ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SECONDA FACOLTÁ DI INGEGNERIA - SEDE DI CESENA
Corso di Laurea Magistrale in Ingegneria Elettronica e delle Telecomunicazioni per lo Sviluppo Sostenibile

Development of Software Tools for the Test of Ultra Wide Band Receivers

Elaborato in Reti di Sensori Wireless per il Monitoraggio Ambientale

Relatore:
Chiar.mo Prof. Davide Dardari

Presentata da:
Sergio Contino

Correlatori:

Dott. Ing. Marco Bottazzi

Dott. Francesco Guidi

Sessione II
Anno Accademico 2011-2012

Contents

Introduction

In the last years, the importance of locating people and objects and communicating with them in real time has become a common occurrence in every day life.

Nowadays, the state of the art of location systems for indoor environments has not a dominant technology as instead occurs in location systems for outdoor environments, where GPS is the dominant technology.

In fact, each location technology for indoor environments presents a set of features that do not allow their use in the overall application scenarios, but due its characteristics, it can well coexist with other similar technologies, without being dominant and more adopted than the others indoor location systems.

In this context, the European project SELECT studies the opportunity of collecting all these different features in an innovative system which can be used in a large number of application scenarios.

The goal of this project is to realize a wireless system, where a network of fixed readers able to query one or more tags attached to objects to be located.

The SELECT consortium is composed of European institutions and companies, including Datalogic S.p.A. and CNIT, which deal with software and firmware development of the baseband receiving section of the readers, whose function is to acquire and process the information received from generic tagged objects.

Since the SELECT project has an highly innovative content, one of the key stages of the system design is represented by the debug phase.

This work aims to study and develop tools and techniques that allow to perform the debug phase of the firmware of the baseband receiving section of the readers.

For this purpose, this work is organized as follows:

1. In chapter 1, the project SELECT is described, with its main goals.

2. In chapter 2, the system that SELECT project aims to realize is pre-

sented, with its structure and its features.

3. In chapter 3, a Matlab simulator implemented for the firmware debug of the baseband receiving section of a SELECT reader is described.

4. In chapter 4, a Matlab simulator to generate ideal input files for FPGA debug is presented.

5. In chapter 5, preliminary feasibility studies are shown and some techniques enabling the debug phase in real time are proposed.

Chapter 1

SELECT European Project and Technologies

SELECT (Smart and Efficient Location, Identification and Cooperation Techniques)[1] [1] is an European project composed of a consortium of European institutions and companies coordinated by Datalogic S.p.A.. In particular, the SELECT partners are [2]:

- Datalogic S.p.A (Italy).
- Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V. (Germany).
- Commisariat à l'Energie Atomique o CEA-LETI (France).
- Consorzio Nazionale Interuniversitario per le Telecomunicazioni o CNIT - Università di Bologna (Italy).
- Centro de Estudios e Investigaciones Tecnicas de Gipuzkoa o CEIT (Spain).
- Ecole Nationale Supérieure de Techniques Avancées o ENSTA-ParisTech (France).
- Novelda AS (Norway).
- ORIA, Poslone Rešitve, D.O.O. (Slovenia).

[1] The research leading to these results has received funding from the [European Union] [European Atomic Energy Community] Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement n.[257544].

select
Smart and Efficient Location, idEntification, and Cooperation Techniques

Figure 1.1: SELECT project logo

Figure 1.2: Seventh Framework Programme logo

Figure 1.3: Datalogic S.p.A. logo

Figure 1.4: CNIT logo

1.1 SELECT Project Objectives

The SELECT project goal is to realize a wireless integrated solution for detection, identification, localization and tracking of objects in real time.

For this purpose, the design of a low-cost system composed by multiple tags and capable to perform secure wireless communications and multiple access with a network of fixed radio devices, cooperative and self-configuring is required [13].

Particular attention is given to the design of the SELECT tags, which have to meet several constraints, such as low cost, low power and small size. These requirements involve design choices oriented towards the realization of passive or semi-passive tags with low complexity architecture.

The SELECT system capability to monitor a generic object in real time, allows to have a continuous visibility in time of the working environment and therefore it lets to fully automate the treatment of desired objects, without requiring any human intervention.

In addition, the low cost, low power and small size requirements tags design, let to equip a large number of objects of different cost and size, in order to provide the possibility to choose the level of granularity of the objects to be monitored.

Although this system may have multiple applications in different economic sectors, the SELECT project focuses its attention on two main application scenarios [2]:

- Intelligent transport logistics solutions: The SELECT system has to guarantee full automation and optimization of the movements of semi-finished and/or finished goods. In particular, the system should be capable to detect, identify and track a generic product and to know its production state, acquiring the information stored in its tag.

- Solutions for supply chain management or SCM of intelligent production: The SELECT system has to allow the full automation of production lines. With the support of automatic guided vehicles, it would be possible to automatically 'call' and move a generic semi-finished product from the warehouse to a certain point of the production line.

1.2 SELECT System Technologies

The SELECT project merge multiple technologies in order to improve the benefits and mitigate the limits of a single solution.

In particular, given the characteristics required of the SELECT system, there are several technologies that can be used [2] [20]:

- Radio Frequency IDentification (RFID): The need of SELECT system to create a wireless network to monitor objects equipped with tags makes the RFID technology the most appealing solution to be applied in SELECT.
- Ultra Wide Band (UWB): The need of SELECT system to work in harsh environments and to obtain high accuracy in objects position estimation makes the UWB technology the best choice for the SELECT system.
- Real Time Location System (RTLS): The need of SELECT system to track objects in real time involves that it has to be a RTLS.

1.2.1 Radio Frequency IDentification

Radio Frequency IDentification (RFID) is the technology of acquisition and automatic identification operating at radio frequency.

The generic structure of a RFID system is shown in figure 1.5, which is composed by [3]:

- RFID tag: Device which contains information data and it usually lets to identify objects, people and animals.
- RFID reader: Device which interrogates tags in order to obtain data stored on them.
- RFID radio interface: Radio frequency transmission channel that allows tag/reader communication.
- RFID system infrastructure management: Infrastructure which processes information circulating in the RFID system.

Furthermore, the RFID structure is characterized by the links:

- reader-central unit link: This type of link is part of the RFID system infrastructure management and serves to allow proper configuration of the infrastructure itself and the circulation of information collected and/or processed. This type of link could be both wired and wireless.
- reader-reader link: This type of link has the same functions as the previous one and could be both wired and wireless.

- reader-tag link: This link serves to enable the RFID system infrastructure management to acquire the information stored on the RFID tags or change the information in them. This link is wireless at radio frequency.

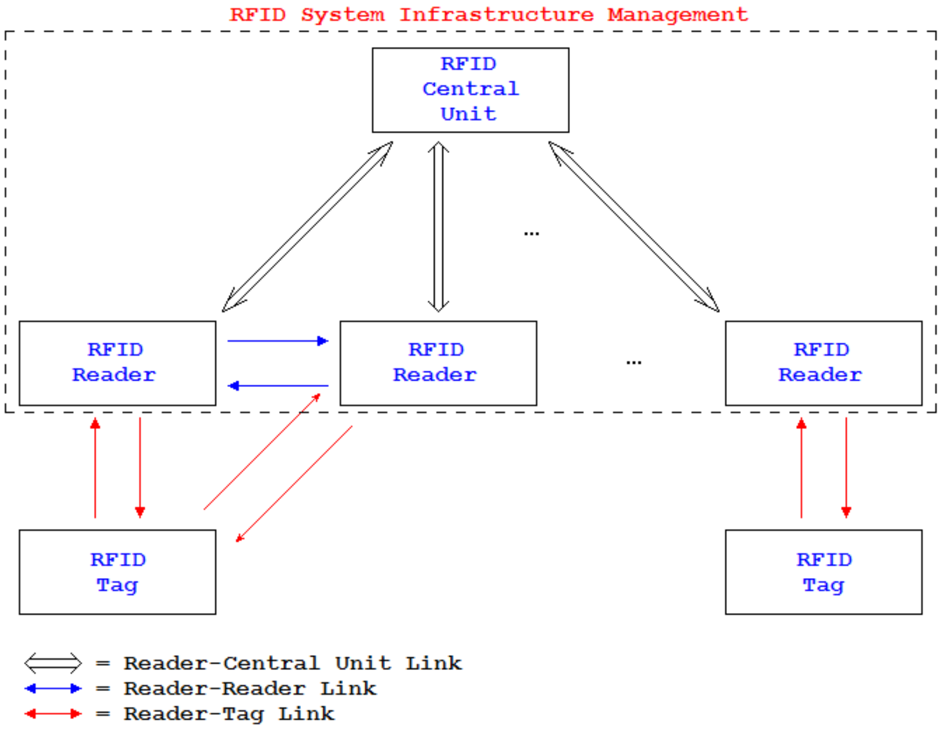


Figure 1.5: General structure of a RFID system

The operation modes of a generic RFID system are:

- RFID system configuration: The RFID central unit and the RFID readers exchange information to guarantee the proper functioning of the overall system.

- RFID tags interrogation: The RFID readers interrogate the RFID tags which respond by sending back the information stored on them. Once received from the RFID readers, this information becomes part of the RFID system infrastructure management, where it can be processed and/or stored.

1.2.1.1 RFID Tag

RFID tag is a device of a RFID system, operating at radio frequency and capable of storing its ID and communicate it to the external environment.

A generic RFID tag is composed by [3]:

- Coupling element: Responsible of the reception and transmission of the radio frequency signal.

- Receiver/demodulator: Responsible of the reception and demodulation of the radio frequency signal.

- Transmitter/modulator: Responsible of the transmission and modulation of the radio frequency signal.

- Microcontroller and memory: Responsible of the device control and of information storage.

- Power supply section: Responsible of the device powering and it is composed by a power supply and/or a battery.

The RFID tags can be classified as:

- Active RFID tags: RFID tags equipped with a battery that powers all its parts. Their operating state is always active, and they can therefore transmit a signal at any time.

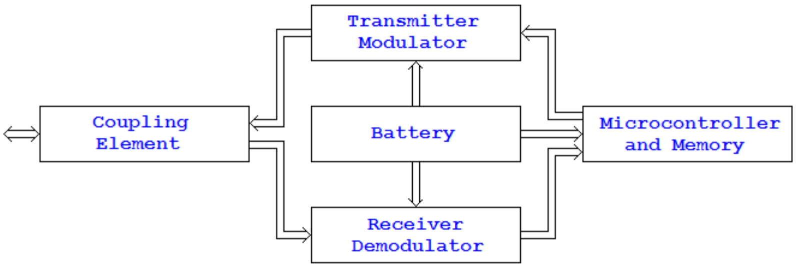


Figure 1.6: General structure of an active RFID tag

- Semi-active RFID tags: RFID tags equipped with a battery that powers all its parts. Their operating state is normally sleeping and can only be activated as a result of an interrogation by a RFID reader.

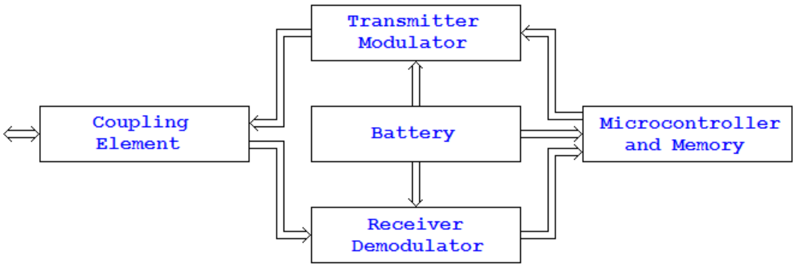


Figure 1.7: General structure of a semi-active RFID tag

- Semi-passive RFID tags: RFID tags equipped with a battery that supplies only the microcontroller and any auxiliary parts, but not the transmitter section. They are able to transmit a signal only as a passive response to an interrogation signal, from which tags acquire the energy necessary to properly supply its own transmitter.

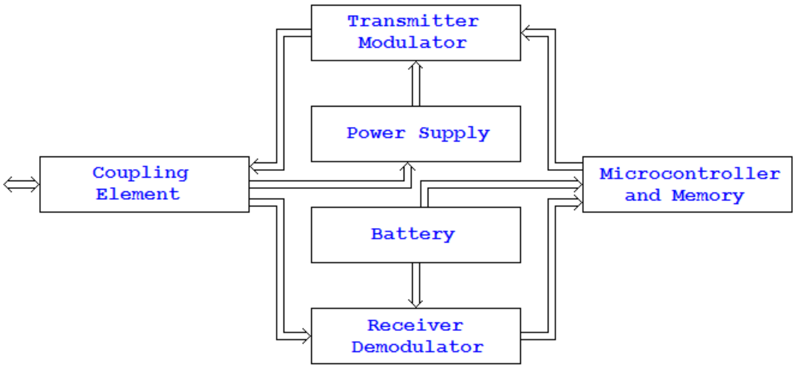


Figure 1.8: General structure of a semi-passive RFID tag

- Passive RFID tags: RFID tags completely devoid of a battery. They are able to transmit a signal only after the RFID reader, as happens for semi-passive tags, and energy harvested from the received signal is used to supply all the tags parts.

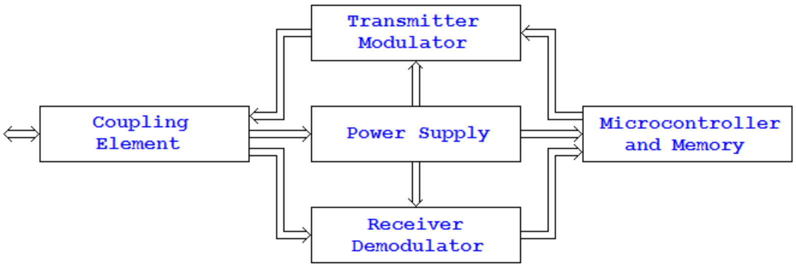


Figure 1.9: General structure of a passive RFID tag

In table 1.1 the performance of the different RFID tags previously outlined is compared.

Type of RFID Tag	Active	Semi-Active	Semi-Passive	Passive
Life Time	Low	Medium	Medium	High
Size	High	High	Medium	Low
Cost	High	High	Low	Low
Energy Consumption	High	Medium	Medium	Low
Coverage	High	High	Low	Low

Table 1.1: Performance of the types of RFID tags

Regardless of its typology, a generic RFID tag can have two different ways of managing its own memory [12]:

- RFID tags with read-only memory: RFID tags that allow only reading their own memory, where it is no possible to update the information contained therein.
- RFID tags with read-write memory: RFID tags that allow both to read and to update/overwrite their own memory.

1.2.1.2 RFID Reader

RFID reader is a device of a RFID system, operating at radio frequency and capable to interrogate a RFID tag and to receive the information sent by the latter after the interrogation.

A generic RFID reader is composed by [3]:

- Coupling element: Responsible of the reception and transmission of the radio frequency signal.
- Transmitter/modulator: Responsible of the transmission and modulation of the radio frequency signal.
- Receiver/demodulator: Responsible of the reception and demodulation of the radio frequency signal.
- Circulator: Responsible of decoupling the transmission/modulation part from the reception/demodulation part.
- Control unit: Responsible of the control of the device.

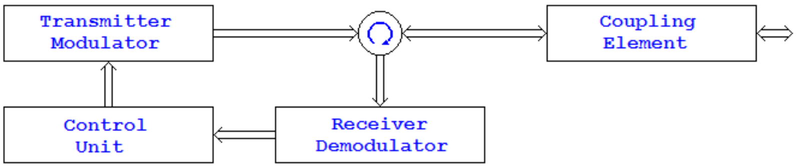


Figure 1.10: General structure of a RFID reader

1.2.1.3 RFID Radio Interface

RFID radio interface represents the radio frequency transmission channel of a generic RFID system where wireless communication between the set of RFID tags and RFID readers takes place.

The presence of the RFID radio interface in a generic RFID system imposes the dependence of the communication operating principle from the operating frequency that characterizes it.

For what the operating frequency concerns RFID systems can be classified as [3]:

- RFID-LF systems: Working at low frequencies (LF).
- RFID-HF systems: Working at high frequencies (HF).
- RFID-UHF systems: Working at ultra high frequencies (UHF).
- RFID-μW systems: Working at microwaves (μW).

In table 1.2 the performance of different types of RFID systems previously outlined is compared.

Type of RFID System	LF	HF	UHF	μW
Transmission Rate	Low	Medium	High	High
Size of RFID Tag	High	High	Low	Low
Energy Consumption	Low	Low	Medium	Medium
Penetration Capacity of Materials	Low	Low	High	High

Table 1.2: Performance of the types of RFID systems

Depending on the operating frequency class of a generic RFID system, the communication operating principle can be [12]:

- Near-field communication: This type of communication is characteristic of the first generation RFID systems, i.e. RFID systems with an operating frequency class equal to LF or HF, for which the limit of the near-far field is comparable to or greater than the maximum operating distance of the system itself. This implies that this type of RFID systems always works in the near-field region, where the communication takes place through inductive coupling of the magnetic field. In this case, the coupling elements of the various devices consist of loops with a suitable number of windings.

- Far-field communication: This type of communication is characteristic of RFID systems of second generation, i.e. RFID systems with an operating frequency class equal to UHF or μW, for which the limit of the near-far field is much lower than the maximum operating distance of the system itself. This implies that this type of RFID systems always works in the far-field region, where the communication takes place through electromagnetic propagation. In this case, the coupling elements of the various devices consist of radio frequency antennas.

1.2.2 Ultra Wide Band

Ultra Wide Band (UWB) is the technology of wireless transmission that allows to generate signals with ultra wide bandwidth. In particular, a generic signal can be defined as UWB if its bandwidth BW is greater than $500MHz$ or if its fractional bandwidth BF is more than 20%:

$$BW = f_H - f_L > 500MHz \qquad or \qquad BF = \frac{f_H - f_L}{\frac{f_H + f_L}{2}} > 20\%$$

where f_H and f_L represent respectively the maximum and the minimum frequency of the signal.

For the generation of a generic UWB signal it is possible to apply two different approaches of radio communication [4]:

- Impulse Radio UWB or IR-UWB.
- Multicarrier UWB or MC-UWB.

The main advantages of UWB technology are:

- Operation even in the absence of direct visibility and/or direct contact.
- Possibility of encrypted communication and multiple access.
- High transfer rate.
- Low power consumption.
- High immunity to noise and interference.
- High channel capacity.
- Excellent ability to penetrate materials.
- High temporal resolution.

Instead, the main disadvantages of UWB technology are:

- Short-range systems.
- Distortion of the signal.
- Synchronization issues.

1.2.2.1 Impulse Radio UWB

The Impulse Radio UWB (IR-UWB) approach is based on the transmission of pulses with very short time duration in order to create a signal with a very wide bandwidth.

Pulses generally used in the IR-UWB approach belong to the Gaussian family and they have the peculiarity of presenting a temporal support extremely reduced.

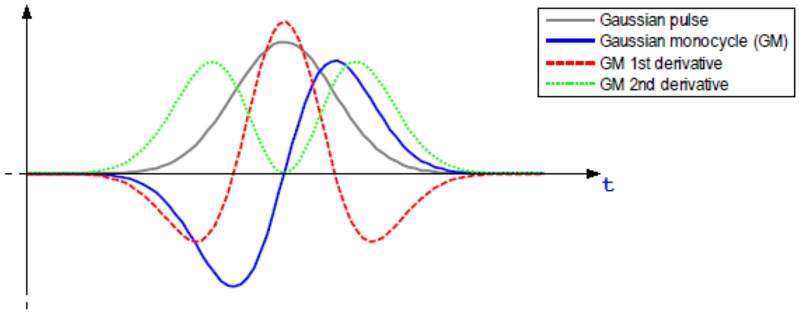


Figure 1.11: Example of Gaussian pulses

The IR-UWB approach allows the modulation of a generic information symbols sequence with the association of a large number of pulses to each symbol of that sequence.

Therefore, the Pulse Repetition Period (PRP) is defined as the time repetition interval of each pulse associated to a generic symbol of the considered information sequence and is given by:

$$T_p \triangleq \frac{T_s}{N_s}$$

where T_s represents the temporal duration of each symbol of the considered information sequence and N_s represents the number of pulses associated to each symbol.

In order to avoid interframe interference among the transmitted pulses, it is necessary that the temporal duration T_i of the received pulse is strictly less than the PRP, i.e.:

$$T_i < T_p$$

Given the temporal duration T_s of each symbol of the considered information sequence, the constraint above imposes an upper bound to the number N_s of pulses associated to each of these symbols:

$$N_s < \frac{T_s}{T_i}$$

The modulation techniques used in the IR-UWB approach are mainly two:

- Pulse Amplitude Modulation (PAM)

- Pulse Position Modulation (PPM)

Pulse Amplitude Modulation The Pulse Amplitude Modulation (PAM) is a modulation technique used in the IR-UWB approach where the pulses are modulated in amplitude according to the generic information sequence $\{b_n\}$ of N_d symbols to be transmitted (see figure 1.12) [14].

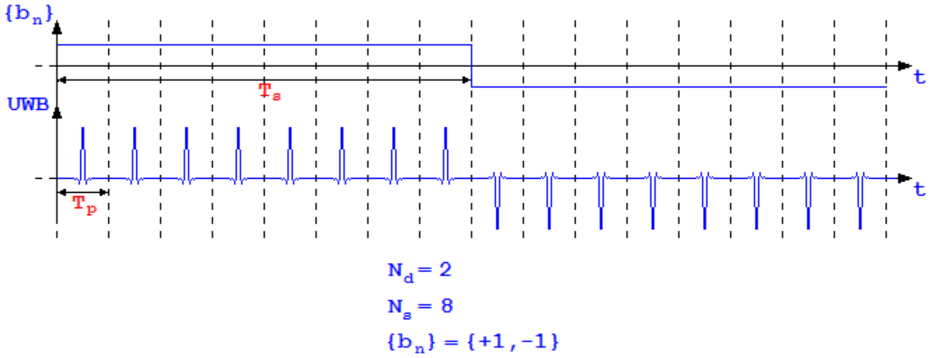


Figure 1.12: Example of PAM for IR-UWB

When the PAM is applied to a system that provides the multiple access and/or encryption of the transmitted signal, it can be combined with the Direct Sequence Spread Spectrum technique (DSSS) [14].

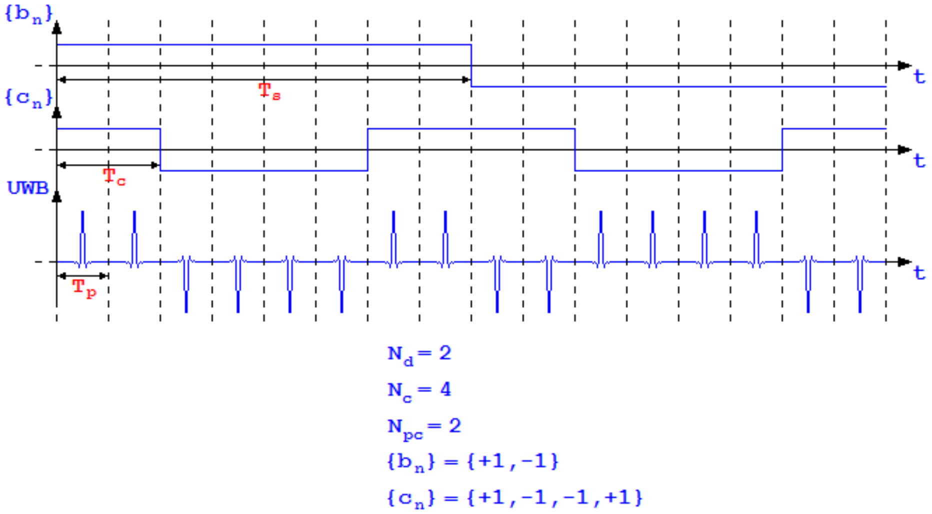


Figure 1.13: Example of PAM-DSSS for IR-UWB

In this case, for each generic information sequence $\{b_n\}$ of N_d symbols to be transmitted is associated a spreading code $\{c_n\}$, which has N_c chips of duration time T_c per chip.

In particular, each chip c_n of the spreading code $\{c_n\}$ modulates the amplitude of N_{pc} pulses associated to a given symbol b_n of the information sequence $\{b_n\}$ (see figure 1.13).

Pulse Position Modulation The Pulse Position Modulation (PPM) is a modulation technique used in the IR-UWB approach where the pulses are modulated in position in the PRP according to the generic information sequence $\{b_n\}$ of N_d symbols to be transmitted (see figure 1.14) [14].

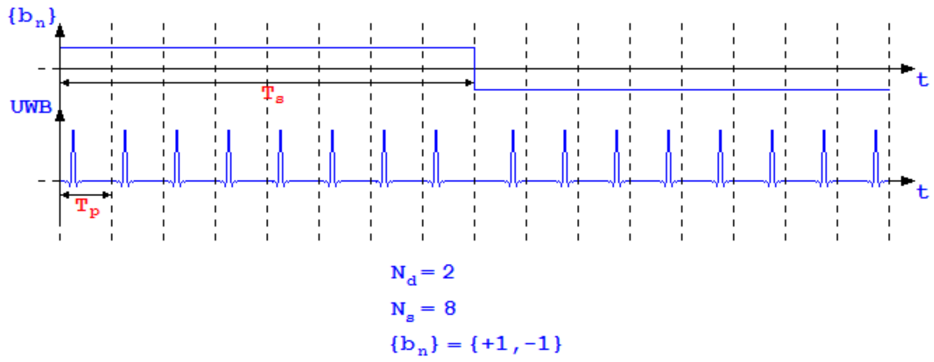


Figure 1.14: Example of PPM for IR-UWB

When the PPM is applied to a system that provides the multiple access and/or encryption of the transmitted signal, it can be combined with the Time Hopping Spread Spectrum technique (THSS) [14].

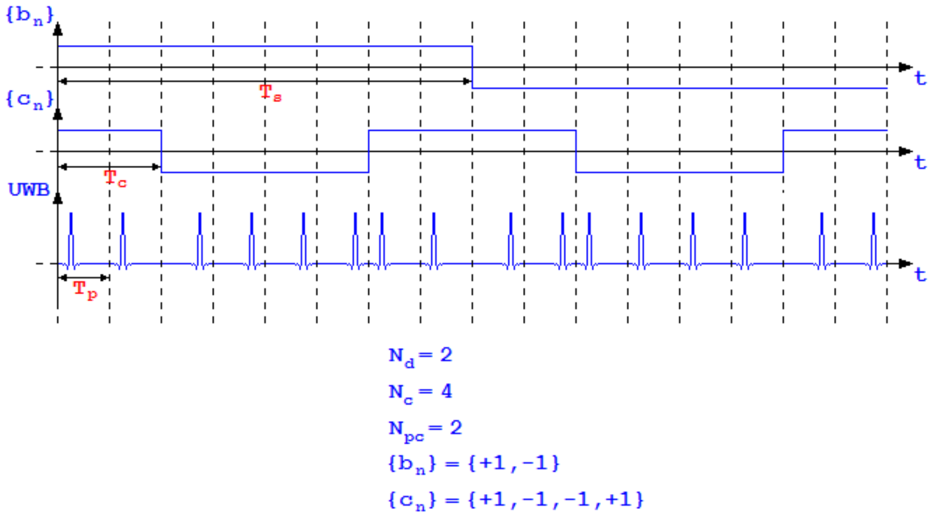


Figure 1.15: Example of PPM-THSS for IR-UWB

In this case, each PRP is divided into a given number of time slot with a duration strictly greater than that of the single pulse, and for each generic information sequence $\{b_n\}$ of N_d symbols to be transmitted is associated a spreading code $\{c_n\}$, which presents N_c chips of duration time T_c per chip.

In particular, each symbol c_n of the spreading code $\{c_n\}$ modulates the position in the PRP of N_{pc} pulses associated with a given symbol b_n of the information sequence $\{b_n\}$ (see figure 1.15).

1.2.2.2 Multicarrier UWB

The MultiCarrier UWB (MC-UWB) approach is based on the transmission of a generic signal on a large number of narrow band subchannels in order to create a signal with a very wide bandwidth.

The MC-UWB approach allows the modulation of a generic information sequence of symbols by means of the subdivision of such symbols on a large number of sinusoidal carriers.

