DIPLOMA THESIS

Development of Receiver Algorithms for Radio Frequency Identification (RFID)

Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology (TU Wien)

Supervisors:
Univ. Prof. Dipl.-Ing. Dr. techn. Markus Rupp (TU Wien)
Dipl.-Ing. Christoph Angerer (TU Wien)

Carlos Fons Cuesta
Student Number 0727373

Vienna, April 2009

Abstract

Radio Frequency Identification (RFID) is a very fast emerging technology for applications demanding identification or tracking purposes. In order to improve RFID technology development, new algorithms on signal processing are needed and new architectures need to be to explored and rated to find the most appropriate implementation. Rapid prototyping provides the flexibility and fast access desired for a wide range of different algorithm evaluations. The communication performance is improved by converting the received signal processing directly to digital processing devices.

Newly developed standards are issued offering new opportunities to develop RFID systems that offer better performance. EPC Global has created widely accepted standards for product identification such as the EPC Class 1 Generation 2 standard for UHF domain RFID and is a field for improvements.

This diploma thesis proposes several implementations designed and generated on electronic design automation tools that operate on a Xilinx Virtex II FPGA following the specifications of the ECP Global standard for the UHF RFID. These implementations include the entrance, image removal and matched filters in the signal processing receiver chain, as well as a synchronization method and its hardware implementation for the symbol detection of Miller codes. Measurements on board and bit error ratio simulations are used to rate the implemented algorithms.

Table of Contents

Index of Figures

Index of Equations

1 Introduction

This diploma thesis proposes several implementations for an RFID receiver operating on a Xilinx Virtex II FPGA following the specifications of the ECP Global standard for UHF RFID [1]. This project is carried out in the Institute of Communications and Radio Frequency Engineering of the Vienna University of Technology by Carlos Fons Cuesta under the direction of Dipl. Ing. Christoph Angerer and Univ. Prof. Dipl.-Ing. Dr. techn. Markus Rupp.

1.1 Background

In radio frequency identification systems (RFID), a reader communicates with several tags using radio waves. These tags can be active or passive. In the second case they do not contain an own battery and receive their energy from an electromagnetic field provided by the RFID reader in form of a continuous wave signal. The RFID reader, also known as interrogator, has to detect a very small received signal strongly interfered by the signal it transmits to supply the tags with energy, and that leaks into the receiver being several magnitudes stronger.

RFID communication technology does not require line of sight connection to identify an item, and many items can be checked almost simultaneously. This technology is currently fast emerging and very potential for application demanding e.g. identification, stock inventory or tracking.

RFID systems operate in three frequency bands: In the low frequency (LF) domain at around 125 kHz, in the high frequency (HF) domain at 13.56MHz and in the ultra high frequency (UHF) domain at 860-960MHz and 2.4 GHz. In the UHF domain the passive RFID tags apply backscatter modulation for communication.

1.1.1 Prototyping Board

An RFID reader test environment is developed with the final goal of improving the signal processing algorithms on the reader to enhance performance and real time testing. The RFID reader system is being implemented on a rapid prototyping board from the Austrian Research Centers. Figure 1.1 shows the basic structure of this board.

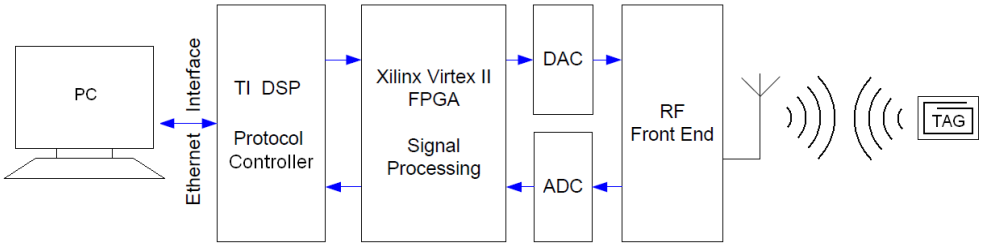


Figure 1.1. Block diagram of the RFID reader prototype.

The main components are a Texas Instruments DSP, a Xilinx Virtex II FPGA [9], two DACs and ADCs and the RF front end.

The DSP controls of the protocol flow. It connects to a user application running on a PC. The signal processing needs to generate the transmitted signal and processes the received signal. This is realized on the FPGA . The DACs and ADCs connect the FPGA to the RF front end, which transmits and receives the radio waves for the communication with the tags.

The trend in the RFID technology is to incorporate many different applications in the same device. For this reason the rapid prototyping board supports two common frequency communication ranges: the 13.56MHz (HF frequency domain) and the 868MHz (UHF frequency domain), further referred to as HF domain and UHF domain.

The FPGA processes the transmit and receive signal at 13.56 MHz either for the HF and the UHF domain to realize a common data processing for the both domains. The FPGA delivers the signal to the DAC at a sample frequency of 40 MHz, which is also the FPGA working frequency. In the UHF mode the RF frontend up-converts the transmit signal to 868 MHz in and down-converts the received UHF signal to 13.56 MHz.

The design flow for configuring the prototyping board is highly automated: The physical layer model is implemented in Matlab/ Simulink [11]. By means of two EDA[1] tools, the Matlab HDL Coder tool[2] [12], and the Xilinx System Generator[3][13], certain blocks can automatically be converted to VHDL code, and directly embedded into the system and synthesized [2], [4].

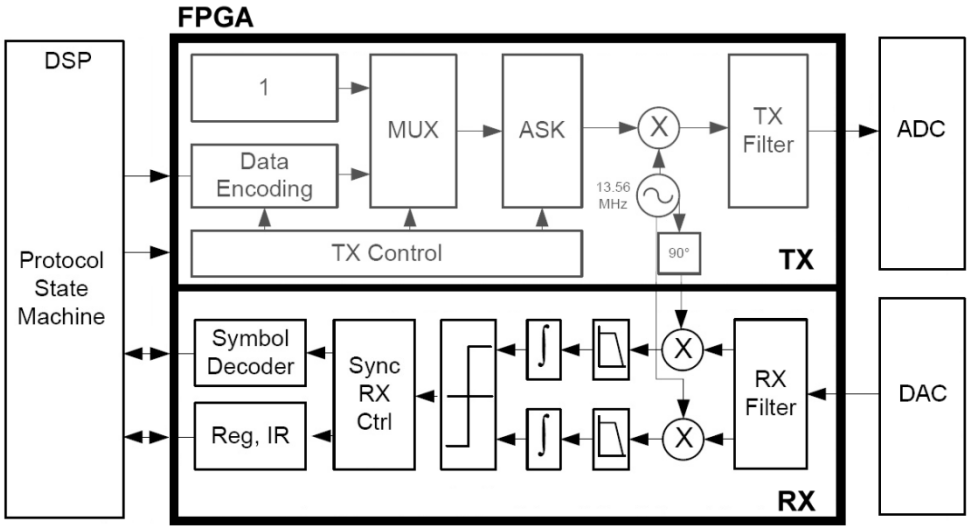


Figure 1.2. FPGA Reader Block Diagram

The FPGA block diagram is shown in Figure 1.2. The FPGA is divided in the transmitter part and the receiver part. This work focuses on the extension of the implemented RFID reader receiver to the EPC global UHF standard.

1.2 Specifications and Impact in the Design

[1] EDA stands for Electronic Design Automation

2 Simulink HDL Coder is a MATLAB tool for conversion of the Simulink block models to VHDL

3 System Generator enables the use of Simulink for designing Xilinx FPGAs

The EPC Global organization [10] has created widely accepted standards for product identification. For this reason the UHF design of this system follows the specifications of the EPC Global standard Class1 Generation2 for UHF RFID [1]. From now on this standard will be referred as the EPC standard for UHF.

The signals defined in the EPC standard for UHF use either the FM0 or the Miller coding. These involve modulating a subcarrier by means of inverting its phase. A subcarrier cycle is shown in Figure 1.3. The frequency of this subcarrier is called backscatter link frequency (BLF)**.**

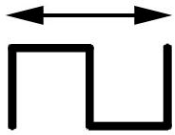
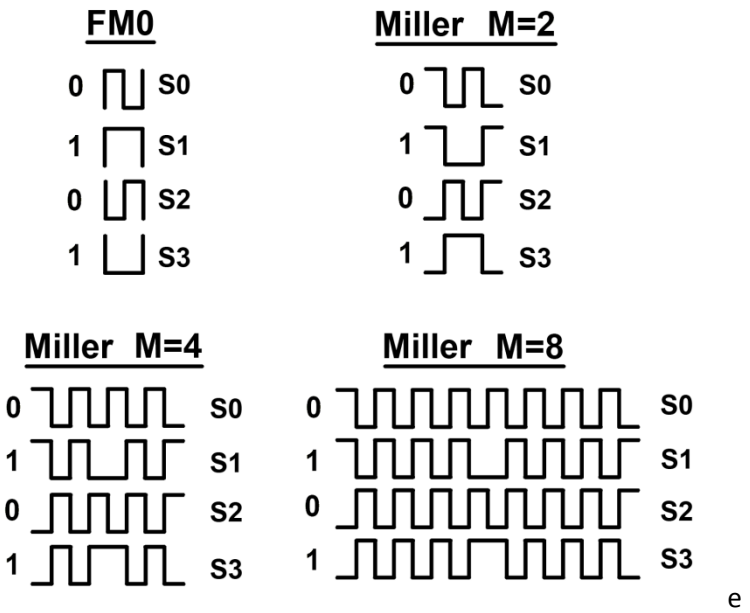


Figure 1.3. BLF subcarrier cycle

The FM0 and Miller symbols consist of a number of concatenated subcarrier cycles.

- FM0: One subcarrier cycle.
- Miller M=2: Two subcarrier cycles
- Miller M=4: Four subcarrier cycles
- Miller M=8: Eight subcarrier cycles

Symbol S1 and S3 represent the logic 1 and have a phase inversion in the middle of the symbol, while S0 and S2 represent logic 0 and don't have phase inversion in the middle. The symbols are shown in Figure 1.4.



e

Figure 1.4. FM0 and Miller symbols

Depending on either FM0 or Miller coding there is an additional phase inversion on the border between some symbols. An example of phase inversion between symbols S0 and S2 is shown in Figure 1.5. The rules for the FM0 and Miller coding are specified in detail in pages 29 and 30 of the EPC standard for UHF RFID [1].

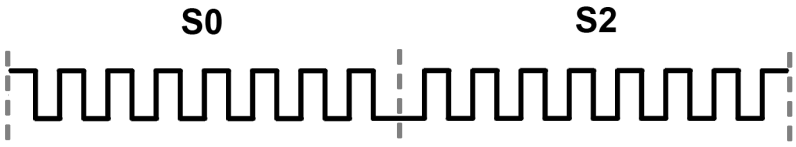


Figure 1.5. Phase inversion in border between symbols S0 and S2.

1.2.1 Summary of Items that Impact the Receiver Design:

Depending on the coding, the symbols have different form and length. The EPC Standard for UHF determines that the BLF that the tags use is freely

chosen by the RFID reader in the range from 40 kHz to 640 kHz and transmitted to the tag at the beginning of the communication. The length of the symbols also varies according to the selected BLF. The UHF standard determines the BLF deviation allowed for the received signal over the indicated by the RIFD reader. This deviation can be up to ±22% on average, which severely complicates synchronization and up to ±2.5% variable during transmission. The exact deviation value each RFID tag commits is unknown to the receiver.

The coding and link frequency selection flexibility together with the big BLF tolerance are the main challenges to overcome an efficient common hardware design.

1.3 Scope of Project

The following is proposed in this project:

- Design and implementation of entrance, image removal, and matched filters. The design must be flexible to fulfill the UHF standard specifications and maintain the already established operation in HF domain by means of a shared hardware structure.
- A synchronization method and its hardware implementation for Miller codes. This method is based in the results by Y.Liu et al. [5] “Digital Correlation Demodulator Design for RFID Reader Receiver” where a receiver system for FM0 codes is developed and C. Mutti et al. [6] “Robust Signal Detection in Passive RFID Systems” where several methods for the application of correlation are studied for these types of coding.

1.4 Chapter description

Chapter 2 details the input filter design. It explains the filtering requirements due to a variable signal bandwidth and the decisions taken to solve them.

Chapter 3 - details the requirements and implementation of the image removal filters**.**

Chapter 4 - details the requirements and implementation of the matched filters.

Chapter 5 – describes the synchronizer design process. In this chapter the detail of the requirements and methods are explained through several stages due to the complexity of the system.

Chapter 6 – shows the tests carried out to verify the correct operation of the designed systems.

Chapter 7 – documents the results produced in this project and presents an assessment of these results.

2 Entrance Filter

The RFID communications in the UHF domain operate in the frequency range from 860 to 960 MHz. There are many other communication systems that utilize frequencies within this range and may interfere in the RFID communication. To avoid these interferences and eliminate any noise beside the operating bandwidth the first block in the RFID receiver is a band pass filter as shown in Figure 2.1.

As mentioned in the introduction the link frequency of the received signal may be within a broad range, meaning the signal may have one of many different bandwidths. In order to support that range of bandwidths it is necessary to reach a balance between high frequency selectivity and an optimized use of the available resources in the FPGA, since these are limited and are to be shared with the other units that integrate the system.

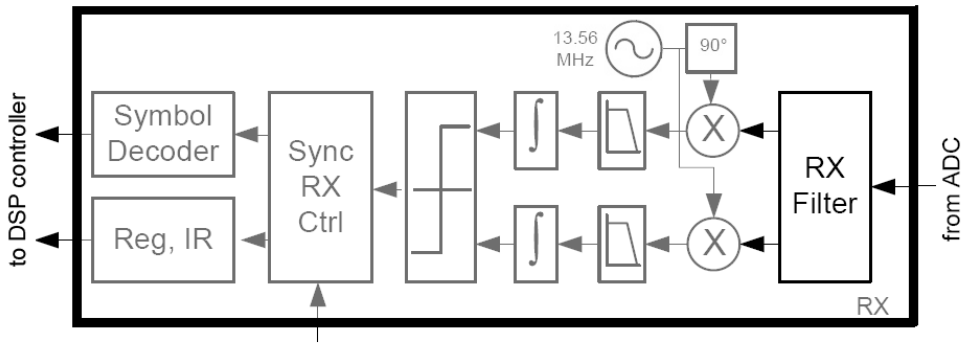


Figure 2.1. Receiver Block Diagram, entrance filter.

This chapter details the design and development of these filters. FDATool[4] [8] is utilized for the frequency response oriented design. The filters are converted into Simulink[5] [11] models, what enables a visual block representation and the simulation of the system. In the last phase of the implementation the

[4] FDATool stands for Filter Design and Analysis. It is a powerful MATLAB tool used for the design of digital filters.

[5] Simulink is a MATLAB tool for Model-Based Design for dynamic and embedded systems

design is automatically converted to VHDL by Simulink HDL Coder[6][12] and integrated into the RFID system.

