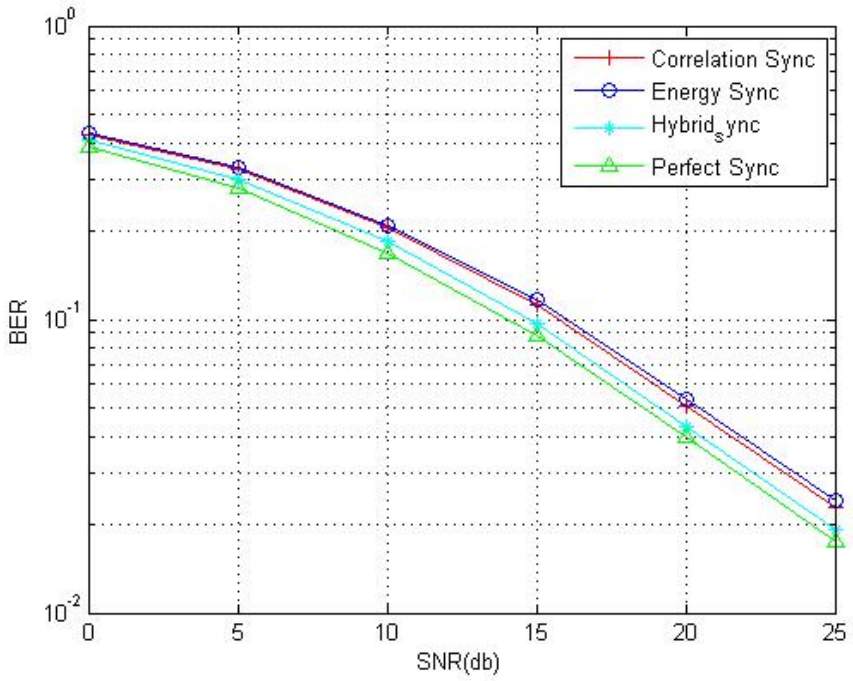

Figure 4.4: FM0 encoding, energy, correlation synchronization and hybrid synchronization BER.

4.1.5 Detection

Regardless of the packet synchronization method used, the received packet's start is known. Now, the known preamble can be used to establish a threshold, in order to determine which bit is sent '0' or '1'. The average value of the samples of the preamble is calculated and then subtracted from the signal as in Figure 4.8. The known preamble contains equal number of '0' and '1' samples, therefore this threshold is considered "fair".

4.1.5.1 FM0 Detection

A simple symbol detection method for FM0 encoding under ideal conditions, i.e. without the noise component and the symbol shift, because of the Tag's data rate variation, would be to adaptive downsample the received waveform so as that one sample corresponds to one bit. Then, by comparing two neighbouring bits, we determine the symbol sent. However, under normal conditions there is a noise component and there is attenu-

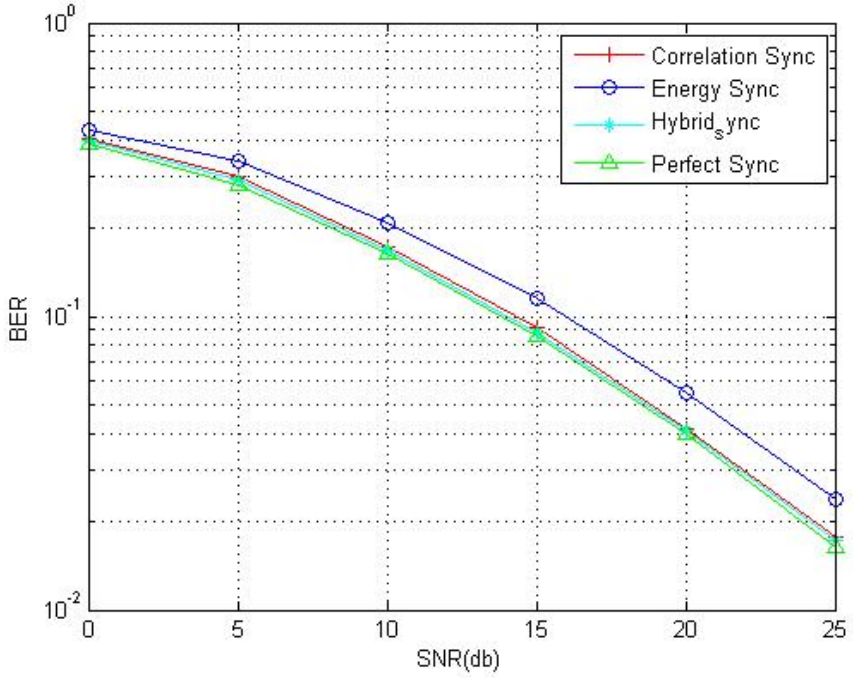

Figure 4.5: FM0 encoding, energy, correlation synchronization and hybrid synchronization BER with pilot tone.

ation at the signal because of the channel, the sample extracted from the downsampling could change, making a proper symbol detection, difficult. Therefore, it is considered safer to examine chunks of bits, to determine which data-x was sent by the Tag, because more bits contain more information.