The modulation technique used in the MC-UWB approach is mainly the Orthogonal Frequency Division Multiplexing (OFDM).

Orthogonal Frequency Division Multiplexing The Orthogonal Frequency Division Multiplexing (OFDM) is a modulation technique used in the MC-UWB approach according where the information sequence of symbols modulate various sinusoidal carriers [14].

In particular, the transmission bandwidth of the modulated signal is divided equally in N_f sub-bands sufficiently large, whose center-band frequencies form the set $\{f_1, ..., f_{N_f}\}$ of available sinusoidal carriers.

A generator of sinusoidal carrier generates a single sinusoidal carrier whose frequency value, which belongs to the set of available sinusoidal carriers, is changed at regular intervals, scanning the bandwidth in an orderly manner (see figure 1.16).

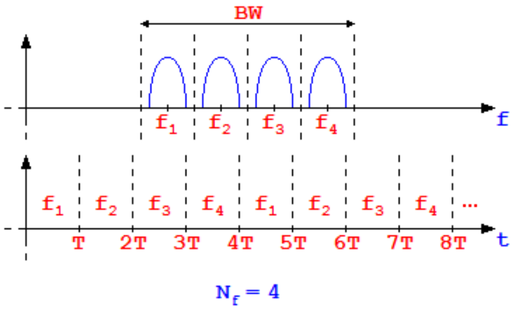


Figure 1.16: Example of OFDM for MC-UWB

When the OFDM is applied to a system that provides the multiple access and/or encryption of the transmitted signal, it can be combined with the Frequency Hopping Spread Spectrum technique (FHSS) [14].

In this case, for each generic information sequence of symbols to be transmitted is associated a spreading code, which is applied to the generator of sinusoidal carriers of the system.

In particular, each symbol of the spreading code fixes the frequency of the sinusoidal carriers that are modulated by the considered information sequence (see figure 1.17).

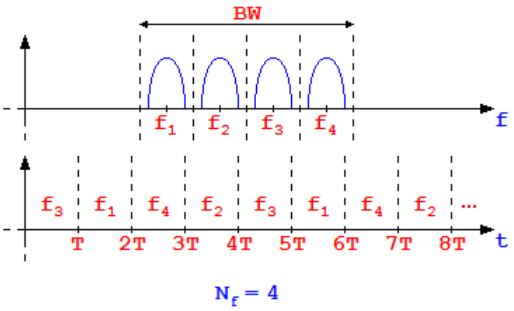


Figure 1.17: Example of OFDM-FHSS for MC-UWB

1.2.2.3 UWB Regulations

Given the high bandwidth occupied by UWB signals and the scarcity of free frequencies, the allocation of bandwidth wide enough to UWB technology is impossible for the frequency spectrum of all countries of the world.

Thus it is necessary to allocate bands sufficiently large for the UWB technology on frequencies already in use. This is possible if and only if the UWB technology is compatible with the primary services already allocated in these bands. Thanks to the high noise interference immunity of the UWB technology, the way to make possible that compatibility is to limit the permitted power spectral density of the UWB signals. In this way, the primary services already allocated in the interest band is not affected significantly from the presence of that signals.

Therefore, world countries have enacted specific regulations for the UWB technology, establishing the frequency bands that can be exploited. These countries set masks that specify the values of maximum transmissible power density.

Considering, for simplicity, the existing regulations in the U.S. and in Europe, the masks for indoor environments are shown in figure 1.18, where the FCC mask is relating to U.S. and the EU mask is relating to Europe [5].

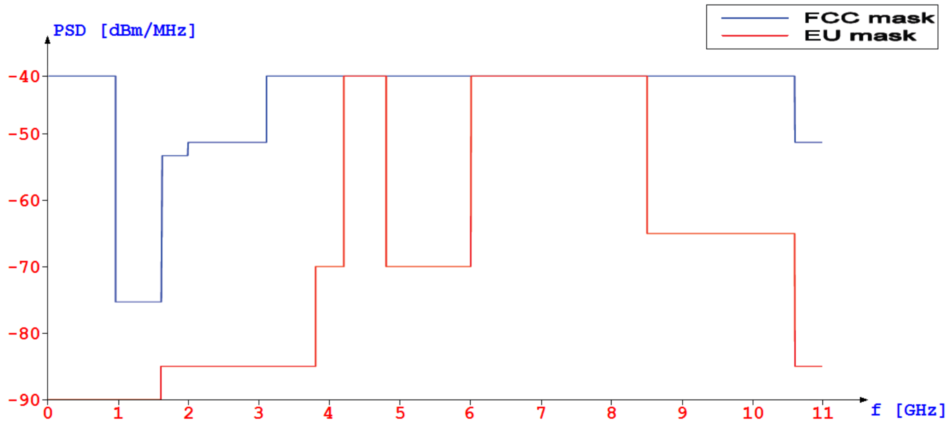


Figure 1.18: UWB regulations masks for U.S. and Europe

From masks in figure 1.18 is deduced that, for the purposes of compatibility between regulations, the frequency bands of most interest are two [2]:

- Band ranging from $3.1GHz$ to $4.8GHz$: In the FCC mask, this band has the higher limit of maximum transmissible power density, i.e. $-41.3\frac{dBm}{MHz}$, while in the EU mask, this band presents the limit of transmissible power density equal to $-70\frac{dBm}{MHz}$, that may become equal to $-41.3\frac{dBm}{MHz}$ for UWB systems that implement specific techniques of interference reduction. These techniques are the low duty cycle or (LDC), which consists in transmitting signals very rarely and with very reduced duration, and the detect and avoid or (DAA), which consists in monitor the transmission channel in order to implement an efficient transmit power control or (TPC) to protect the primary systems already allocated to that band.

- Band ranging from $6GHz$ to $8.5GHz$: Both in the FCC mask and in the EU mask, this band has the higher limit of maximum transmissible power density, i.e. $-41.3\frac{dBm}{MHz}$.

1.2.3 Real Time Location System

Real Time Location System or RTLS is a generic local positioning system for automatically localization in real time.

The general structure of an RTLS is composed of two blocks [6]:

- RTLS Nodes: Generic objects to be localized in real time.

- RTLS Network: System with function to determine the position of the RTLS nodes.

1.2.3.1 Localization in the RTLS

The localization in a RTLS is the process of the spatial position estimation of a generic RTLS node and it is the fundamental phase of the RTLS operation.

A generic localization process involves three phases [16]:

- Acquisition: Acquisition of the necessary information to locate the considered object.
- Positioning: Coarse estimation of the position of the considered object.
- Refining: Refining of the position estimation of the considered object.

There are three main modes of localization:

- Relative localization: The position of an object is estimated by comparison to known positions.
- Absolute localization: The position of an object is estimated with respect to the terrestrial globe.
- Symbolic localization:The position of an object is estimated on the basis of indirect information, applying criteria of proximity or association.

In addition, there are three main types of localization:

- Range Based.
- Angle Based.
- Free Range.

Range Based Localization The range based localization is based on the estimation of RTLS nodes distances in order to determine their positions.

Therefore, the acquisition phase of the range based localization process corresponds to the distances estimation. The main techniques of distances estimation are [16] [17]:

- Time of arrival (TOA): Technique based on the estimation of the signal propagation time between a RTLS network point and the desired RTLS node. This temporal measurement can be converted into a distance estimation knowing the propagation speed of the transmitted signal.

- Time difference of arrival (TDOA): Technique based on the estimation of the signal propagation time difference between two points in the RTLS network and the desired RTLS node. This temporal measurement can be converted into a distance estimation knowing the propagation speed of the transmitted signal.

- Received Signal Strength (RSS): Technique based on the estimation of the signal power received from the RTLS network and transmitted from the desired RTLS node. This power measurement can be converted into a distance estimation knowing the path-loss of the transmission channel.

For the localization of a particular RTLS node, the main algorithm for the position estimation used in the range based localization is the trilateration, which allows to estimate the position of a generic point knowing the distances between the latter and other three different points (see figure 1.19).

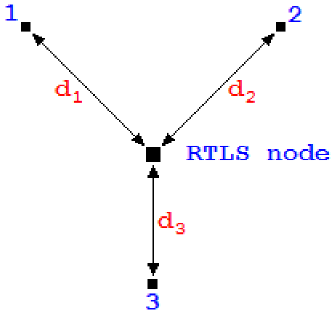


Figure 1.19: Example of trilateration

Angle Based Localization The angle based localization is based on the estimation of angles of RTLS nodes in order to determine their position.

Therefore, the acquisition phase of the angle based localization process correspond to the angles estimation. The main technique of angles estimation is the angle of arrival (AOA). This technique is based on the estimation of the direction of arrival of a signal transmitted from the desired RTLS node [16].

For the localization of a particular RTLS node, the main algorithm for the position estimation used in the angle based localization is the triangulation, which allows to estimate the position of a generic point knowing the angles between the latter and three other different points (see figure 1.20).

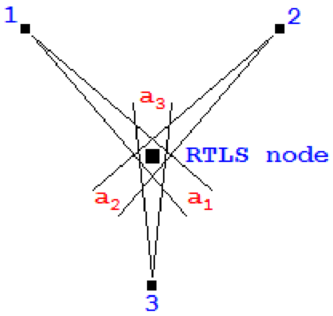


Figure 1.20: Example of triangulation

Range Free Localization The range free localization is based on the acquisition of information about RTLS nodes not directly related to distances or angles.

The main techniques of acquisition of the positioning information are [16]:

- Cell identification or Cell-ID: Technique based on the determination of the RTLS network cell in which the desired RTLS node is located.
- Proximity: Technique based on the determination of the presence of the desired RTLS node in the proximity of known points of the RTLS network.
- Line of sight or LOS: Technique based on the determination of the visibility of the desired RTLS node by certain points in the RTLS network.

For the localization of a particular RTLS node, the algorithms for the position estimation used in the range free localization are multiple and strictly dependent on the acquisition technique used.

1.2.3.2 Technologies for RTLS

The types of technologies used in RTLS are mainly three [6]:

- Radio Frequency Technologies.
- Optical Technologies.
- Acoustic Technologies.

Radio Frequency Technologies for RTLS: The main radio frequency technologies used in RTLS are:

- Bluetooth.
- Cellular Networks.
- Global Positioning System (GPS).
- Radio Frequency IDentification (RFID).
- Ultra Wide Band or UWB.
- Wi-Fi.
- ZigBee.

Optical Technologies for RTLS: The main optical technologies used in RTLS are:

- Infrared.
- Optical Locating.

Acoustic Technologies for RTLS: The main acoustic technologies used in RTLS are:

- Surface Acoustic Wave (SAW).
- UltraSound IDentification (USID).

Chapter 2

SELECT System

2.1 SELECT System Operation

The SELECT system has a low-cost system architecture consisting of a network of fixed readers able to query, via a secure wireless communications and with multiple access, one or more tags attached to objects (see figure 2.1).

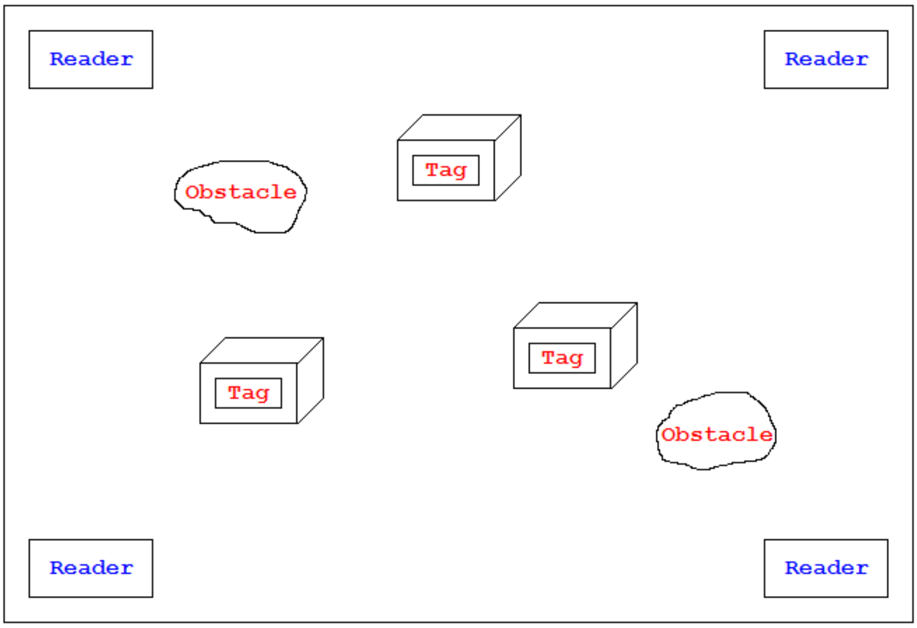


Figure 2.1: Example of SELECT application scenario

In order to meet the low cost, low power and small size requirements, the SELECT system involves:

- Semi-passive tags that exploits backscatter modulation to transmit data, i.e. directly modulating the received signal by varying the antenna load properties.
- Wireless IR-UWB transmission technology with PAM modulation.

In addiction, to meet the secure communication and multiple access requirements, the SELECT system adopts:

- Technique of DSSS or Direct Sequence Spread Spectrum.

Since the SELECT system has to operate in real time, it has to work in continuous mode, repeating seamlessly a given interrogation cycle to which the various tags respond [7] [8].

The interrogation signal consists of a packet $\{r_n\}$ of N_r data symbols composed of all logical 1, as readers do not transmit any information to the tags.

The UWB interrogation signal transmitted by the reader is modulated according to the specific reader code $\{d_n\}$ of N_c chips, where each chip is repeated for N_{pc} time to improve the SNR and thus the reader-tag link budget (see figure 2.2).

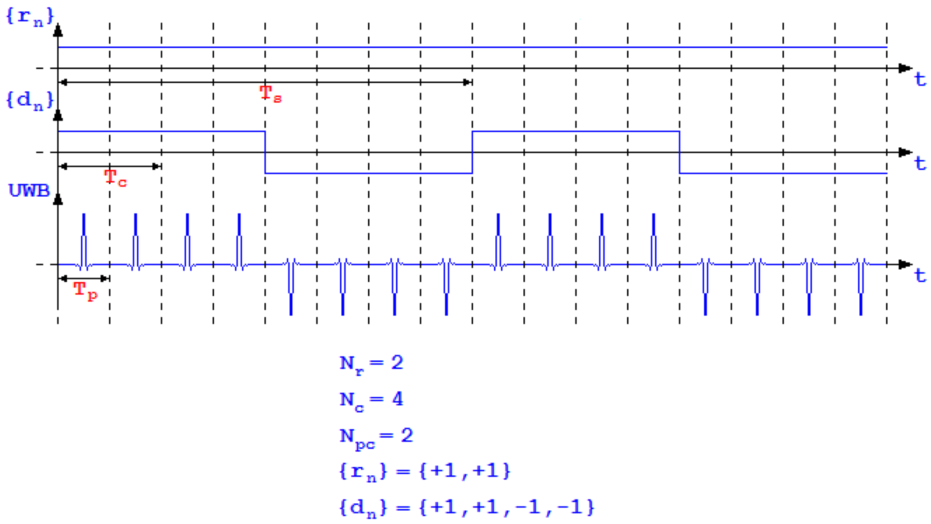


Figure 2.2: Example of structure of an interrogation signal

Then the UWB interrogation signal is received by each tag, which modulates it according to the composite sequence given by the tag code $\{c_n\}$ and the information symbol $\{b_n\}$ to be transmitted [18] [19] (see figure 2.4).

The symbols packet $\{b_n\}$ is composed by two parts (see figure 2.3):

- Preamble: Sequence of preliminary known data symbols whose purpose is to allow synchronization and tag distance estimation. Therefore, this sequence consist of all logical 1, i.e. $b_n \in \{+1\}$.
- Payload: Data symbols sequence which contains the information data stored on the tag. It is thus composed of a specific sequence of $\{-1, +1\}$, i.e. $b_n \in \{-1, +1\}$.

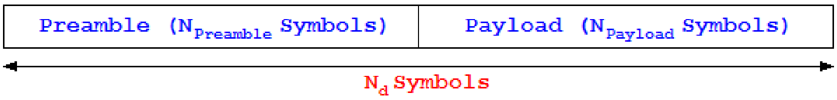

Preamble (N$_{Preamble}$ Symbols)
Payload (N$_{Payload}$ Symbols)
N$_d$ Symbols

Figure 2.3: Structure of a data packet

The UWB signal backscattered by each tag is composed of two components [19]:

1. Antenna mode scattering: This component stems from the capability of an antenna to radiate when excited at its parts, and thus depends on how the antenna is loaded. By proper variations of the load, it is possible to modulate this signal component.
2. Structural mode scattering: It is the backscattered component that only depends on the antenna given shape and material, and does not depend on how the antenna is loaded.

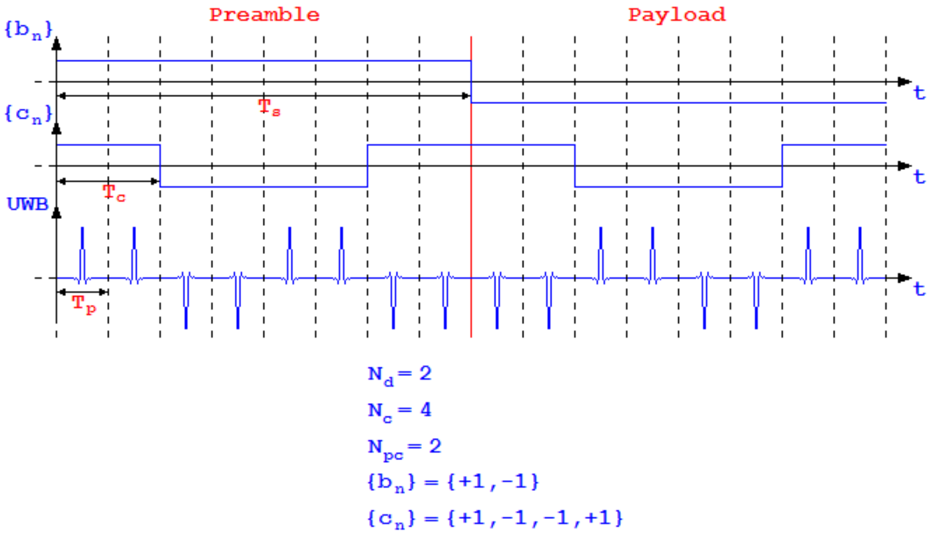


Figure 2.4: Example of structure of the response signal to an interrogation

Table 2.1 shows the parameters values relating to the reader-tag communication of the SELECT system [9].

Symbol	Description	Value
N_d	Number of tag data symbols per packet	256
N_r	Number of reader data symbols per packet	256
N_c	Number of chips per spreading code	1024
N_{pc}	Number of pulses per chip	8
T_s	Symbol period	$1.048ms$
T_c	Chip period	$1.024\mu s$
T_p	PRP	$128ns$

Table 2.1: Reader-tag communication parameters adopted in SELECT

Since the UWB technology requires low transmission power, in order to transmit a generic information packet it is necessary to apply techniques that allow to increase the SNR of the received UWB signal [7] [8].

The technique used in SELECT consists in the transmission of several pulses per chip.

In this manner, for each chip, more than one pulses is accumulated at the receiving section, with a consequent increase of the SNR. On the other side, the number of pulses per chip should be not too large, in order to prevent the system from suffering for the presence of clock drift (see figure 2.5).

Figure 2.6 shows the functional scheme of a general communication between a reader and tags in the SELECT system. The resulting signal from the accumulator of the reader it is used from the reader itself for:

- Tag Detection.

- Demodulation of received data symbols.

- Ranging distance estimation between readers and tags by TOA estimation.

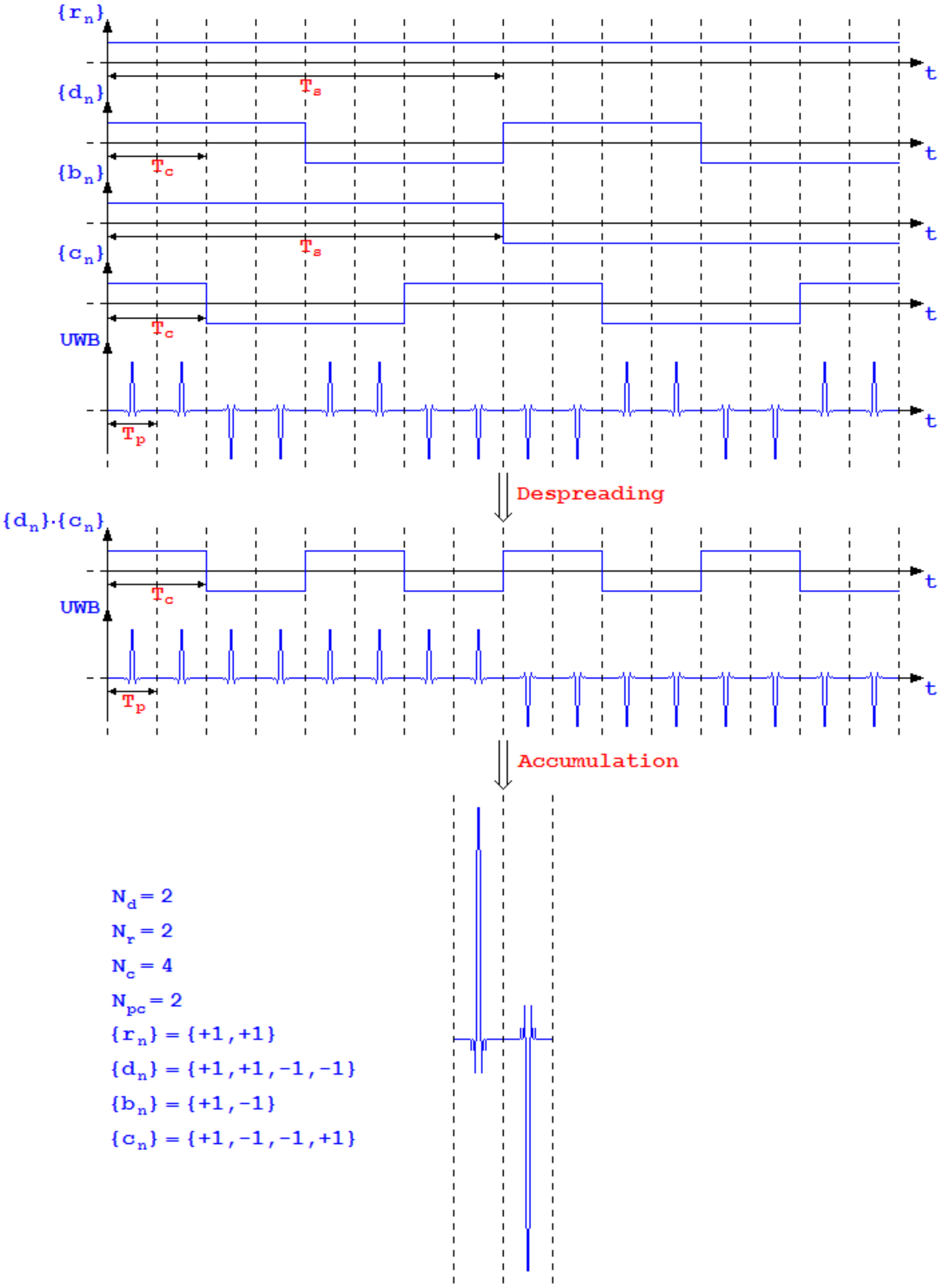


Figure 2.5: Example of the SNR increase technique of the received UWB signal

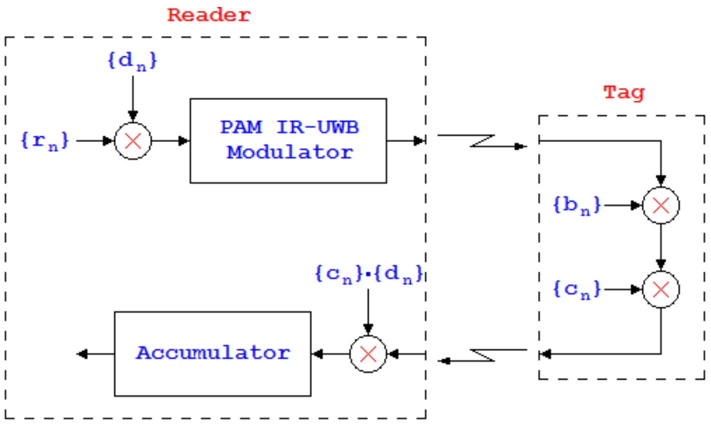


Figure 2.6: Functional scheme of the reader-tag communication

2.2 SELECT System Issues

The main issues that arise in SELECT are [7] [8]:

- Tags Activation.
- Devices Synchronization.
- Tags Interference and Clutter Contribution.

2.2.1 Tags Activation

Since tags employed in SELECT are semi-passive devices, they require to be activated in order to start communicating. In particular, before the activation phase, tags are in sleeping mode to preserve the battery consumption.

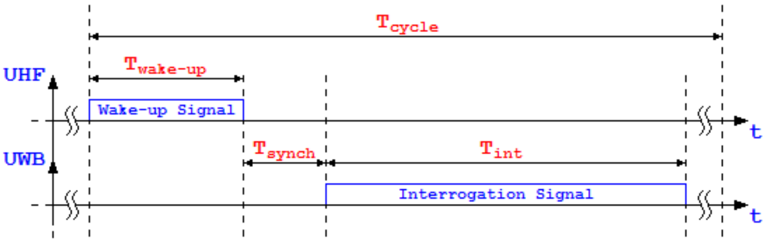


Figure 2.7: Structure of an interrogation cycle

Since UWB technology does not allow to transmit sufficiently high powers for the activation phase, the SELECT system implements a tags wake-up procedure in UHF. In particular, at the beginning of each interrogation cycle, the reader transmits a wake-up signal in continuous wave in order to provide sufficient energy to activate the tags (see figure 2.7) [9].

Table 2.2 reports the parameters constraints relative to an interrogation cycle of the SELECT system.

Symbol	Description	Constraints
$T_{wake-up}$	Time of wake-up	$T_{wake-up} < 10\mu s$
T_{synch}	Time of synchronization	$T_{synch} \ll T_{wake-up}$
T_{int}	Time of interrogation	$T_{int} = T_s \cdot N_r$
T_{cycle}	Time of interrogation cycle	$T_{cycle} > T_{wake-up} + T_{synch} + T_{int}$

Table 2.2: Interrogation cycle parameters adopted in SELECT

2.2.2 Devices Synchronization

Since the devices in SELECT do not share the same reference time, they can not be synchronized with the risk of having a not proper communication.

Thus, in the SELECT system the following two main synchronization process have to be assured:

- Reader-Reader Synchronization
- Reader-Tag Synchronization.

2.2.2.1 Reader-Reader Synchronization

For this type of synchronization, the SELECT system adopts an Ethernet link. Through this wired communication it is possible to implement a communication protocol that performs a tight synchronization between readers. A finer synchronization could be achieved performing correlation among reader codes (this study is still ongoing in SELECT).

2.2.2.2 Reader-Tag Synchronization

For this type of synchronization, the SELECT system adopts the wake-up signal as synchronization signal. This signal marks the beginning of the interrogation cycle and therefore the beginning of the communication (see figure 2.7).

However, despite this precaution, the reader-tag synchronization is not guaranteed with sufficient accuracy due to the reader-tag distance, which involves a certain signals propagation time, and due to the clock drift between the local oscillators of the devices [22]. Therefore, the wake-up signal is received with a certain delay from the various tags, leading out synchronization readers and tags.

In order to refine the reader-tag synchronization, and thus obviate the above issue, the SELECT system uses spreading codes with auto-correlation functions that present a single correlation peak in a range sufficiently wide in the neighborhood of the synchronism instant, allowing to obtain, in output from the despreader, a strong signal only in correspondence with perfect synchronism of the despreading code with the received UWB signal.

Since the wake-up signal shares the same time base of the spreading code of the reader, this code is in perfect synchronism with the received UWB signal, and therefore, in the despreading phase, it may be applied directly, without synchronism problems. The spreading code of the tag, instead, suffers the effects described above, so it has a different time base than the wake-up signal and thus presents problems of synchronization.

In addiction, when strong tag clock drift is present due to the adoption of low cost tag local oscillators, the initial offset can significantly increase after the transmission of several symbols and thus another finer procedure is required.