2.1 Filter Requirements

The tag to reader transmission is performed in a frequency band, commonly used in many other applications, which might interfere in the RFID communication. Therefore, and taking into account the wide range of link frequencies that the UHF EPC for UHF RFID allows to use, it is necessary to develop a filter which bandwidth is as tight as possible to the spectral width of the received RFID signal. The EPC standard for UHF RFID permits the communication from the RFID tag to the RFID reader in a modulation frequency that ranges from 40 kHz to 640 kHz. Therefore a unique filter would not suit a selective filtering in a single way for all possible transmission link frequencies.

The required filtering flexibility is achieved by using a filter bank with bandwidths adequately distributed along the operating range.

In addition, the EPC standard for UHF permits a frequency tolerance that depends on the link frequency requested by the transmitter. These values are listed in the table 6.9 of the EPC standard for UHF [1]. This table shows that the allowed tolerance is assigned by frequency intervals. Especially for BLF 640 kHz and BLF 320 kHz the link frequency deviation tolerance allowed is more restricted. It is meaningful that the filter bank is equipped with filters optimized to these concrete frequencies. It is also reasonable to design the other filter bandwidths matching the interval borders. These link frequencies are 40 kHz, 107 kHz, 160 kHz, 256 kHz, 320 kHz, 465 kHz and 640 kHz, which happen to be spread in a quite regular form along the permitted link frequency range, therefore these are the frequencies finally selected to implement the set of filters.

In this design it is considered that in future an RFID receiver in the UHF domain will operate simultaneously with the one already implemented in the HF domain. The frequencies permitted in the HF domain are 424 kHz and 847 kHz. The first one is within the range of link frequencies permitted in the UHF domain. The second frequency is slightly higher than the UHF domain. In this project an

[6] Simulink HDL Coder is a MATLAB tool for conversion of the Simulink block models to VHDL

additional filter is implemented for the link frequency of 847 kHz, thus making possible to utilize the filter bank in both domains, HF and UHF.

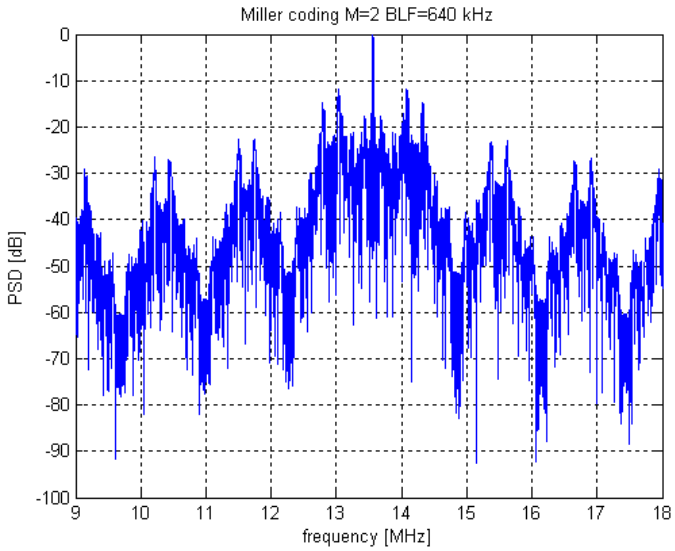


Figure 2.2. PSD of Miller M2, BLF 640 kHz

When using MATLAB to analyze the frequency distribution of the signals used for the RFID communication in UHF, 96% of the power is allocated within the first 3 harmonics. Based on this result, the decision is taken to use this region to delimit the bandwidth required by the filters. The power distribution around the carrier can be seen in Figure 2.2 and Figure 2.3 for a Miller signal with M=2 and BLF 640kHz.

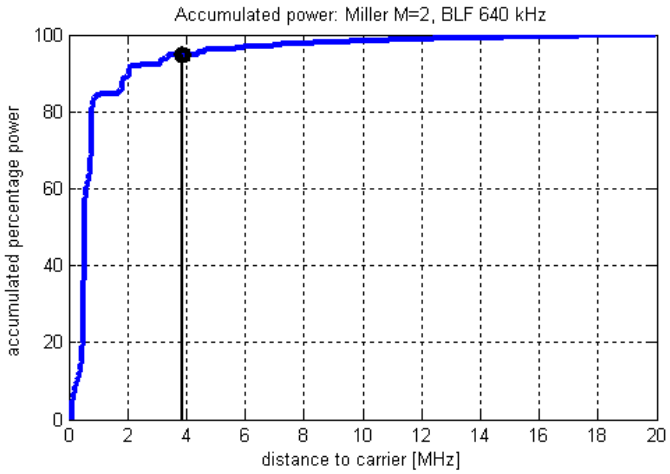


Figure 2.3. Accumulated power

With the above solution the bandwidth and cut-off frequency of each filter results according to the Table 2.1

Table 2.1. BW and Cut-off frequencies of the filter bank

BLF	BW	cut-off frequency	
		Low	High
847. kHz	10.16 MHz	8.48 MHz	18.64 MHz
640. kHz	7.68 MHz	9.72 MHz	17.4 MHz
465. kHz	5.58 MHz	10.77 MHz	16.35 MHz
320. kHz	3.84 MHz	11.64 MHz	15.48 MHz
256. kHz	3.07 MHz	12.02 MHz	15.1 MHz
160. kHz	1.92 MHz	12.6 MHz	14.52 MHz
107. kHz	1.28 MHz	12.92 MHz	14.2 MHz
40. kHz	.48 MHz	13.32 MHz	13.8 MHz

2.2 Filter Design

The MATLAB FDATool is utilized to design the filters. This is a specialized tool in filter design and analysis of FIR and IIR digital filters. The tool offers among other features, the comparison of a large variety of filter types, as well as control of their parameters.

2.2.1 FIR Filter Solution

FIR type filters are chosen initially due to the advantages such as liners phase response, so that no distortion is present in group delay, inherently stable, or a fast response time.

The used FPGA device has 56 multipliers some of which are already in use by the previously implemented RFID reader and some of them should be reserved for future design stages. At present the need for a minimum of two more filters in the receiver belonging to the in-phase and quadrature demodulator is already anticipated and requires a number of multipliers as detailed in the next chapter. It is advisable not to consume too many of the available resources of this type. In these circumstances it makes sense to use approximately 8 multipliers for the set of entrance filters.

In order to reduce the need of multipliers by the FIR filters, these filters have to be created in symmetric form, making it possible to create order 15 filters with the assigned coefficients.

Various tests using different types of FIR filters and different parameters are carried out, leading to the result that the FIR filters created with this order do not achieve the desired balance between the frequency selectivity desired and the attenuation in the pass band. Figure 2.4 Shows as an example, the amplitude response achieved by means of a Kaiser window FIR filter order 15 that consumes 8 multipliers.

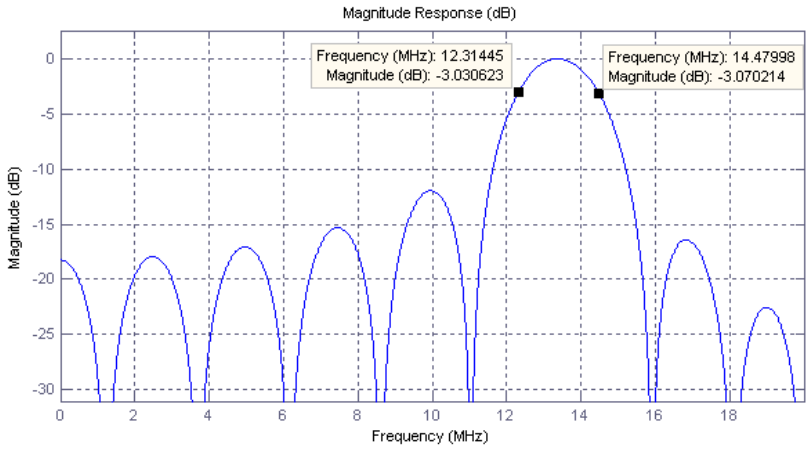


Figure 2.4. Amplitude Response of order 15 FIR filter by Kaiser Window

2.2.2 IIR Filter Solution

The required order for a selective filtering when utilizing the lowest link frequencies defined by the EPC standard for UHF would exceed the amount of available resources, therefore the IIR model filters are considered. The advantage of these filters is a greater selectivity in frequency and other aspects such as low computational complexity. Nevertheless they have other disadvantages, such as nonlinear phase and stability is not ensured a priori. These factors need to be taken into account during design.

The solution chosen is the implementation of Butterworth filters of order 6 and direct form. These filters have a more linear phase response in the pass band than other filters such as Chebyshev or elliptical filters. This minimizes distortion of group delay and in addition, the response in amplitude is flat in that band. Figure 2.5 displays the amplitude and phase response of the filter implemented for the link frequency of 320 kHz. The values in the pass band limit are noted over the curves. Even if it cannot be appreciated in the figure due to the closeness of the curves, the figure displays the filter response for the theoretical values of coefficients and variables as well as the final response of the filter after quantization of the coefficients, as explained in Simulink Model Generation (section 2.4).

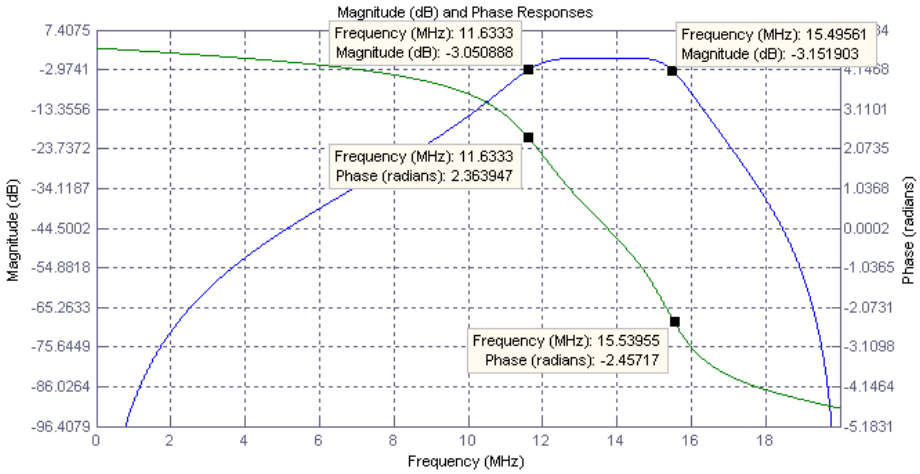


Figure 2.5. Amplitude and phase response of Butterworth filter order 6 for 320 kHz

Figure 2.6 shows the filter for the link frequency of 40 kHz that is the one with the narrowest pass band. The response in phase and amplitude that exhibits this filter is adequate for the requirements, as shown in the image.

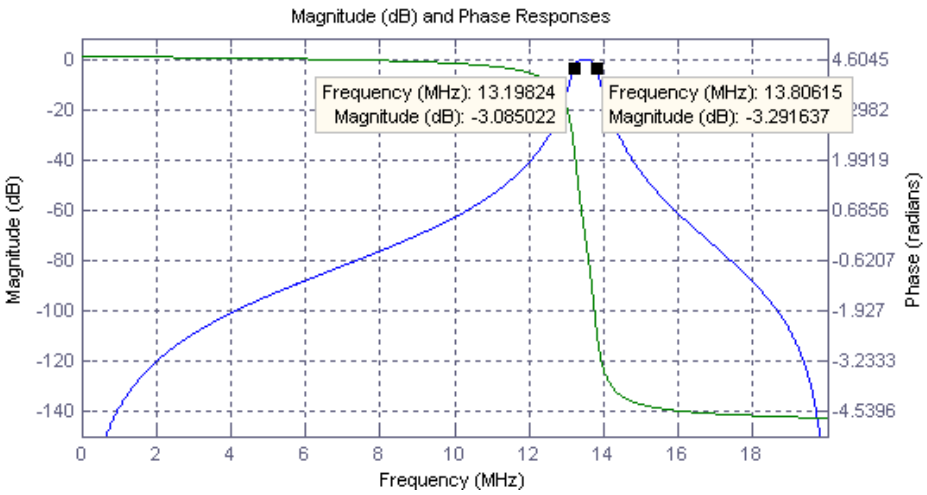


Figure 2.6. Amplitude and phase response of Butterworth filter order 6 for 40 kHz

For the design of the IIR filters, it is necessary to verify that the filter is stable at the operating point. In our case, given that the filters are causal, it is enough to verify that all poles are located within the unit radius circumference to warranty stability, Figure 2.7.

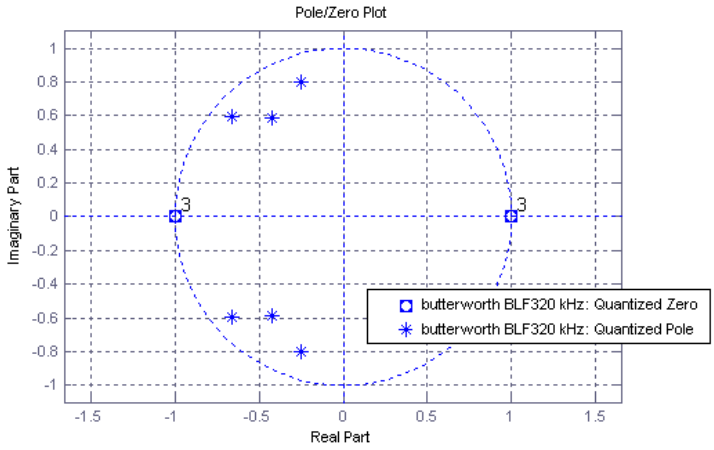


Figure 2.7. Pole/zero diagram

Figure 2.8 displays the transition time of the response of the filter for a step and it can be noticed that it is small enough in comparison with the time of the entrance signal pulse. In the case of this figure, the time of the signal pulse is of 1.5 µs for a link frequency of 320 kHz.

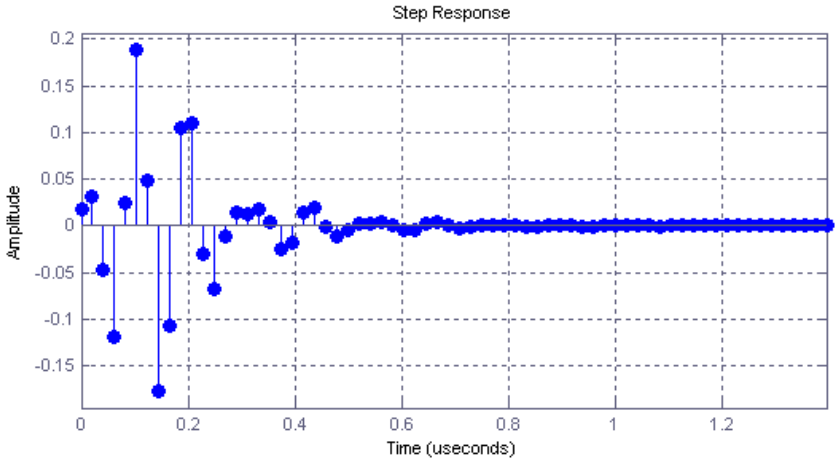


Figure 2.8. Step response

2.3 Simulink Model Generation

The FDATool is used to convert the designed filters to a Simulink model. The model uses only fixed point representation. In order to achieve the first implementation in an automatic way, the following factors are specified in the FDATool:

- dimension specification and value scaling of input and output signals, multipliers, adders, coefficients, etc,
- blocks that have to saturate its output,
- the way the blocks perform approximations.