Using the observations in [9], where the memory of the FM0 encoding is exploited, we detect the symbols received by shifting the received waveform at $b_{max} + P - \frac{T}{2}$, where P is the preamble length and T is the duration of one symbol. Initial symbol synchronization occurs, using a packet synchronization method. Assuming that the symbol period is stable, which means we will always be successful, in finding the correct start of the k_{th} symbol $x_{th}[t]$, that has the form shown in Figure 4.9 and Equation 4.14.

These two symbol waveforms are orthogonal and reduce the amount of calculations for symbol synchronization in half. The functions D and D' are defined for the FM0

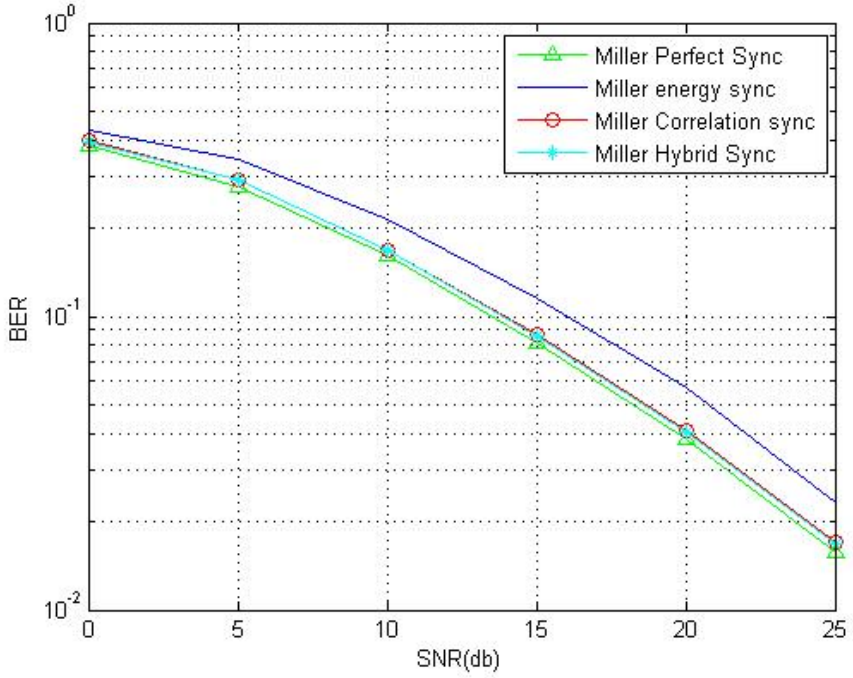

Figure 4.6: Miller encoding, energy, correlation synchronization and hybrid synchronization BER.

decoding:

$$D = \begin{cases} 1 & 0 < m \leq \frac{M}{2} \\ -1 & \frac{M}{2} < m \leq M \\ 0 & elsewhere \end{cases} \tag{4.14}$$

$$D' = -D \tag{4.15}$$

We determine the symbol sent by correlating $x[n]$ with the two waveforms of Equations 4.14 and 4.15 and comparing the two integrals, the waveform that gave the maximum integral, corresponds to the symbol sent. One of the waveforms corresponds to an initial, known to the receiver, symbol that is contained in the preamble.

The first waveform extracted, will always be that of the second half of the last symbol 1 of the preamble. The following waveforms will be either D or D'. The waveforms extracted are stored in a matrix L. We determine which symbol was sent by comparing two neighbouring symbols. A change in waveforms, i.e. if the first symbol is D and the