To accomplish this requirement, the despreading is performed considering the composite sequences given by the reader code (not shifted) and local shifted replicas of the tag code.

In this way, testing in parallel different tag code shifts, it is possible to evaluate and correct the entity of the tag clock drift by taking the maximum of the despreading correlations.

In addiction, in order to have a more easy and effective synchronization, the SELECT system allows to associate multiple pulses to each chip of the spreading code. This involves an enlargement of the correlation peak of the auto-correlation function of each spreading code used, in order to easier locate that correlation peak during the despreading phase [7] [8].

2.2.3 Tags Interference and Clutter Contribution

The UWB signal backscattered to the reader is composed of two components (see figure 2.8):

- Useful component: It is the received UWB signal component due to the useful tag backscatter modulation, i.e. the amount of the antenna mode scattering component of the intended tag.

- Interference component: It is the received UWB signal component due to the amount of the antenna mode scattering components of the other tags.

- Clutter component: It is the received UWB signal component due to the reflections of the environment, which comprises the reflections of the obstacles and the reflections of the tag antennas, i.e. the amount of the structural mode scattering components [25].

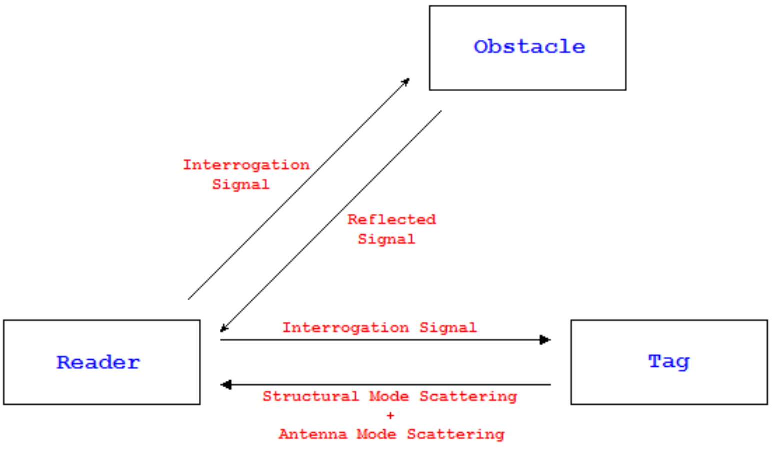


Figure 2.8: Example of transmission with backscatter modulation

2.2.3.1 Tags Interference

As several tags may access the transmission channel simultaneously, the interference created could be detrimental for the communication.

In order to minimize the interference, SELECT adopts a code division multiple access (CDMA) solution, that consists to assign orthogonal and quasi-orthogonal spreading codes, which have good correlation properties in both synchronous and asynchronous scenario.

In this way it is possible to reduce the multiple access interference (MAI), i.e. the interference between tags during simultaneous access the transmission channel [24].

The number of tags that can access the transmission channel simultaneously corresponds to the amount of available spreading codes.

For this reason, in SELECT it is worthwhile to adopt long sequences, which implies a huge number of available codes [7].

The chosen spreading sequences in SELECT are the Orthogonal Gold Codes [8].

2.2.3.2 Clutter Contribution

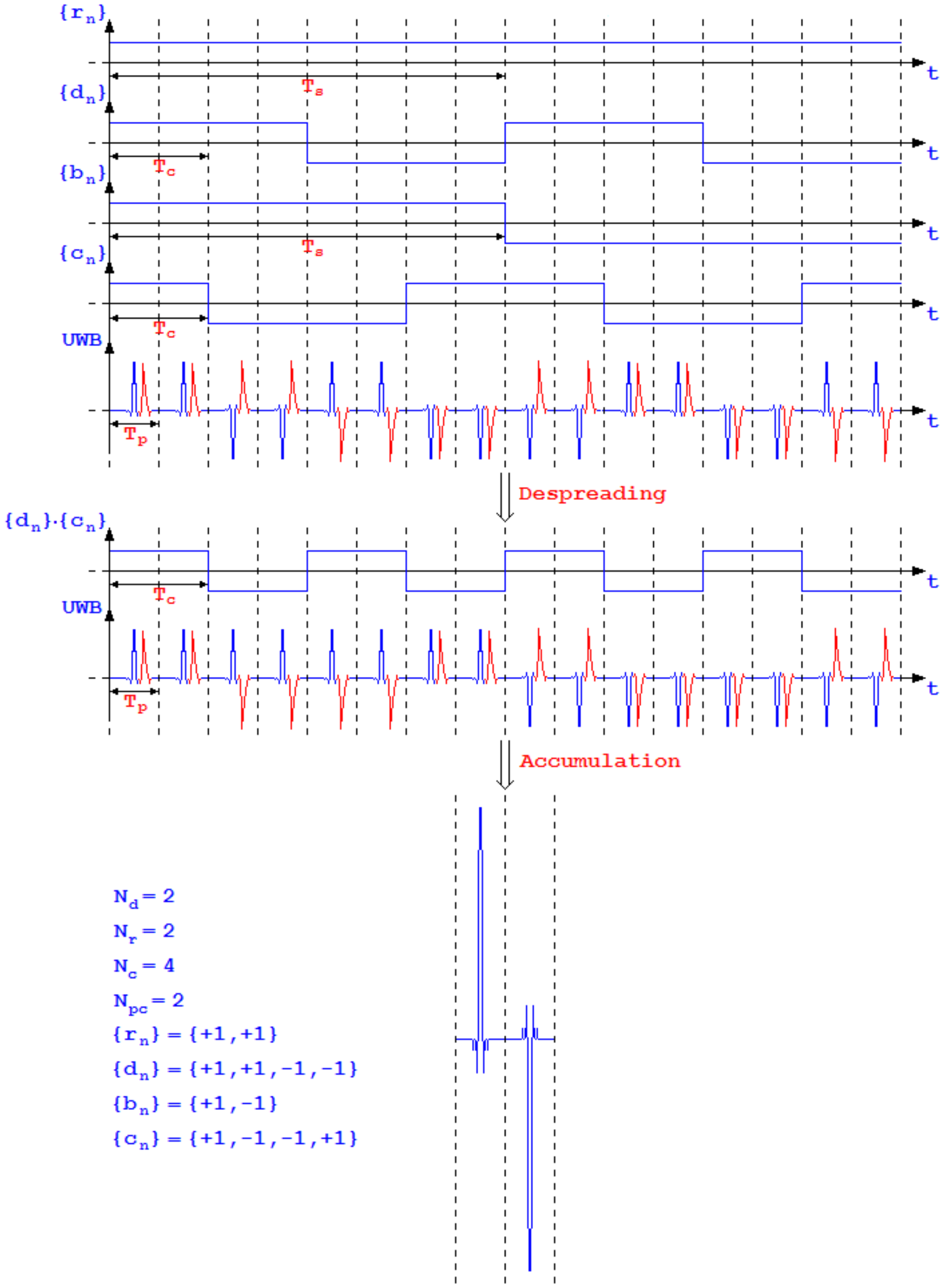


Figure 2.9: Example of clutter removal

In general, the clutter component is always dominant with respect to the antenna mode scattering component and thus it is necessary to strongly compensate its contribution, i.e. it is necessary to implement clutter removal techniques.

In the SELECT system, the clutter is removed using balanced tags spreading codes [19].

This technique relies on the fact that the antenna scattering mode component is modulated by tags and reader spreading codes, while the clutter component is modulated only by the reader spreading code.

Thus, applying balanced tags spreading codes and assuming the stationarity of the transmission channel on each data symbol during the accumulation phase, the clutter component is perfectly compensated (see figure 2.9) [7] [8].

2.3 SELECT System Architecture

The general structure of the SELECT system is composed by four parts (see figure 2.10) [9]:

- SELECT Central Unit.

- SELECT Reader.

- SELECT Relay.

- SELECT Tag.

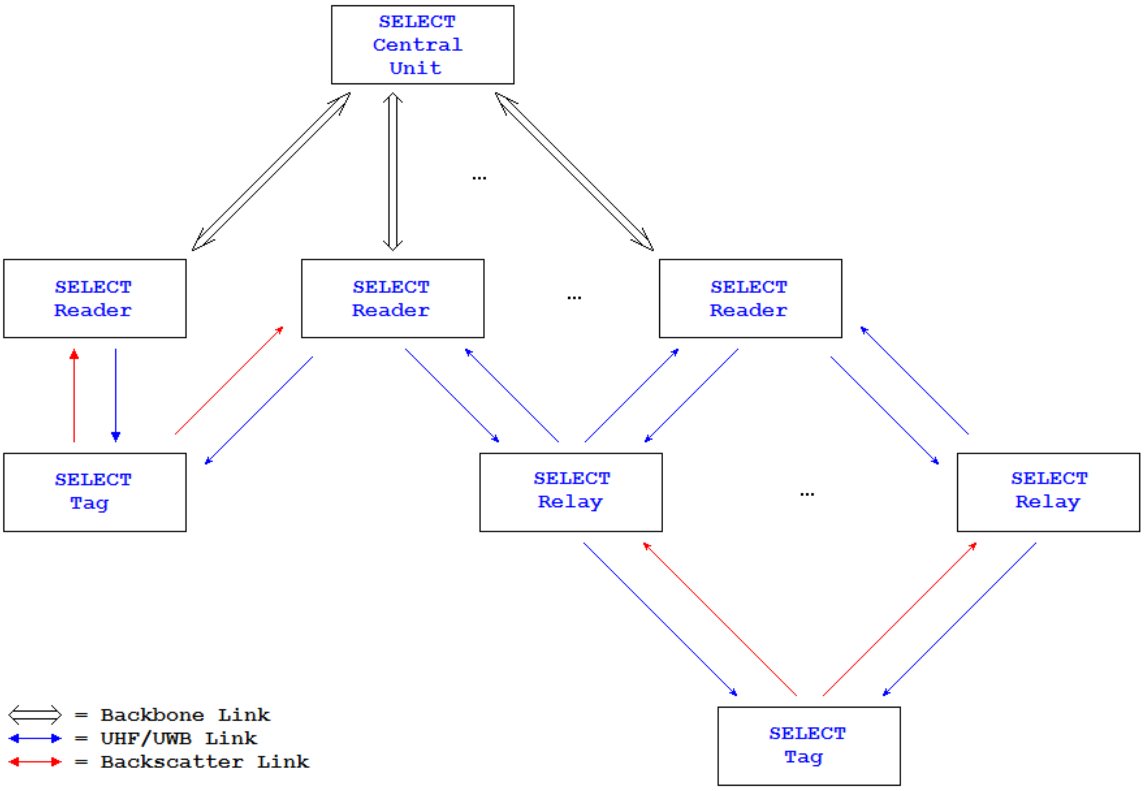


Figure 2.10: General structure of the SELECT system

2.3.1 SELECT Central Unit

The SELECT central unit is the control part of the SELECT system and can only communicate with the SELECT readers. It implements two main functions:

- Control and management of the SELECT system.

- Processing of data sent by the SELECT readers for localization and tracking of the SELECT tags.

This part of the SELECT system is composed of two main sections:

- Processing unit: Processing section of the SELECT central unit.

- Application software: Application programs that implements the SELECT central unit functionality.

2.3.2 SELECT Reader

The SELECT reader is the SELECT device whose main functions are:

- Interrogation of SELECT tags.
- Communication with the SELECT central unit.
- Communication with the SELECT relay.

The functional structure of a SELECT reader is shown in figure 2.11 and it is composed of two main sections:

- Transmitting section.
- Receiving section.

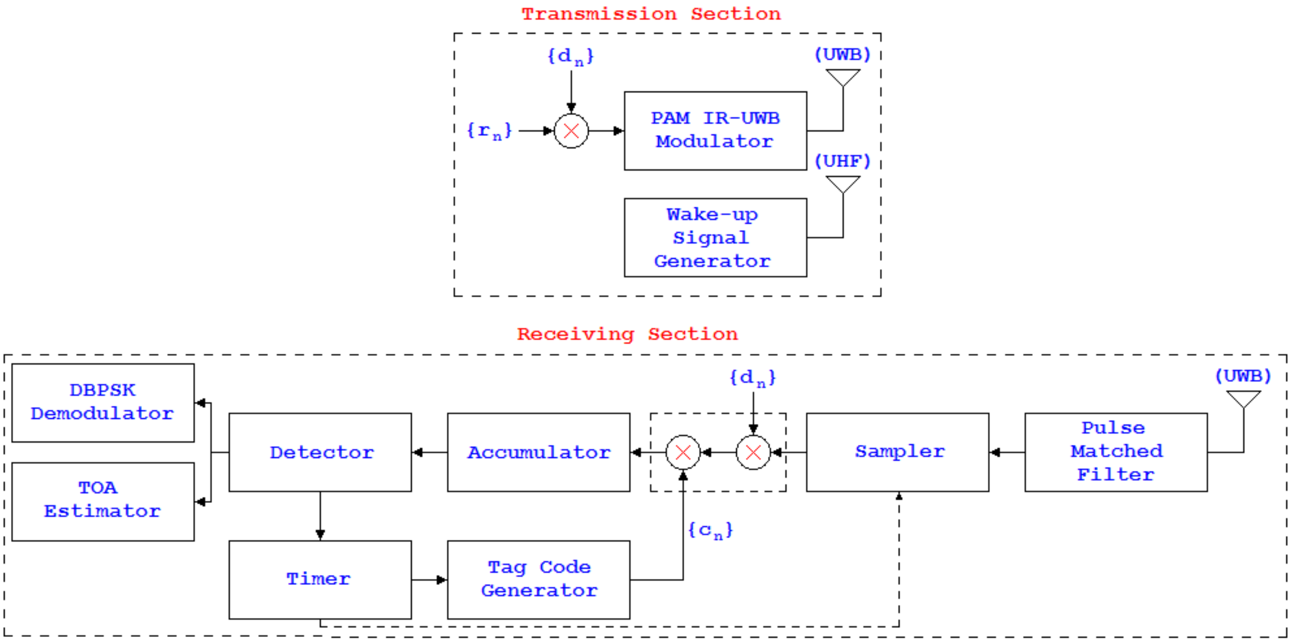


Figure 2.11: Functional structure of a SELECT reader

2.3.2.1 Transmitting Section of a SELECT Reader

The transmitting section of a SELECT reader has the function of generating and transmitting signals during an interrogation cycle depicted in figure 2.7.

The functional structure of that section it is composed by two blocks (see figure 2.11):

- UWB block: Part relating to the generation of the interrogation signal. It generates the data symbols packet to be transmitted, after the spreading operation and the PAM modulation, through a UWB antenna.

- UHF block: Part relating to the generation of the wake-up signal. It generates the synchronization signal, which is transmitted by a UHF antenna.

2.3.2.2 Receiving Section of a SELECT Reader

The receiving section of a SELECT reader has the function of receiving and processing the UWB backscattered signal.

The functional structure of that section is composed of (see figure 2.11):

- Pulse Matching Filter: Filtering block of the received UWB signal.

- Sampler: Sampling block of the output signal from the pulse matching filter controlled by a timer.

- Despreader: Despreading block of the signal coming out of the sampler, where the EX-OR operation with the related SELECT reader code and the useful SELECT tag code is applied.

- Accumulator: Accumulation block for each data symbol of the signal outgoing from the despreader.

- Detector: Detection block of the desired SELECT tag, where the signal outgoing from the accumulator is compared with a threshold.

- Timer: Timing block that provides the synchronization of the desired SELECT tag spreading code.

- Tag Code Generator: Generation block of the desired SELECT tag spreading code.

- Demodulator: Demodulation block of the signal outgoing from the detector, where it is possible to extract the information symbols transmitted by the desired SELECT tag.

- TOA Estimator: TOA estimation block of the received UWB signal.

The operation of the receiving section of a SELECT reader can be divided into two phases:

1. Synchronization phase of the useful SELECT tag spreading code: Operating phase where the timer is used to properly shift the local replicas of the intended useful tag code.
2. TOA estimation phase and data symbols demodulation: Operating phase where the despreading of the received UWB signal is performed by applying the desired SELECT tag spreading code generated on the time base determined by the previous phase. Therefore, at this stage it is possible to perform the TOA estimation and data symbols demodulation.

The implementative structure of the receiving section of a SELECT reader is shown in figure 2.12, where there are two differences from the functional structure of figure 2.11:

- Integrate and Dump: This block replaces the pulse matched filter, as the latter turns out to be complex to be implemented, especially in case of UWB signals, where the estimation of the transmission channel is extremely complex.
- Code Generator: This block brings together the generation of spreading codes relating to the SELECT reader itself and the desired SELECT tag, generating a unique despreading code in order to perform a single despreading operation.

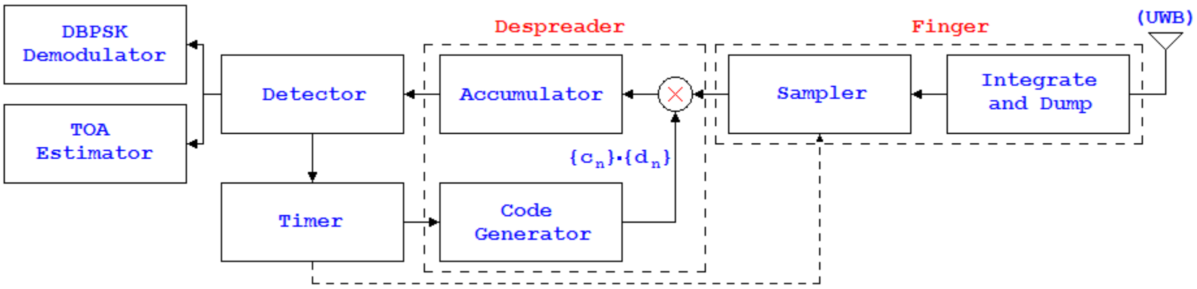


Figure 2.12: Implementative structure of the receiving section of a SELECT reader

2.3.3 SELECT Relay

The SELECT relay is the SELECT device whose function is to extend the geographical system coverage.

They cannot communicate with the SELECT central unit and they have reduced functionality compared to SELECT readers, as they are required to be less complex and expensive [23].

2.3.4 SELECT Tag

The SELECT tag is the SELECT device whose function is to communicate its stored information to the SELECT reader.

The functional structure of a SELECT tag is composed by two main parts [21] (see figure 2.13):

- UHF section.
- UWB section.

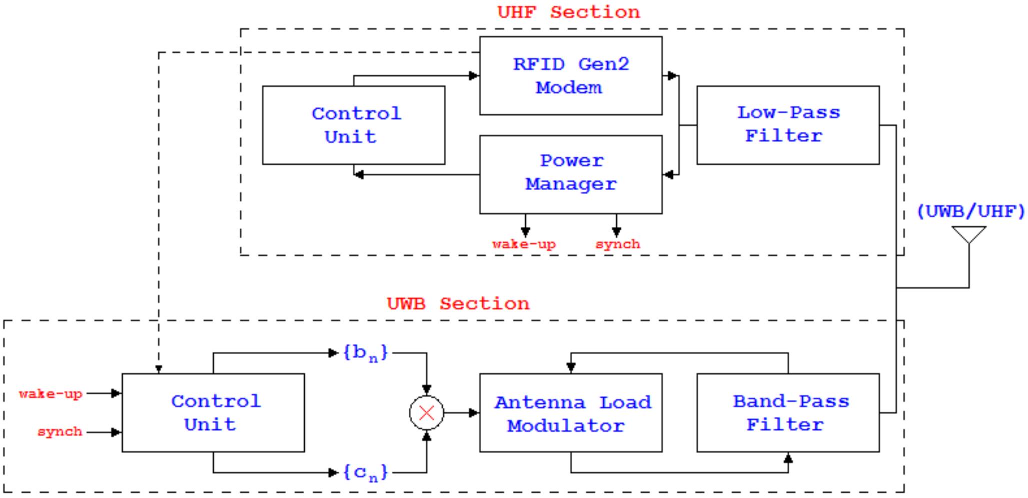


Figure 2.13: Functional structure of a SELECT tag

2.3.4.1 UHF Section of a SELECT Tag

The UHF section of a SELECT tag has two main functions (see figure 2.13):

- Detect the wake-up signal in order to activate the UWB section and provide it the synchronism signal.
- Implement the functionality of a passive UHF RFID tag according to the Gen2 standard.

This section is completely passive, because it is powered by the energy extracted from the received UHF signal.

2.3.4.2 UWB Section of a SELECT Tag

The UWB section of a SELECT tag has the function of modulating in backscatter the received interrogation signal by proper variations of the antenna load, in order to send data to the SELECT reader (see figure 2.13).

Antenna load variations actually depend on the SELECT tag code and to the information symbol to be transmitted. The modulator consists of a simple switch that, commutates at each chip time in order to connect the antenna respectively to a short circuit or open circuit, thus varying its radiation properties [18] [19].

This section is semi-passive, since part of the reception/transmission section is supplied by the received UWB signal, while the control part is supplied by a battery.

However, in order to minimize the energy consumption and thus maximize the life time of the entire device [13], the SELECT tag is normally in sleeping mode and is triggered by the UHF section only in correspondence of an interrogation signal.

Chapter 3

Baseband SELECT Reader Receiver Simulator

3.1 SELECT Reader Receiver Implementation to be Simulation

The implementation of a SELECT reader receiver is shown in figure 3.1, which is composed by two modules [9]:

- Receiving UWB Module: Receiving module of the SELECT tags response signals. This module is composed by the UWB front end LORELEI, an analog chip designed by CEA-LETI.

- Processing Module: Processing module of the data flow outgoing from the UWB receiving module. This module is composed by an FPGA, which is responsible for processing of this data flow, and by a microprocessor, which takes care of the communication with the SELECT central unit.

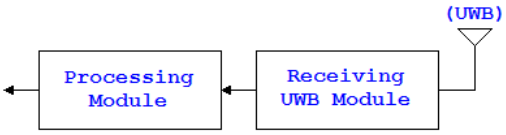


Figure 3.1: Implementation of the receiver of a SELECT reader

The implementative architecture of the structure of the SELECT reader receiver in figure 2.12 is shown in figure 3.2, which is composed of four parts:

- UWB Front End: Block performs the analog processing of frequency converting of the received UWB signal.
- Analog Core: Block structured in N_{af} branches, called analog fingers, each of which is composed by an integrate and dump and a sampler (see figure 2.12). Each analog finger can hook the position in the PRP of pulses relating to a single SELECT tag, and therefore, this block allows to demodulate in parallel up to N_{af} payloads of the SELECT tags response signals.
- A/D Converter: Block making the analog to digital conversion of the signal coming out from the analog core.
- Digital Core: Block structured in N_{dd} branches, called digital despreaders, each of which is composed of a despreader, an accumulator and a code generator (see figure 2.12). Each digital despreader can detect a single SELECT tag, and therefore, this block allows to detect in parallel up to N_{dd} SELECT tags.

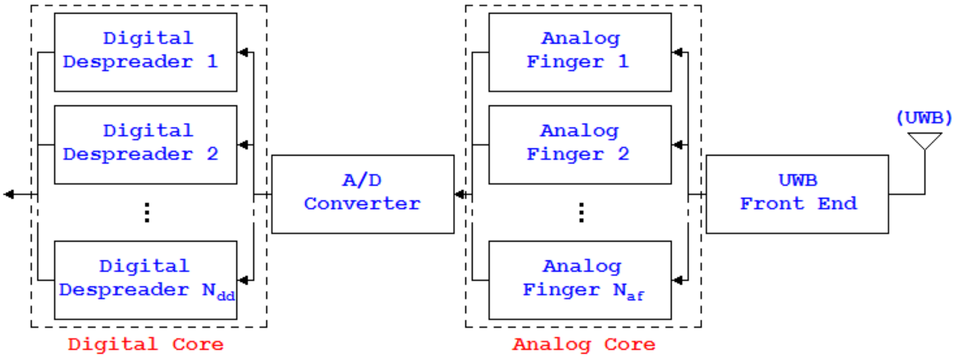


Figure 3.2: Implementative architecture of the receiving section of a SELECT reader

The architecture of figure 3.2 is an hybrid architecture, since the UWB front end and the analog core are implemented in analog while the digital core is implemented digitally.

3.1.1 UWB Front End LORELEI Architecture

The UWB front end LORELEI is an analog chip designed by CEA-LETI that implements the UWB receiving module of a SELECT reader and in particular the block of UWB front end and analog core of the architecture of figure 3.2.

This chip has 4 analog fingers, whose positions on individual PRP must be managed by an external logic. In particular, the hardware constraints of the architecture of the chip itself require that the position of the 4 analog fingers on the single PRP should remain available for $64ns$, i.e. the external control logic may update their position every $64ns$.

The structure of a single finger of the UWB front end LORELEI is shown in figure 3.3. The incoming UWB signal in that structure, after being reported in baseband through an operation of down conversion and a low-pass filtering, is translated in frequency in correspondence of an intermediate frequency to $500MHz$. This latter operation projects the baseband UWB signal on 4 sinusoidal components, forming 4 channels, denominated II, IQ, QI, and QQ. Subsequently, this operation allow to perform a refinement of the TOA estimation. The 4 signals outgoing from the mixers of frequency translation are then integrated on integration windows of $2ns$, whose temporal position depends on where the external control logic have placed the considered analog finger. Finally, the signal from each integrator is converted to digital by a 5 bits ADC.

The scanning of an entire PRP takes place by means of 128 integration windows of $2ns$ superposed on each other of 50%. Since each available analog finger must be kept fixed for $64ns$, for each PRP, which has time duration of $128ns$, it is possible to obtain up to 8 integrations per channel, corresponding to the same number of samples in the output from each ADC.

In particular, the architecture of the LORELEI chip makes available, to the input of each ADC, a sample every $16ns$, implying a sampling frequency equal to $62.5MHz$ and not allowing the full scan of each PRP of the UWB received signal [9].

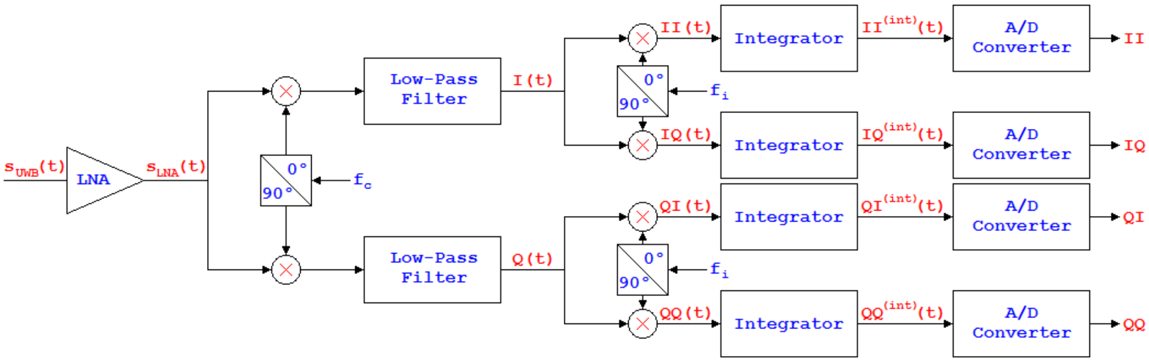


Figure 3.3: Structure of a finger of the UWB front end LORELEI

Depending on the portion of the response signal of figure 2.4, the changes mode of fingers positioning, in particular:

- Preamble: During the reception of that portion of the response signal, the SELECT reader has no information on the pulses position and therefore must be able to scan the entire signal in order to identify the position of the latter. Since the LORELEI chip can not completely scan a PRP, it is necessary to apply a method that allows to estimate that full scan. Whereas, during that portion of the received signal, the data symbols are all 1 and assuming the stationarity of the transmission channel, the SELECT reader apply a scheduling policy in order to make possible to obtain the full scan of a data symbol by collecting the scans of 16 data symbols.

- Payload: During the reception of that portion of the response signal, the SELECT reader knows, thanks to the previous phase, the position in the PRP of the pulses relative to the SELECT tags that have responded to the interrogation. Therefore, it may fix each finger on the integration windows relative to that pulses positions and thus make it possible to perform the demodulation of that portion of the received signal.

3.2 Baseband SELECT Reader Receiver Simulator Implementation

One of the fundamental phases of the design of a generic system is the debug phase, i.e. the phase in which is performed the search of any errors in the system itself in order to correct them.