Some parameters such as the size of the multiplier factors are fixed by the FPGA device or imposed by the output and input signals of the previous and next blocks in the receiver.

The implemented Butterworth filters are made of three sections set up in series. Figure 2.9[7] shows the Simulink model generated in the first section of one of the filters. The number of sections is directly related to the order for the filter, in this case order 6. The filters are implemented in the direct form in order to utilize the minimum number of multipliers. In this way 3 multipliers are used by section, making a total of 9 multipliers.

The filter model generated in Simulink is ready for simulation, either with Simulink, or integrated in the MATLAB workspace, exchanging input and output signals.

[7] . Figure 2.9 shows the values of the used signals. As example, notation Sfix16_En15 means notation in fixed point representation of 16 bits with sign and decimal point in bit number 15.

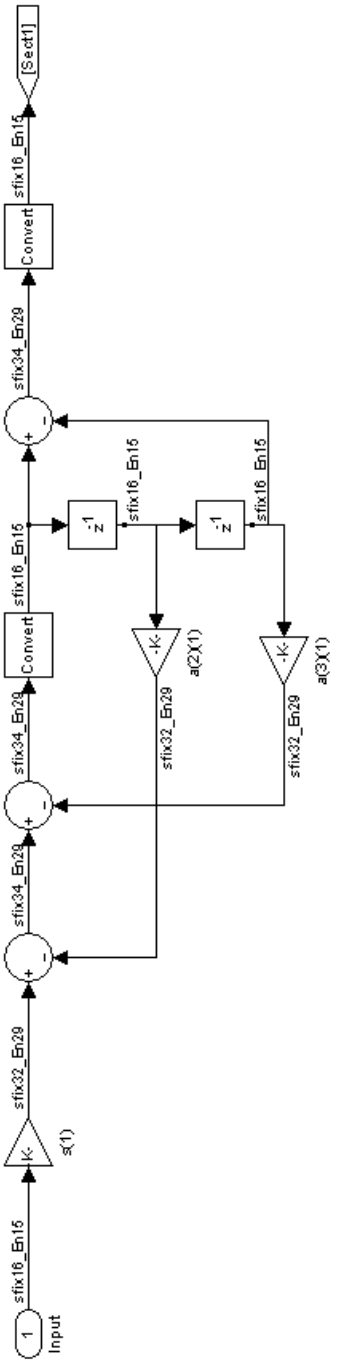


Figure 2.9. First section of Butterworth filters direct form order 6.

2.4 Shared Hardware Structure

Because the filters are never used simultaneously, the most efficient way of implementing the filter bank to minimize the hardware resource consumption is to design in a way that all filters share the same hardware. Each time a filter is to be used, the corresponding coefficients are loaded.

The signal representation scale is fixed for all filters, so that all of them can use the same hardware. To achieve that, a common scale needs to be chosen to provide the best possible precision that allows all the signal values to be represented independently of the selected filter. So, in the first instance, the design of the filters is made separately in order to study the best dimensions and scaling for each one. Afterwards the optimum representation is chosen.

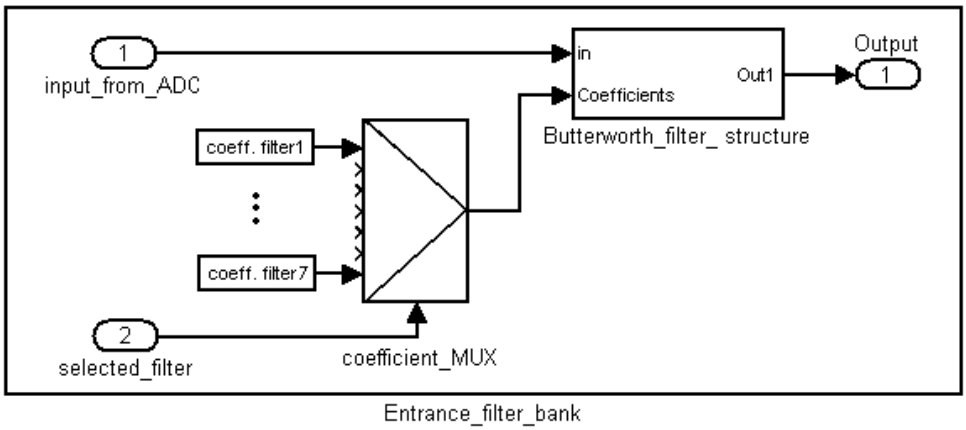


Figure 2.10. Filter bank block

The controlling DSP software of the RFID reader has to select one of the implemented filters depending on the link frequency in use. This is provided by adding a new register to the register block, Figure 2.10 that acts as interface between the controller and the FPGA receiver so that the controller can store the selected filter value on it. The receiver is wired so that the selected filter is sent to the filter bank and the corresponding coefficients are activated, Figure 2.11.

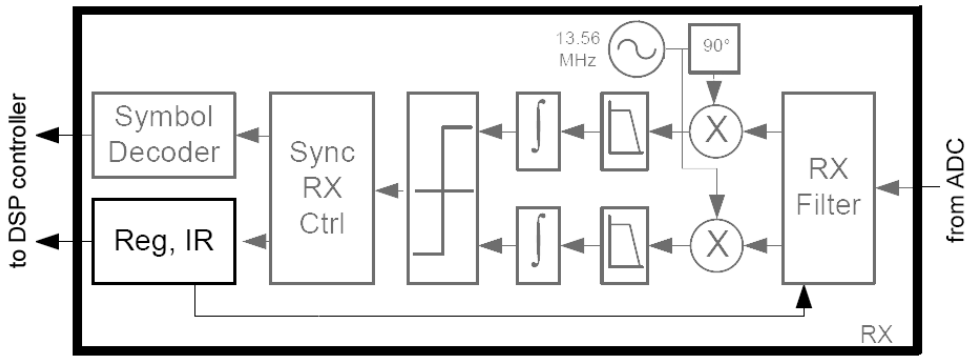


Figure 2.11. Block diagram of the FPGA, register bank.

2.5 HDL Implementation

The Simulink HDL Coder of MATLAB offers the great advantage on FPGA design that converts in a direct and automated way a Simulink model to VHDL. Simulink HDL Coder converts the structure, dimensions and scaling values the same as defined in the Simulink model. Once the conversion is completed, the VHDL generated files are instantiated in the receiver.

The ModelSim simulator is used to verify the correct operation of the generated device. Therefore, simulation files already used in tests before this project are modified to deliver a signal captured in an experimental test to the input to the receiver system. The signal was recorded with an RFID tag transmitting in the HF frequency domain. This enables the comparative study of the attenuation and distortion in the signal when using a filter with narrower or wider bandwidth. The real performance of the filter is analyzed in chapter 5. A spectrum analyzer is used to verify the correct operation once the design has been loaded in the device.

The simulation file also generates the signaling to configure the parameters desired by the controller. The configuration is modified to select the various filters one after another and verify the correct operation of the filter bank.

The last step consists of the preparation of the configuration file for the FPGA. The design with FDATool or Simulink tools does not account during simulation for the delay of the signals when implementing the FPGA device, producing some errors in the timing that could not been foreseen until this point.

It is necessary to add certain delays in the implemented VHDL code, or modify it in the Simulink model to have a model that truly approaches real implementation.

3 Image Frequency Removal Filters

This chapter details the design of the filters involved in the demodulation process of the received signal. The IQ demodulation, realized by multiplying this signal with a sine and cosine signal, generates the baseband signal but also a component at 12.88. This component has to be removed by the filters. The requirements are not very demanding, as no steep slopes are needed and the signal characteristics involved in this process are known in advance. Furthermore there are two filters needed, what make the hardware consumption more sensitive to the resource utilization efficiency.

3.1 Function of the Filter

The received signal will be demodulated to obtain its base band equivalent. For this purpose a phase and quadrature demodulator is used. It is composed of two multipliers and two low pass filters, as shown in Figure 3.1.

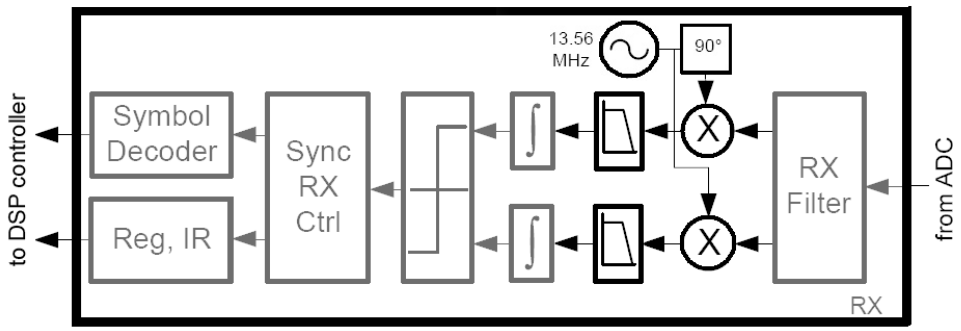


Figure 3.1. Block diagram of the FPGA receiver.

The output of the entrance filter block is divided in two branches and multiplied by two sinusoids with 90 degrees shifted phase in order to generate the in-phase and quadrature components. In each branch, the resulting signal after the multipliers is composed of the base band component and the image component. The last one needs to be removed by means of filtering.

The received signal is down-converted to 13.56 MHz by the radio frequency interface, external to the FPGA. The transmitter part of the RFID reader provides the sinusoids at the required frequency for demodulation, 13.56 MHz.

Given that the sampling frequency is 40 MHz and therefore the maximum frequency is 20 MHz, the image component generated is centered in 12.88 MHz, as calculated in equation 3.1 and 3.2:

$$f_c + f_l = 2 \times 13.56 \text{ MHz} = 27.12 \text{ MHZ} \quad (3.1)$$

$$f_s - 27.12 \text{ MHz} = 40 \text{ MHz} - 27.12 \text{ MHz} = 12.88 \text{ MHz} \quad (3.2)$$

Where f_c means carrier frequency and f_l means local frequency, both equal to 13.56 MHz. f_s is the sample frequency of the received signal which is the same as the board clock frequency.

The received signal has been previously filtered by the band pass entrance filter, therefore it can be assumed that the out-of-band noise has already been removed and only the image component generated by the multipliers needs to be eliminated. To realize that the receiver can work for both the UHF domain and also the HF domain, the BLFs of the HF domain will be considered. Then, the broadest image bandwidth occurs for the HF domain for BLF 847 kHz independently of the encoding type. A higher BLF on either domain generates a bandwidth tighter around the image carrier frequency, 12.88 MHz. According to the bandwidths generated for the entrance filter bank in the previous chapter, the image component is located in this case beyond 8.5 MHz and the base band component below 5 MHz.

3.2 Implemented Filters

Order 3 equiripple low pass filters have been designed in the direct symmetric structure, fulfilling in a single way the requirements needed for every link frequency configuration. Then, it is not necessary for this task to implement a set of filters with different bandwidths. As further advantages to low resource consumption, this filter configuration presents linear phase response, inherited stability and fast response time.

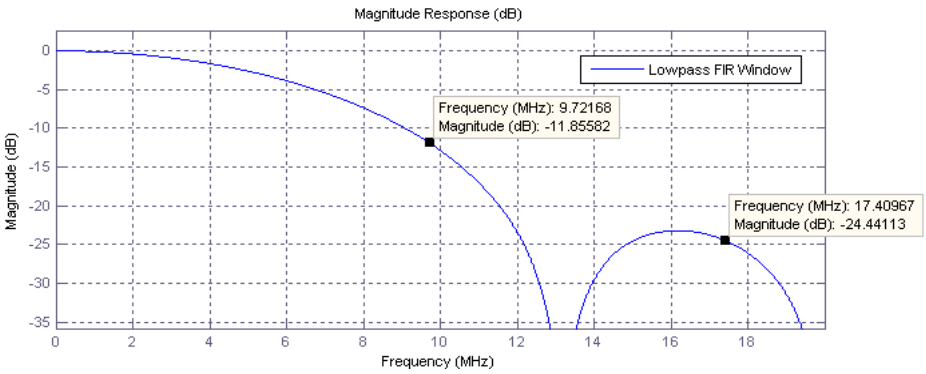


Figure 3.2. Magnitude response of the low pass filters

The symmetric structure enables the use of just 2 multipliers for the 4 coefficients that this filter order requires. This makes a total amount of 4 multipliers taking into account the two filters needed in the receiver. The filter amplitude response of these filters is shown in Figure 3.2. In the graph, the frequency limits of the broadest UHF image, BLF 640 kHz, are indicated by two points on the curve.

The filter configuration selected generates additionally two maximum attenuation frequency points. One of them is fixed at the highest frequency, 20 MHz. The other is located at 12.88 MHz. As explained for the received signal in section 2 of chapter 2, the spectral power of the image component increases when the frequency is closer to the image carrier frequency. For this reason the filter is designed so that the second maximum attenuation point coincides with the position of the image carrier frequency, in order to remove the maximum possible amount of power and improve the filter performance.

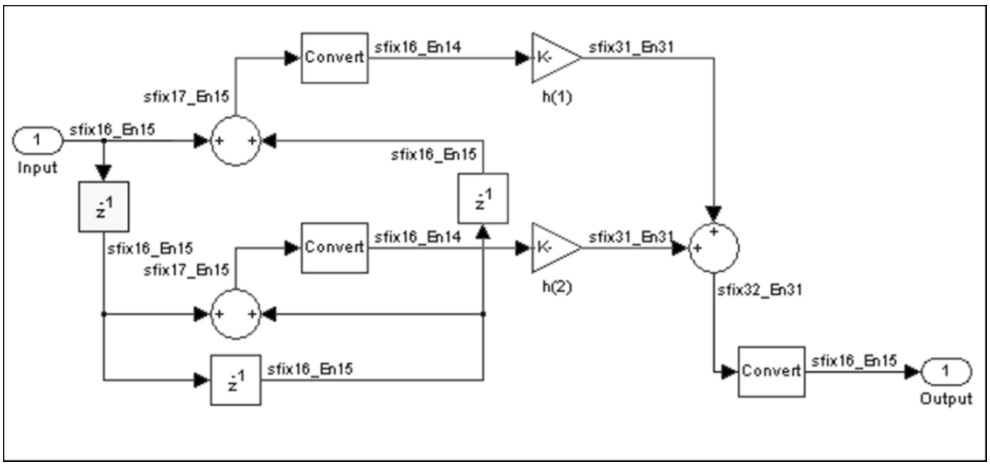


Figure 3.3. Structure of the low pass filters

By means of the same procedure as explained in the chapter 2, the dimension of the filter signals are configured both in the FDATool and afterwards adjusted in the Simulink model to obtain the maximum precision possible. The limitations are again multiplier factor dimensions and the required input and output dimension. The structure of the Simulink model of the filters is shown in Figure 3.3.