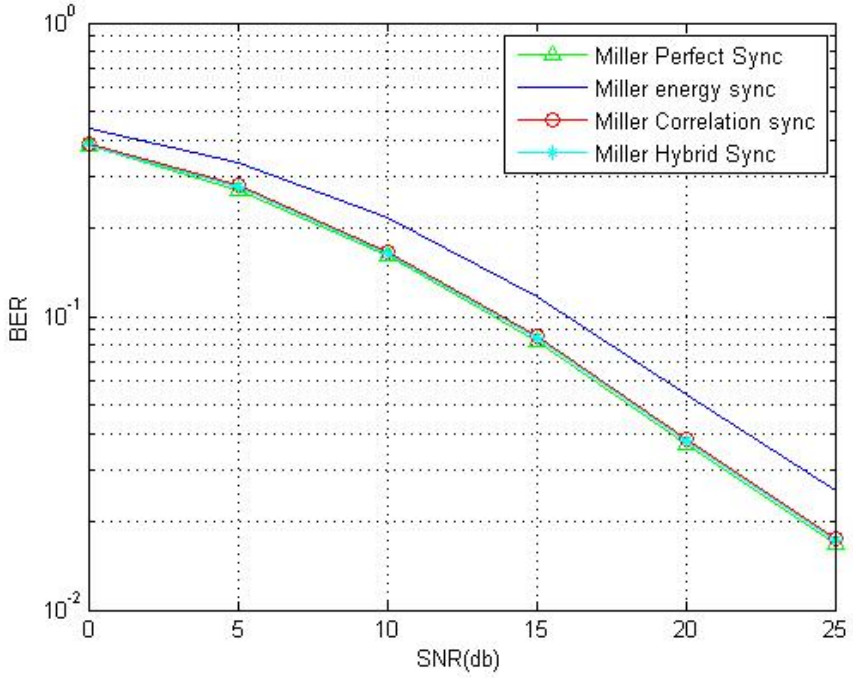

Figure 4.7: Miller encoding, energy, correlation synchronization and hybrid synchronization BER with pilot tone.

second is D' and vice versa, means that a symbol 1 was sent, otherwise we determine a symbol 0. The procedure is expressed in the equation below:

$$L_k = \begin{cases} 1 & \sum_{i=1}^{N} x_k[i]D[i] > \sum_{i=1}^{N} x_k[i]D'[i] \\ 0 & \text{else} \end{cases} \tag{4.16}$$

$$a_{k+1} = \begin{cases} 0 & \text{if } L_k = L_{k+1} \\ 1 & \text{else} \end{cases} \tag{4.17}$$

Where N is the number of samples in a symbol and a_k is the detected symbol.

For example consider the packet received, shown in Figure 4.10, the transmitted RN16 is "1111000000110111", we easily detect it by seeing the figure's waveforms and comparing them with the FM0 encoding symbols shown in chapter 2. As stated before, the detection begins at time $\frac{T}{2}$ of the previous symbol sent which is the last symbol '1' of the preamble. The first '1', that of the preamble, is detected and is a D waveform which means that $L_1 = 1$. The second waveform is located between the vertical lines, that are

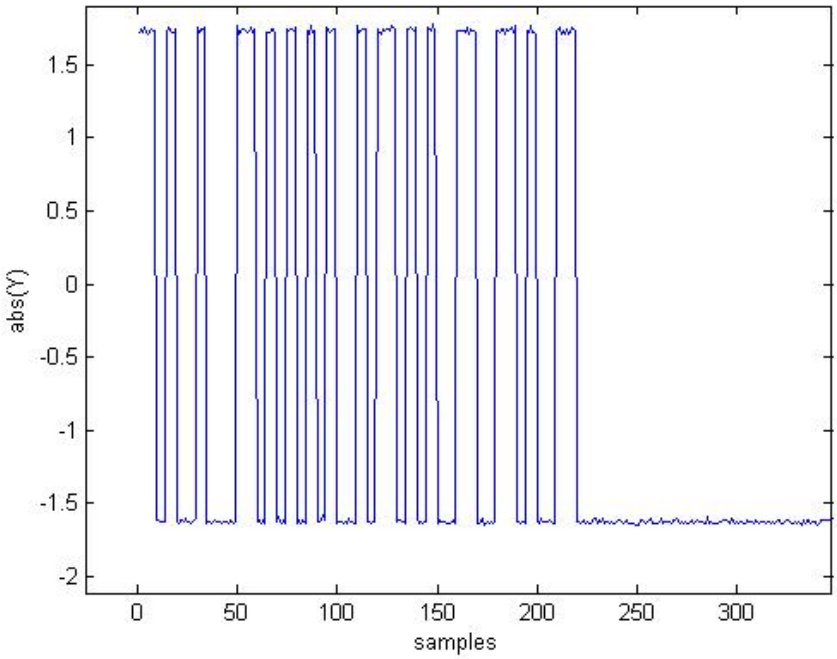

Figure 4.8: A synced RN16 reply (FM0 encoding).

located between the 60_{th} and 80_{th} samples, it is a D',which means that $L_2 = 0$ therefore $a_2 = 1$ because $L_1 \neq L_2$. Then we compare the second waveform to the third which is a D, which means that $L_3 = 1$, therefore $a_3 = 1$. This procedure is continued, until all the waveforms are compared, including that of the 'dumb' bit at the end of the packet. Then, the received RN16 will be all the symbols from a_2 to a_i, where i is the last useful bit.