Given the highly innovative content of the SELECT system, the debug phase becomes more than essential to the success of the project.

One of the debugging methods which allow to obtain the best compromise between economy, cost and effectiveness is certainly the computer simulation.

In particular, the debug phase via simulator consists, given the same input, in the comparison of the simulator outputs with the debugged system outputs. If these outputs are the same, then there is a reasonable certainty that both implementations are correct, and if the outputs are discordant, errors in at least one of the two implementations have to be identified. The use of a simulator for the debug phase is very effective only if its implementation is significantly different from that of the debugged system, because, in this way, it is possible to avoid any correlation between the architectures that could manifest the same errors that would not be detectable.

Considering the implementation of the SELECT reader receiver of figure 3.1, the section that requires more debug is the processing part of the baseband UWB signal coming out from the UWB front end LORELEI, which is implemented by the FPGA firmware of the processing module of the SELECT reader receiver implementation.

Therefore, for the debugging purposes of this firmware, I have implemented a simulator in Matlab code of this processing part.

In particular, the portion of the firmware that is currently available for debugging is the part relating to the reception and processing of the preamble of the received UWB signal. Therefore, the realized Matlab simulator simulates only that portion of the FPGA firmware.

In order to an easier and faster use of the data flow outgoing from the UWB front end LORELEI, CEA-LETI has created the text files in which is stored that data flow.

In this way, it is not necessary to have an hardware acquisition interface, making possible to perform the debug phase entirely via software both for the FPGA firmware and for the Matlab simulator.

Therefore, the outgoing signal from the UWB front end LORELEI is placed as input in the form of text file for both implementations.

Each of this text files contains a data flow related to a specific acquisition mode and they are structured in rows of 32 bits, each of which is composed of 4 samples coming out from the 4 ADCs of UWB front end LORELEI, i.e. the 4 samples of the channels II, IQ, QI and QQ. In particular, each sample is composed of 5 LSB bits relating to the output of an ADC and of 3 MSB bits set to 0. In addiction, the order of arrangement of the samples in each row of these text files is II, IQ, QQ and QI.

The Matlab simulator of the baseband SELECT reader receiver is called `Baseband_Receiver_Simulator` and it is divided into three sections:

- Despreading Section.
- Detection and coarse TOA estimation section.
- Fine TOA estimation section.

3.2.1 Despreading Section

This section of `Baseband_Receiver_Simulator` performs the despreading and the accumulation of the signal outgoing from the UWB front end LORELEI that, as mentioned above, is available in the form of text files.

The Matlab code of that section of `Baseband_Receiver_Simulator` is shown in figure 3.4.

One of the features which must be implemented in this section is the synchronization of the desired SELECT tag spreading code with that of the SELECT reader itself in order to create the proper despreading code to be applied to the received signal.

For this purpose it is necessary an iterative process that allows to apply different despreading codes, each created by the product between the SELECT reader spreading code and a replica of the desired SELECT tag spreading code appropriately shifted in time.

To do this, `Baseband_Receiver_Simulator` uses two parameters:

- `Nspan`: Sets the amount of replicas of the desired SELECT tag spreading code to use, establishing how many different despreading codes apply to the received signal in order to find the correlation peak and then achieve synchronization.

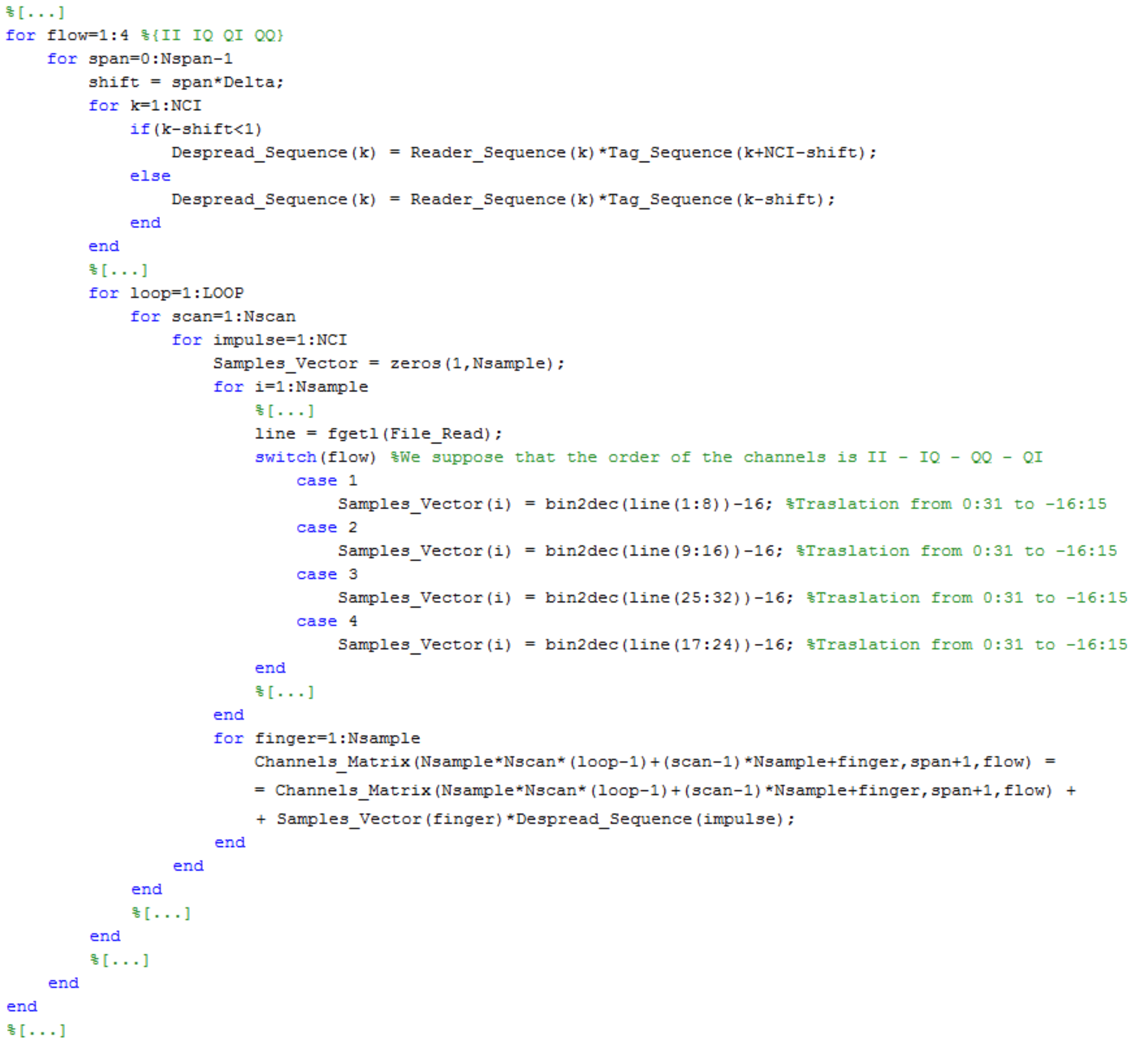
- `Delta`: Sets the amount of time shift between two consecutive replicas of the desired SELECT tag spreading code. It must be a natural number and its unit of measurement is the PRP.

After generating a despreading code, at each iteration the accumulation is performed of the received signal pulses, which, due to the scheduling policy set by the UWB front end LORELEI, must be performed in groups of 8 samples, i.e. it must be done by reading the input files by 8 rows at a time.

In particular, this accumulation operation involves the reading of 8 rows from the input file, applying the correct despreading code and summing the result to an accumulation matrix called `Channels_Matrix`.

The size of this matrix is $128 \times N_{span} \times 4$, where 128 corresponds to the number of integration windows in which is subdivided a single PRP, N_{span} corresponds to the number of despreading codes to be applied to the input samples flow and 4 corresponds to the number of channels on which is divided this flow, i.e. II, IQ, QI, and QQ [10].

Finally, it has to be noted that, before the processing, the input samples undergo an offset equal to 16 units. This is because the 5 bits ADCs of UWB front end LORELEI have an input dynamics range centered on 0 and an output dynamics range ranging from 0 to 31. Therefore, in order to make centered on 0 also the dynamics range of the samples outgoing from these components, it is necessary to perform that translation, obtaining an output dynamics range ranging from -16 to 15.



```
%[...]
for flow=1:4 %{II IQ QI QQ}
    for span=0:Nspan-1
        shift = span*Delta;
        for k=1:NCI
            if(k-shift<1)
                Despread_Sequence(k) = Reader_Sequence(k)*Tag_Sequence(k+NCI-shift);
            else
                Despread_Sequence(k) = Reader_Sequence(k)*Tag_Sequence(k-shift);
            end
        end
        %[...]
        for loop=1:LOOP
            for scan=1:Nscan
                for impulse=1:NCI
                    Samples_Vector = zeros(1,Nsample);
                    for i=1:Nsample
                        %[...]
                        line = fgetl(File_Read);
                        switch(flow) %We suppose that the order of the channels is II - IQ - QQ - QI
                            case 1
                                Samples_Vector(i) = bin2dec(line(1:8))-16; %Traslation from 0:31 to -16:15
                            case 2
                                Samples_Vector(i) = bin2dec(line(9:16))-16; %Traslation from 0:31 to -16:15
                            case 3
                                Samples_Vector(i) = bin2dec(line(25:32))-16; %Traslation from 0:31 to -16:15
                            case 4
                                Samples_Vector(i) = bin2dec(line(17:24))-16; %Traslation from 0:31 to -16:15
                        end
                        %[...]
                    end
                    for finger=1:Nsample
                        Channels_Matrix(Nsample*Nscan*(loop-1)+(scan-1)*Nsample+finger,span+1,flow) =
                        = Channels_Matrix(Nsample*Nscan*(loop-1)+(scan-1)*Nsample+finger,span+1,flow) +
                        + Samples_Vector(finger)*Despread_Sequence(impulse);
                    end
                end
            end
            %[...]
        end
        %[...]
    end
end
%[...]
```

Figure 3.4: Matlab code for the despreading

3.2.2 Detection and Coarse TOA Estimation Section

This section of `Baseband_Receiver_Simulator` performs the detection of the desired SELECT tag and, when that is present, the coarse TOA estimation [15].

Matlab code for that section of `Baseband_Receiver_Simulator` is shown in figure 3.5.

First, the detection algorithm involves the calculation of an energy matrix, called `Energy_Matrix`, through proper combination of the samples present in the matrix `Channels_Matrix`.

In particular, considering the $i-th$ integration window and the $k-th$

despreading code, the generic element $E_{i,k}$ of the matrix `Energy_Matrix` is created considering the samples $II_{i,k}$, $IQ_{i,k}$, $QI_{i,k}$, $QQ_{i,k}$ of the 4 channels II, IQ, QI, QQ and combining them as follows:

$$E_{i,k} = I_{i,k}^2 + Q_{i,k}^2$$

with:

$$I_{i,k} = II_{i,k} - QQ_{i,k}$$

$$Q_{i,k} = IQ_{i,k} + QI_{i,k}$$

Thus, the size of this matrix is $128 \times N_{span}$, where 128 corresponds to the number of integration windows which is subdivided a single PRP and N_{span} corresponds to the number of despreading codes applied to the input samples flow.

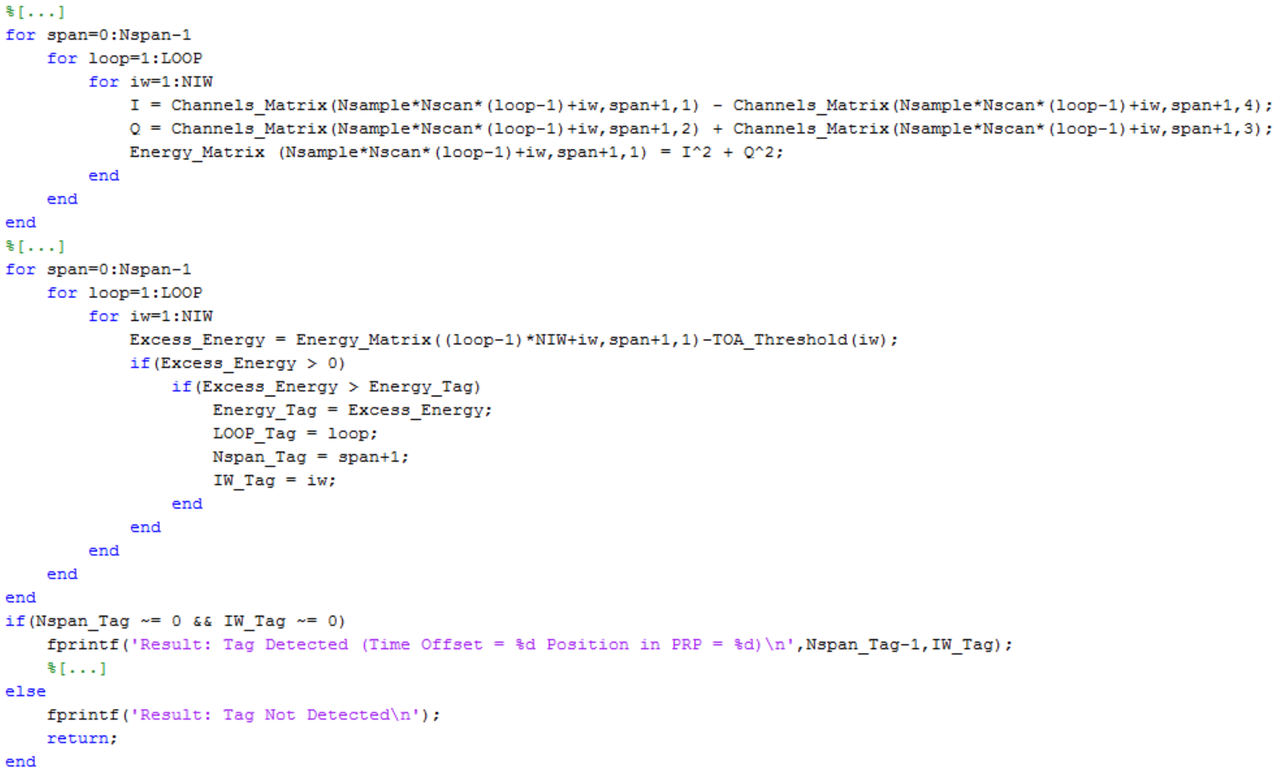
Once calculated the matrix `Energy_Matrix`, the detection algorithm involves to apply on that matrix a thresholds vector, called `TOA_Threshold` and size 128×1, which has a variable profile in order to mitigate the near-far effects. In particular, this profile depend on the specific considered scenario, but in general must be decreasing at the increasing of the integration window number. This is because, the signals present in the integration windows with reduced number tend to have greater amplitude than the signals present in the integration windows with greater numbers [8].

After applying the vector `TOA_Threshold` at the matrix `Energy_Matrix`, the detection algorithm involves the detection of the presence of the desired SELECT tag if and only if at least one of the bin energy of the matrix `Energy_Matrix` has exceeded the threshold corresponding to it.

If the detection phase has detected the presence of the desired SELECT tag, it is possible to proceed to the coarse TOA estimation, which consists in identifying the integration window in which the response pulse of that SELECT tag itself is present [10].

This is because, at the transmission time of the interrogation signal, all pulses that compose it are positioned in a certain integration window within its PRP. However, at the receiving time of this signal, the position of the pulses is different from that at the their transmission time, due to the propagation time required for the interrogation signal to get to the SELECT tag and return to the SELECT reader. Therefore, estimating the final integration window of the desired SELECT tag pulses and knowing the position of these pulses at the time of their transmission, it is possible to estimate the TOA, which is equal to the time difference between these positions.

The coarse TOA estimation algorithm involves the determination of the integration window of desired SELECT tag pulses determining, in the matrix `Energy_Matrix`, the integration window of the widest energy bin among of all those that have passed the corresponding threshold.



```
%[...]
for span=0:Nspan-1
    for loop=1:LOOP
        for iw=1:NIW
            I = Channels_Matrix(Nsample*Nscan*(loop-1)+iw,span+1,1) - Channels_Matrix(Nsample*Nscan*(loop-1)+iw,span+1,4);
            Q = Channels_Matrix(Nsample*Nscan*(loop-1)+iw,span+1,2) + Channels_Matrix(Nsample*Nscan*(loop-1)+iw,span+1,3);
            Energy_Matrix (Nsample*Nscan*(loop-1)+iw,span+1,1) = I^2 + Q^2;
        end
    end
end
%[...]
for span=0:Nspan-1
    for loop=1:LOOP
        for iw=1:NIW
            Excess_Energy = Energy_Matrix((loop-1)*NIW+iw,span+1,1)-TOA_Threshold(iw);
            if(Excess_Energy > 0)
                if(Excess_Energy > Energy_Tag)
                    Energy_Tag = Excess_Energy;
                    LOOP_Tag = loop;
                    Nspan_Tag = span+1;
                    IW_Tag = iw;
                end
            end
        end
    end
end
if(Nspan_Tag ~= 0 && IW_Tag ~= 0)
    fprintf('Result: Tag Detected (Time Offset = %d Position in PRP = %d)\n',Nspan_Tag-1,IW_Tag);
    %[...]
else
    fprintf('Result: Tag Not Detected\n');
    return;
end
```

Figure 3.5: Matlab code for the detection and the coarse TOA estimation

3.2.3 Fine TOA Estimation Section

This section of `Baseband_Receiver_Simulator` performs the fine TOA estimation of the desired SELECT tag if it has been detected.

Matlab code for that section of `Baseband_Receiver_Simulator` is shown in figure 3.9.

The coarse TOA estimation allows to estimate the integration window of the desired SELECT tag pulses, but does not allow to determine the exact position of these within that integration window. Therefore, the precision with which it is possible to estimate the TOA during this operation can not fall below the temporal duration of an integration window, i.e. $1ns$.

However, that accuracy is not sufficient to satisfy the expected performance for the SELECT system and therefore it is necessary a fine TOA estimation phase in order to estimate the position of the pulses inside the integration window determined during the coarse TOA estimation phase.

The algorithm of fine TOA estimation is based on the functionalities offered by the UWB front end LORELEI, exploiting the information obtainable from the components resulting from the projection on the sinusoidal basis executed by that chip, i.e. the availability of the four channels II, IQ, QI and QQ.

Assuming that the coarse TOA estimation phase has estimated that the integration window of the desired SELECT tag pulses is the $k-th$ and that the despreading code that has allowed to obtain the maximum correlation peak is the $i-th$, the fine TOA estimation algorithm involves the refinement of the coarse TOA estimate by processing the samples $II_{i,k}$, $IQ_{i,k}$, $QI_{i,k}$, $QQ_{i,k}$ of the matrix `Channels_Matrix`.

Once identified the samples $II_{i,k}$, $IQ_{i,k}$, $QI_{i,k}$, $QQ_{i,k}$, the fine TOA estimation algorithm involves three steps [7] [8]:

1. Determination of the energy of the two orthogonal components outgoing from the down conversion block of the UWB front end LORELEI by calculating:
$$Energy_1 = II_{i,k}^2 + IQ_{i,k}^2$$
$$Energy_2 = QI_{i,k}^2 + QQ_{i,k}^2$$

2. Determination of the orthogonal component with more energy, defining:
$$\begin{cases} \begin{cases} I_{i,k} = II_{i,k} \\ Q_{i,k} = IQ_{i,k} \end{cases} & Energy_1 \geq Energy_2 \\ \begin{cases} I_{i,k} = QI_{i,k} \\ Q_{i,k} = QQ_{i,k} \end{cases} & Energy_2 \geq Energy_1 \end{cases}$$

3. Estimation of the pulses position in the $k-th$ integration window by calculating the arctangent as follows:
$$\varphi = \arctan\left(\frac{Q_{i,k}}{I_{i,k}}\right)$$

The projection on sinusoidal bases by the UWB front-end LORELEI allows to decompose on orthogonal basis the two outgoing components from the down conversion block.

In this way it is possible to associate a phasor at these orthogonal components of which it is possible to determine the angle φ by calculating the arctangent (see figure 3.6).

Now, associating at the $k-th$ integration window a domain equal to $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, it is possible to estimate the position of the desired SELECT tag pulses by applying a simple linear relationship between the angle φ and the real position (see figure 3.7).

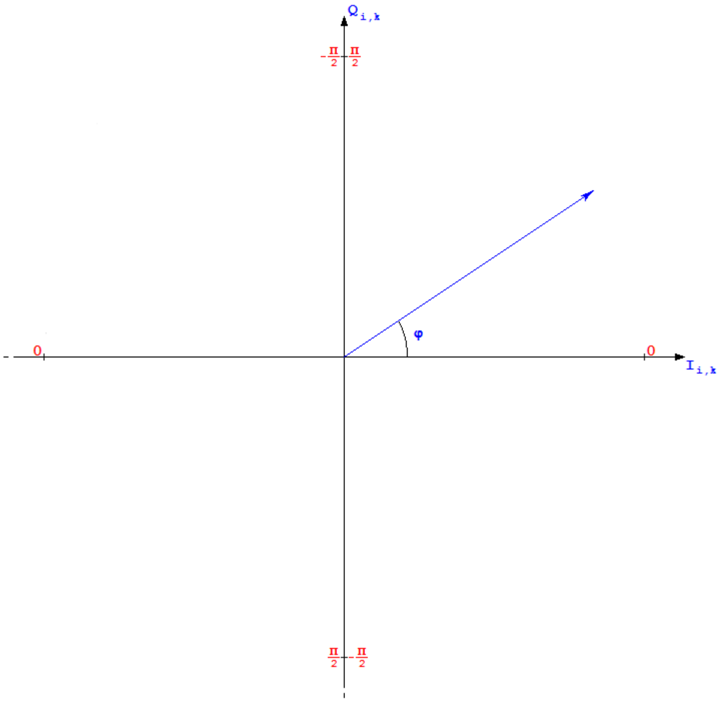


Figure 3.6: Phasor of the orthogonal components

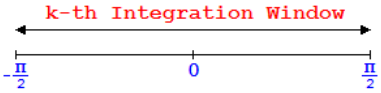


Figure 3.7: Domain in angles of an integration window

This fine TOA estimation algorithm is very effective but also is very expensive in terms of computing resources. From the point of view of `Baseband_Receiver_Simulator`, which is executed by a computer, this latter aspect is not a problem, while it is an issue for the FPGA that performs the firmware that implements this algorithm.

In particular, from the point of view of the computing resources necessary for the execution of the fine TOA estimation algorithm, the more expensive operations for a FPGA are the squaring and the calculation of the arctangent.

Therefore, in order to make much less expensive that algorithm in terms of resources, the FPGA, and thus also `Baseband_Receiver_Simulator`, implements a simplified version. This simplified version of the fine TOA estimation algorithm involves:

1. Determination of the energy of the two orthogonal components outgoing from the down conversion block of the UWB front end LORELEI by calculating:

$$Energy_1 = |II_{i,k}| + |IQ_{i,k}|$$

$$Energy_2 = |QI_{i,k}| + |QQ_{i,k}|$$

2. Determination of the orthogonal component with more amplitude, defining:

$$\begin{cases} \begin{cases} I_{i,k} = II_{i,k} \\ Q_{i,k} = IQ_{i,k} \end{cases} & Energy_1 \geq Energy_2 \\ \begin{cases} I_{i,k} = QI_{i,k} \\ Q_{i,k} = QQ_{i,k} \end{cases} & Energy_2 \geq Energy_1 \end{cases}$$

3. Estimation of the pulses position within the $k - th$ integration window by reading a lookup table containing approximate values of the arctangent.

The quantization of the arctangent was accomplished by splitting the codomain of this function, which is the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, in 16 portions and associating to each of these its central value (see figure 3.8). In this way, each window has a width equal to $\frac{\pi}{16}$ and the maximum quantization error associated with the approximate arctangent function is equal to $\frac{\pi}{32}$, i.e. $31.25ps$.

Therefore, the lookup table associated with that approximate function is shown in table 3.1, in which has been defined:

$$signIQ = sign(I_{i,k}) \cdot sign(Q_{i,k})$$

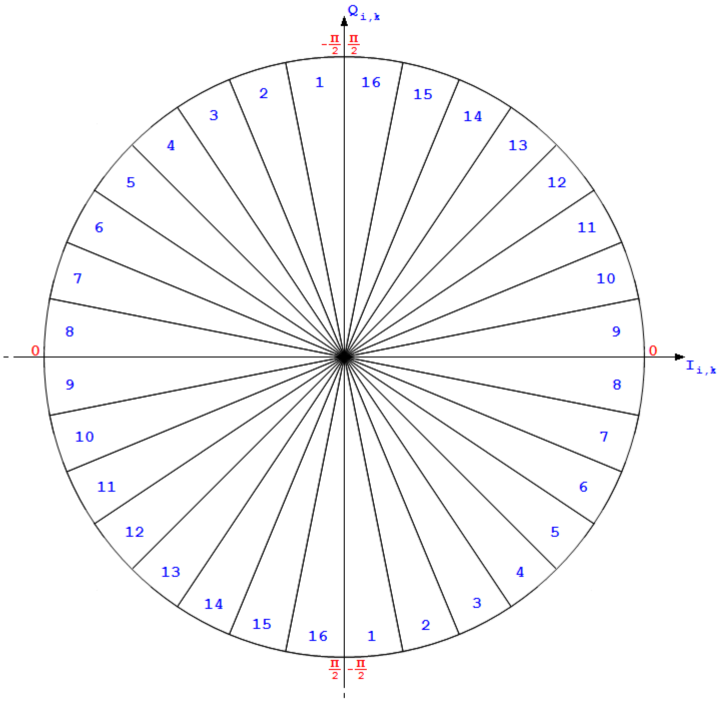
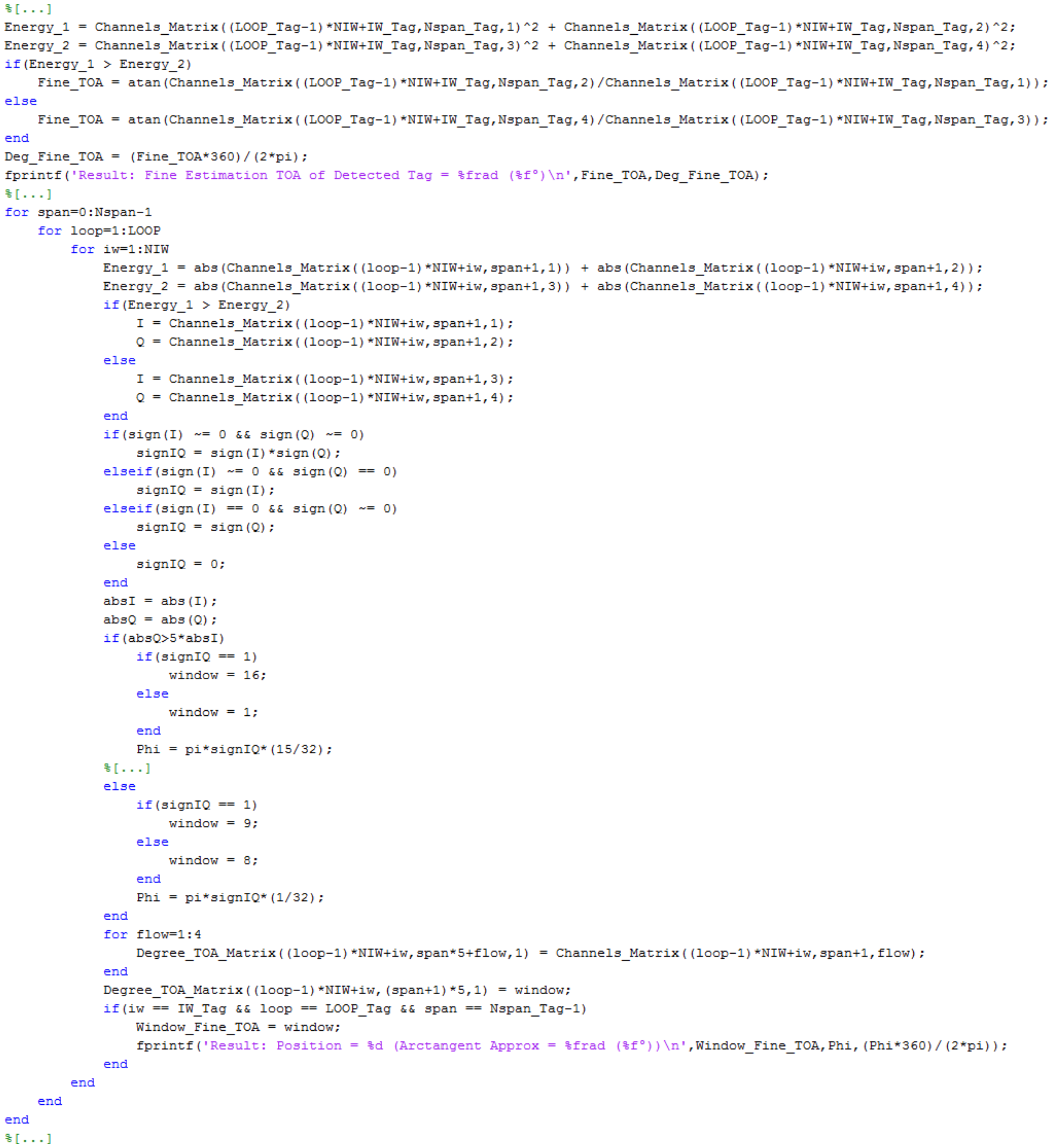