3.3 HDL Implementation

In the next step, Simulink HDL Coder converts the Simulink model of the filters to VHDL code and they are instantiated at the in-phase and quadrature demodulator branches. Next, a functional verification is carried out using ModelSim. For this task the same test bench as in chapter 2 is used, which simulates a real tag response with BLF 424 kHz. The input and output signal of the filter in the in-phase and quadrature branches are shown in Figure 3.4.

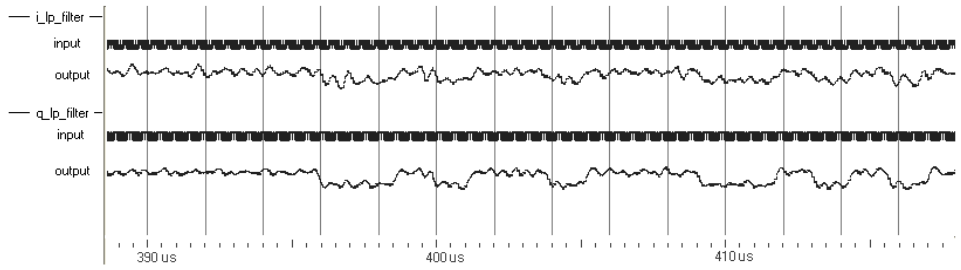


Figure 3.4. Simulation of low pass filters, BLF 424 kHz.

The continuous wave emitted by the transmitter to supply the RFID tag during the response time generates an interference much higher that the response of the tag. This becomes apparent in the input signal of these filters as an offset and a sinusoidal component dominate over the baseband signal. In the output of the filters, in Figure 3.4, the offset has been manually eliminated and both offsets amplified by 1000 with respect to the input signal in order to appreciate the baseband signal, which results from the demodulation process. The power is distributed in randomly, depending on the phase difference between the local sinusoids used in demodulation and the carrier of the received signal. It can be appreciated in this concrete case that there is more signal power in the quadrature branch than in the in-phase branch, so that the baseband signal is more clearly appreciable in the quadrature branch, as shown in Figure 3.4.

4 Matched Filter

The demodulated signal is passed through the matched filter on each branch of the receiver in order to maximize the SNR. The assumption of ideal square pulses of the received signal permits a great simplification of the matched filter algorithm. However the high oversampling of the input signal requires storing a high number of samples. For this reason the design will require the utilization of RAM memory of the FPGA device.

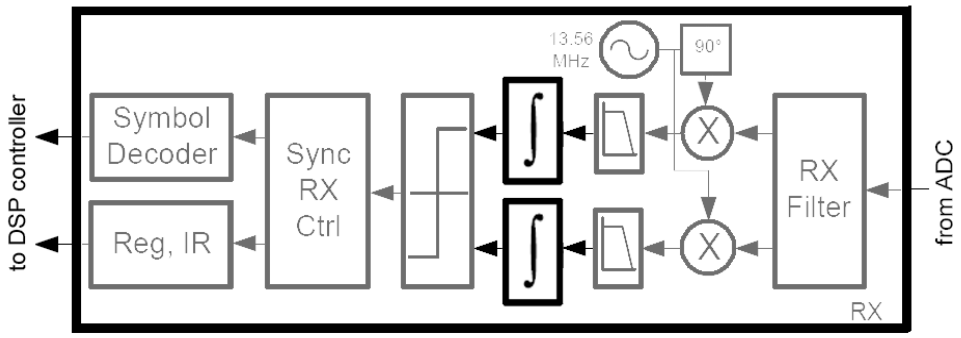


Figure 4.1. Receiver block diagram, matched filters.

4.1 Filter Implementation

The in-phase and quadrature demodulator deliver the baseband signal, as shown in Figure 3.4 in chapter 3. The FM0 and Miller signals are made up by the phase-modulation of a subcarrier signal consisting of square pulses as specified in the section 1 of chapter 2. Assuming that these pulses have a perfectly rectangular shape, the matched filter to this signal is a rectangular pulse of the same duration.

The filtering consists in the convolution of the input signal $x(n)$ with the matched filter $f_{a(n)}$, which is a rectangular pulse of L samples long. The power of $f_{a(n)}$ has not been normalized because it has no impact in the slicer performance [3]. So, the matched filter is simplified to samples of constant amplitude 1. This

convolution is equivalent to a summation and for this reason it is represented as an integration sign in the receiver block diagram.

$$y(n) = \sum_{i=1}^{L} x(n-i) * fa(i) = \sum_{i=1}^{L} x(n-i) \qquad (4.1)$$

Since the scaling is not critical the summation result is shifted right the minimum value, which according to the summation length avoids overflow risk.

For the highest UHF link frequency, 640 MHz, and at the sample frequency of 40 MHz, the length of a BLF cycle is 62 samples, which corresponds to pulses of 31 samples long, and therefore a length of summation L equal to 31. While for the smallest UHF link frequency, 40 KHz, the length of that pulse corresponds to 500 samples, and therefore a summation length L of 500 samples. The pulse lengths in the HF domain are 24 and 48 samples, corresponding in the same way to L 24 and 48. Then, integration lengths 500 and 24 samples determine the maximum and minimum summation lengths required for the matched filter.

It is computationally expensive to perform such a large amount of additions in one single cycle at the frequency of 40 MHz, fortunately we can optimize it and implement it very efficiently when taking into account that once the value of this summation has been calculated for a certain instant n the value for the next instant $n+1$ is:

$$y(n+1) = y(n) + x(n+1) - x(n-L) \qquad (4.2)$$

When initializing the algorithm, at n=1, the value for y(0) is 0 as well as the values for x(-i) for i=0… L-1l. These values simulate the last L samples received are 0. In any case Y(n) is not valid until L cycles have passed.

4.2 RAM Solution

The filtering becomes a simple operation of addition and subtraction. Nevertheless, it is necessary to know the sample received L cycles before. This

implies to store the L last samples to use the oldest as subtrahend in the equation 4.2.

The input signal consists of samples of 16 bits, and the length L of the pulses may be up to 500 samples. This requires storing $16 * 500 = 8000$ bits at each one of the receiver branches. Due to the amount of required memory an implementation utilizing the embedded memory blocks is advantageous. The device used has 56 18-Kbits memories. As 2 independent filters must be used for the branches of Phase and quadrature, one of those memories is used for each of the filter branches.

The RAM memory is always accessed in the same way. At each cycle, the new samples received are stored in the memory. After L cycles the oldest sample is read and afterwards removed, since it is not required anymore. This storage mode is known as FIFO (First In First Out).

The sample frequency of the received signal is the same as the clock frequency, 40 MHz. In each of these cycles it is required to store the new samples received and also to read the sample received L cycles ago. The embedded memory blocks are dual ported memories allowing accessing the memory for reading and writing within one clock cycle [7].

4.3 HDL Implementation

LogiCORE FIFO Generator [7] is a tool specialized for the design of Xilinx FIFO Generator[8]. It customizes the design of the core for a wide range of design parameters like width, depth, memory type, etc. and provides the control logic to ensure a FIFO storage behavior of the core. For this reason the design is directly programmed by means of the Xilinx ISE software.

To implement the matched filter it is necessary to include the logic to control the timing for accessing the FIFO core to ensure that the read data will be provided at the correct time according to the summation length required. All together the matched filter is composed by the FIFO core, the timing control logic for it and the summation. Together they are instantiated in each receiver branch

8 The FIFO Generator core is a fully verified first-in first-out memory queue for use in any application requiring in-order storage and retrieval, enabling high-performance and area optimized designs. [7]

of the demodulator. The protocol stack DSP controller indicates the summation length that is required at each time. For this purpose the receiver is modified to enable a new register and to connect it to the matched filter. To complete this step of the design, a functional verification of the filters is carried out using ModelSim

Figure 1.2 shows the input and output of the matched filter in both the in-phase and the quadrature branches of the receiver for BLF 424 kHz. The input is the base band signal demodulated from the received signal. At the outputs of both matched filters the new signal shape shows the signal levels more clearly because of the higher SNR. The offset has been removed for this image either at the input and the output of the matched filter.

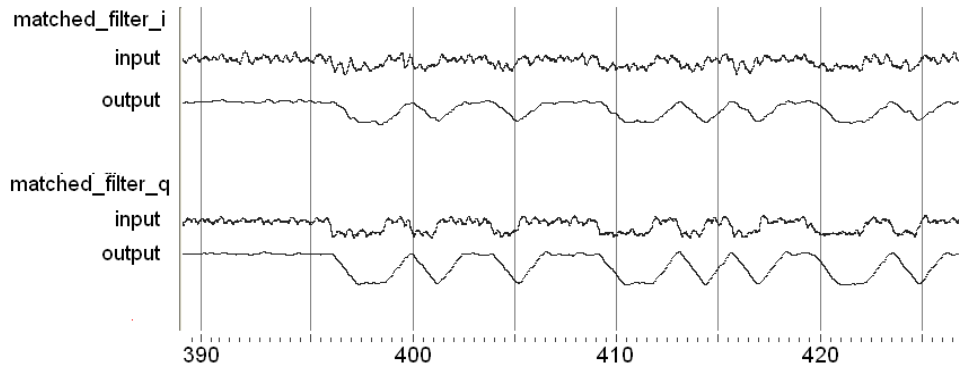


Figure 4.2. ModelSim Simulation of the matched filter, BLF 424 kHz

5 Synchronization Unit

This chapter describes design and development of a Miller coding synchronizer. Its task is to recover the symbol clock in the received sequence and provide the information about the symbols the next block of the receiving chain requires. To achieve this, the synchronizer performs sliding correlations of the received signal and the ideal reference symbols of the Miller modulation. The position of the synchronizer in the receiving chain is shown in Figure 5.1.

The work of Y. Liu et al.[5] and C. Mutti et al. [6] are taken as a basis for this thesis. Their work uses the correlation among symbols for the detection in UHF systems with FM0 coding, and studies the application of correlation to the decoding of DBP[9] codes respectively.

The design of this synchronizer takes into account the combination of two demanding requirements, a large range of link frequencies and a wide tolerance margin permitted for link frequency of the received transmission. This tolerance requires that the real link frequency of the received transmission is estimated. This chapter describes the steps completed to resolve each one of these requirements and the implemented blocks to perform them.

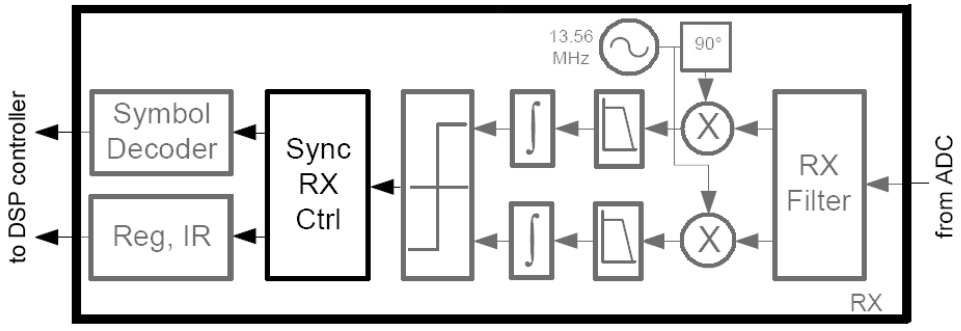


Figure 5.1. Receiver Block Diagram, synchronizer.

[9] DBP stands for Differential Bi-Phase codes.

5.1 Requirements

The design requirements depend directly on the next block in the receiver chain. This block is a decoder that will use the redundancy of the Miller modulation to correct possible errors and estimate the correct transmitted sequence.

The selection of this decoder system is based in the results published by C. Mutti et al. [6]. This study analyses the performance of various detection schemes for DBP codes. The decoder decides on the received symbol sequence as a function of their occurrence probability. The probability is taken as the result of the correlation of the input signal and an ideal reference of each symbol.

The document [6] studies various ways to perform the received sequence estimation. The paper concludes the performance is higher for the maximum likelihood sequence decoding (Viterbi decoding) selecting the maximum values achieved by using sliding correlation. For this reason this project implements correlations of length equal to one symbol, and then, by means of these correlations, determine the intervals where the occurrence of these symbols is expected and provide the decoder with the highest correlation values obtained at each interval, as well as a synchronization signal in concordance with the position of these symbols.

5.2 Miller symbol correlations

In the UHF receiver, the block before the synchronizer is the slicer. It provides at its output a two level baseband estimation of the received signal. This is the signal that the synchronizer has to correlate with the reference symbols.

The first step for the correlation design is the study of the correlation between the ideal base band signal and the symbols of Miller modulation. Hence, the feasibility of synchronization by means of correlations is verified and the most appropriate realization form is decided.

As an example the correlation between a symbol S0 received in a Miller M=4 coded signal and its ideal reference pattern is shown in Figure 5.2. For this

purpose, an input sequence, $S_e(n)$, is used for stimulation, as shown in section a) of this figure. This signal contains the symbol sequence S2, S0, S2. The green line indicates the symbol boundaries. The symbols S2 previous and next to the symbol S0 in the input sequence are included because the phase change at the symbol borders depends on them and therefore they influence the result of the correlation. The reference symbol S0 is shown in section b) of the same Figure.

The correlation is calculated as:

$$R_{n_1}^{S0} = \sum_{n=1}^{N} S_e(n - n_1) * S0(n) \qquad (5.1)$$

Where n_1 is the shift of the input signal in the time axis and N is the length of the reference S0 symbol, denoted as S0 in the equation.

The correlation is shown in Fig in section c) of Figure 5.2. This representation shows a clear and unique peak indicating the position of the symbol S0 and various smaller peaks in various positions within the symbol interval.