4.1.6 Miller Detection

The same logic, as in FM0 detection, is applied to Miller encoding detection. The waveform is shifted at $P - \frac{T}{2}$, where P is the size of the preamble and T is the period of one symbol, meaning that the decoding starts at the second half of the last symbol of the preamble.

$$Dmil_1 = \begin{cases} D & 0 < m \leq \frac{K}{2} \\ D' & \frac{K}{2} < m \leq K \\ 0 & elsewhere \end{cases} \tag{4.18}$$

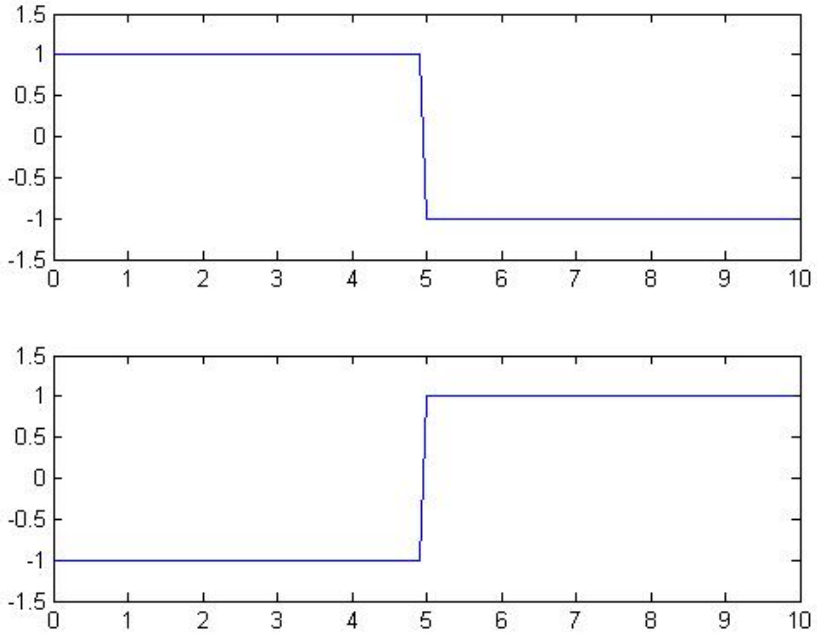

Figure 4.9: Possible symbol waveforms after shifting the signal at $\frac{T}{2}$, FM0 encoding.

$$Dmil_2 = \begin{cases} D & 0 < m \leq \frac{K}{2} \\ D & \frac{K}{2} < m \leq K \\ 0 & elsewhere \end{cases} \tag{4.19}$$

$$Dmil_3 = \begin{cases} D' & 0 < m \leq \frac{K}{2} \\ D & \frac{K}{2} < m \leq K \\ 0 & elsewhere \end{cases} \tag{4.20}$$

$$Dmil_4 = \begin{cases} D' & 0 < m \leq \frac{K}{2} \\ D' & \frac{K}{2} < m \leq K \\ 0 & elsewhere \end{cases} \tag{4.21}$$

Where $K = 2T$ in the above equations and T is the length of the waveforms D and D'. These waveforms are shown in Figure 4.11. In case $M = 4$ or $M = 8$ the $Dmil$ waveforms take a form to match the transmitted symbols of Miller encoding as shown in chapter 2.

The symbols to be detected for $M = 2$ are shown in equations 4.18 to 4.21, the Interrogator has to correlate all the known waveforms with the synchronized received waveform, the function that leads to the maximum correlation corresponds to the received

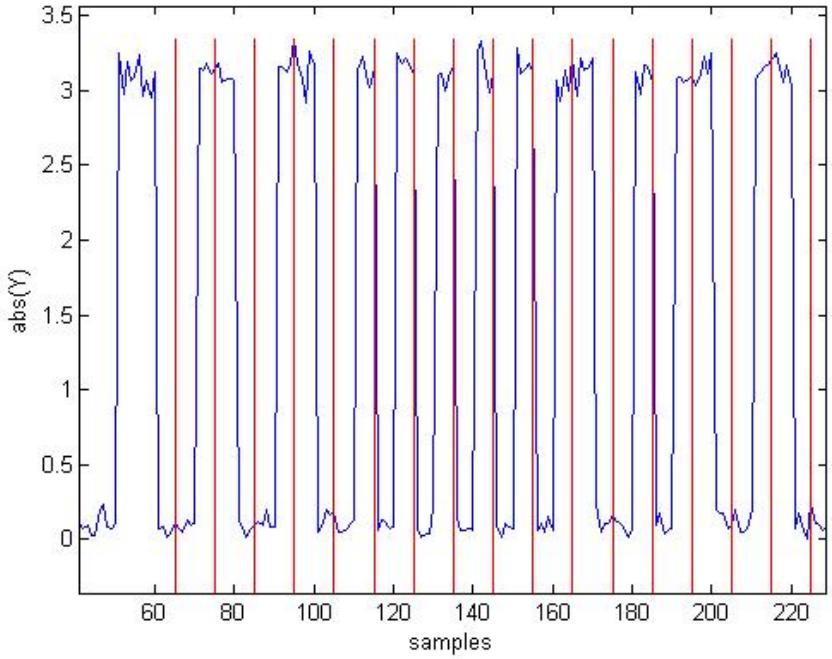

Figure 4.10: Example received FM0 packet.

symbol, then we compare the neighbouring symbols, if the neighbouring symbols change amplitude, we detect a symbol '1', otherwise we detect a symbol '0'.

$$bit_k = \arg\max_j (\sum_{i=1}^{K} x_k[i] Dmil_j[i]), j \in \{1, 2, 3, 4\} \tag{4.22}$$

where:

- K is the number of samples in a symbol.
- $Dmil_1$ to $Dmil_4$ are the waveforms expressed in equations 4.18 to 4.21, respectively.
- x_k is the synchronized k_{th} received waveform, that corresponds to a symbol.
- bit_k is the decoded half symbol.