Figure 3.8: Quantization of the arctangent for FPGA

Condition	Approximation	Portion in Integration Window
$Q_{i,k} > 5I_{i,k}$	$\frac{15}{32}\pi \cdot signIQ$	$1st \quad signIQ = +1$ $16th \quad signIQ = -1$
$2.5I_{i,k} \leq Q_{i,k} \leq 5I_{i,k}$	$\frac{13}{32}\pi \cdot signIQ$	$2nd \quad signIQ = +1$ $15th \quad signIQ = -1$
$1.5I_{i,k} \leq Q_{i,k} \leq 2.5I_{i,k}$	$\frac{11}{32}\pi \cdot signIQ$	$3rd \quad signIQ = +1$ $14th \quad signIQ = -1$
$I_{i,k} \leq Q_{i,k} \leq 1.5I_{i,k}$	$\frac{9}{32}\pi \cdot signIQ$	$4th \quad signIQ = +1$ $13th \quad signIQ = -1$
$Q_{i,k} \leq I_{i,k} \leq 1.5Q_{i,k}$	$\frac{7}{32}\pi \cdot signIQ$	$5th \quad signIQ = +1$ $12th \quad signIQ = -1$
$1.5Q_{i,k} \leq I_{i,k} \leq 2.5Q_{i,k}$	$\frac{5}{32}\pi \cdot signIQ$	$6th \quad signIQ = +1$ $11th \quad signIQ = -1$
$2.5Q_{i,k} \leq I_{i,k} \leq 5Q_{i,k}$	$\frac{3}{32}\pi \cdot signIQ$	$7th \quad signIQ = +1$ $10th \quad signIQ = -1$
$I_{i,k} \geq 5Q_{i,k}$	$\frac{1}{32}\pi \cdot signIQ$	$8th \quad signIQ = +1$ $9th \quad signIQ = -1$

Table 3.1: Lookup table of the approximation arctangent for FPGA



```matlab
%[...]
Energy_1 = Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,1)^2 + Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,2)^2;
Energy_2 = Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,3)^2 + Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,4)^2;
if(Energy_1 > Energy_2)
    Fine_TOA = atan(Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,2)/Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,1));
else
    Fine_TOA = atan(Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,4)/Channels_Matrix((LOOP_Tag-1)*NIW+IW_Tag,Nspan_Tag,3));
end
Deg_Fine_TOA = (Fine_TOA*360)/(2*pi);
fprintf('Result: Fine Estimation TOA of Detected Tag = %frad (%f°)\n',Fine_TOA,Deg_Fine_TOA);
%[...]
for span=0:Nspan-1
    for loop=1:LOOP
        for iw=1:NIW
            Energy_1 = abs(Channels_Matrix((loop-1)*NIW+iw,span+1,1)) + abs(Channels_Matrix((loop-1)*NIW+iw,span+1,2));
            Energy_2 = abs(Channels_Matrix((loop-1)*NIW+iw,span+1,3)) + abs(Channels_Matrix((loop-1)*NIW+iw,span+1,4));
            if(Energy_1 > Energy_2)
                I = Channels_Matrix((loop-1)*NIW+iw,span+1,1);
                Q = Channels_Matrix((loop-1)*NIW+iw,span+1,2);
            else
                I = Channels_Matrix((loop-1)*NIW+iw,span+1,3);
                Q = Channels_Matrix((loop-1)*NIW+iw,span+1,4);
            end
            if(sign(I) ~= 0 && sign(Q) ~= 0)
                signIQ = sign(I)*sign(Q);
            elseif(sign(I) ~= 0 && sign(Q) == 0)
                signIQ = sign(I);
            elseif(sign(I) == 0 && sign(Q) ~= 0)
                signIQ = sign(Q);
            else
                signIQ = 0;
            end
            absI = abs(I);
            absQ = abs(Q);
            if(absQ>5*absI)
                if(signIQ == 1)
                    window = 16;
                else
                    window = 1;
                end
                Phi = pi*signIQ*(15/32);
            %[...]
            else
                if(signIQ == 1)
                    window = 9;
                else
                    window = 8;
                end
                Phi = pi*signIQ*(1/32);
            end
            for flow=1:4
                Degree_TOA_Matrix((loop-1)*NIW+iw,span*5+flow,1) = Channels_Matrix((loop-1)*NIW+iw,span+1,flow);
            end
            Degree_TOA_Matrix((loop-1)*NIW+iw,(span+1)*5,1) = window;
            if(iw == IW_Tag && loop == LOOP_Tag && span == Nspan_Tag-1)
                Window_Fine_TOA = window;
                fprintf('Result: Position = %d (Arctangent Approx = %frad (%f°))\n',Window_Fine_TOA,Phi,(Phi*360)/(2*pi));
            end
        end
    end
end
%[...]
```

Figure 3.9: Matlab code for the fine TOA estimation

3.3 Application Example of the Baseband SELECT Reader Receiver Simulator

As an application example of `Baseband_Receiver_Simulator` we consider one of the input file provided by CEA-LETI, which involves the presence of a single SELECT tag and an interrogation signal composed by a data symbol $b_n = 1$ which undergoes the spreading operation by a spreading codes characterized by a number of chips $N_c = 1024$, at each of which are associated a number of pulses $N_{pc} = 1$.

The simulation parameters are:

$$N_{span} = 3$$

$$\Delta = 1$$

Furthermore, since it is not available a characterization of the transmission channel in which the acquisitions have been made to create the input file, each of the bin energy threshold is set to 1.

Positioning the considered file in input to `Baseband_Receiver_Simulator`, after the despreading operation and the accumulation, the samples shown in figure 3.10 and the energy bins shown in figure 3.11 are obtained.

Finally, the results obtained by `Baseband_Receiver_Simulator` in response to the example input file are:

Parameter	Value
$Span$	0
$Coarse\,TOA$	62
$Fine\,TOA$	$-0.774rad\,(-44.349°)$
$Portion\,in\,Integration\,Window$	$5th$
$Arctangent\;Approximation$	$-0.687rad\,(-39.375°)$

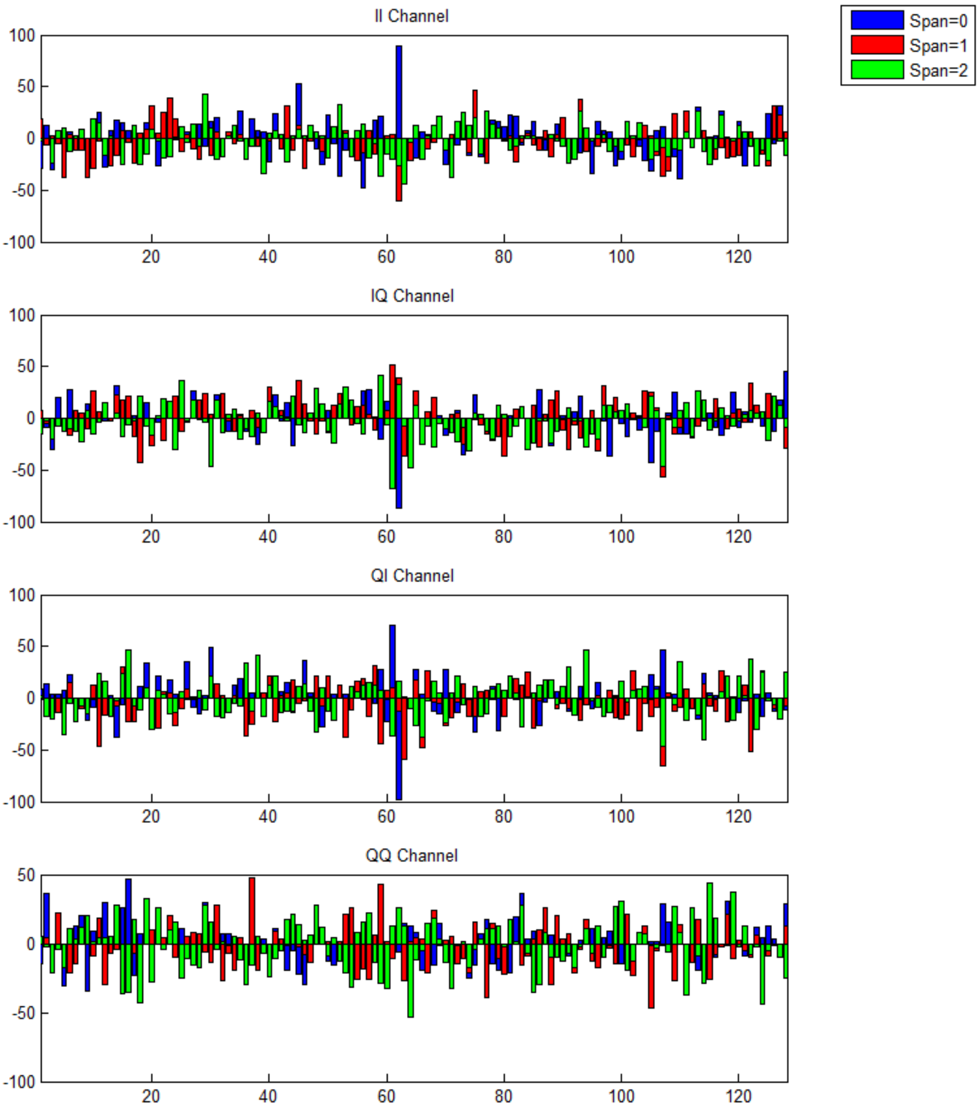


Figure 3.10: Accumulation samples of the channels II, IQ, QI e QQ

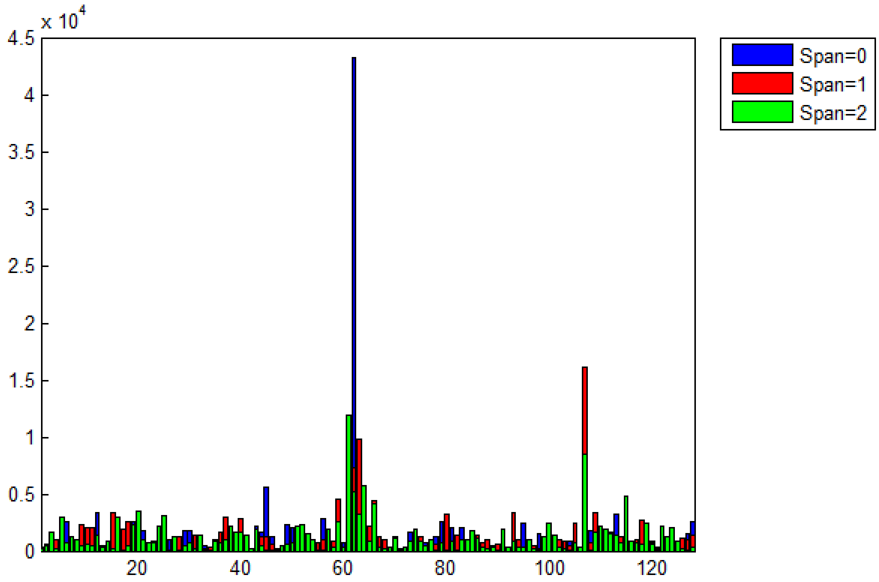


Figure 3.11: Energy bins

3.4 Conclusions and Open Issues

By making various comparisons, with the same input, between the results obtained by the execution of the FPGA firmware and `Baseband_Receiver_Simulator`, results show an exact match, allowing to conclude that, with high probability, the two implementations are correct.

However, there may still be present conceptual errors common to both implementations that could generate identical results but incorrect. Therefore, it is necessary to continue the debugging for the purpose to detect the possible presence of these type of errors.

Chapter 4

Input File Generator for Baseband SELECT Reader Receiver

4.1 Implementation of Input File Generator for Baseband SELECT Reader Receiver

In order to validate the baseband SELECT reader receiver implementation, the use of `Baseband_Receiver_Simulator` allows to successfully perform a significant part of the FPGA firmware debug.

However, given the highly innovative content of the SELECT system, the debug phase can not stop at simple comparison of the results obtained by `Baseband_Receiver_Simulator` and by the FPGA firmware, because there may be present conceptual errors common to both implementations that generate the same results but incorrect.

Therefore, in order to detect the possible presence of this errors type, it is necessary to know a priori which results expected from the application of a given input file, in order to verify if those obtained by `Baseband_Receiver_Simulator` and by the FPGA firmware are reliable.

This is not always possible using real input file due to measurement errors and uncertainty of propagation environments and of acquisition system components.

Also in cases where the expected results can be determined, the non-idealities of the input file may imply, by the two implementations, the generation of the results agree with each other but different from those expected. In this case, it is very difficult determine if these differences are due to conceptual errors of that implementations or to the non-idealities present in the input file itself.

Since they are the real files, the input files provided by CEA-LETI have the problems mentioned above and thus are not usable for the purpose of conceptual errors detection present in `Baseband_Receiver_Simulator` and in the FPGA firmware.

Therefore, in order to solve these issues, I have implemented an input file generator in Matlab code that allows to synthesize the input files according to simple channel model and without any non-idealities, allowing a complete and precise determination of the expected results.

The Matlab generator of the input files is called `Input_File_Generator` and it emulates three parts of the SELECT system:

- Section of the SELECT reader transmission: Generation of the interrogation signal by reproduction of the behavior of the SELECT reader transmission section.

- Transmission channel and SELECT tags: Generation of the signal received by the SELECT reader receiver by reproduction of the behavior of the transmission channel and of the SELECT tags.

- UWB Front End LORELEI: Generation of the input files by reproduction of the ideal behavior of the UWB Front End LORELEI.

The basic pulse used by `Input_File_Generator` is a root raised cosine pulse as follows [7] [8]:

$$p(t) = \frac{4\beta}{\pi\sqrt{t_p}}\left(cos\left((1+\beta)\,\pi\frac{t}{t_p}\right) + \frac{1}{4\beta\frac{t}{t_p}}sin\left((1-\beta)\,\pi\frac{t}{t_p}\right)\right)\frac{1}{1-\left(4\beta\frac{t}{t_p}\right)^2}$$

having defined:

$$t_p = \frac{1+\beta}{BW}$$

where β and BW represent respectively the roll-off factor and the bandwidth of the pulse itself.

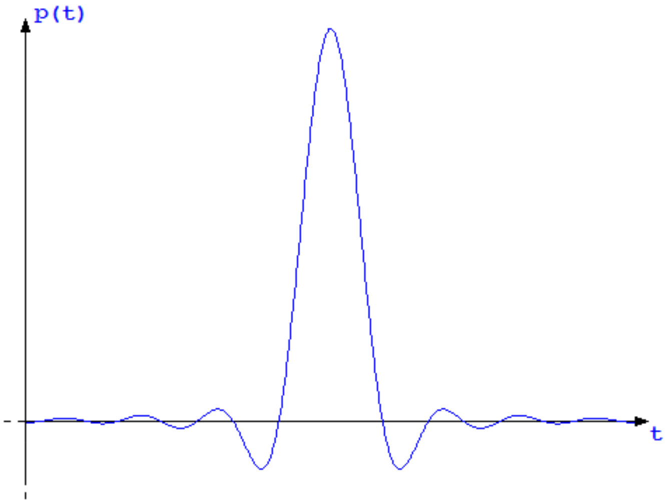


Figure 4.1: Baseband root raised cosine pulse

Considering the useful UWB band ranging from $3.1GHz$ to $4.8GHz$, it is necessary perform an up conversion by a sinusoidal carrier $f_c = 3.95GHz$, obtaining:

$$p_{RF}(t) = \frac{4\beta}{\pi\sqrt{t_p}}\left(cos\left((1+\beta)\,\pi\frac{t}{t_p}\right) + \frac{1}{4\beta\frac{t}{t_p}}sin\left((1-\beta)\,\pi\frac{t}{t_p}\right)\right)\frac{1}{1-\left(4\beta\frac{t}{t_p}\right)^2}cos\left(2\pi f_c t + \alpha\right)$$

where α represents the starting angle of the sinusoidal carrier f_c.

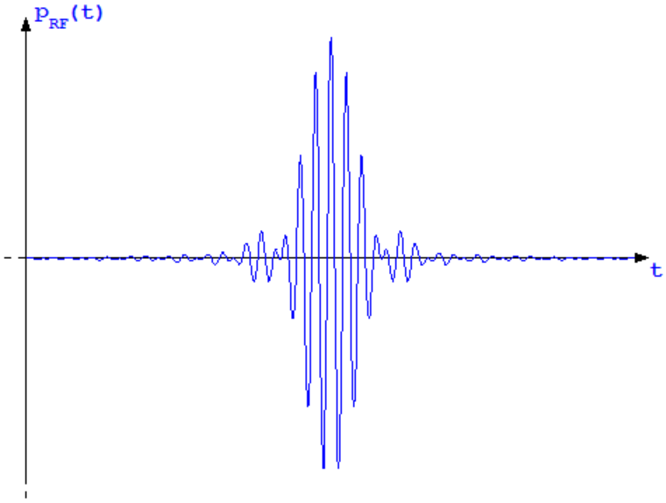


Figure 4.2: Radio frequency root raised cosine pulse

To make this pulse, meet the EU mask of the European UWB signals regulations in the band ranging from $3.1GHz$ to $4.8GHz$, it is necessary that the basic pulse $p(t)$ present a bandwidth $BW = 1.6GHz$ and a roll-off factor $\beta = 0.6$. In this way, the power spectrum of the pulse $p_{RF}(t)$ full respect the EU mask (see figure 4.3).

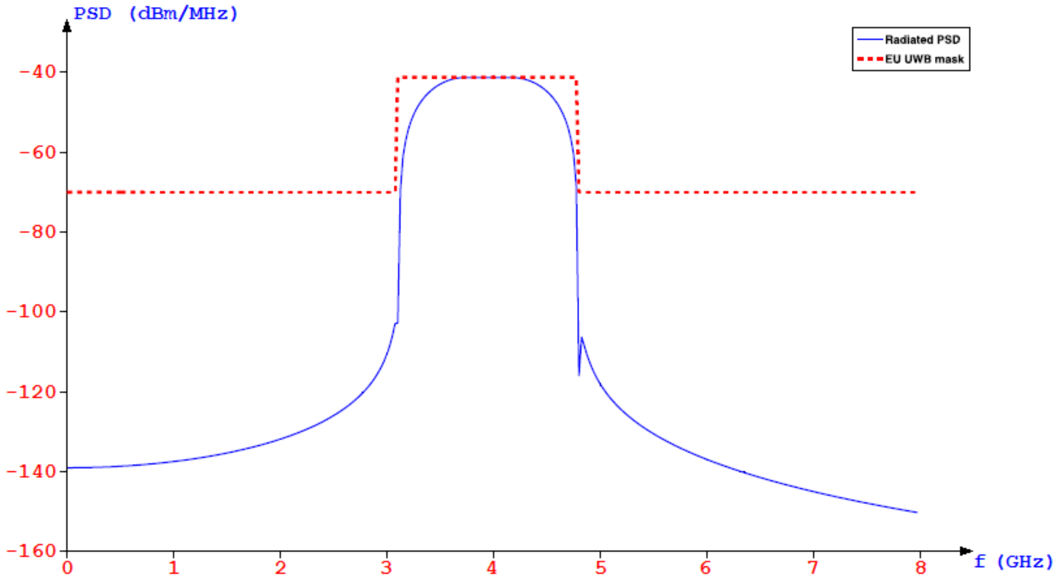


Figure 4.3: Power spectrum of a radio frequency root raised cosine pulse

The Matlab code relating to the generation of the single root raised cosine pulse transmitted by the SELECT reader transmitter is shown in figure 4.4.

This implementation takes into account two types of special cases:

- $\frac{t}{t_p} = 0$: In this case, the direct implementation of the theoretical formula of the root raised cosine pulse $p(t)$ returns an indeterminate form of type NaN. Therefore it is necessary to determine the closed-form expression of the maximum by manipulation of the theoretical formula, in particular:

$$p(t) = \frac{4\beta}{\pi\sqrt{t_p}}\left(cos\left((1+\beta)\,\pi\frac{t}{t_p}\right) + \frac{1}{4\beta\frac{t}{t_p}}sin\left((1-\beta)\,\pi\frac{t}{t_p}\right)\right)\frac{1}{1-\left(4\beta\frac{t}{t_p}\right)^2} =$$

$$= \frac{\frac{4\beta}{\pi\sqrt{t_p}}}{1-\left(\frac{4\beta t}{t_p}\right)^2}\left(cos\left(\frac{(1+\beta)\,\pi t}{t_p}\right) + \frac{1}{\frac{4\beta t}{t_p}}\left(sin\left(\frac{\pi t}{t_p}\right)cos\left(\frac{\beta\pi t}{t_p}\right) - sin\left(\frac{\beta\pi t}{t_p}\right)cos\left(\frac{\pi t}{t_p}\right)\right)\right) =$$

$$= \frac{\frac{4\beta}{\pi\sqrt{t_p}}}{1-\left(\frac{4\beta t}{t_p}\right)^2}\left(cos\left(\frac{(1+\beta)\,\pi t}{t_p}\right) + \frac{\pi}{4\beta}sinc\left(\frac{\pi t}{t_p}\right)cos\left(\frac{\beta\pi t}{t_p}\right) - \frac{\pi}{4}sinc\left(\frac{\beta\pi t}{t_p}\right)cos\left(\frac{\pi t}{t_p}\right)\right)$$

from which, by placing $\frac{t}{t_p} = 0$, can be seen:

$$p\left(0\right) = \frac{4\beta}{\pi\sqrt{t_p}}\left(1 + \frac{\pi}{4\beta} - \frac{\pi}{4}\right) = \frac{4\beta}{\pi\sqrt{t_p}} + \frac{1-\beta}{\sqrt{t_p}}$$

- $\frac{t}{t_p} = \frac{1}{4\beta}$: In this case, the direct implementation of the theoretical formula of the root raised cosine pulse $p(t)$ returns an indeterminate form of type $+\infty$. Therefore it is necessary to determine the closed-form expression of the zero by manipulation of the theoretical formula, in particular:

$$p(t) = \frac{4\beta}{\pi\sqrt{t_p}}\left(cos\left(\left(1+\beta\right)\pi\frac{t}{t_p}\right) + \frac{1}{4\beta\frac{t}{t_p}}sin\left(\left(1-\beta\right)\pi\frac{t}{t_p}\right)\right)\frac{1}{1-\left(4\beta\frac{t}{t_p}\right)^2} =$$

$$= \frac{4\beta}{\pi\sqrt{t_p}}\frac{1}{1-\left(4\beta\frac{t}{t_p}\right)^2}\left(cos\left(\pi\frac{t}{t_p}\right)cos\left(\beta\pi\frac{t}{t_p}\right) - sin\left(\pi\frac{t}{t_p}\right)sin\left(\beta\pi\frac{t}{t_p}\right)\right) +$$

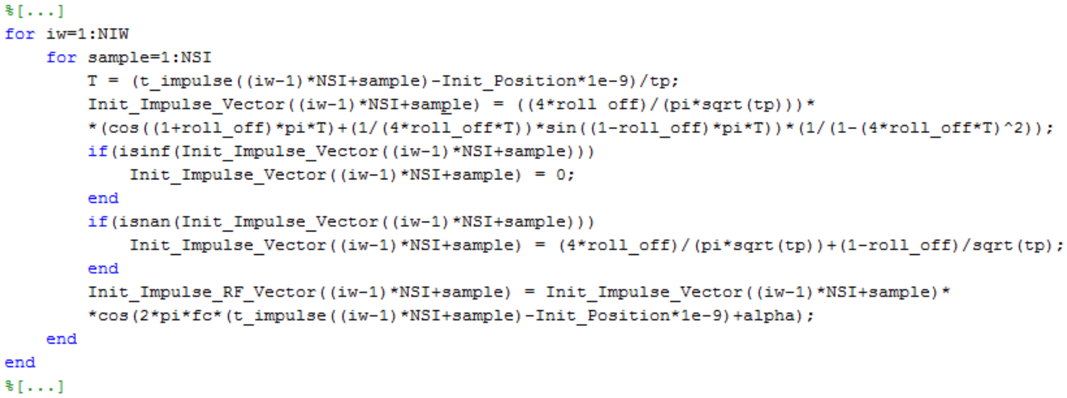
$$+\frac{4\beta}{\pi\sqrt{t_p}}\frac{1}{4\beta\frac{t}{t_p}}\frac{1}{1-\left(4\beta\frac{t}{t_p}\right)^2}\left(sin\left(\pi\frac{t}{t_p}\right)cos\left(\beta\pi\frac{t}{t_p}\right) - cos\left(\pi\frac{t}{t_p}\right)sin\left(\beta\pi\frac{t}{t_p}\right)\right)$$

from which, by placing $\frac{t}{t_p} = \frac{1}{4\beta}$, can be seen:

$$p\left(\frac{t_p}{4\beta}\right) = \frac{4\beta}{\pi\sqrt{t_p}}\frac{1}{1-\left(\frac{4\beta}{4\beta}\right)^2}\left(cos\left(\frac{\pi}{4\beta}\right)cos\left(\frac{\pi}{4}\right) - sin\left(\frac{\pi}{4\beta}\right)sin\left(\frac{\pi}{4}\right)\right) +$$

$$+\frac{4\beta}{\pi\sqrt{t_p}}\frac{1}{1-\left(\frac{4\beta}{4\beta}\right)^2}\left(sin\left(\frac{\pi}{4\beta}\right)cos\left(\frac{\pi}{4}\right) - cos\left(\frac{\pi}{4\beta}\right)sin\left(\frac{\pi}{4}\right)\right) =$$

$$= \frac{32\sqrt{2}\beta^3}{\pi\sqrt{t_p}}\frac{\left(cos\left(\frac{\pi}{4\beta}\right) - sin\left(\frac{\pi}{4\beta}\right)\right) - \left(cos\left(\frac{\pi}{4\beta}\right) - sin\left(\frac{\pi}{4\beta}\right)\right)}{\left(4\beta\right)^2 - \left(4\beta\right)^2} \to 0$$



```
%[...]
for iw=1:NIW
    for sample=1:NSI
        T = (t_impulse((iw-1)*NSI+sample)-Init_Position*1e-9)/tp;
        Init_Impulse_Vector((iw-1)*NSI+sample) = ((4*roll off)/(pi*sqrt(tp)))*
        *(cos((1+roll_off)*pi*T)+(1/(4*roll_off*T))*sin((1-roll_off)*pi*T))*(1/(1-(4*roll_off*T)^2));
        if(isinf(Init_Impulse_Vector((iw-1)*NSI+sample)))
            Init_Impulse_Vector((iw-1)*NSI+sample) = 0;
        end
        if(isnan(Init_Impulse_Vector((iw-1)*NSI+sample)))
            Init_Impulse_Vector((iw-1)*NSI+sample) = (4*roll_off)/(pi*sqrt(tp))+(1-roll_off)/sqrt(tp);
        end
        Init_Impulse_RF_Vector((iw-1)*NSI+sample) = Init_Impulse_Vector((iw-1)*NSI+sample)*
        *cos(2*pi*fc*(t_impulse((iw-1)*NSI+sample)-Init_Position*1e-9)+alpha);
    end
end
%[...]
```

Figure 4.4: Matlab code for the generation of the pulse transmitted by the SELECT reader

Since the input files must be free of non-idealities, the transmission channel considered is an ideal free space. In particular, the emulation of the transmission channel includes:

- Total absence of noise and interference.
- Simplified path gain $PG(d)$ equal to:

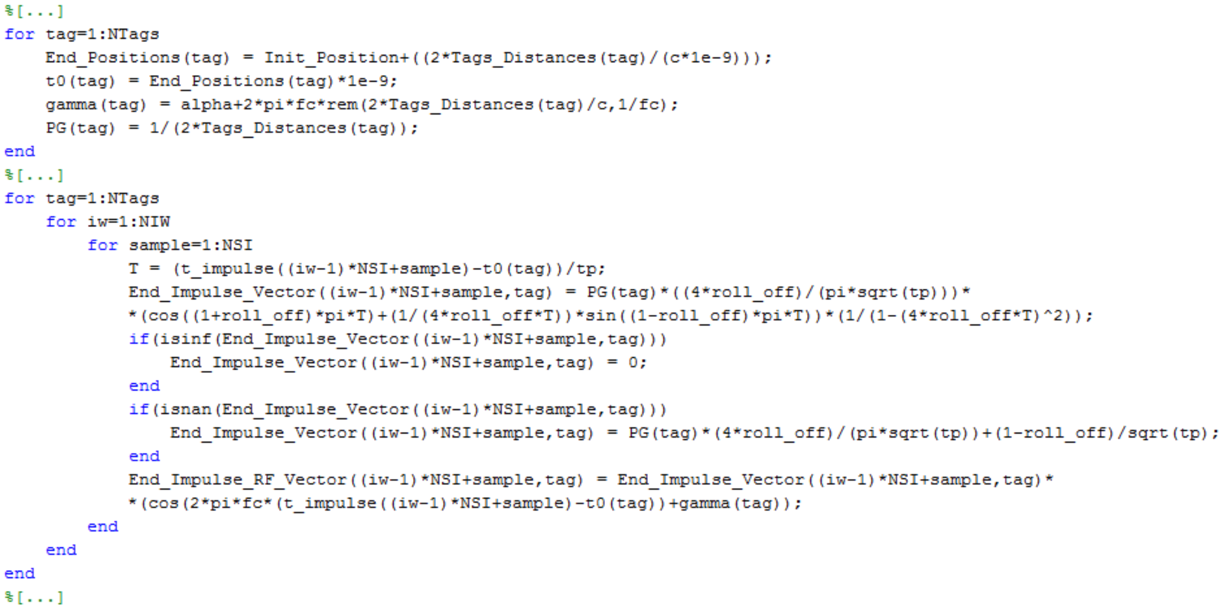
  $$PG(d) = \frac{1}{(2d)^2}$$

  where d represents the distance between SELECT reader and SELECT tag.
- Phase angle γ of the transmitted signal due only to the distance traveled, i.e.:

  $$\gamma = \alpha + 2\pi f_c mod \left(\frac{2d}{c}, \frac{1}{f_c}\right)$$

  where α represents the initial phase angle of the transmitted signal, f_c represents the frequency of the transmitted signal and c represents the propagation speed, that is supposed to be equal to the light speed.