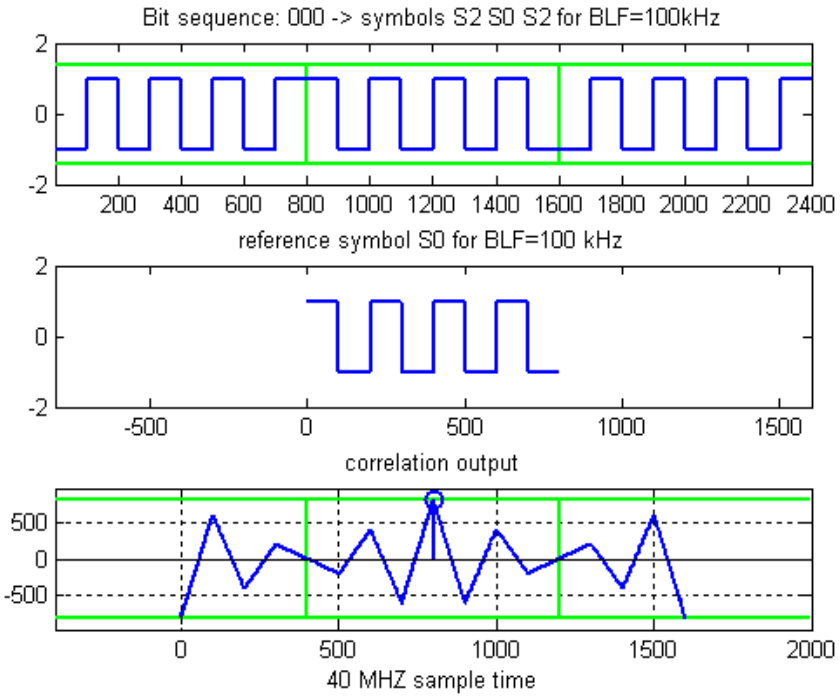


Figure 5.2. Correlation study between S0 symbols Miller M=4 coding

The correlation with the symbol S2 is obtained by just changing the sign of the correlation obtained with the S0 symbol, as S2 = -S1. It is the same for symbols S1 and S3. So, as from this point, only the correlation with one of the symbols of each pair, and the negative result will be taken to know the correlation with the other symbol of the pair.

Figure 5.3 shows the effect of the correlation of the S1, S3, S1 symbols sequence with the reference symbol S3. In this example the Miller coding M=8 is used. This figure demonstrates that the number of peaks next to the main maximum increase when increasing the parameter M.

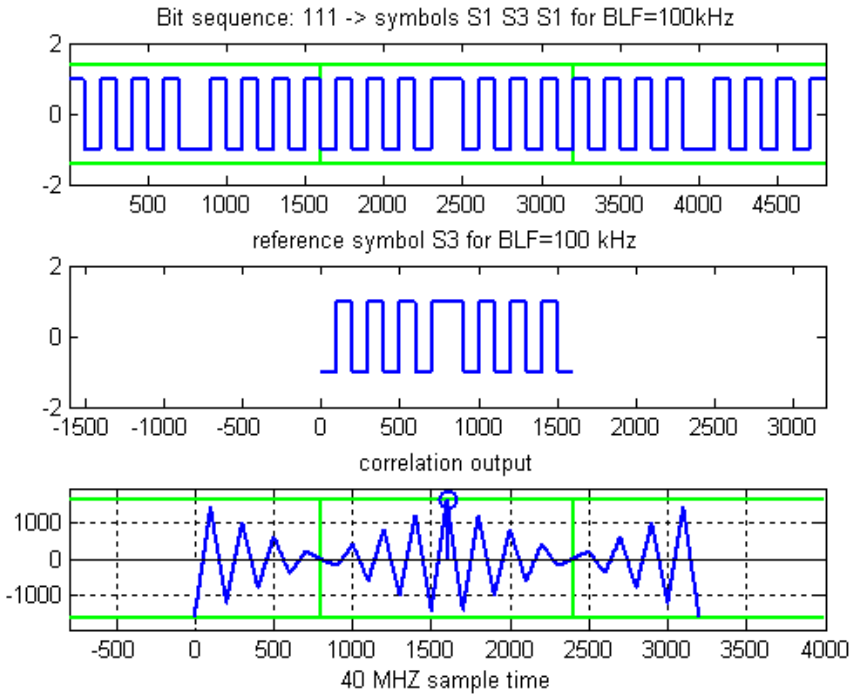


Figure 5.3. Correlation between S3 with S3 symbols Miller M=8 coding

Figure 5.4 displays the result of the correlation with S3 symbol when receiving the symbol sequence S2, S0, S2. Despite the fact that no symbols S1 or S3 exists in the input sequence, the correlation result shows that in the borders between symbols an S1 and S3 symbols pattern appears. This is due to the phase inversion in Miller modulation. This fact must be taken into account to avoid an error in the symbol detection.

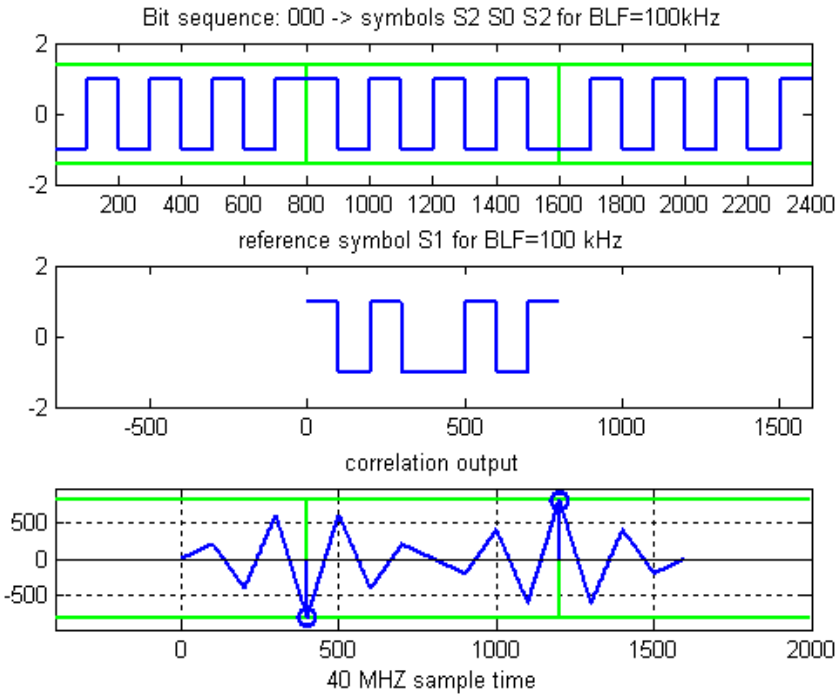


Figure 5.4. Correlation of S0 and S1 symbols, Miller M=4

There are some specific combinations as the one shown Figure 5.5 where the correlation of the symbols S1, S2, S3 sequence with the reference symbol S0 is shown. Here, multiple positive and negative correlation peaks can be seen. They indicate the pattern of S0 and S2 symbols in various positions. This happens in some cases only for the correlation with the symbol S0 (or S2).

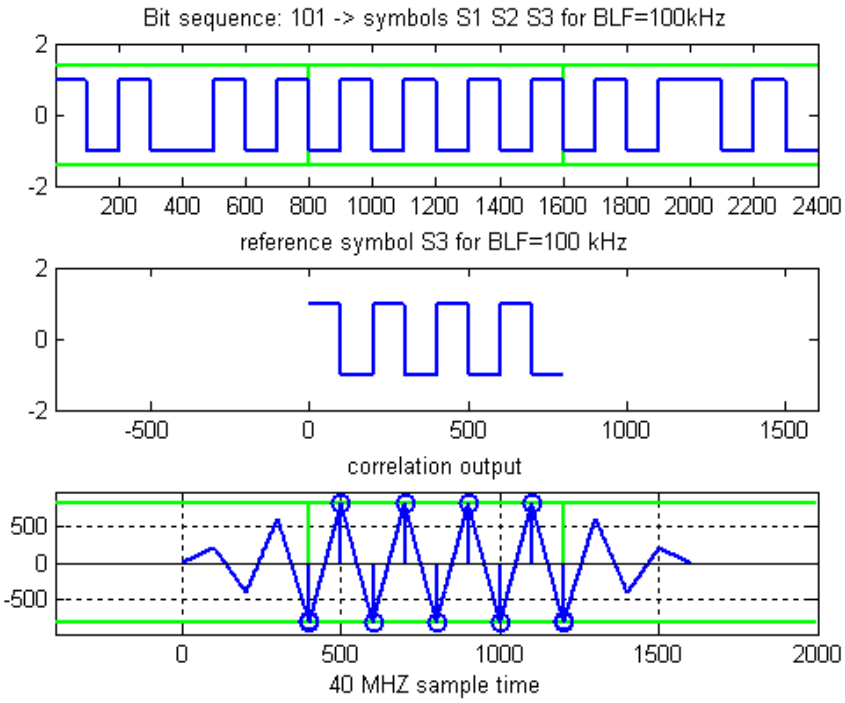


Figure 5.5. Correlation of S0 and S0 symbols, Miller M=8

Since the UHF standard allows some tolerance between the link frequency of an RFID tag and the link frequency indicated by the transmitter, the impact on the correlation of this deviation is studied. Figure 5.6 shows an example where the received signal has a frequency deviation of +10%, so that the symbols are shorter in time versus the reference ones. In this case it can be noticed that the main signal maximums and minimums are kept in the correct position with respect to the received signal, although somewhat attenuated and enlarged, making its recognition more difficult. On the other hand, some peaks that don't indicate any symbol position are distorted, in some cases magnified, so that in the case of a large frequency deviation would make the recognition of symbols uncertain even within the limits of tolerance specified in the standard.

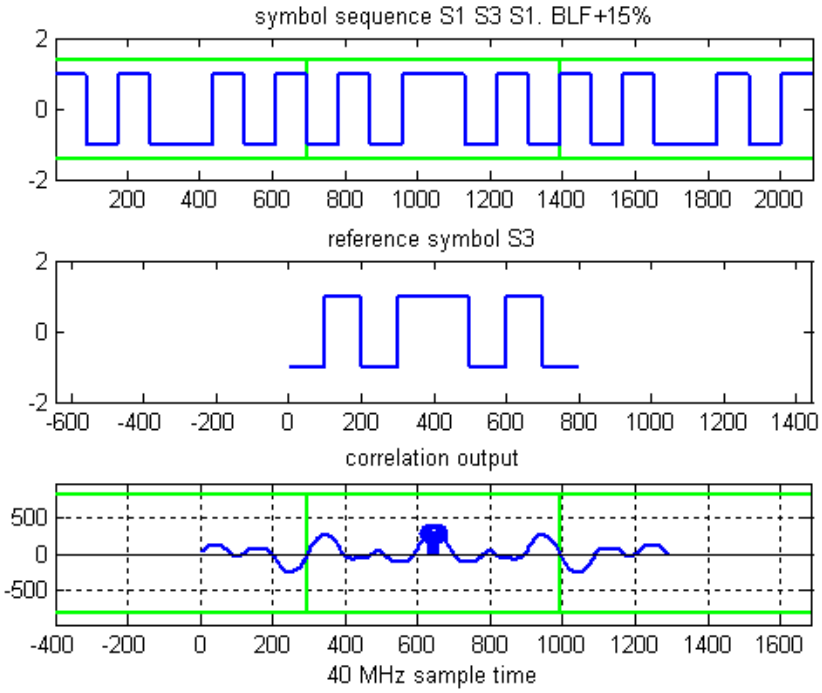


Figure 5.6. Correlation with a deviation of +18% in the link frequency

It has been observed that the pattern of the S1 symbol, that generates a clear an isolated peak, appears at least once every second symbol interval, although it may appear either in the symbol position or in the border between two symbols. Given this fact, the following rules: are established to perform the required synchronization:

- The search of symbols in the correlation signals is divided into intervals of a length of half an interval. The correlation values are not taken as valid output of correlation between symbols, as they don't correspond to real symbols.

- The synchronizer will adjust its temporal reference clock in the intervals where there appear positive or negative maximums of the correlation with reference symbol S1 only.

- This intervals will be distinguished because the maximum positive or negative correlation value with the symbol S1, is greater than the maximum correlation values with either the symbols S0 or S2 in that interval.

Figure 5.7 indicates over the correlation signals with the S1 and S2 symbols for the Miller symbol sequence [S2 S0 S1 S2 S0 S1] the intervals where a

symbol pattern is searched, those that correspond to real symbols and not real symbols, and the points where the temporary synchronization is performed.

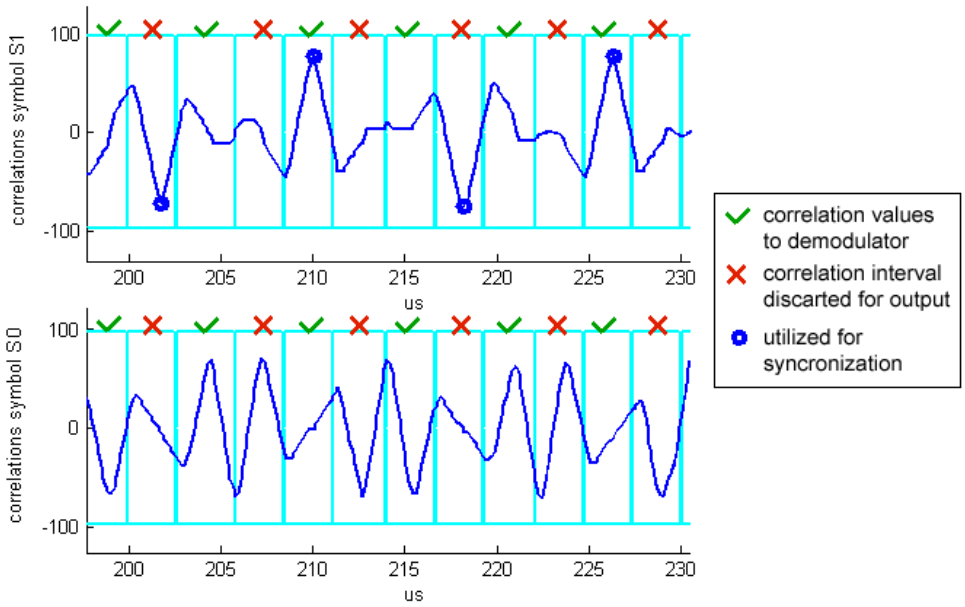


Figure 5.7. Time synchronization and discarded intervals, Miller M=2.

5.3 Implementation of the correlation

According to formula 5.1, a correlation with the last N samples received must be done at each cycle. So, this is the number of samples required to record in the memory. For this purpose, a shift register will be used. The shift register of length N stores each new sample received, shifting the previous ones. A correlation calculation will be performed each new sampling cycle with the information of the register.

The block that computes the correlation with the reference symbols S1 and S0 will be referred to as correlator. A graphic scheme of this block is shown in Figure 5.8.