The transitions, for a symbol '1', allowed by the encoding, are shown in Figure 4.12. If any other transition is detected a symbol '0' is detected. Therefore, the detection

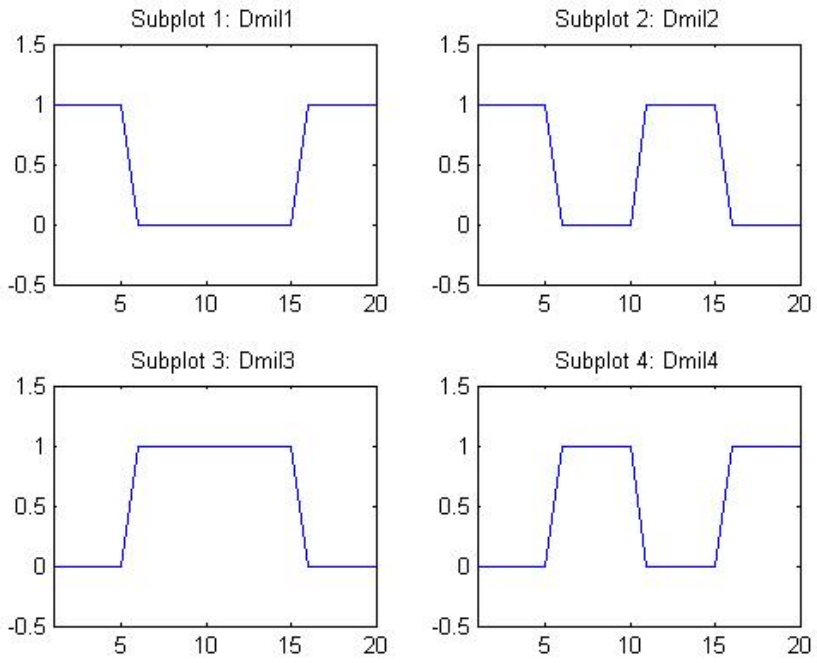

Figure 4.11: Miller modulated encoding symbol '1s' (left column) and '0s' (right column) for M=2.

algorithm is:

$$a_k = \begin{cases} 1 & \text{if } bit_{k-1} = 4 \text{ and } bit_k = 1 \text{ or } bit_k = 2 \text{ or} \\ & \text{if } bit_{k-1} = 2 \text{ and } bit_k = 3 \text{ or } bit_k = 4 \text{ or} \\ & \text{if } bit_{k-1} = 3 \text{ and } bit_k = 4 \text{ or } bit_k = 3 \text{ or} \\ & \text{if } bit_{k-1} = 1 \text{ and } bit_k = 1 \text{ or } bit_k = 2 \\ 0 & else \end{cases} \tag{4.23}$$

Where a_c is the decoded symbol. The same method is applied for $M = 4$ and $M = 8$, where the only difference is that the waveforms $Dmil_j, j \in \{1, 2, 3, 4\}$ change, accordingly.

For example, consider the Miller received packet, that is shown in Figure 4.13 for $M = 2$, the payload sent is '1111010011011111' and the symbols sent are between the blue vertical lines. We know that the last half symbol of the preamble is a $Dmil_4$. It is easy to notice, that the first half symbol in the payload corresponds to $Dmil_2$. However, we know that if the phase changes, a symbol '1' was sent. Then, we compare the first half symbol of the payload with the second and the procedure is continued, until all the symbols of the payload are detected.