The Matlab code relating to the generation of the single root raised cosine pulse received by the SELECT reader receiver is shown in figure 4.5.



```
%[...]
for tag=1:NTags
    End_Positions(tag) = Init_Position+((2*Tags_Distances(tag)/(c*1e-9)));
    t0(tag) = End_Positions(tag)*1e-9;
    gamma(tag) = alpha+2*pi*fc*rem(2*Tags_Distances(tag)/c,1/fc);
    PG(tag) = 1/(2*Tags_Distances(tag));
end
%[...]
for tag=1:NTags
    for iw=1:NIW
        for sample=1:NSI
            T = (t_impulse((iw-1)*NSI+sample)-t0(tag))/tp;
            End_Impulse_Vector((iw-1)*NSI+sample,tag) = PG(tag)*((4*roll_off)/(pi*sqrt(tp)))*
            *(cos((1+roll_off)*pi*T)+(1/(4*roll_off*T))*sin((1-roll_off)*pi*T))*(1/(1-(4*roll_off*T)^2));
            if(isinf(End_Impulse_Vector((iw-1)*NSI+sample,tag)))
                End_Impulse_Vector((iw-1)*NSI+sample,tag) = 0;
            end
            if(isnan(End_Impulse_Vector((iw-1)*NSI+sample,tag)))
                End_Impulse_Vector((iw-1)*NSI+sample,tag) = PG(tag)*(4*roll_off)/(pi*sqrt(tp))+(1-roll_off)/sqrt(tp);
            end
            End_Impulse_RF_Vector((iw-1)*NSI+sample,tag) = End_Impulse_Vector((iw-1)*NSI+sample,tag)*
            *(cos(2*pi*fc*(t_impulse((iw-1)*NSI+sample)-t0(tag))+gamma(tag));
        end
    end
end
%[...]
```

Figure 4.5: Matlab code for the generation of the pulse received by the SELECT reader

With reference to the structure of a single finger of the UWB front end LORELEI of figure 3.3, its emulation must avoid the introduction of non-idealities. To do this, the emulation of the various sections of the UWB front end LORELEI involves:

- Emulation of low noise amplification section: The emulation of that section involves the amplification, with a gain equal to G_{LNA}, of the input signal without introducing noise.
  Therefore, given as input the following signal $s_{UWB}(t)$:

  $$s_{UWB}(t) = PG(d)p(t)cos\left(2\pi f_c t + \gamma\right)$$

  in output is obtained the following signal $s_{LNA}(t)$:

  $$s_{LNA}(t) = G_{LNA}s_{UWB}(t) = G_{LNA}PG(d)p(t)cos\left(2\pi f_c t + \gamma\right)$$

- Emulation of down conversion section: The emulation of that section involves, for debug purposes, a perfect locking of the phase of the input signal carrier f_c, a mixing with two carriers at frequency f_c in perfect quadrature and a perfect low-pass filtering which eliminates completely the high frequency component and leave unchanged the baseband component.
  Therefore, given as input the signal $s_{LNA}(t)$ coming from the low noise amplification section, in output are obtained the following signals $I(t)$ and $Q(t)$:

  $$\begin{cases} I(t) = & s_{LNA}(t)cos\left(2\pi f_c t\right)\Big|_{Passa-Basso} = G_{LNA}PG(d)p(t)cos\left(2\pi f_c t_0\right) \\ Q(t) = & s_{LNA}(t)sin\left(2\pi f_c t\right)\Big|_{Passa-Basso} = -G_{LNA}PG(d)p(t)sin\left(2\pi f_c t_0\right) \end{cases}$$

- Emulation of section related to the projection to intermediate frequency: The emulation of that section involves the mixing of each input signal with two carriers at intermediate frequency f_i in perfect quadrature.
  Therefore, given as input the signals $I(t)$ and $Q(t)$ coming from the down conversion section, in output are obtained the following signals $II(t)$, $IQ(t)$, $QI(t)$ and $QQ(t)$:

  $$\begin{cases} II(t) = I(t)cos\left(2\pi f_i t\right) = G_{LNA}PG(d)p(t)cos\left(2\pi f_c t_0\right)cos\left(2\pi f_i t\right) \\ IQ(t) = I(t)sin\left(2\pi f_i t\right) = G_{LNA}PG(d)p(t)cos\left(2\pi f_c t_0\right)sin\left(2\pi f_i t\right) \\ QI(t) = Q(t)cos\left(2\pi f_i t\right) = -G_{LNA}PG(d)p(t)sin\left(2\pi f_c t_0\right)cos\left(2\pi f_i t\right) \\ QQ(t) = Q(t)sin\left(2\pi f_i t\right) = -G_{LNA}PG(d)p(t)sin\left(2\pi f_c t_0\right)sin\left(2\pi f_i t\right) \end{cases}$$

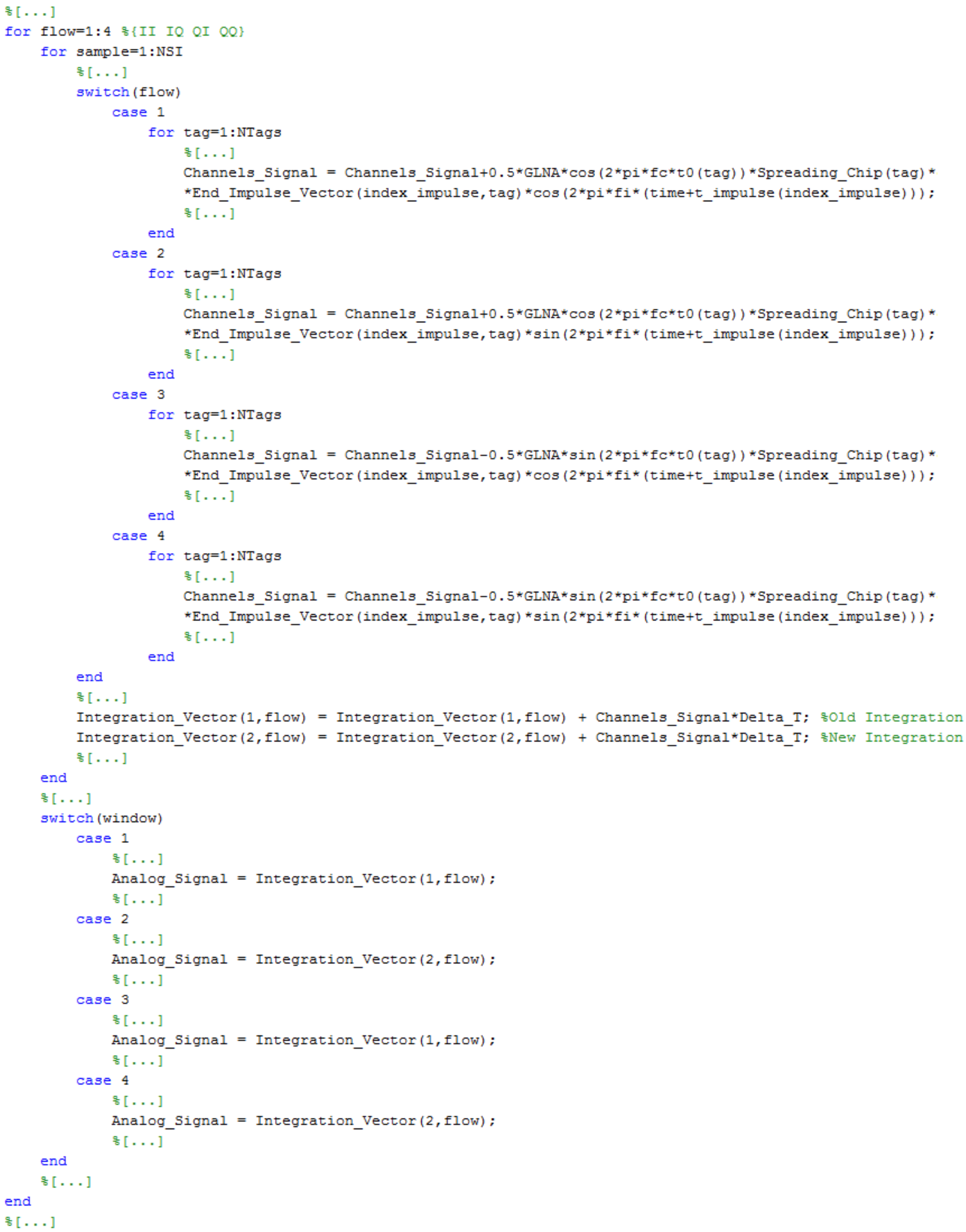
- Emulation of integration section: The emulation of this section involves the integration of each input signal on rectangular windows of width $T = \frac{1}{f_i}$ overlapping by 50%.
  Therefore, given as input the signals $II(t)$, $IQ(t)$, $QI(t)$ and $QQ(t)$ coming from the section of projection to intermediate frequency, in output are obtained the following signals $II^{(int)}(t)$, $IQ^{(int)}(t)$, $QI^{(int)}(t)$ and $QQ^{(int)}(t)$:

$$\begin{cases} II^{(int)}(t) = \int_0^T II(t)\,dt = \int_0^T G_{LNA}PG(d)p(t)cos\,(2\pi f_c t_0)\,cos\,(2\pi f_i t)\;dt \\ IQ^{(int)}(t) = \int_0^T IQ(t)\,dt = \int_0^T G_{LNA}PG(d)p(t)cos\,(2\pi f_c t_0)\,sin\,(2\pi f_i t)\;dt \\ QI^{(int)}(t) = \int_0^T QI(t)\,dt = -\int_0^T G_{LNA}PG(d)p(t)sin\,(2\pi f_c t_0)\,cos\,(2\pi f_i t)\;dt \\ QQ^{(int)}(t) = \int_0^T QQ(t)\,dt = -\int_0^T G_{LNA}PG(d)p(t)sin\,(2\pi f_c t_0)\,sin\,(2\pi f_i t)\;dt \end{cases}$$

- Emulation of A/D conversion section: The emulation of this section involves the A/D conversion of the input signals using 5 bits ADCs with automatic gain control to maximize the input dynamics.

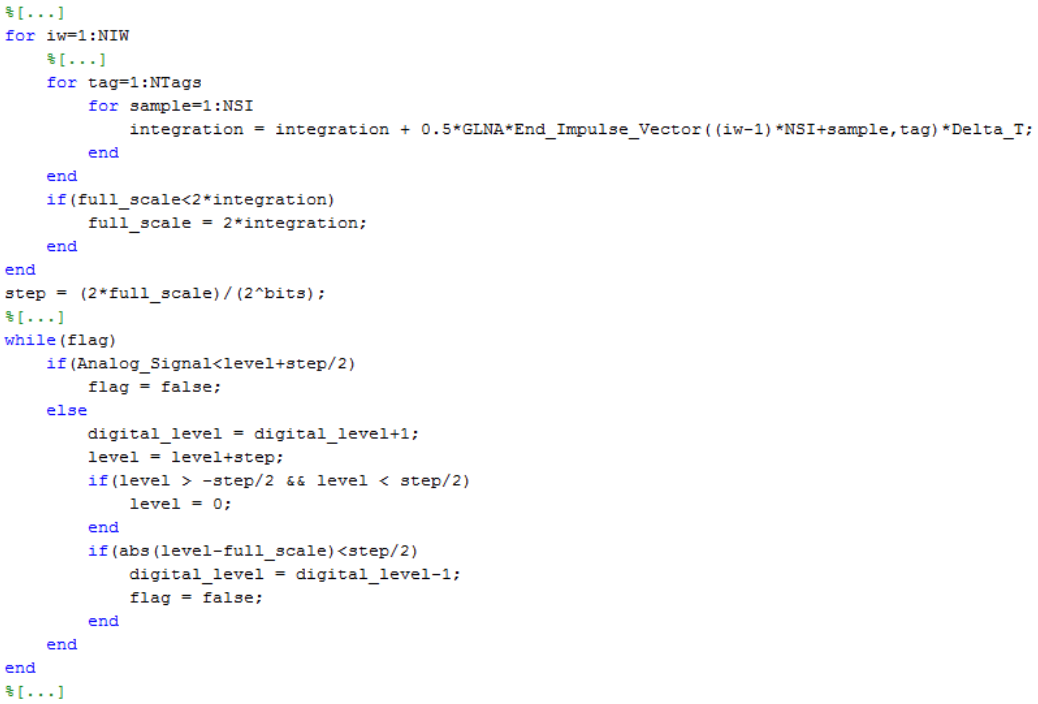
The Matlab code relating to the analog processing section of the UWB front end LORELEI is shown in figure 4.6, in which it has to be noted that this section also implements the spreading of the UWB signal with the spreading codes of the SELECT reader and the SELECT tags. In particular, since the input file must be used by implementations that consider only the preamble of the response signal, the data symbols, having all be put to 1, does not appear explicitly inasmuch they were implied.

The Matlab code relating to the section of A/D conversion of the UWB front end LORELEI is shown in figure 4.7, in which it has to be noted also the implementation of the automatic gain control, where arises the full scale equal to the maximum amplitude of the input signal to the ADCs.



```
%[...]
for flow=1:4 %{II IQ QI QQ}
    for sample=1:NSI
        %[...]
        switch(flow)
            case 1
                for tag=1:NTags
                    %[...]
                    Channels_Signal = Channels_Signal+0.5*GLNA*cos(2*pi*fc*t0(tag))*Spreading_Chip(tag)*
                    *End_Impulse_Vector(index_impulse,tag)*cos(2*pi*fi*(time+t_impulse(index_impulse)));
                    %[...]
                end
            case 2
                for tag=1:NTags
                    %[...]
                    Channels_Signal = Channels_Signal+0.5*GLNA*cos(2*pi*fc*t0(tag))*Spreading_Chip(tag)*
                    *End_Impulse_Vector(index_impulse,tag)*sin(2*pi*fi*(time+t_impulse(index_impulse)));
                    %[...]
                end
            case 3
                for tag=1:NTags
                    %[...]
                    Channels_Signal = Channels_Signal-0.5*GLNA*sin(2*pi*fc*t0(tag))*Spreading_Chip(tag)*
                    *End_Impulse_Vector(index_impulse,tag)*cos(2*pi*fi*(time+t_impulse(index_impulse)));
                    %[...]
                end
            case 4
                for tag=1:NTags
                    %[...]
                    Channels_Signal = Channels_Signal-0.5*GLNA*sin(2*pi*fc*t0(tag))*Spreading_Chip(tag)*
                    *End_Impulse_Vector(index_impulse,tag)*sin(2*pi*fi*(time+t_impulse(index_impulse)));
                    %[...]
                end
        end
        %[...]
        Integration_Vector(1,flow) = Integration_Vector(1,flow) + Channels_Signal*Delta_T; %Old Integration
        Integration_Vector(2,flow) = Integration_Vector(2,flow) + Channels_Signal*Delta_T; %New Integration
        %[...]
    end
    %[...]
    switch(window)
        case 1
            %[...]
            Analog_Signal = Integration_Vector(1,flow);
            %[...]
        case 2
            %[...]
            Analog_Signal = Integration_Vector(2,flow);
            %[...]
        case 3
            %[...]
            Analog_Signal = Integration_Vector(1,flow);
            %[...]
        case 4
            %[...]
            Analog_Signal = Integration_Vector(2,flow);
            %[...]
    end
    %[...]
end
%[...]
```

Figure 4.6: Matlab code for the emulation of the analog processing section of the UWB front end LORELEI



```
%[...]
for iw=1:NIW
    %[...]
    for tag=1:NTags
        for sample=1:NSI
            integration = integration + 0.5*GLNA*End_Impulse_Vector((iw-1)*NSI+sample,tag)*Delta_T;
        end
    end
    if(full_scale<2*integration)
        full_scale = 2*integration;
    end
end
step = (2*full_scale)/(2^bits);
%[...]
while(flag)
    if(Analog_Signal<level+step/2)
        flag = false;
    else
        digital_level = digital_level+1;
        level = level+step;
        if(level > -step/2 && level < step/2)
            level = 0;
        end
        if(abs(level-full_scale)<step/2)
            digital_level = digital_level-1;
            flag = false;
        end
    end
end
%[...]
```

Figure 4.7: Matlab code for the emulation of the A/D conversion section of the UWB front end LORELEI

4.2 Application Example of the Input File Generator for Baseband SELECT Reader Receiver

As an application example of `Input_File_Generator` we consider a scenario composed of a SELECT reader and 2 SELECT tags, characterized by a distance from the first equal to:

$$Tag\, n.1:\ d_1 = 5.25m$$

$$Tag\, n.2:\ d_2 = 5.95m$$

The interrogation signal is composed of a data symbol $b_n = 1$ which undergoes the spreading operation by spreading codes characterized by a number of chips $N_c = 32$, to each of which are associated a number of pulses $N_{pc} = 2$.

The temporal structure of a single PRP in transmission is that shown in figure 4.8, which is composed of a root raised cosine pulse characterized by:

Parameter	Value
β	0.6
BW	$1.6GHz$
t_p	$1ns$
f_c	$3.95GHz$
α	$0rad\,(0°)$
$Position\,in\,PRP$	$20ns$

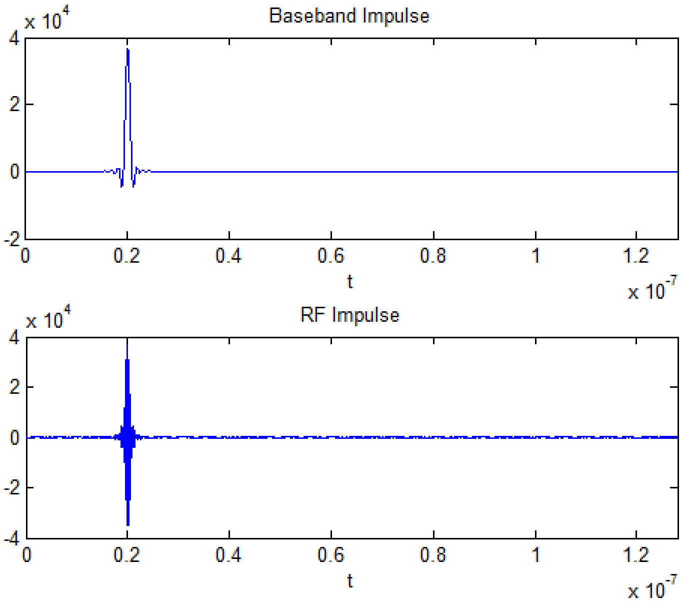


Figure 4.8: Single PRP in transmission

Given the scenario taken into consideration, the temporal structure of a single PRP in reception is that shown in figure 4.9, which is composed of two root raised cosine pulses characterized by:

Parameter	Value
PG	$\begin{cases} 9\cdot 10^{-3} & Tag\,n.1 \\ 7\cdot 10^{-3} & Tag\,n.2 \end{cases}$
γ	$\begin{cases} 2.2rad\,(124.5°) & Tag\,n.1 \\ 5.0rad\,(285.0°) & Tag\,n.2 \end{cases}$
$Position\,in\,PRP$	$\begin{cases} 55.02ns & Tag\,n.1 \\ 59.69ns & Tag\,n.2 \end{cases}$

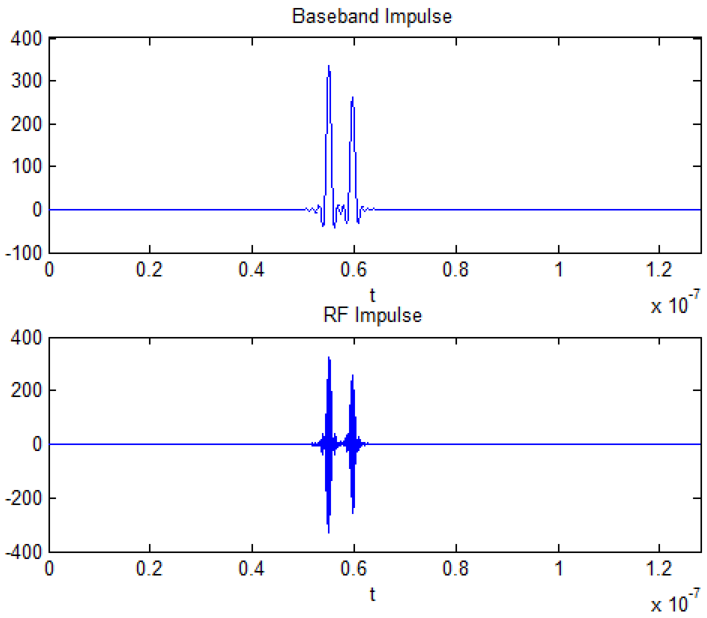


Figure 4.9: Single PRP in reception

The input file relating to that scenario is given by digitization of the signals of figure 4.10, each of which is composed of $N_c \cdot N_{pc} = 64\, PRP$. The temporal structure of each of that PRP is shown in figure 4.11, which is obtained by projecting to intermediate frequency, integration and quantization of the temporal structure of figure 4.9.

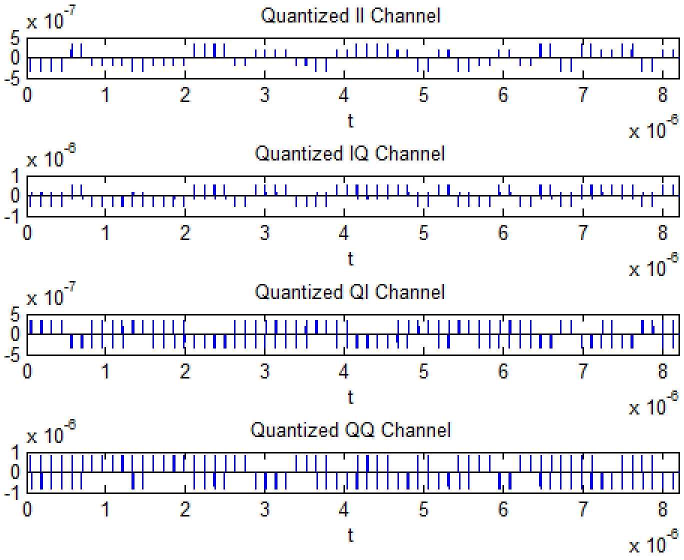


Figure 4.10: Total Signal to be digitized

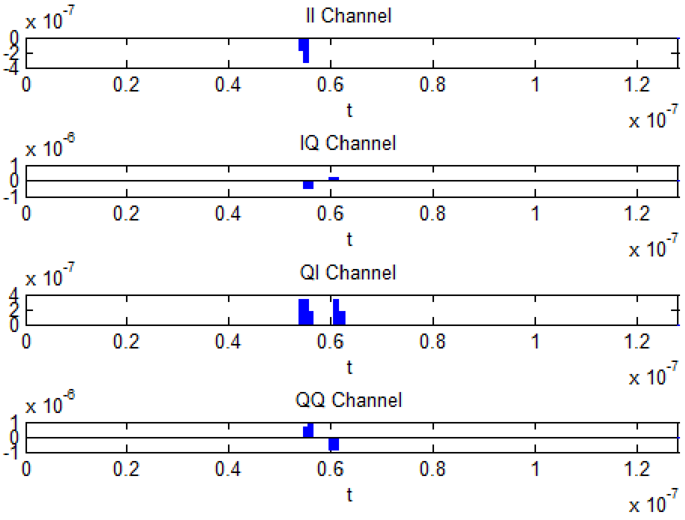


Figure 4.11: Single PRP to be digitized

Putting that input file in input to `Baseband_Receiver_Simulator`, the expected results are:

Parameter	Value
$Coarse\,TOA$	$\begin{cases} 55 \quad Tag\,n.1 \\ 60 \quad Tag\,n.2 \end{cases}$
$Fine\,TOA$	$\begin{cases} 0.076rad\,(4.361°) \qquad Tag\,n.1 \\ -0.961rad\,(-55.057°) \quad Tag\,n.2 \end{cases}$
$Portion\,in\,Integration\,Window$	$\begin{cases} 9th \quad Tag\,n.1 \\ 4th \quad Tag\,n.2 \end{cases}$
$Arctangent\;Approximation$	$\begin{cases} 0.098rad\,(5.625°) \qquad Tag\,n.1 \\ -0.884rad\,(-50.625°) \quad Tag\,n.2 \end{cases}$

The actual results obtained by `Baseband_Receiver_Simulator` in response to the example input file are:

Parameter	Value
$Coarse\,TOA$	$\begin{cases} 55 & Tag\,n.1 \\ 60 & Tag\,n.2 \end{cases}$
$Fine\,TOA$	$\begin{cases} 0.000rad\,(0.000°) & Tag\,n.1 \\ -0.960rad\,(-55.008°) & Tag\,n.2 \end{cases}$
$Portion\,in\,Integration\,Window$	$\begin{cases} 9th & Tag\,n.1 \\ 4th & Tag\,n.2 \end{cases}$
$Arctangent\ Approximation$	$\begin{cases} 0.098rad\,(5.625°) & Tag\,n.1 \\ -0.884rad\,(-50.625°) & Tag\,n.2 \end{cases}$

From the comparison of the obtained results with the expected results can be seen that, in this case, there is an almost perfect correspondence, i.e. `Baseband_Receiver_Simulator` provides results completely in agreement with those expected.