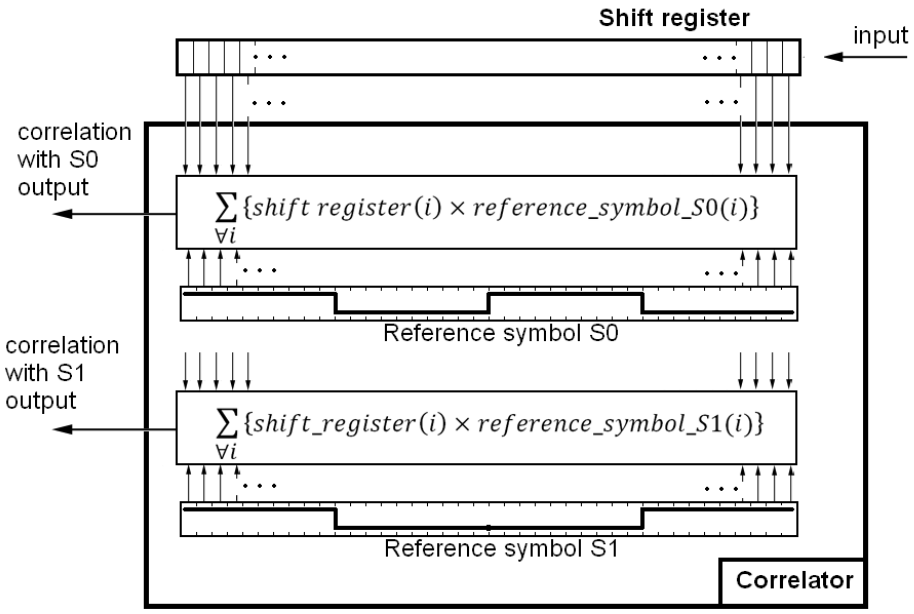


Figure 5.8. Correlator structure.

5.3.1 Correlator operation

The synchronizer input signal is composed of two levels, +1 or -1. The Miller symbols used as reference are also composed by two levels and it can be decomposed in a number of squared pulses depending on the level as shown in Figure 5.9.

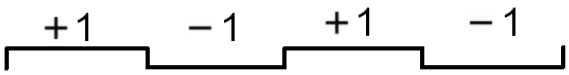


Figure 5.9. Pulse decomposition of the reference symbol S0.

Taking these values into account the correlation summation can be decomposed in as many parts as square pulses the reference symbols have and simplify the calculations for each pulse in the same way as the matched filter summation, detailed in chapter 4 in the formulas 4.1 and 4.2. This calculation is represented in a schematic manner in Figure 2.7. After this simplification, the

correlation operation is reduced to a few additions and subtractions, requiring access to two only points of the shift register for each pulse of the reference register, which means an enormous complexity and resource reduction.

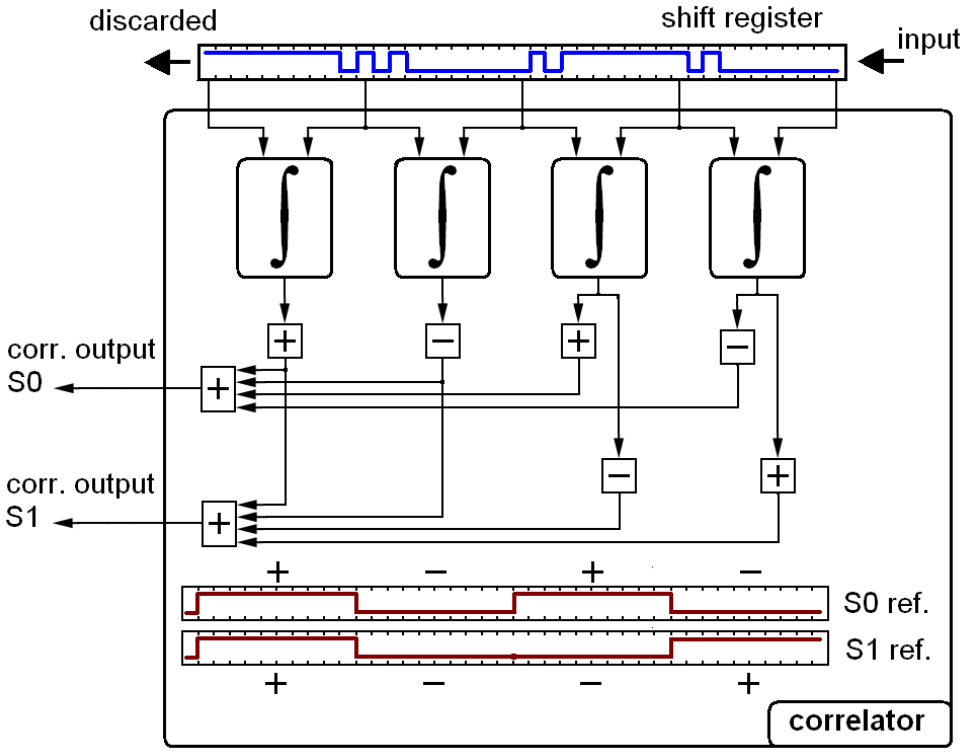


Figure 5.10. Correlator scheme.

5.3.2 Compression

The synchronization hardware must be prepared to work in the whole range of link frequencies allowed by the UHF standard. Due to the wide range of link frequencies and coding types allowed, the symbols may reach from 120 samples, for Miller M=2 at 640 kHz, up to 8000 samples, for miller M=8 at BLF=40 kHz, at the sampling frequency at this point of the receiver. To achieve an efficient

hardware resource usage, all signals should be treated by the same hardware structure independent of the selected link frequency.

To achieve this, decimation to a common link frequency is performed. The common link frequency is that one that generates pulses of less temporary length. This corresponds to the greater link frequency permitted in the case of maximum deviation permitted, this is 640 kHz link frequency + 22% tolerance. The final average value is 780 kHz. Since the pulses that compose the Miller signals are half as long as a BLF cycle, this means pulse lengths of 25 samples at the sampling frequency.

The lowest frequency that can be requested from an RFID tag is 40kHz. At that frequency only +-4% tolerance in frequency is permitted. This gives pulses of a length from 480 to 521 samples. To process these signals in the correlator system, the signal must be decimated by a minimum factor of 20 samples. This means to use only one out of every 20 samples of the received signal. In order to avoid loss of information, a process is performed that transforms the set of removed samples in a single sample of a value proportional to the average value of the removed samples. In this way the information is used in an efficient manner saving hardware resources.

The input signal is made up of one-single-bit samples. To achieve a decimation from 20 to 1, the 20 samples can be added and represented by a integer number of 5 bit. In this way the usage of memory is reduced from 20 to just 5. If from this value only the 3 most significant bits are taken, the memory usage is further reduced, while a great part of the information is kept.

The number of bits that must be recorded in the whole shift register is now calculated as follows:

$$2 * M * n_m * n_b$$

Where 2*M represents the number of pulses per symbol as a function of the Miller repetition parameter M, n_m the number of samples per pulse, and n_b the number of bits that are used to represent each one of the new samples.

In the implemented synchronizer, the sampling frequency will be reduced to 10 samples per pulse, using 3 bits for each sample. This means 480 bits in total for Miller M=8.

The number of samples that must be grouped depends on the input signal link frequency. Therefore, to perform this operation, a controller is

designed that computes the number of samples to group as a function of requested link frequency.

The diagram of this operation is shown in, Figure 5.11. The number of grouped samples is used as summation length and a clock enable signal is generated in accordance with the new sampling frequency. From this block on, the correlation system will take into account this clock enable to work on the received signal.

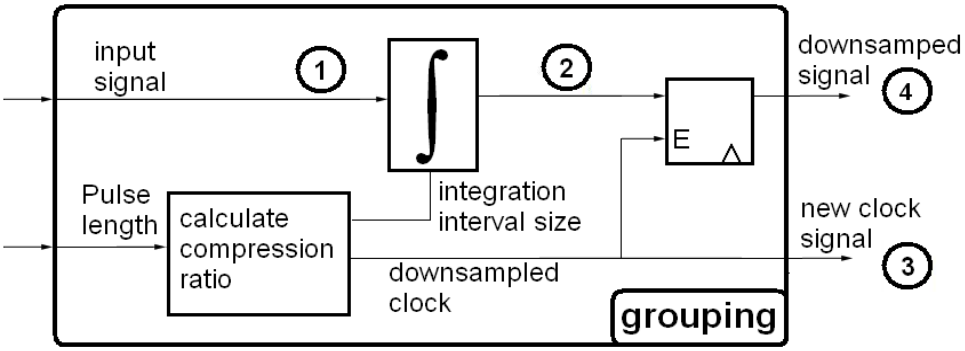


Figure 5.11. Grouping block.

Figure 5.12 displays the signals that take part in this process for an initial link frequency of 250 kHz, which means pulse lengths of 80 samples.

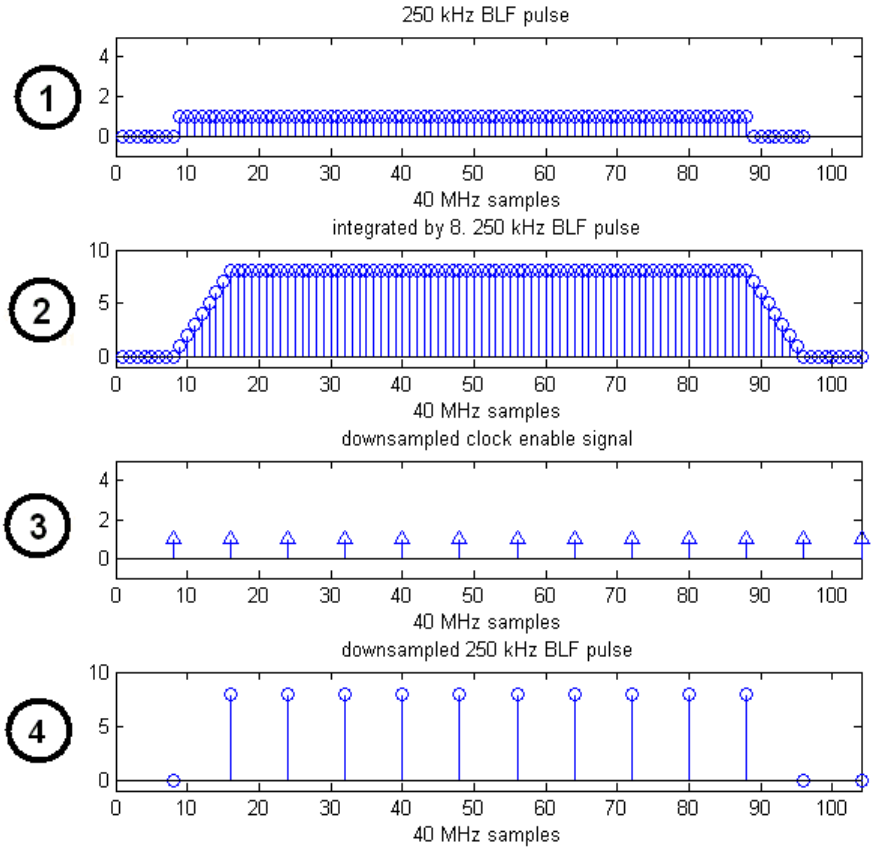


Figure 5.12. Sample grouping process.

5.4 Frequency Estimation

A method has been proposed to handle signals emitted at any link frequency by the transmitter. To be able to obtain the information from the correlations it is necessary to take into account the link frequency deviation of the received signal over the indicated link frequency.

The work of Liu et al. [5] proposes the use of a correlation bank. The correlators process the input signal preamble simultaneously with many reference symbols. The reference link frequency of each correlator is deviated positively or negatively over the one requested by the transmitter covering the whole

tolerance range. The correlator with the highest value indicates the estimation of the BLF of the received sequence.

As a difference of the FM0 coding, Miller coding includes a long pilot sequence at the beginning of the preamble destined for improving the synchronization. The synchronizer implemented in this Project utilizes this pilot sequence to find the link frequency of the received signal in a progressive way.

The pilot sequence consists of a number of subcarrier cycles emitted in succession without any phase inversion during a period of 4 or 16 times the symbol duration. The RFID reader decides whether to request one pilot length or the other to the Tags.

Since there are not phase inversions along the pilot sequence, it can be considered the pilot sequence is a repetition of symbol S0 in succession (or similarly of symbol S2). Then, if there is no link frequency deviation among the received signal and the reference symbol S0 the correlation between them presents maximum positive and negative peaks, while the sequence is orthogonal in all the interval to the symbols S1 and S3, and the correlation with S1 is theoretically zero. In Figure 5.13 is shown an example of the correlation of the pilot sequence with symbols S1 in the upper part and S0 in the lower part respectively for BLF of 380 kHz. The negative amplitude offset of -300 in the Symbol S0 correlation has been additionally added to avoid overlapping in the figure.

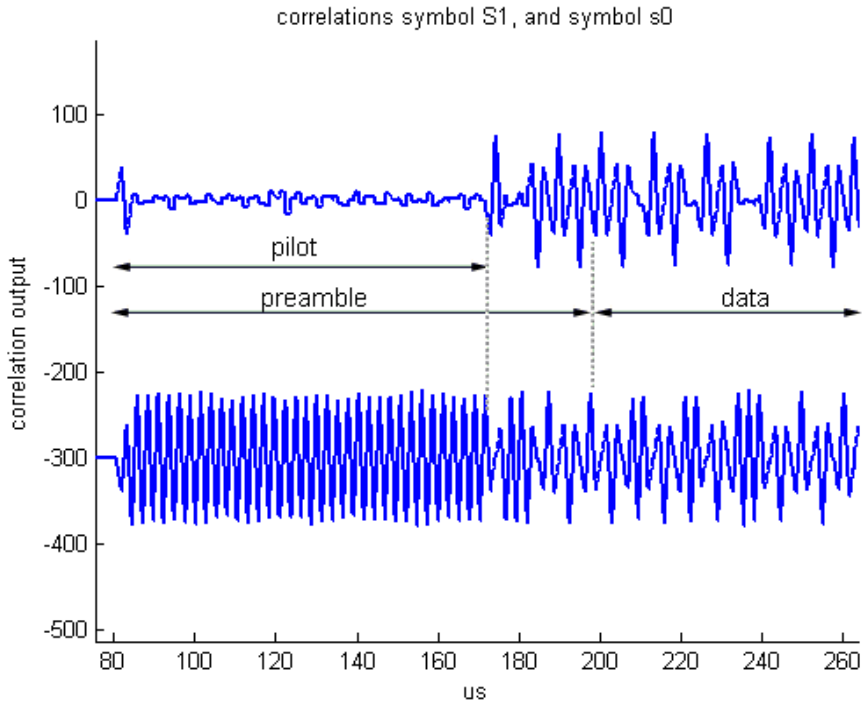


Figure 5.13. Symbols S1 (up) and S0 (down), correlation for a preamble and data sequence, BLF 380 kHz, SNR 3dB.

Computing the pilot sequence with the same hardware as the one utilized for the symbols means an important resource consumption saving, especially regarding memory. To estimate the frequency the pilot sequence segment contained in the shift register is correlated simultaneously with two additional correlators. The link frequency reference of these correlators is deviated positively and negatively over the expected one. This deviation means the symbol duration utilized for correlation are longer or shorter. Figure 5.14 shows a representative sketch of the reference symbols utilized. Since all the correlators share the same shift register each one accesses the data at different positions according to their lengths.

The correlator in the center of the Figure is not deviated in respect to the requested BLF. This correlator will be called main correlator, while the others will be referred to by indicating their respective deviation over the main correlator.