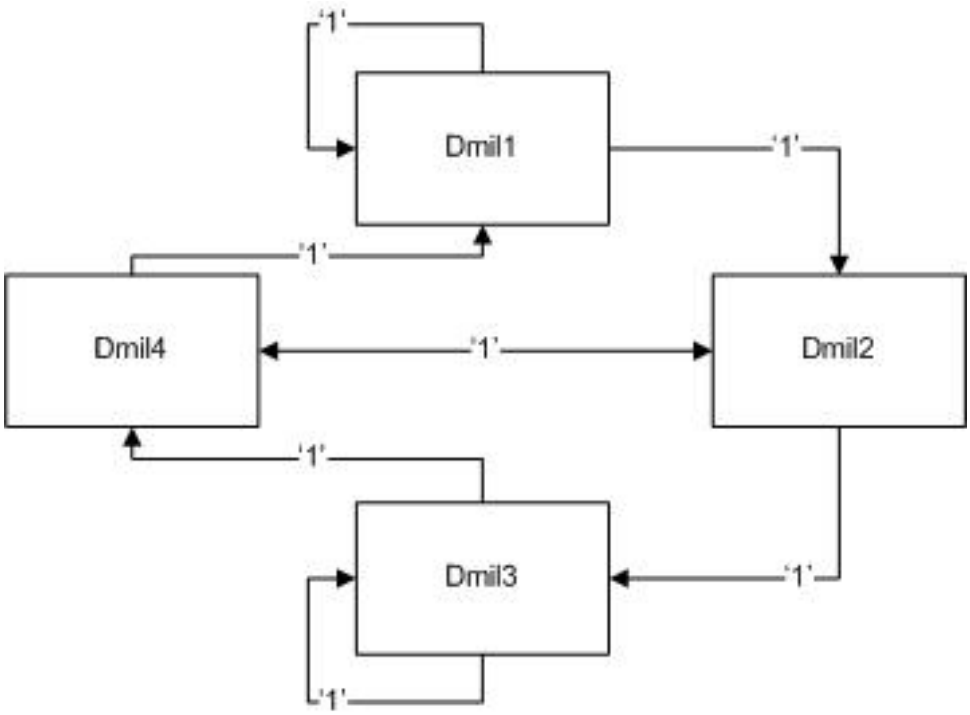

Figure 4.12: Miller allowed transitions for a bit '1', according to the symbol received.

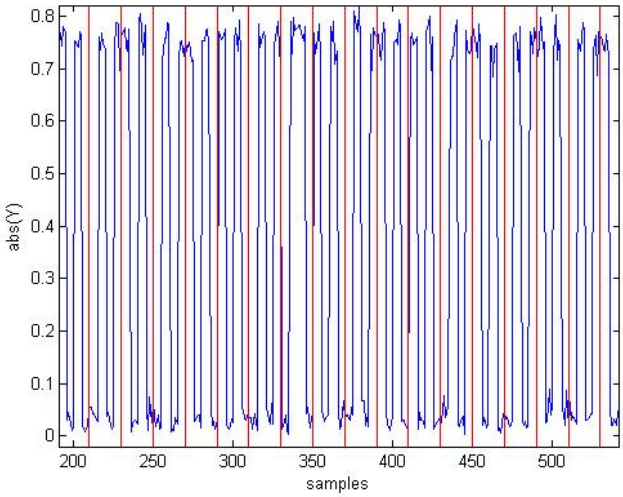

Figure 4.13: A received Miller modulated packet.

Chapter 5

Packet Collision Detection

As explained in chapter 3, when an Interrogator issues a Query command, it specifies the number of time slots, N, that exist for the given Query, meaning the number of times it will issue a QuerRep command. When a Tag detects a Query command, it produces a random number s where $s \in [0, 1, ..., N-1]$. If $s = 0$ the Tag backscatters an RN16, otherwise it waits for the next QueryRep command. When a QueryRep command is transmitted, the Tag decreases s by 1, then it assesses if $s = 0$ and if the condition is true, it transmits its RN16, otherwise it waits for the next QueryRep command. This procedure is continued, until the Interrogator seizes to transmit.
In case where two or more different Tags produce the same random number s, they will attempt to transmit in the same time slot resulting in a packet collision and a possible erroneous detection of the packet by the Interrogator. The erroneous detection of the RN16 is undesired, because if an Interrogator transmits back erroneous RN16 for both Tags in its ACK command, then both Tags will not transmit their EPC, resulting in time loss, as the Interrogator will have to send a new Query to receive their EPCs. Therefore, algorithms for detecting an RN16 in two or more collided packets are required, in order to enhance the protocol. However to decode an RN16 in a collision, the collision must be detected first.

In this chapter a Tag collision occurrence scheme is presented and tested, through simulations.

5.1 Collision Detection

Given that tag transmission (via backscatter) in commercial RFID protocols is, always, initiated and directed by the Interrogator, while the typical range of such systems is on the order of a few meters and the minimum bit duration is on the order of a few microseconds, one would expect the two collided tag signals to arrive at the Interrogator with negligible time difference, compared to the bit duration and aligned bit boundaries [10].

If a Tag collision occurs, meaning that two or more Tags attempt to transmit their RN16 in the same timeslot, a signal as in Figure 5.1 is received. Because of the same time of arrival of the packets at the receiver, an enhanced preamble can be seen, followed by 16 random symbols that collide and result in four separate levels, instead of two. Two of these levels are the same as in the preamble, while the remaining two are collided data-1s and data-0s.