4.3 Conclusions and Open Issues

The application of the input file generated by `Input_File_Generator` shows that, in general, the implementation of `Baseband_Receiver_Simulator` and of the FPGA firmware are correct.

However, in many cases it happens that the obtained results related to the coarse TOA estimation differ from those expected, although always for a single integration window.

From a first analysis of possible causes, it would seem that the excessive quantization error introduced by the 5 bits ADCs of the UWB front end LORELEI lead to a significant distortion of the received UWB signal, which, in turn, may lead to an erroneous coarse TOA estimation.

This problem still requires further studies in order to determine the exact causes. Its resolution is desirable inasmuch the non certainty on the estimation of the received pulses integration window makes useless the entire process of fine TOA estimation.

Chapter 5

Preliminary Studies for Functional Improvement of the SELECT Reader Receiver Implementation for Debugging Purposes

In order to make the debugging of a SELECT reader receiver fully reliable, it is necessary that a generic remote user can calculate the expected results in real time to make possible a direct and immediate verification of the obtained results.

To do this, it is therefore necessary that the implementative hardware of a SELECT reader receiver allows to the generic remote user, which could be for example `Baseband_Receiver_Simulator`, to acquire in real time the same information flow processed by the device, i.e. the information flow coming out of the UWB front end LORELEI.

To obtain reliable results at the remote user side, it is necessary that the information flow acquired by that user is identical, from the point of view of the information content, to that processed by the SELECT reader receiver.

Thus, it must be ensured that there is no loss of useful information during the transfer of the information flow from the UWB front end LORELEI to the remote user.

The implementative hardware of the SELECT reader receiver was designed by Fraunhofer and it is composed of three main sections (see figure 5.1):

- Section of communication with the SELECT central unit: This section performs the communication with the SELECT central unit and the transmission of the acquired data. It is substantially composed by a microprocessor.

- Section of processing: This section performs the processing of data received from the UWB front end LORELEI. It is substantially composed by a FPGA.
- Section of remote communication: This section performs the sending of the data received from the UWB front end LORELEI to a remote user. It is substantially composed by a FPGA and an EMIF interface.

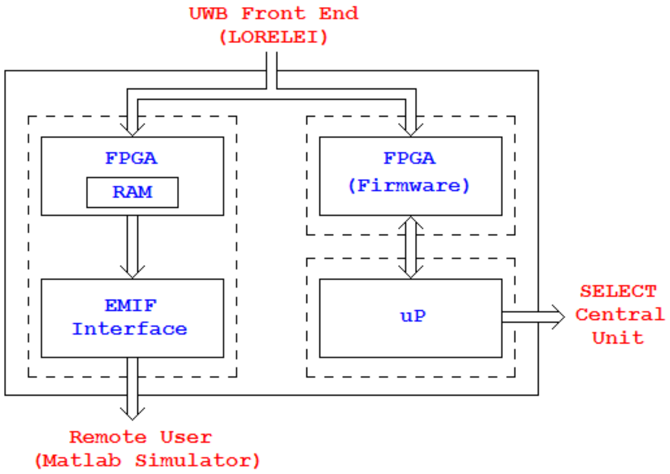


Figure 5.1: Implementative hardware of a SELECT reader receiver

Sections set forth above, the first two are used to implement parts strictly necessary for the receiver of a SELECT reader, while the last is used to allow the transfer of information flow from the UWB front end LORELEI to the remote user for debugging purposes.

In order to the correct transfer of this information flow, it is required a buffering sufficient to compensate the time delay introduced by the EMIF interface during transmission to the remote user.

For properly satisfy the debugging purposes, the information flow to be transferred in real time to the remote user must be equal at least to an entire data symbol.

Therefore, since the EMIF interface can not guarantee sufficiently high transmission rate and especially can not guarantee the constancy of that transmission, it is necessary to buffer at least the information flow relating to an entire data symbol.

Therefore, considering the communication parameters in the table 2.1, the size of the information flow relating to an entire data symbol is equal to:

$$N_c \cdot N_{pc} \cdot N_{sample} \cdot N_{bit} \cdot N_{channel} = 2.62MB$$

with:

- $N_c = 1024$: Number of chips per spreading code.
- $N_{pc} = 8$: Number of pulses per chip of spreading code.
- $N_{sample} = 128$: Number of samples per PRP.
- $N_{bit} = 5$: Number of quantization bits.
- $N_{channel} = 4$: Number of channels (II, IQ, QI and QQ).

The implementative hardware of a SELECT reader receiver does not have an external RAM memory, and thus the buffering must necessarily take place in the internal RAM memory to the FPGA, which is composed by 268 blocks from $18Kb$ each. Therefore, the maximum available memory space is equal to:

$$268 \cdot 18Kb = 603KB$$

This memory space, for architecture reasons, it is almost never completely usable and therefore it is well to consider to have available a RAM memory of size equal to $500KB$.

Comparing the available RAM memory space and the size of the minimum information flow to perform the debugging phase, it is evident that the full buffering of this information flow is not possible.

Not being able to make hardware modifications on the SELECT reader receiver board, the only possible solution consists in data compression.

For this purpose, the minimum useful compression percentage CP is equal to:

$$CP = 1 - \frac{500KB}{2.62MB} = 0.81 = 81\%$$

which corresponds to a minimum useful compression factor CF equal to:

$$CF = \frac{2.62MB}{500KB} = 5.24$$

To improve the functional of a SELECT reader receiver implementation for debugging purposes, I performed a preliminary feasibility study aimed at understanding if it is really possible to get the required compression levels and, if it is possible, trying to conceive techniques that achieve this goal.

Furthermore, the development of these techniques must meet requirements of minimum expenditure of FPGA resources, both in terms of calculation and in terms of memory occupation.

5.1 Information Flow Compression By Lossless Techniques

Given the need to compress the information flow received from the UWB front end LORELEI without altering the information content, the more suitable compression techniques are those lossless, i.e. the compression techniques that eliminates the redundancy of information flow without altering its information content.

In the family of the lossless compression codes, the more efficient is the Huffman code, which, therefore, it is the ideal candidate to be used for compression of the information flow received from the UWB front end LORELEI.

The compression of a generic information flow through Huffman code involves two steps [11]:

1. Construction of a binary tree dependent by the probability distribution of the symbols to be encoded.

2. Construction of the codewords dictionary by reading the binary tree.

The construction of the binary tree involves three steps:

1. Descending order sorting of probability of the symbols to be encoded.

2. Union of the two lower probabilities in a single node, to which is associated the sum of these probabilities.

3. Repeat the previous steps until to depletion nodes.

Instead, the construction of the codewords dictionary involves two steps:

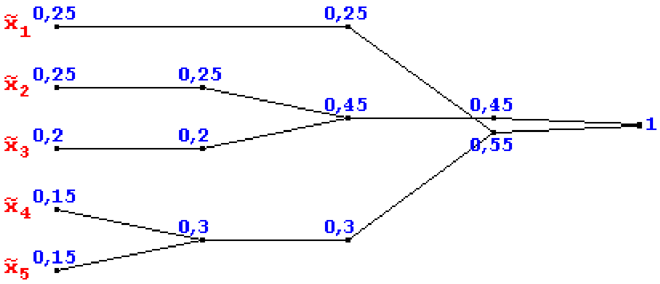
1. Attribution, in an arbitrary manner, of the binary values 0 and 1 at the bifurcations of the obtained binary tree.

2. Reading, from the root to each leaf, of the binary values of the various branches in order to creation the codewords to associate at each symbol to be encoded.

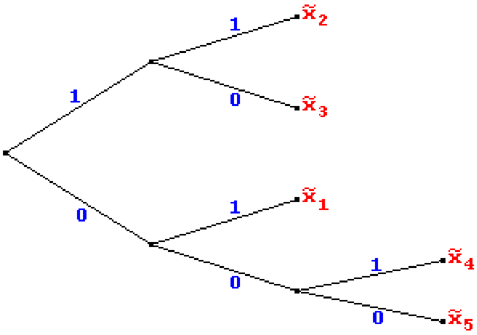
Example: Given an information flow composed by symbols $X \in \mathcal{X} = \{\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_3, \widetilde{x}_4, \widetilde{x}_5\}$ with probability distribution $p(x)$ given by:

Symbol	$p(x)$
$\widetilde{x}_1$	0.25
$\widetilde{x}_2$	0.25
$\widetilde{x}_3$	0.20
$\widetilde{x}_4$	0.15
$\widetilde{x}_5$	0.15

Applying the Huffman code algorithm, we get:



from which is deduced the binary tree:



Therefore, the codewords dictionary relating to the considered information flow is:

Symbol	Codeword
$\widetilde{x}_1$	01
$\widetilde{x}_2$	11
$\widetilde{x}_3$	10
$\widetilde{x}_4$	001
$\widetilde{x}_5$	000

whereby it is possible to perform the compression of that information flow associating to each of its symbols the corresponding codeword of that dictionary.

In order to determine the average compression level of the information flow received from the UWB front end LORELEI that can be achieved with Huffman encoding, I have developed a compressor in Matlab code based on that encoding and I apply it to the files provided by CEA-LETI.

The Matlab compressor with Huffman encoding is called `Huffman_Encoder` and, in place of the theoretical binary tree of the Huffman code algorithm, I created a matrix, called `Words_Matrix`, which has n columns, equal to the cardinality of the set of symbols to be encoded, and $n-1$ rows, equal to the number of the theoretical binary tree bifurcations. Since each symbols to be compressed is composed of 5 bits, the matrix `Words_Matrix` has size equal to 32×31.

Applying the Huffman code algorithm, in correspondence with each sum of probabilities is written a row of the matrix `Words_Matrix`, in which are assigned the binary values 0 and 1 to the symbols corresponding to each of the branches united by this sum. In particular, the binary value 1 is assigned to the symbols corresponding to the branch with greater probability, while the binary value 0 is assigned to the symbols corresponding to the branch with lower probability.

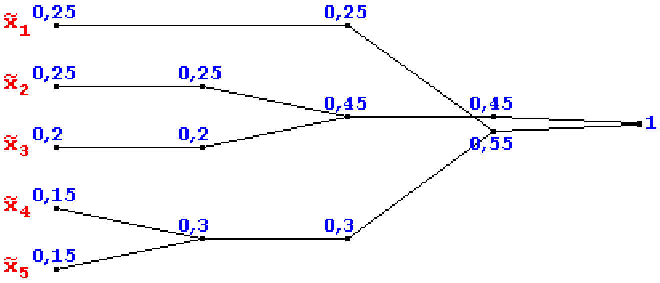
After the writing of the matrix `Words_Matrix`, the codewords dictionary is created by assigning to each symbol to be encoded the codeword that is obtained by reading from bottom to top the column corresponding to that symbol.

The Matlab code relative to the implementation of the Huffman code algorithm is shown in figure 5.2.

Example: Given an information flow composed by symbols $X \in \mathcal{X} = \{\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_3, \widetilde{x}_4, \widetilde{x}_5\}$ with probability distribution $p(x)$ given by:

Symbol	$p(x)$
$\widetilde{x}_1$	0.25
$\widetilde{x}_2$	0.25
$\widetilde{x}_3$	0.20
$\widetilde{x}_4$	0.15
$\widetilde{x}_5$	0.15

Applying the Huffman code algorithm, we get:

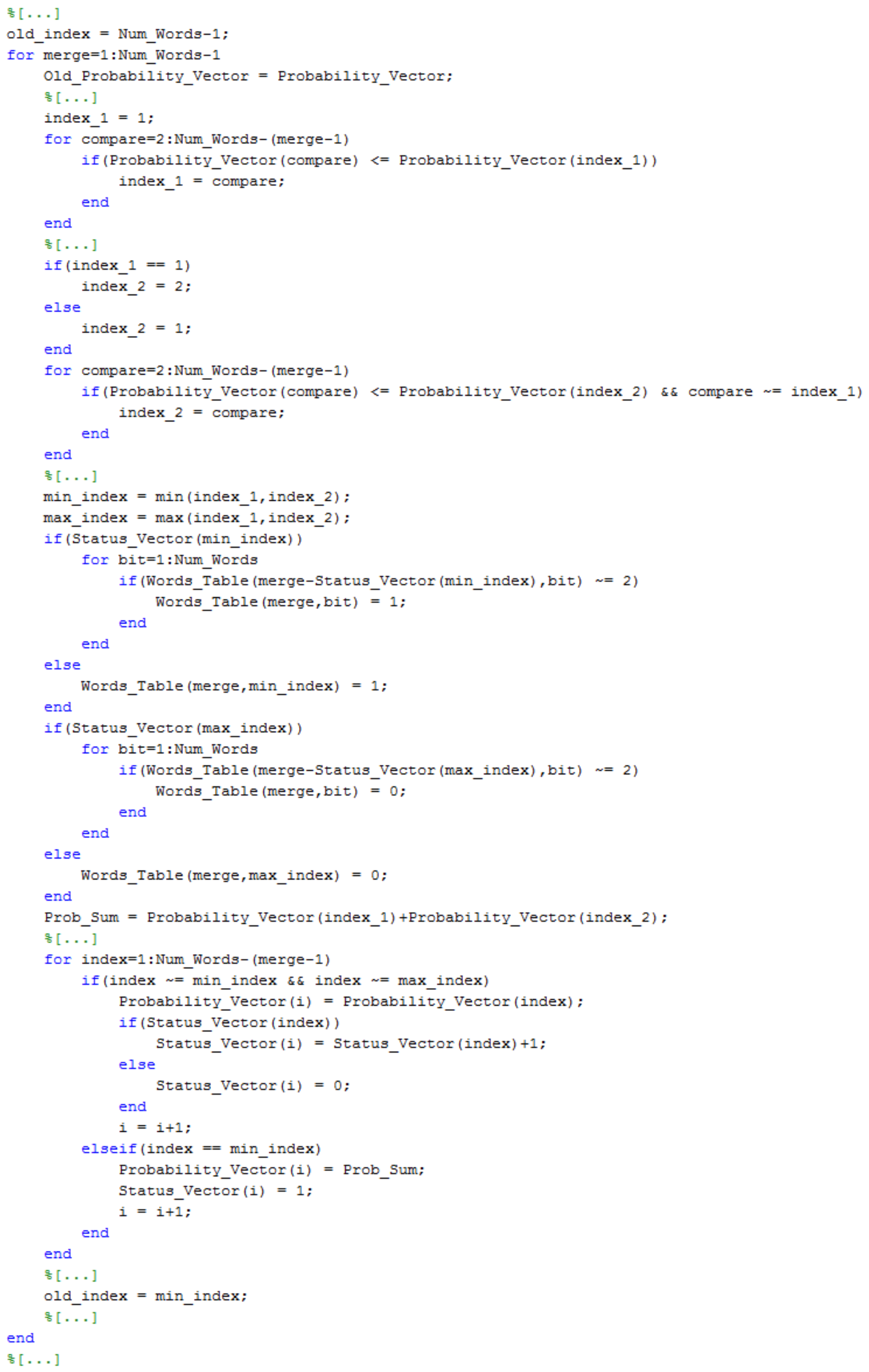


from which is deduced the matrix `Words_Matrix`:

	$\widetilde{x}_1$	$\widetilde{x}_2$	$\widetilde{x}_3$	$\widetilde{x}_4$	$\widetilde{x}_5$
1	×	×	×	1	0
2	×	1	0	×	×
3	1	×	×	0	0
4	0	1	1	0	0

Therefore, reading per columns the matrix `Words_Matrix` are obtained the codewords dictionary relating to the considered information flow:

Symbol	Codeword
$\widetilde{x}_1$	01
$\widetilde{x}_2$	11
$\widetilde{x}_3$	10
$\widetilde{x}_4$	001
$\widetilde{x}_5$	000



```
%[...]
old_index = Num_Words-1;
for merge=1:Num_Words-1
    Old_Probability_Vector = Probability_Vector;
    %[...]
    index_1 = 1;
    for compare=2:Num_Words-(merge-1)
        if(Probability_Vector(compare) <= Probability_Vector(index_1))
            index_1 = compare;
        end
    end
    %[...]
    if(index_1 == 1)
        index_2 = 2;
    else
        index_2 = 1;
    end
    for compare=2:Num_Words-(merge-1)
        if(Probability_Vector(compare) <= Probability_Vector(index_2) && compare ~= index_1)
            index_2 = compare;
        end
    end
    %[...]
    min_index = min(index_1,index_2);
    max_index = max(index_1,index_2);
    if(Status_Vector(min_index))
        for bit=1:Num_Words
            if(Words_Table(merge-Status_Vector(min_index),bit) ~= 2)
                Words_Table(merge,bit) = 1;
            end
        end
    else
        Words_Table(merge,min_index) = 1;
    end
    if(Status_Vector(max_index))
        for bit=1:Num_Words
            if(Words_Table(merge-Status_Vector(max_index),bit) ~= 2)
                Words_Table(merge,bit) = 0;
            end
        end
    else
        Words_Table(merge,max_index) = 0;
    end
    Prob_Sum = Probability_Vector(index_1)+Probability_Vector(index_2);
    %[...]
    for index=1:Num_Words-(merge-1)
        if(index ~= min_index && index ~= max_index)
            Probability_Vector(i) = Probability_Vector(index);
            if(Status_Vector(index))
                Status_Vector(i) = Status_Vector(index)+1;
            else
                Status_Vector(i) = 0;
            end
            i = i+1;
        elseif(index == min_index)
            Probability_Vector(i) = Prob_Sum;
            Status_Vector(i) = 1;
            i = i+1;
        end
    end
    %[...]
    old_index = min_index;
    %[...]
end
%[...]
```

Figure 5.2: Matlab code for the implementation of the Huffman code algorithm

The input files provided by CEA-LETI on which to perform the evaluation of the average compression level of the information flow received from the UWB front end LORELEI through Huffman encoding are 12:

- CODE_OFF_L5PAR_PXI_Scan_8EP4f_IG1_CI1024_131080.
- CODE_ONN_MP1800_L5PAR_PXI_Scan_8EP4f_IG1_CI1024_131080.
- SELECT_TEST_CABLE_OFF_L2PAR_PXI_Scan_8EP4f_IG1_CI1_136.
- SELECT_TEST_CABLE_OFF_L2PAR_PXI_Scan_8EP4f_IG1_CI16_2056.
- SELECT_TEST_CABLE_OFF_L2PAR_PXI_Scan_8EP4f_IG3_CI1_136.
- SELECT_TEST_CABLE_OFF_L5PAR_PXI_Scan_8EP4f_IG1_CI1_136.
- SELECT_TEST_CABLE_OFF_L5PAR_PXI_Scan_8EP4f_IG1_CI16_2056.
- SELECT_TEST_CABLE_ONN_L2PAR_PXI_Scan_8EP4f_IG1_CI1_136.
- SELECT_TEST_CABLE_ONN_L2PAR_PXI_Scan_8EP4f_IG1_CI16_2056.
- SELECT_TEST_CABLE_ONN_L2PAR_PXI_Scan_8EP4f_IG3_CI1_136.
- SELECT_TEST_CABLE_ONN_L5PAR_PXI_Scan_8EP4f_IG1_CI1_136.
- SELECT_TEST_CABLE_ONN_L5PAR_PXI_Scan_8EP4f_IG1_CI16_2056.

The first evaluation that I performed is to create a codewords dictionary based on one of the available input files and apply that statistics on each of these input files. The obtained results are shown in table 5.1, in which the red values correspond to the compression percentages relating to an input file compressed with his own statistics, while the black values correspond to the compression percentage relating to an input file compressed with the statistics of a different input file.

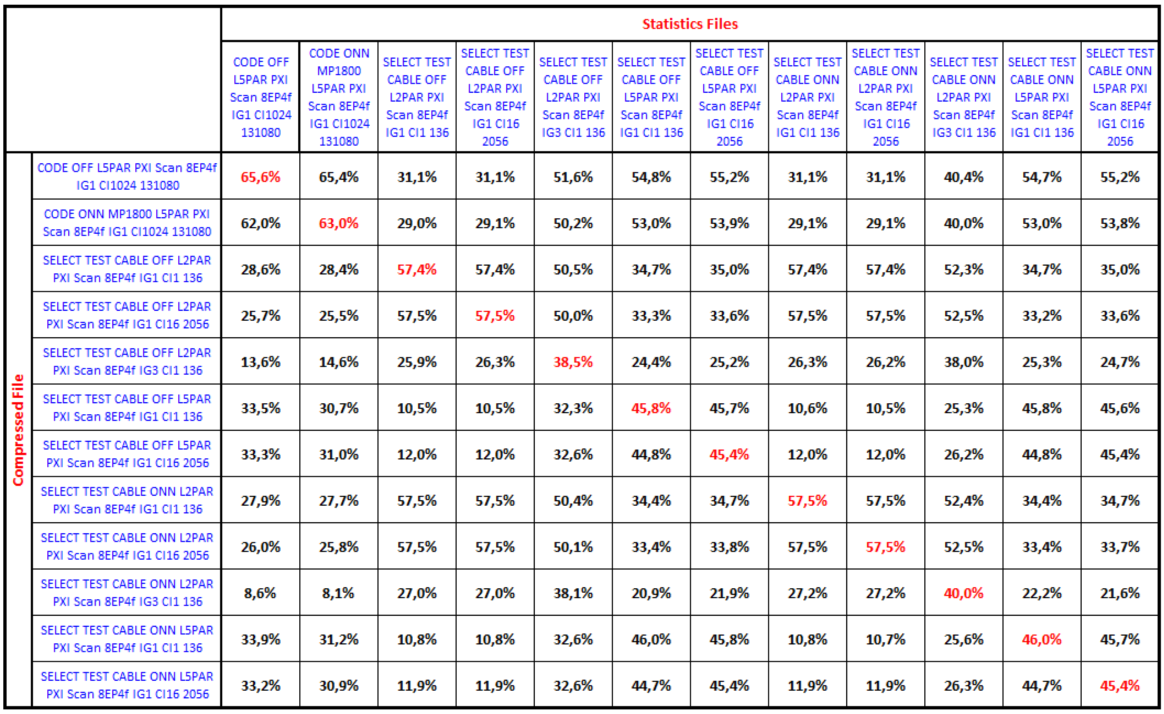
From these results I was able to conclude that:

- The compression levels relating to each input file compressed with his own statistics, which correspond to the highest compression level can be obtained with the Huffman encoding for the given input file, average around to:

$$CP = 50\%$$

$$CF = 2$$

- The compression levels relating to each input file compressed with the statistics of a different input file in some cases tend not to decrease excessively, while in other cases there is a more than appreciable decrement.



		Statistics Files											
		CODE OFF L5PAR PXI Scan 8EP4f IG1 CI1024 131080	CODE ONN MP1800 L5PAR PXI Scan 8EP4f IG1 CI1024 131080	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG1 CI1 136	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG1 CI16 2056	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG3 CI1 136	SELECT TEST CABLE OFF L5PAR PXI Scan 8EP4f IG1 CI1 136	SELECT TEST CABLE OFF L5PAR PXI Scan 8EP4f IG1 CI16 2056	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG1 CI1 136	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG1 CI16 2056	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG3 CI1 136	SELECT TEST CABLE ONN L5PAR PXI Scan 8EP4f IG1 CI1 136	SELECT TEST CABLE ONN L5PAR PXI Scan 8EP4f IG1 CI16 2056
Compressed File	CODE OFF L5PAR PXI Scan 8EP4f IG1 CI1024 131080	65,6%	65,4%	31,1%	31,1%	51,6%	54,8%	55,2%	31,1%	31,1%	40,4%	54,7%	55,2%
	CODE ONN MP1800 L5PAR PXI Scan 8EP4f IG1 CI1024 131080	62,0%	63,0%	29,0%	29,1%	50,2%	53,0%	53,9%	29,1%	29,1%	40,0%	53,0%	53,8%
	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG1 CI1 136	28,6%	28,4%	57,4%	57,4%	50,5%	34,7%	35,0%	57,4%	57,4%	52,3%	34,7%	35,0%
	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG1 CI16 2056	25,7%	25,5%	57,5%	57,5%	50,0%	33,3%	33,6%	57,5%	57,5%	52,5%	33,2%	33,6%
	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG3 CI1 136	13,6%	14,6%	25,9%	26,3%	38,5%	24,4%	25,2%	26,3%	26,2%	38,0%	25,3%	24,7%
	SELECT TEST CABLE OFF L5PAR PXI Scan 8EP4f IG1 CI1 136	33,5%	30,7%	10,5%	10,5%	32,3%	45,8%	45,7%	10,6%	10,5%	25,3%	45,8%	45,6%
	SELECT TEST CABLE OFF L5PAR PXI Scan 8EP4f IG1 CI16 2056	33,3%	31,0%	12,0%	12,0%	32,6%	44,8%	45,4%	12,0%	12,0%	26,2%	44,8%	45,4%
	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG1 CI1 136	27,9%	27,7%	57,5%	57,5%	50,4%	34,4%	34,7%	57,5%	57,5%	52,4%	34,4%	34,7%
	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG1 CI16 2056	26,0%	25,8%	57,5%	57,5%	50,1%	33,4%	33,8%	57,5%	57,5%	52,5%	33,4%	33,7%
	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG3 CI1 136	8,6%	8,1%	27,0%	27,0%	38,1%	20,9%	21,9%	27,2%	27,2%	40,0%	22,2%	21,6%
	SELECT TEST CABLE ONN L5PAR PXI Scan 8EP4f IG1 CI1 136	33,9%	31,2%	10,8%	10,8%	32,6%	46,0%	45,8%	10,8%	10,7%	25,6%	46,0%	45,7%
	SELECT TEST CABLE ONN L5PAR PXI Scan 8EP4f IG1 CI16 2056	33,2%	30,9%	11,9%	11,9%	32,6%	44,7%	45,4%	11,9%	11,9%	26,3%	44,7%	45,4%

Table 5.1: Compression percentage with crisscross single statistics

In order to apply the Huffman encoding on the FPGA, its implementation must require the minimum possible amount of resources.

The implementation applied by `Huffman_Encoder` ensures the highest level of compression with Huffman code but is very costly in terms of resources.

Therefore, from the point of view of the FPGA implementation, that solution does not represent the best trade-off between compression level and used resources.

One possible sub-optimal solution that may be suitable for implementation on FPGA is to create a codewords dictionary which represent the average statistics of a generic information flow received from the UWB front end LORELEI and then loaded into a lookup table that may be used by the FPGA in order to compress the information flow.

So that this solution can be considered optimal for a FPGA, it is necessary to verify that the compression levels obtained by this method are not excessively lower than the maximum performance obtainable with classic Huffman encoding.

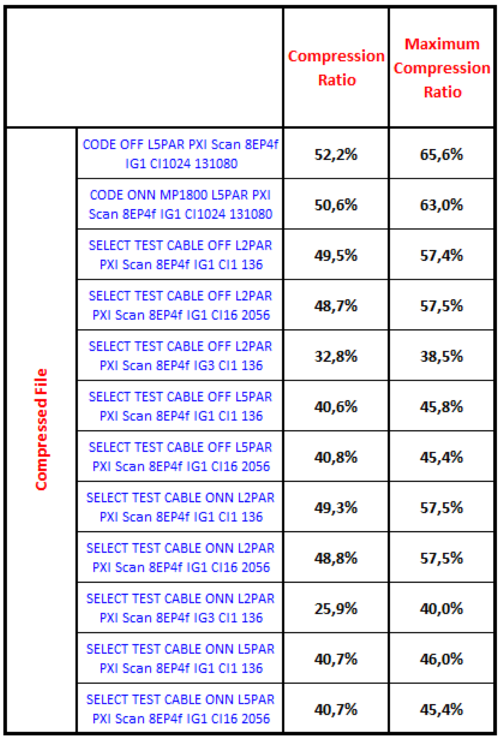
In order to evaluate the validity of the exposed sub-optimal solution, I have evaluated, by means of `Huffman_Encoder`, the compression level obtainable for the various available input files using a codewords dictionary based on the statistics relative to the complete set of these input files.

In this way it is possible to evaluate whether the compression levels of the various available input files do not decrease excessively using an average statistics and therefore whether it is possible the effective application of a solution based on an average statistics stored in a lookup table.