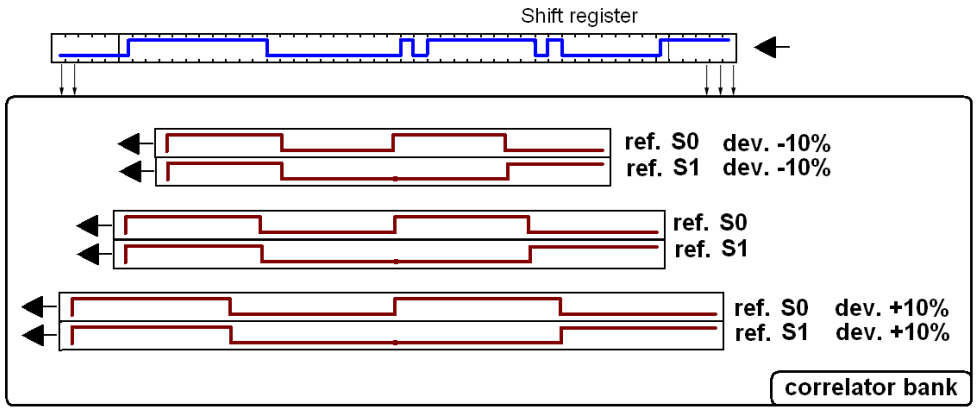


Figure 5.14. Correlator bank for 3 link frequency references

During the pilot sequence processing there is a correlator that produces higher correlation values because its reference frequency is closer to the received sequence link frequency than the other correlators. With this information the synchronizer knows if the link frequency deviation is positive or negative.

To compensate the link frequency deviation an additional decimation is included before the correlators. Its decimation ratio depends on the deviation information the correlator bank is calculating as shown in Figure 5.15.

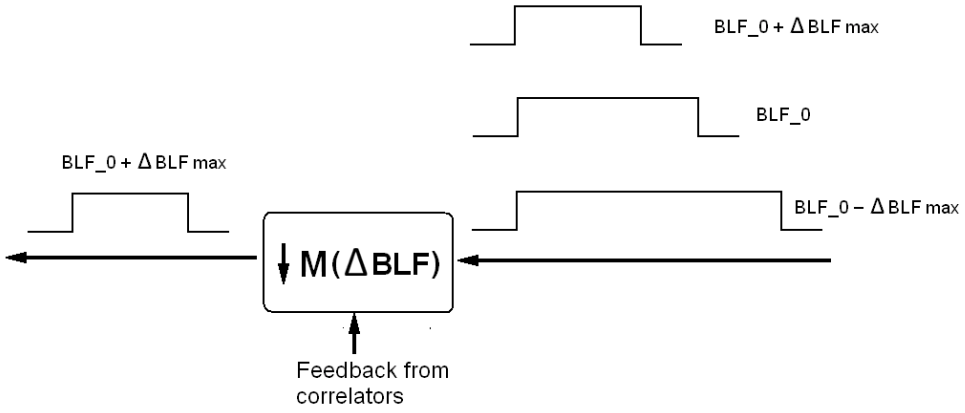


Figure 5.15. Link frequency deviation compensation

The decimation ratio is initialized to the value that exactly transforms the time length of the symbols at the requested BLF to the minimum length allowed by the tolerance. If the real BLF utilized is lower, the decimation ratio will decrease. The opposite will do if the link frequency is higher. At the end of the preamble pilot the symbol length at the output of the decimator coincides with the minimum value.

During the pilot sequence evaluation only the correlations with the symbol S0 are taken into account since it is orthogonal to the symbol S1. Figure 5.16 shows the maximum correlation values for the symbol S0 over the deviation between a correlator reference and the link frequency of the signal received. In this Figures three points indicate the values the correlators would indicate if their frequency deviation over the main correlation were -10%, 0% y and +10%, and assuming the main correlator frequency coincides with the frequency of the received signal.

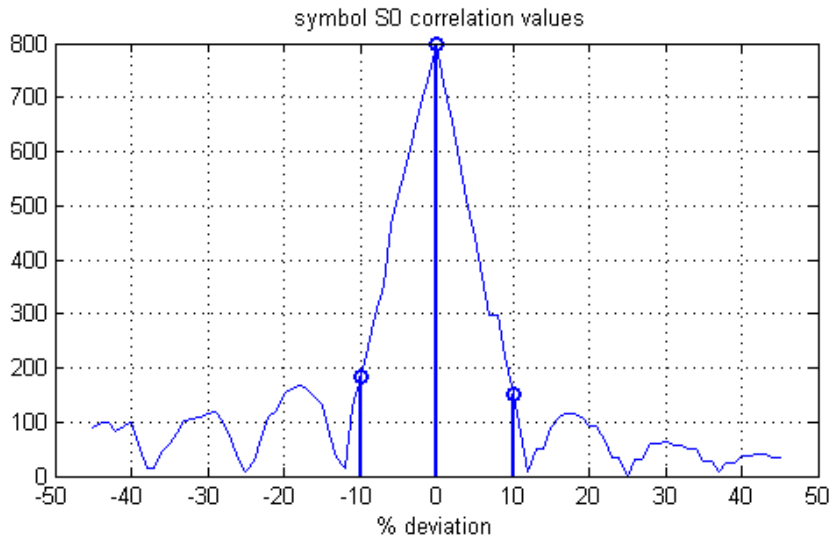


Figure 5.16. Correlation peak attenuation versus link frequency deviation, Miller M=8.

Figure 5.16 shows the value of the ±10% deviated correlators is approximately the same. This fact will be utilized to recognize the link frequency synchronization with the input signal. Since the side correlations do not produce exactly the same value, this difference has to be compensated applying some weights.

Figure 5.17 shows the maximum correlation values achieved if the link frequency is 8% deviated under the requested one. This example shows that the -10% deviated correlation value is much higher than the -10% indicating the decimation ratio has to be varied.

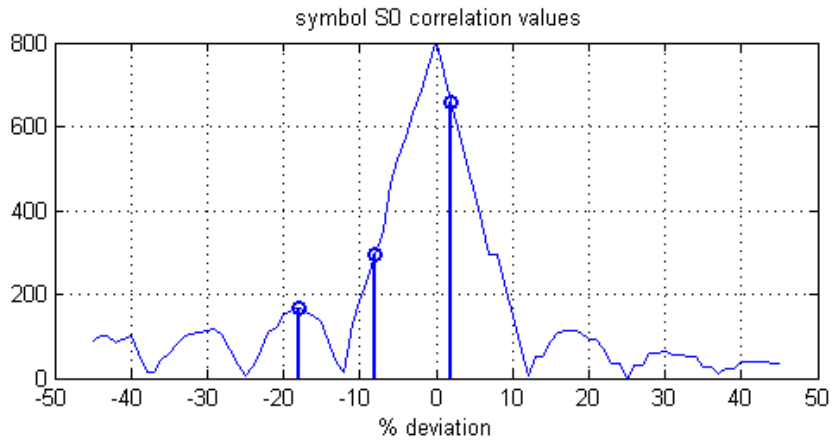


Figure 5.17. Correlation peak attenuation versus link frequency deviation, Miller M=8.

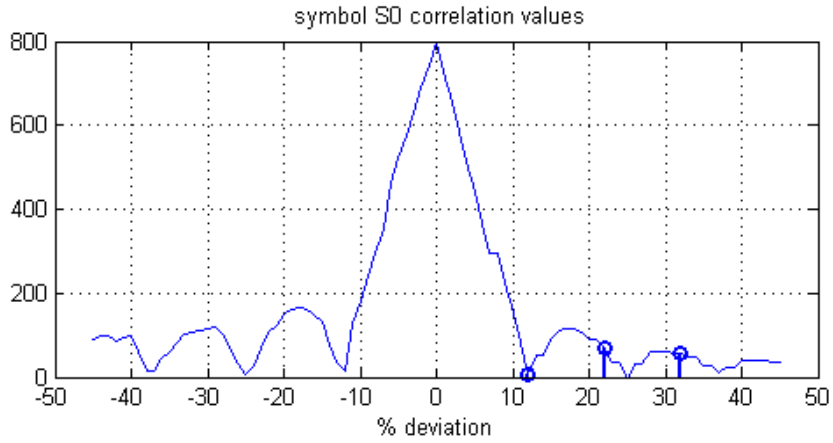


Figure 5.18. Correlation peak attenuation versus link frequency deviation, Miller M=8.

Figure 5.19 shows the maximum values for the 3 correlators for a deviation of +22%. This example shows a case where the synchronizer wound not be able to distinguish the direction to vary the decimation ratio, for this reason two additional correlators have been added to the correlator bank. The reference symbols sketch of the new correlator bank is shown in Figure 5.20.

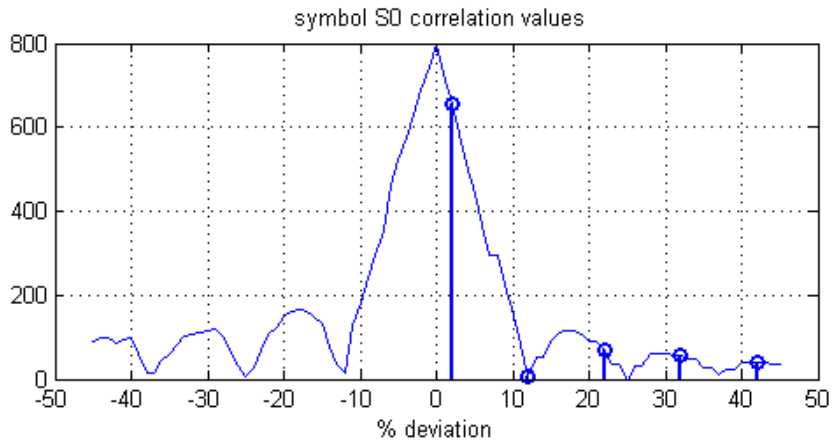


Figure 5.19. Correlation peak attenuation versus link frequency deviation, Miller M=8.

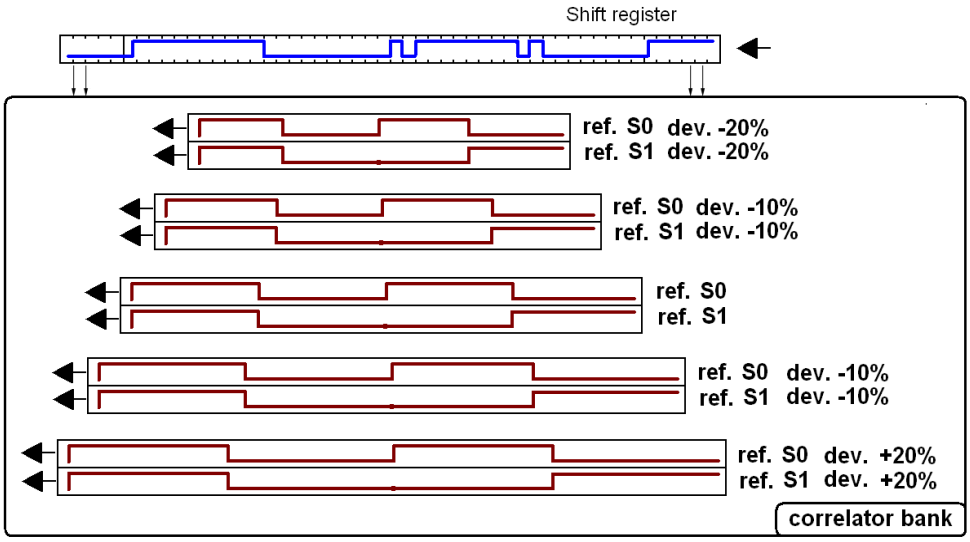


Figure 5.20. Correlator bank for 5 link frequency references

The decimator is placed before the grouping unit to be independent of the grouping conditions. The block sketch is shown in Figure 5.21. This figure shows also the position of the correlators bank, the comparison block, that is in

charge for deciding the highest correlations and the decimation controller. The discontinuous square embraces the blocks working at the down sampled frequency specified by the grouping block.

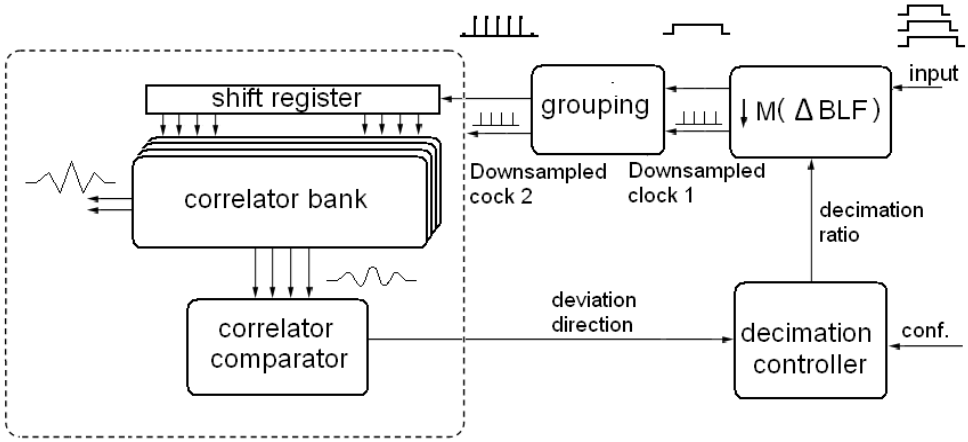


Figure 5.21. Synchronizer model, Frequency synchronization.

5.4.1 Preamble processing

The slicer detects the instant when a tag signal is received [3]. Until that moment the slicer is providing the logic 0 at its output. The synchronizer waits for a variation in the slicer output to start the frequency synchronization.

The correlation comparator block is in charge for studying the results of the deviated correlations. First, it waits until the shift register is full. Then it memorizes the values of the maximum positive and negative peaks during a time interval of 1.28 times a subcarrier cycle time to ensure a symbol pattern has entered the shift register, even in the worst deviation case. At the end of this time interval it decides the deviation direction depending on which correlator has achieved the highest maximum and send this information to the decimator controller. The decimator controller changes the decimation ratio, meaning a new sample frequency. The cycle starts again waiting the shift register to be filled with a section of input signal with the new sample frequency. This cycle is shown in Figure 5.22.