These four levels after calculating the absolute value of the signal and subtracting the value of the carrier will be a,b,c and 0. Where a and 0 are located only in the preamble and all of them are located in the mean value of the signal. Therefore, the mean energy of the samples of the preamble will be $\frac{a}{2}$, because it is consisted of the same number of data-1s and data-0s. The the mean energy, per sample, of the rest of the signal before the dummy, meaning the payload, is expected to be $\frac{a+b+c}{4}$, as the symbols sent by the Tags are equally probable to be sent, while if there is no collision present the energy of the preamble is expected to be $\frac{a}{2}$ in the average case. Thus, a collision occurrence can be detected, by comparing the ratio of energy of the preamble, with the energy of the payload and establishing a threshold, which indicates there is a collision. If there is no collision, we expect the ratio aforementioned to be close to 1, however there will be deviations to this value, because of the channel, the noise and the values of the payload.

$$det = \begin{cases} 0 & \text{if } \frac{E_{preamble}}{E_{payload}} \simeq 1 \\ 1 & \text{else} \end{cases} \tag{5.1}$$

Where:

- When a collision is detected, $det = 1$.
- $E_{preamble}$ is the energy, per sample, of the preamble.

- $E_{payload}$ is the energy, per sample, of the payload.

5.2 Simulation Results

This section shows the simulation results, for the detection of a collision, when none, one, two, three and five Tags ,that use FM0 encoding, participate in it. The average preamble energy to the average payload energy ratio thresholds are set to 0.95 and 1.05 meaning that, if the ratio is not somewhere in between, a collision is detected. As seen in Figure 5.2, as more Tags participate in a collision, the scheme improves in accuracy to above 80% for simple two Tags collision, to above 90% for the five Tags collision scenario. That is because as the SNR ratio is increased and the noise affects less the average preamble energy to the average payload energy ratio ratio. For the same reasons, as the SNR ratio is increased, the failure of the algorithm is decreased, when only one Tag is present, to below 3%. Furthermore, by increasing the size of the preamble, using pilot tone, the failure rate of the algorithm is decreased, as shown in Figure 5.3, because with more samples, the preamble energy can be estimated more accurately.

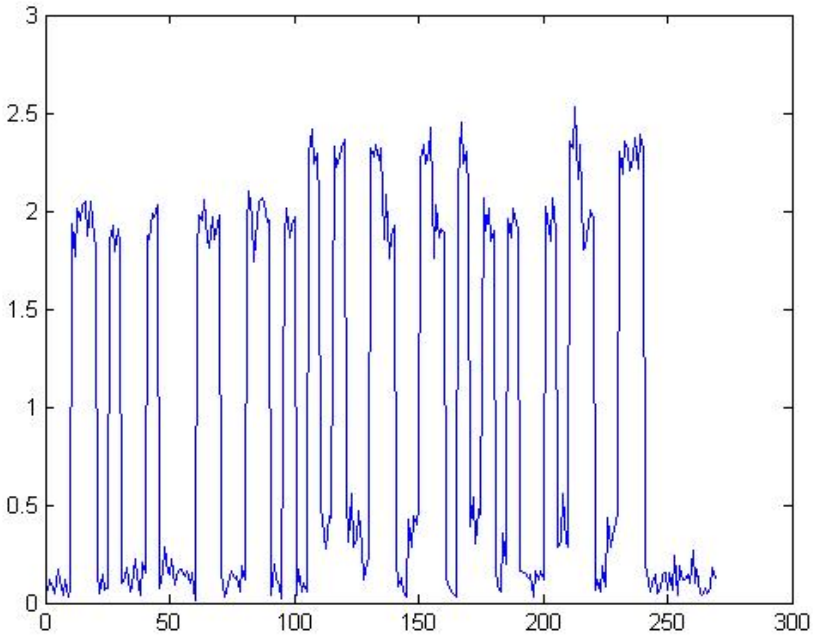

Figure 5.1: Sample collision FM0 encoding.

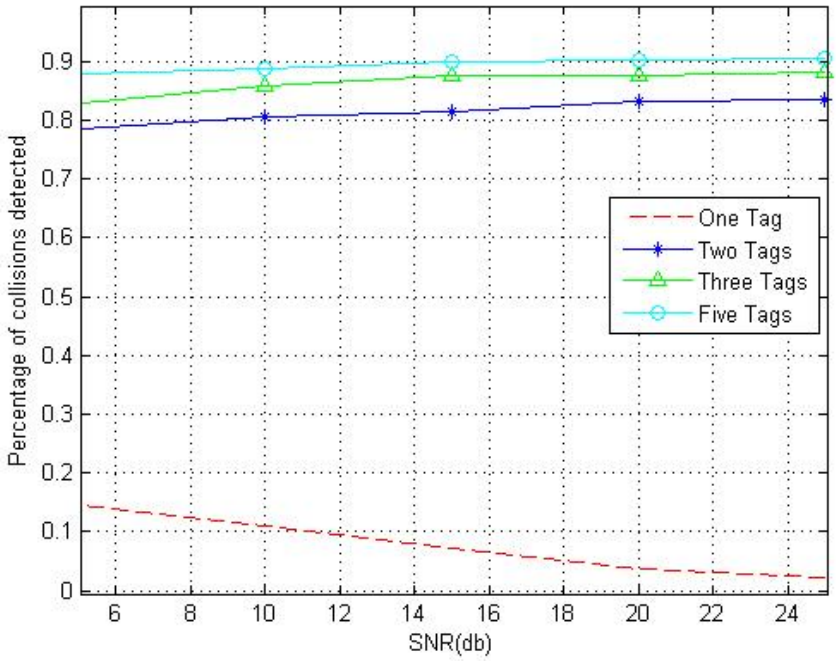

Figure 5.2: Success rate of packet collisions with multiple Tags.