However, compressing each input file with the codewords dictionary based on the statistics of the complete set of that input files, the result would not be fully reliable, since, in this statistics is also present the one of the input file to be compressed.

Therefore, in order to obtain reliable results, the compression of each input file is performed with a codewords dictionary based on the statistics of the remaining input files.

The obtained results are shown in table 5.2, in which is shown, for each input file, the compression percentage obtained with the codewords dictionary based on the statistics of the remaining available input files and the maximum compression percentage of the input file itself.



		Compression Ratio	Maximum Compression Ratio
Compressed File	CODE OFF L5PAR PXI Scan 8EP4f IG1 CI1024 131080	52,2%	65,6%
	CODE ONN MP1800 L5PAR PXI Scan 8EP4f IG1 CI1024 131080	50,6%	63,0%
	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG1 CI1 136	49,5%	57,4%
	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG1 CI16 2056	48,7%	57,5%
	SELECT TEST CABLE OFF L2PAR PXI Scan 8EP4f IG3 CI1 136	32,8%	38,5%
	SELECT TEST CABLE OFF L5PAR PXI Scan 8EP4f IG1 CI1 136	40,6%	45,8%
	SELECT TEST CABLE OFF L5PAR PXI Scan 8EP4f IG1 CI16 2056	40,8%	45,4%
	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG1 CI1 136	49,3%	57,5%
	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG1 CI16 2056	48,8%	57,5%
	SELECT TEST CABLE ONN L2PAR PXI Scan 8EP4f IG3 CI1 136	25,9%	40,0%
	SELECT TEST CABLE ONN L5PAR PXI Scan 8EP4f IG1 CI1 136	40,7%	46,0%
	SELECT TEST CABLE ONN L5PAR PXI Scan 8EP4f IG1 CI16 2056	40,7%	45,4%

Table 5.2: Compression percentage with average statistics

From these results I was able to conclude that:

- The compression percentage relating to each compressed input file with the average statistics tend not to decrease excessively.

- Due to the good compression level achieved with the application of a codewords dictionary based on an average statistics, the implementation of a Huffman encoding based on lookup table seems to be the best trade-off between compression level and used resources for as regards the application on FPGA.

By means of `Huffman_Encoder`, a first statistics usable for the implementation of the sub-optimal Huffman encoding based on look-up table is given by the statistics of the complete set of input files provided by CEA-LETI (see figure 5.3). Given this statistics, the codewords dictionary to be loaded into the FPGA lookup table is therefore:

w_n	h_n	w_n	h_n
16	11	0	00000100011
15	01	7	00000100010
17	101	24	00000100001
14	100	27	00000100000
18	0011	6	00000001001
13	0010	25	00000001000
19	0001	26	00000000011
12	00001	31	00000000010
11	0000011	3	000000000011
20	00000101	4	000000000010
10	00000011	5	000000000001
9	00000010	28	0000000000001
21	000001001	1	00000000000001
23	000000011	2	000000000000001
22	000000001	29	0000000000000001
8	0000000101	30	0000000000000000

in which $w_n \in [0, 31]$ represents the symbols to be encoded, i.e. the words coming out the 4 ADCs of the UWB front end LORELEI, and h_n represents the corresponding codewords given by the sub-optimal Huffman encoding.

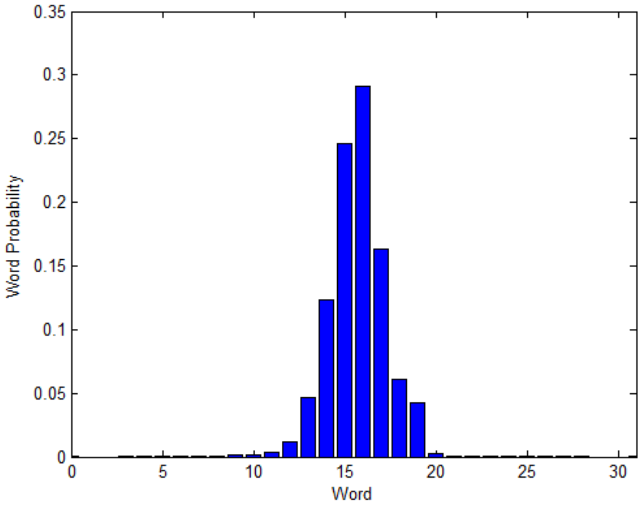


Figure 5.3: Average statistics

5.2 Information Flow Compression By Lossy Techniques

From the preliminary study with lossless compression techniques, it is deduced that the maximum obtainable compression factor is approximately equal to 2. Given the minimum acceptable compression factor required by buffering the information flow received from the UWB front end LORELEI, the compression factor achieved by lossless compression is therefore not sufficient.

Therefore, despite one of the constraints required by the compression techniques to be used in the solving of this problem is to preserve the information integrity of the information flow to be compressed, the insufficiency of the lossless compression techniques led me to consider also lossy compression techniques.

As previously mentioned, in general, if the information content of the information flow sent to the remote user is altered during compression, the results of the debug phase become unreliable, making useless the presence of the remote user himself.

The only way to apply lossy compression techniques to the information flow sent to the remote user without precluding the reliability of the debug phase results, is to identify and eliminate the unnecessary information content in the considered information flow, i.e. removing all or part of the information that it is not used neither by SELECT reader receiver nor by remote user.

That information content, in fact, from the theoretical point of view does not constitute redundancy, and therefore can not be eliminated by the lossless compression techniques, but from the practical point of view it is, since it is not in any way useful.

Since the results obtained from the SELECT reader receiver are based on the results of the despreading and accumulation processes, then, a given compressed information flow may provide the same final results, it is sufficient that this information flow give the same results from the despreading and accumulation processes.

Given the number N_{span} of correlators of the SELECT reader receiver and the step Δ of the despreading codes temporal shift, each of the N_{span} despreading codes is composed of the product between the spreading code of the SELECT reader itself and that of the desired SELECT tag suitably shifted in time of a temporal entity equal to a multiple of Δ PRPs (see figure 5.4). Therefore, the granularity characteristic of the temporal visibility of the despreading process is equal to Δ PRPs, because this process is performed on groups of Δ PRPs, i.e. on groups of Δ pulses.

Therefore, if for each group of Δ PRPs the average PRP is calculated and replaced it in each of these, the results of the despreading and accumulation processes remain unchanged (see figure 5.5).

From these considerations I have designed a lossy compression technique that can not preclude the reliability of the results of the debug phase. This technique consists in generating a compressed file by replacing the respective average PRP at each group of Δ PRPs. The remote user that receives the compressed file, before performing the despreading and accumulation processes, must perform the decompression of this file, which consists in the consecutively reproduction for Δ times of each average PRP content in the compressed file itself (see figure 5.6).

The compression levels characteristic of this lossy compression technique are therefore given by:

$$CP = 1 - \frac{1}{\Delta} \qquad CF = \Delta$$

From the preliminary study conducted by CNIT, the optimal values of N_{span} and Δ are equal to [8]:

$$N_{span} = 11 \qquad \Delta = 4$$

Therefore, from these optimal values, the compression levels are deduced of the exposed lossy compression technique, i.e.:

$$CP = 75\% \qquad CF = 4$$

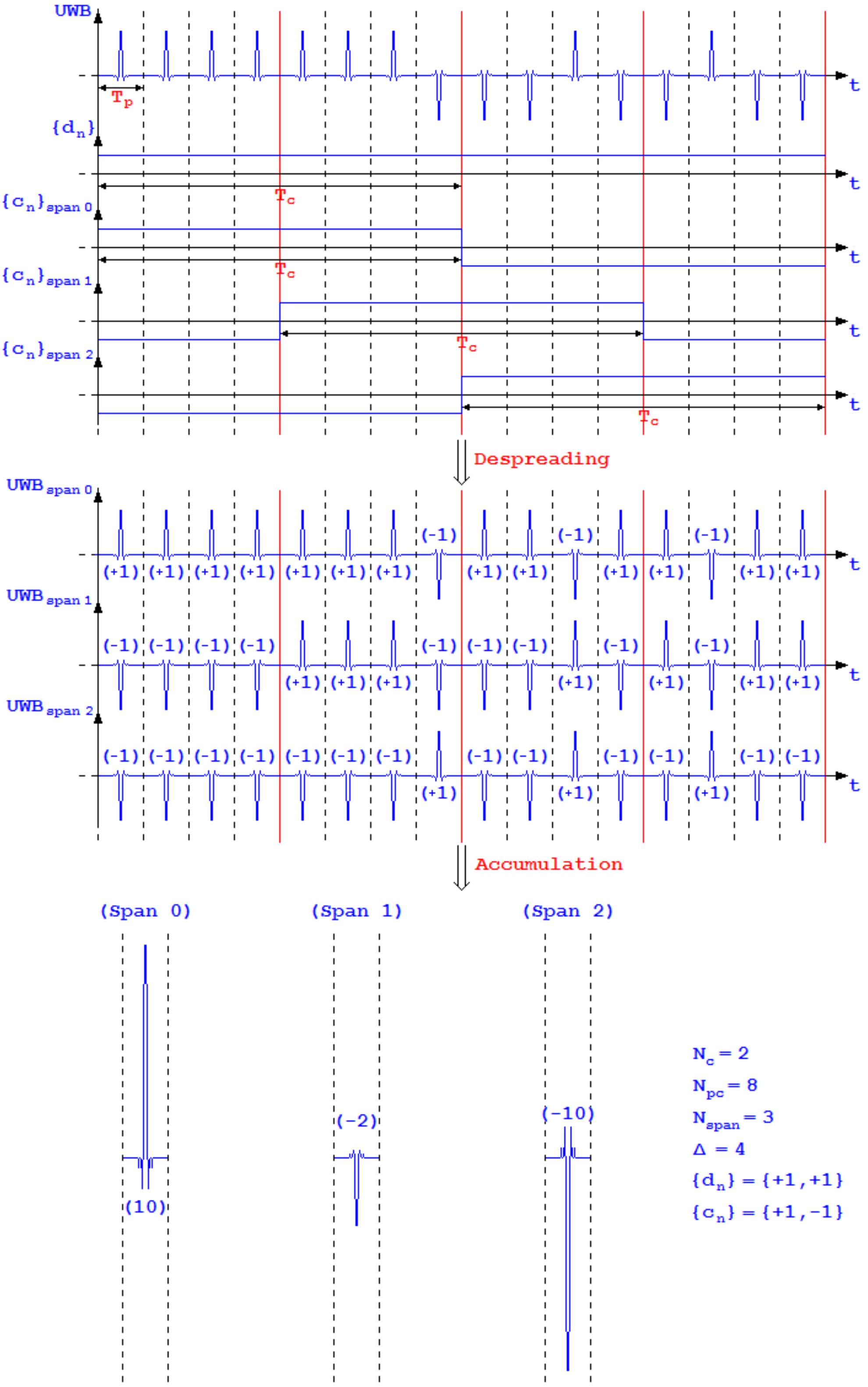


Figure 5.4: Despreading and accumulation process of a SELECT reader

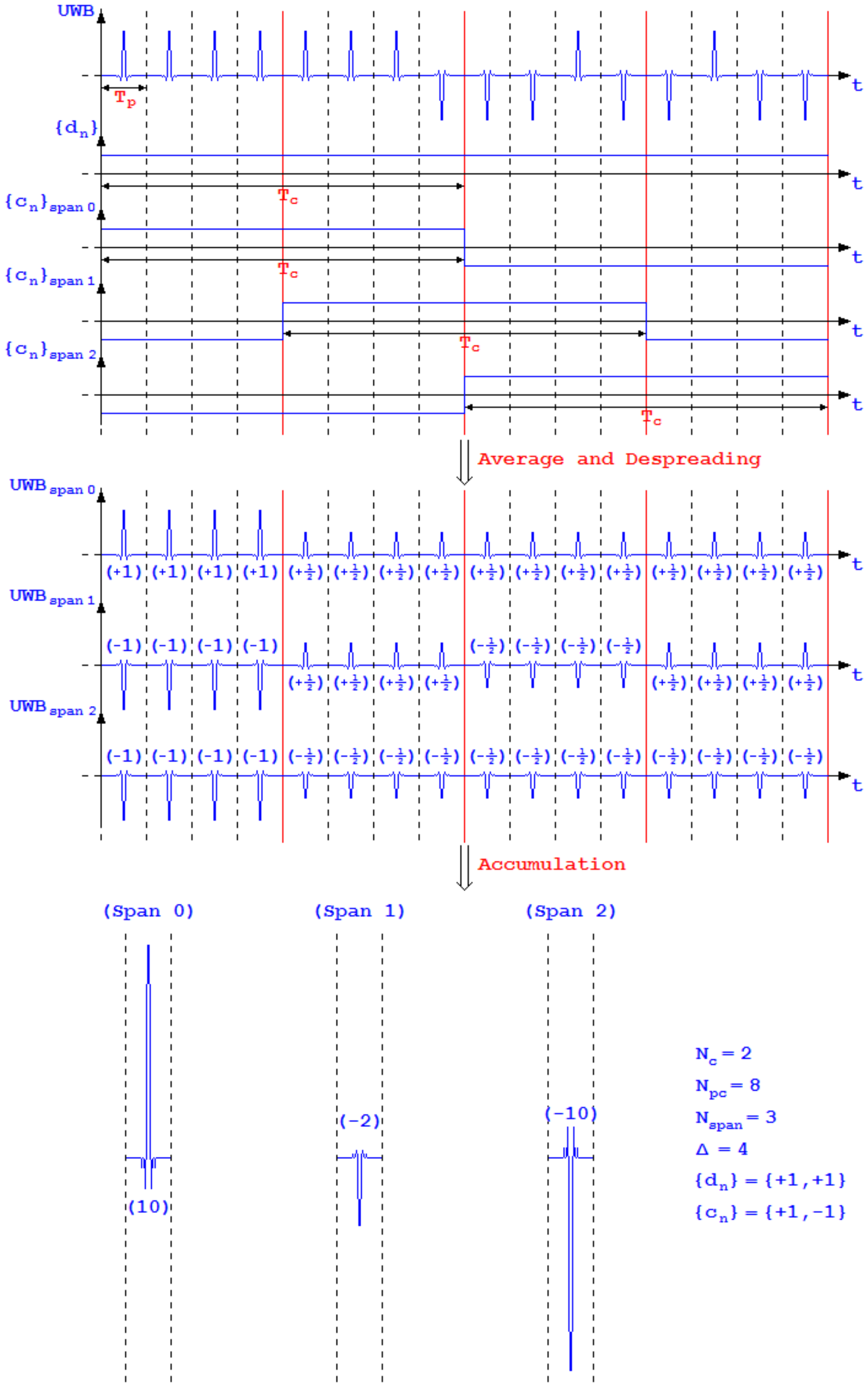


Figure 5.5: Despreading and accumulation process of a SELECT reader with average PRP

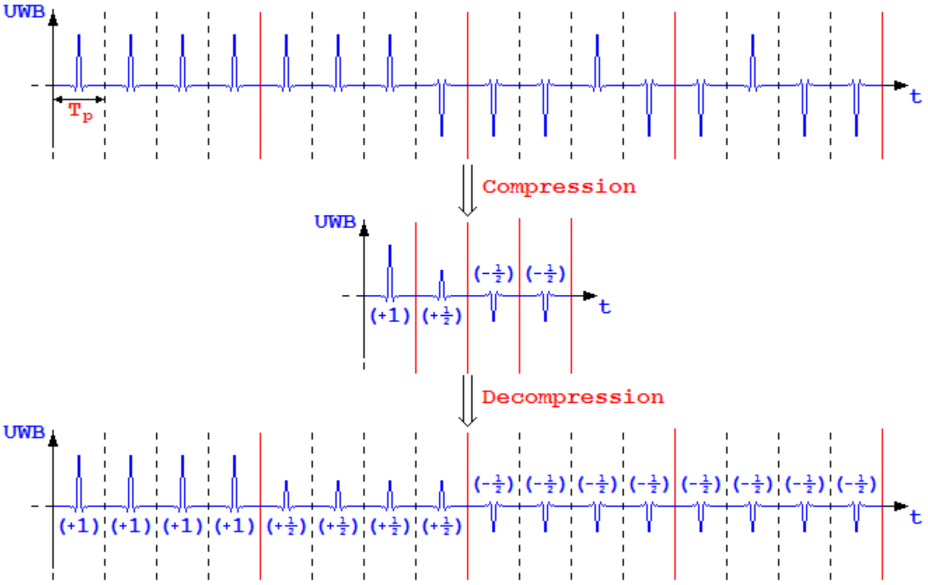


Figure 5.6: Lossy compression technique

5.3 Conclusions and Open Issues

Conclusions and Open Issues for the Lossless Compression Technique: The preliminary studies related to the lossless compression technique are based on a very small number of input files. Therefore, the obtained results may not be fully reliable. In order to improve these results, it is necessary to have other input files for the purpose to continue and enhance these preliminary studies and thus consolidate their results.

Conclusions and Open Issues for the Lossy Compression Technique: The exposed lossy compression technique is based on a purely theoretical concept, which is supposed to work but that is yet to be verified in practice.

Furthermore, so that this compression technique can operate correctly, it is necessary that the FPGA of the remote communication section and the FPGA of the processing section share the same time base. In theory, this problem should be solved easily since the two FPGAs, residing on the same board, can communicate among themselves and, therefore, the synchronization between these devices may take place by a simple communication protocol aimed at this purpose.

However, the resolution of the synchronization problem remains to be explored and resolved.

Conclusions and Open Issues for the Information Flow Compression: From preliminary studies it is clear that none of the developed compression techniques can, by itself, be able to meet the requirements for the resolution of the problem of buffering the information flow received from the UWB front end LORELEI, since the compression levels characteristic of the developed compression techniques are equal to:

$$Lossless\,Compression\,Technique = \begin{cases} CP_{loss-less} = 50\% \\ CF_{loss-less} = 2 \end{cases}$$

$$Lossy\,Compression\,Tecnique = \begin{cases} CP_{lossy} = 75\% \\ CF_{lossy} = 4 \end{cases}$$

However, a possible solution that would achieve the requirements involves the application of both compression techniques to the information flow to be compressed.

In particular, this solution involves to first apply the lossy compression technique and, subsequently, to apply the lossless compression technique.

In theory, the compression levels characteristic of this solution should be:

$$CP = 1 - \frac{1}{CF_{lossy} \cdot CF_{loss-less}} = 87.5\%$$

$$CF = CF_{lossy} \cdot CF_{loss-less} = 8$$

However, these values are yet to be verified, since the probability distribution characteristic of the information flow compressed by the lossy compression technique is different from that characteristic of the original information flow, even if qualitatively should be very similar to each other. Therefore, the performance of the lossless compression technique applied to the compressed information flow may also be very different from those relating to the application to the original information file.

Conclusions

The European project SELECT aims to realize a low-cost system architecture consisting of a network of fixed readers able to query one or more tags attached to objects.

Given the highly innovative content of this project, a complete and effective debug phase of the implemented system is fundamental to the success of the project.

Thus, this work was devoted to the debugging of the baseband receiving section of a SELECT reader, that is one of the devices of the SELECT system, resulting in the need to realize a set of three software tools that allow an efficient implementation of this test stage.

The first developed debugging tool consisted of a Matlab simulator of the baseband receiving section of a SELECT reader. This approach allowed to detect possible errors in the FPGA firmware developed by Datalogic S.p.A., the SELECT consortium coordinator. From the results obtained through this tool it has been possible to conclude that the implementation of the FPGA firmware developed by the SELECT partner is correct, i.e. it does not present any significant implementative error.

The second debugging tool, instead, consisted of an input file generator for the baseband receiving section of a SELECT reader. In particular, this software emulates the behavior of the whole transmission chain of the SELECT system in absence of non-idealities and noise, in order to control the expected results.

In this way, using these files as input for the previous described tool and FPGA firmware, it was possible to compare the two performance and verify the correct functioning of the firmware.

Thus, the output files were created neglecting implementation impairments. Debug results showed that in many cases small errors in the TOA estimate were present, helping Datalogic S.p.A. in their correction. This error, always equal to an integration window, is still unacceptable for the performance required by SELECT. However, from a preliminary analysis, it appeared that this difference in the TOA estimate is due to the reduced number of quantization bits adopted in LORELEI ADCs, that is 5 bits. This small number of bits may cause an high signal degradation, especially when a dynamic range is present at the ADCs parts.

Finally, the last developed debugging tool consisted of a preliminary study of techniques for the functional improving of the real-time debugging of the SELECT reader receiving section.

In particular, the aim of this preliminary study was the development of techniques for the compression of the data flow coming out from LORELEI to guarantee a proper real-time debug in presence of a reduced available memory space.

This preliminary study led to the development of two compression techniques, lossless and lossy, both applicable to the considered system.

The two proposed compression techniques showed a good data compression, but they could not be adopted for the real-time debugging due to the still not enough memory space. However, a possible compromise could be the application of both compression techniques at the same time.

The development of such software tools allowed a complete and reliable debugging of the baseband receiving section of a SELECT reader. However, the tests were conducted in ideal conditions, as the FPGA firmware and the real hardware requires time for their set-up, and at this moment they are still at a beginning phase. For what concerns this work, the tools developed can be simply adopted to fully test, the FPGA firmware also considering real working conditions, requiring a minimum effort.

Bibliografia

[1] www.selectwireless.eu

[2] G. Micheletti, M. Bottazzi. *Small or Medium-Scale Focused Research Project (STREP) Proposal ICT Call 5.* SELECT Project, 26 Nov 2009.

[3] F. Fuschini. *Sistemi di Radio Comunicazione per Automatic Data Collection.* Course of Tecniche Elettromagnetiche per la Localizzazione ed il Controllo Ambientale, University of Bologna at Cesena, 2012.

[4] Centre Tecnològic de Telecomunicacions de Catalunya - CTTC. *Ultra Wideband Technology for High Precision Ranging and Location.*

[5] D. Dardari. *Technologies for Wireless Sensor Networks.* Course of Reti di Sensori Wireless per Monitoraggio Ambientale LM, University of Bologna at Cesena, 2012.

[6] A. Malik. *RTLS for Dummies.* Wiley, 2009.

[7] D. Dardari et Al. *Signal Processing Techniques: Interim Report.* SELECT Deliverable 2.2.1, 13 Aug 2011.

[8] François Dehmas et Al. *Signal Processing Techniques: Final Report.* SELECT Deliverable 2.2.2, 29 Jun 2011.

[9] A. Sibille et Al. *Multi-Functional Network Design: Intermediate System Specification.* SELECT Deliverable 2.3.1, 25 Oct 2011.

[10] D. Dardari et Al. *Detection Algorithms.* SELECT Report, 19 Mar 2012.

[11] L. Calandrino, M. Chiani. *Lezioni di Comunicazioni Elettriche.* Pitagora Editrice Bologna, 2006.

[12] K. Finkenzeller. *RFID Handbook.* Wiley, 2003.

[13] R. Verdone, D. Dardari, G. Mazzini, A. Conti. *Wireless Sensor and Actuator Networks.* Elsevier Academic Press, 2008.

[14] A. Goldsmith. *Wireless Communications.* Cambridge University Press, 2005.

[15] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, M. Z. Win. *Ranging With Ultrawide Bandwidth Signals in Multipath Environments.* Proceedings of the IEEE, Vol.97, n.2, Feb 2009.

[16] L. Pappalardo. *Localizzazione.* Informatics department - University of Pisa, 2011.

[17] D. Dardari, E. Falletti, M. Luise. *Satellite and Terrestrial Radio Positioning Techniques - A Signal Processing Perspective.* Elsevier Ltd, London, 2011.

[18] D. Dardari, F. Guidi. *The Novel Concept of Passive Ultrawide Bandwidth RFID for the Internet of Things.* WiLAB (DEIS), University of Bologna, 30 Nov 2011.

[19] D. Dardari, F. Guidi, C. Roblin, A. Sibille. *Ultra-Wide Bandwidth Backscatter Modulation: Processing Schemes and Performance.* EURASIP Journal on Wireless Communications and Networking, vol. 2011:47, no. 1, 2011.

[20] D. Dardari, R. D'Errico, C. Roblin, A. Sibille, M. Z. Win. *Ultrawide Bandwidth RFID: The Next Generation?.* Proceedings of the IEEE, Special Issue on RFID - A Unique Radio Innovation for the 21st Century, vol. 98, no. 9, pp. 1570-1582, Sep 2010.

[21] R. D'Errico, M. Bottazzi, F. Natali, E. Savioli, S. Bartoletti, A. Conti, D. Dardari, N. Decarli, F. Guidi, F. Dehmas, L. Ouvry, U. Alvarado, N. Hadaschik, C. Frankek, Z. Mhanna, M. Sacko, Y. Wei, A. Sibille. *An UWB-UHF Semi-Passive RFID System for Localization and Tracking Applications.* IEEE International Conference on RFID-Technology and Applications, (RFID-TA 2012), Nice, France, Nov 2012, pp. 1-6.

[22] N. Decarli, F. Guidi, A. Conti, D. Dardari. *Interference and Clock Drift Effects in UWB RFID Systems Using Backscatter Modulation.* IEEE International Conference on Ultra-Wideband, ICUWB 2012, Syracuse, USA, Sep 2012, pp. 1-5.

[23] D. Dardari, N. Decarli, A. Guerra, A. Conti. *Enhanced Localization Coverage with Non-Regenerative UWB Relays.* 20th European Signal Processing Conference 2012 (EUSIPCO 2012), Bucharest, Romania, Aug 2012, pp. 1-5.

[24] F. Guidi, N. Decarli, D. Dardari, C. Roblin, A. Sibille. *Performance of UWB Backscatter Modulation in Multi-Tag RFID Scenario Using Experimental Data.* IEEE International Conference on Ultra-Wideband, ICUWB 2011, Bologna, Italy, Sep 2011.

[25] F. Guidi, A. Sibille, D. Dardari, C. Roblin. *UWB RFID Backscattered Energy in the Presence of Nearby Metallic Reflectors.* European Conference on Antennas and Propagation 2011, Rome, Italy, Apr 2011.

Acknowledgements

Questo elaborato rappresenta il coronamento di un lungo percorso di crescita sotto tutti i punti di vista. Tuttavia, tutto ciò non sarebbe stato possibile senza l'aiuto delle persone che mi sono state vicino durante tale percorso e che, pertanto, desidero ringraziare. In particolare, uno speciale ringraziamento va a:

Chiar.mo Professor Davide Dardari: Per il sostegno dato e la conoscenza che ha saputo trasmettermi non solo durante l'esperienza di tesi ma anche durante i suoi corsi di studio che ho avuto la fortuna di seguire.

Dottor Francesco Guidi: Per tutti i preziosi consigli e ed il sostegno che ha saputo darmi durante l'esperienza di tesi.

Dottor Marco Bottazzi: Per avermi dato la possibilità di vivere l'esperienza di tesi presso in un ambiente fantastico e stimolante come l'azienda Datalogic S.p.A. ed il sostegno umano e professionale datomi durante questa esperienza.

Dottor Federico Natali e Dottor Enrico Savioli: Per il prezioso sostegno umano e professionale datomi durante l'esperienza di tesi.

Mia Madre Esther e Mio Padre Vincenzo: Per il fondamentale supporto che hanno saputo darmi. Senza di loro nulla di tutto questo sarebbe stato possibile. In particolare, oltre all'importante supporto economico, hanno sempre saputo darmi supporto morale e psicologico, fornendomi la possibilità di avere sempre qualcuno su contare ed a cui chiedere consiglio, aiutandomi sempre a superare i momenti difficili che inevitabilmente si presentano nella vita di ognuno di noi.

Mia Sorella Ida: Per il sostegno che ha sempre dato alla mia causa, dandomi, insieme ai miei genitori, la forza per superare i momenti difficili che a volte l'esperienza universitaria presenta.

La Mia Ragazza Deborah: Per il sostegno e la pazienza che mi ha dimostrato durante i miei studi universitari. Non è sempre facile per una persona riuscire a capire le necessità di studio di uno studente di ingegneria, ma lei è sempre riuscita a farlo.

Il Mio Amico Valerio Casadei: Per il sostegno reciproco e le spensierate risate durante la nostra vita universitaria. Un grande amico che ho avuto la fortuna di conoscere e frequentare durante la mia esperienza universitaria e che mi ha dato la possibilità di renderla piacevolmente speciale.