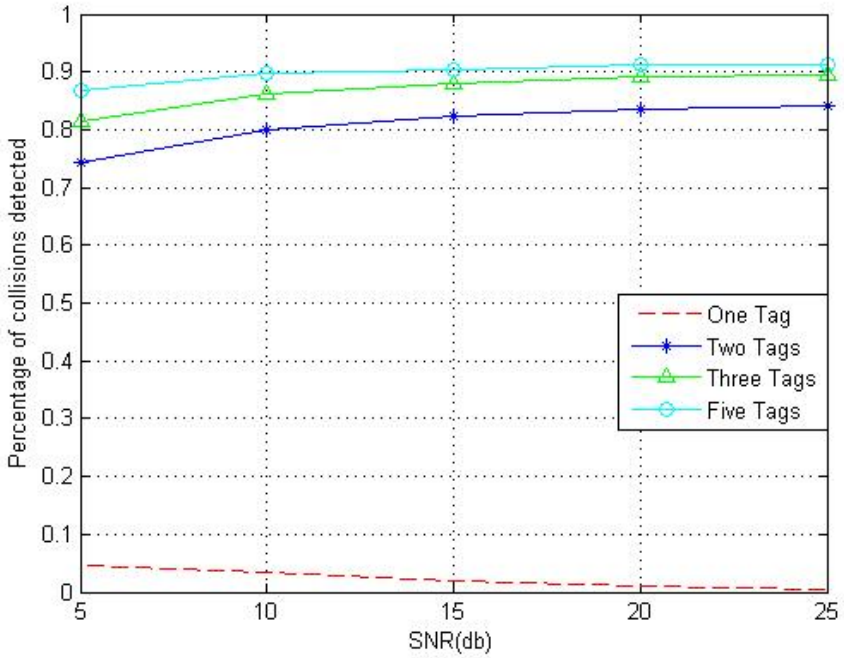

Figure 5.3: Success rate of packet collisions with multiple Tags, using pilot tone.

Chapter 6

Conclusion

6.1 Conclusion

This thesis explored the noncoherent reception and signal processing schemes for the EPC Gen2 Protocol. Three synchronization schemes were compared for two different encodings and we reached a conclusion that the cross-correlation synchronization method works better than the energy synchronization for all SNRs. Also a hybrid synchronization, combination of the energy and the correlation synchronization was tested and was found more efficient and faster than the other two, because it exploits their advantages. Also, the detection schemes, for two encodings, that were used to evaluate the synchronization schemes were shown, as well as, a symbol synchronization scheme, that was suggested to eliminate the problem of the non-stable symbol period. Furthermore, a collision occurrence detection scheme was presented and shown in simulations. It proved efficient, because it improves as more Tags try to transmit in one timeslot, while the failure, i.e. if one Tag transmits and is perceived as a collision, drops below 3% as the SNR grows.

References

[1] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Increased range bistatic scatter radio," *IEEE Transactions on Communications (TCOM)*, vol. 62, no. 3, pp. 1091–1104, March 2014. 3, 7, 8

[2] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. New Jersey: John Wiley and Sons,, 2005. 6, 8

[3] A. Bletsas, A. Dimitriou, and J. Sahalos, "Improving backscatter radio tag efficiency," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 6, pp. 1502–1509, June 2010. 6, 8

[4] C. Angerer, "Design and exploration of radio frequency identification systems by rapid prototyping," July 2010. 8

[5] A. P. Liavas, "Coursenotes for lesson: Telecommunication systems I." 10

[6] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '12. New York, NY, USA: ACM, 2012, pp. 61–72. 15

[7] "EPCTM radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz 960 MHz version 1.2.0," October 2008. 16

[8] A. P. Liavas, "Coursenotes for lesson: Telecommunication systems II." 33

[9] M. Simon and D. Divsalar, "Some interesting observations for certain line codes with application to RFID," *Communications, IEEE Transactions on*, vol. 54, no. 4, pp. 583–586, April 2006. 40

[10] A. Bletsas, J. Kimionis, A. G. Dimitriou, and G. N. Karystinos, "Single-antenna coherent detection of collided FM0 RFID signals," *IEEE Transactions on Communications (TCOM)*, vol. 60, no. 3, pp. 756–766, March 2012. 50