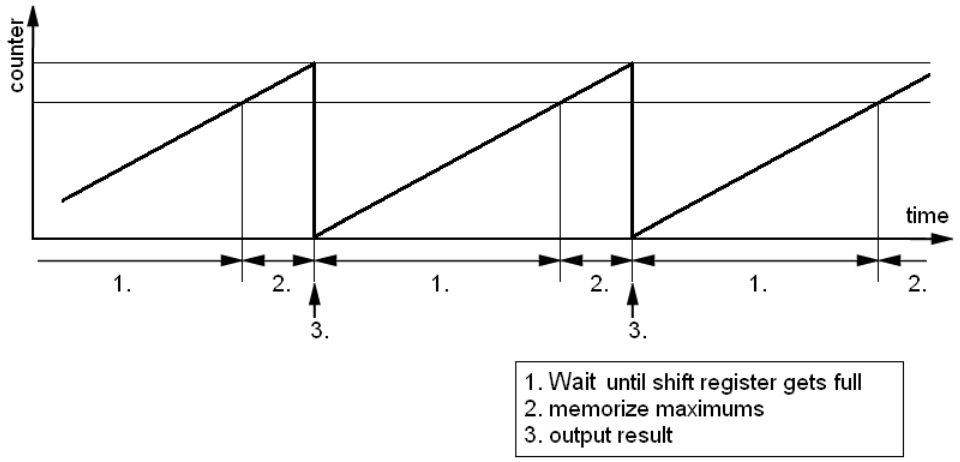


Figure 5.22. Frequency synchronization cycle.

The time to perform this adjustment depends on the length of the pilot sequence. The length varies depending on the configuration and also depending of the link frequency deviation. For example, if the link frequency is higher as indicated, the pilot time interval is shorter.

Figure 5.23 shows in dark blue the correlation output of the main correlators for an input signal with the long preamble, 16 symbols and time deviation of -18%. The correlation with symbol S1 is in the upper part, while the correlation with the symbol S0 has a -300 offset. The light blue lines indicate the different stages for the received signal processing.

This Figure shows how at the beginning there is a bar frequency synchronization and the correlation with symbol S1 is not zero and the correlation with symbol S0 does not produce optimum values. As time proceeds, the pilot sequence the synchronizer corrects the deviation showing the desired correlation values.

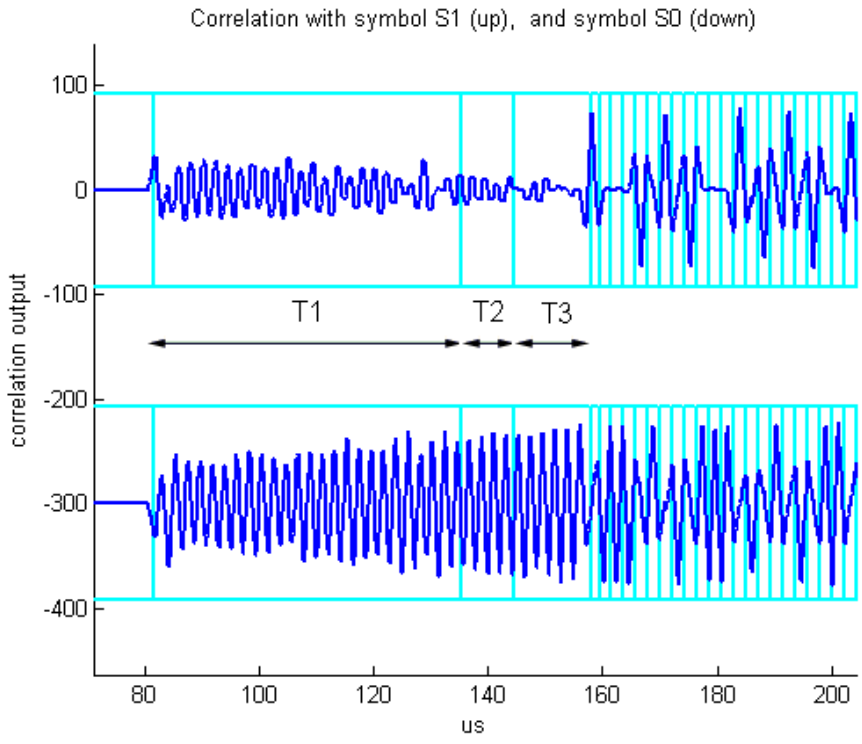


Figure 5.23. Frequency synchronization, BLF 380 kHz, SNR 3 dB, temporal deviation -18%, extended preamble.

Figure 5.23 shows the time intervals T1, T2 and T3 of the pilot evaluation. During T1 the frequency is being progressively synchronized. This time is as big as possible to allow this algorithm to get the best result.

In the time interval T2, the frequency synchronization doesn't stop, but during this interval the highest absolute values for the correlation of symbol S1 and S0 are memorized in order the compute a threshold in the mean value.

The time interval T3 starts to await for the beginning of the first symbols S1 of the preamble. This is decided when the correlation with symbol S1 grows beyond the threshold level just calculated. When this happens the pilot interval finishes and the synchronizer starts looking for the symbol positions.

When the symbol detection interval starts the frequency synchronization algorithms continues working the same way but the decimation ratio corrections are much smaller in order to get if possible a finer approximation.

The remaining part of the preamble after the pilot and the data segment follow the Miller coding rules, so that the preamble can be detected and decoded together with the data.

5.4.2 Evaluation of the estimated frequency

As a function of the total variation the decimation ratio has accumulated, the frequency the decimator would be compensated correctly can be deduced. This frequency is defined here as the estimated frequency.

This estimated frequency is represented in the next Figures over time for several deviations of an input signal. The deviation is represented temporarily, this means the relative variation of the sample length of the pulses or symbols. This variation, ΔT, is related to the link frequency deviation, ΔBFL, as

$$1+\Delta T=1/(1-\Delta BFL)$$

As an reference, the limit frequency deviations, +22% and -22%, correspond to deviations in time, ΔT , of -18% and +28% respectively.

Figure 5.24 shows in blue line the relative temporal deviation over time for the realization already shown in Figure 5.23. The red line indicates the average time deviation of the input signal utilized for stimulation, -18%, and in black lines the deviation limits, at -18% and +28%, and the no deviation line.

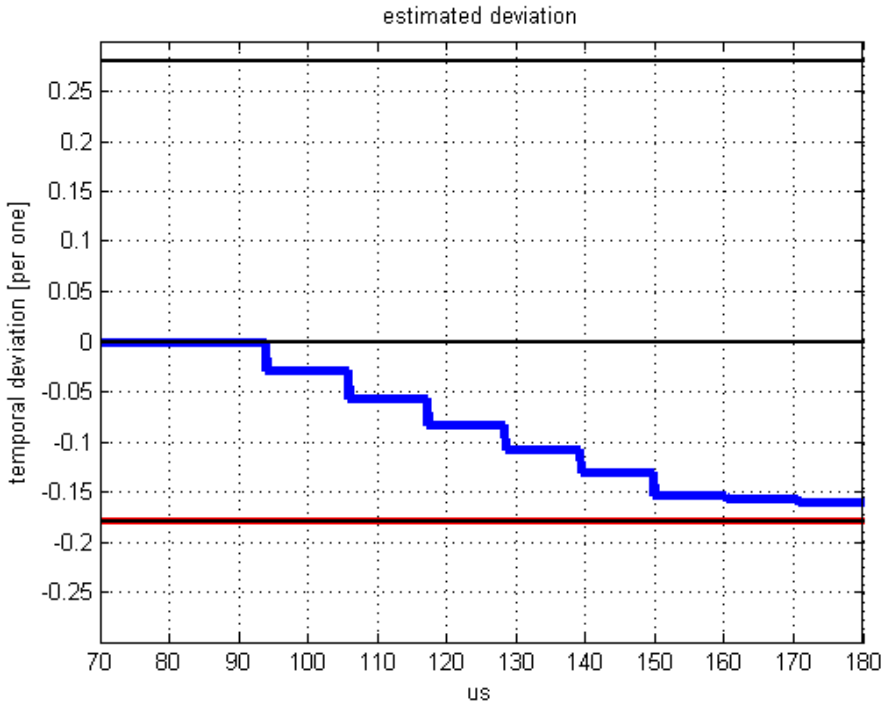


Figure 5.24. Blue: Estimated frequency, Red: Average received signal frequency, BLF 380 kHz, SNR 3 dB, Miller M=2.

Figure 5.25 shows an example for a +28% time deviated input signal. In this Figure can be observed that T3 is much longer than in the Figure 5.23 because the link frequency deviations difference.

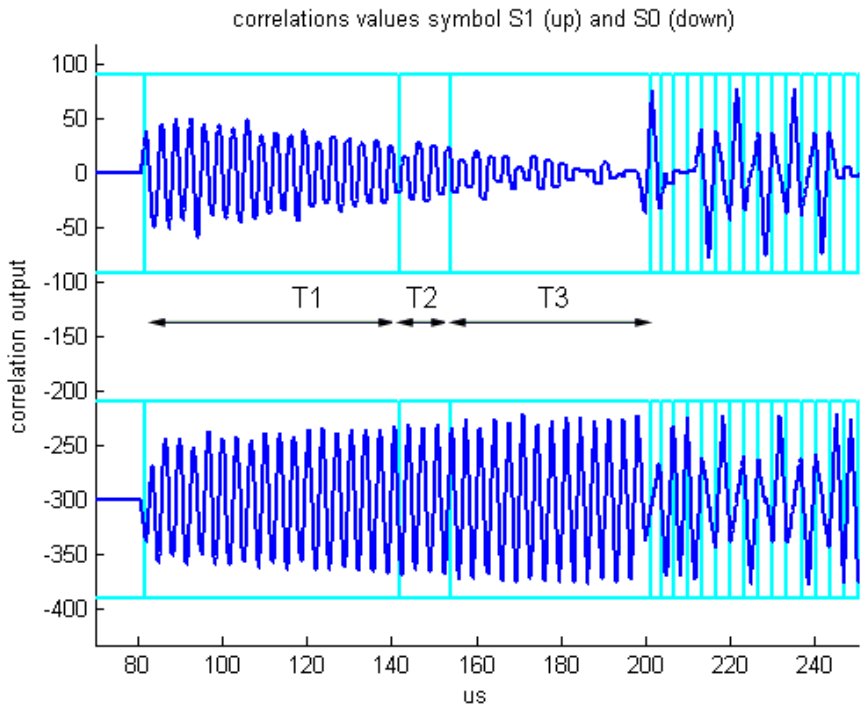


Figure 5.25. Frequency synchronization, BLF 380 kHz, SNR 3 dB, temporal deviation +0.18%, long preamble.

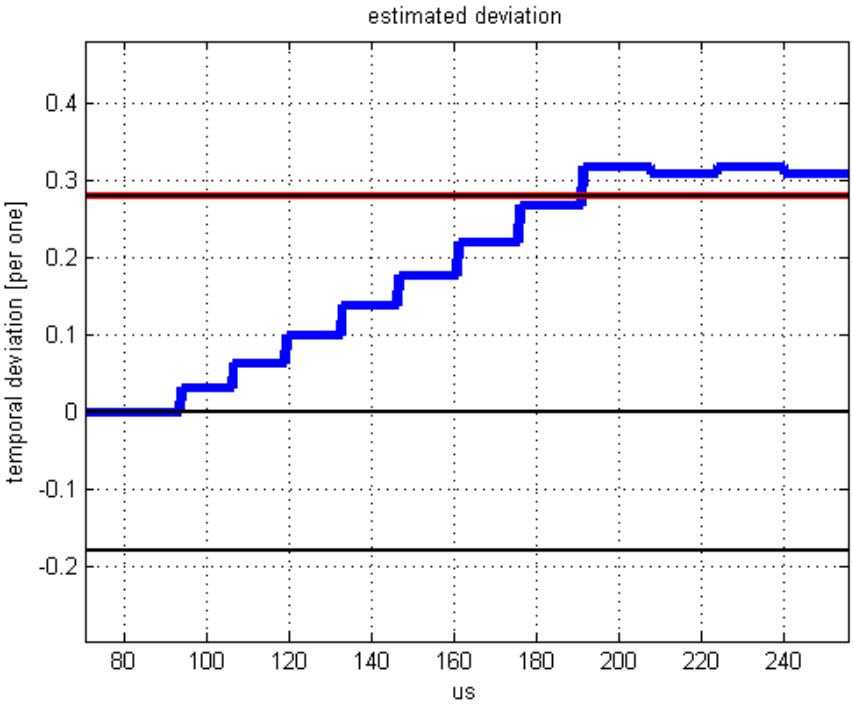


Figure 5.26. Blue: Estimated frequency, red: average received signal frequency, BLF 380 kHz, SNR 3 dB.

Figure 5.27 shows the estimated link frequency shows an example of +12% in time deviation but a short preamble. T1 is much smaller in this case because of the shorter preamble. In this case the preamble time is so short that the frequency adjusting jumps have to be too big to reach the farest link frequency deviations in time. For this reason the synchronizer saves time by deciding the freqcuency deviation direction before the shift register is completely filled with signal at the new sample frequency. This way more frequency adjusts are done in the same time but smaller frequency jumps, enough to fulfil all deviation cases.

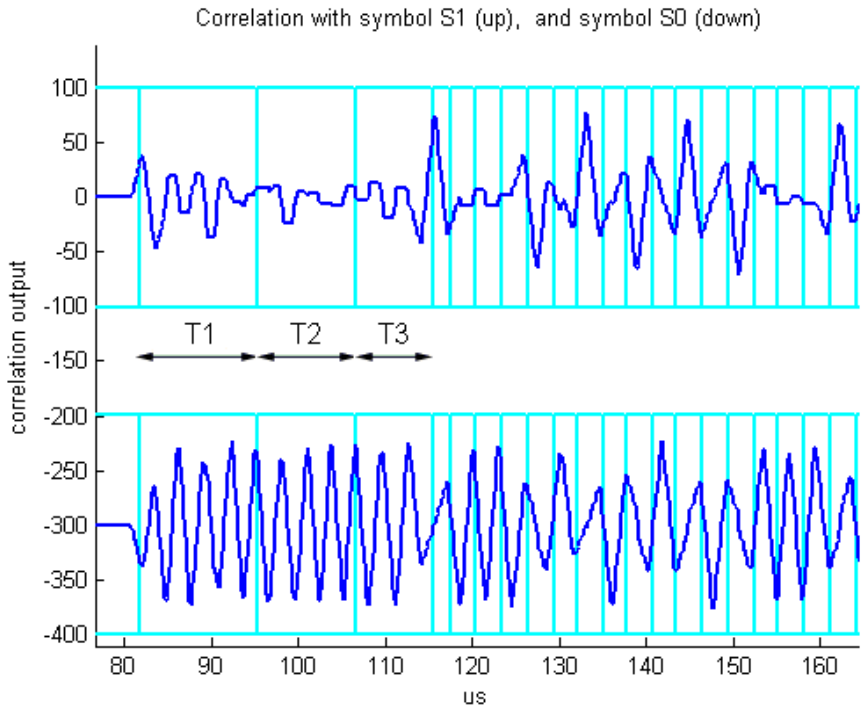


Figure 5.27. Frequency synchronization, BLF 380 kHz, SNR 3 dB, temporal deviation -18%, short preamble.

Figure 5.28 shows the time deviation estimation that corresponds to the example in Figure 5.27. It is shown here that the time deviation estimation changes more abruptly to compensate the shorter length of the preamble.

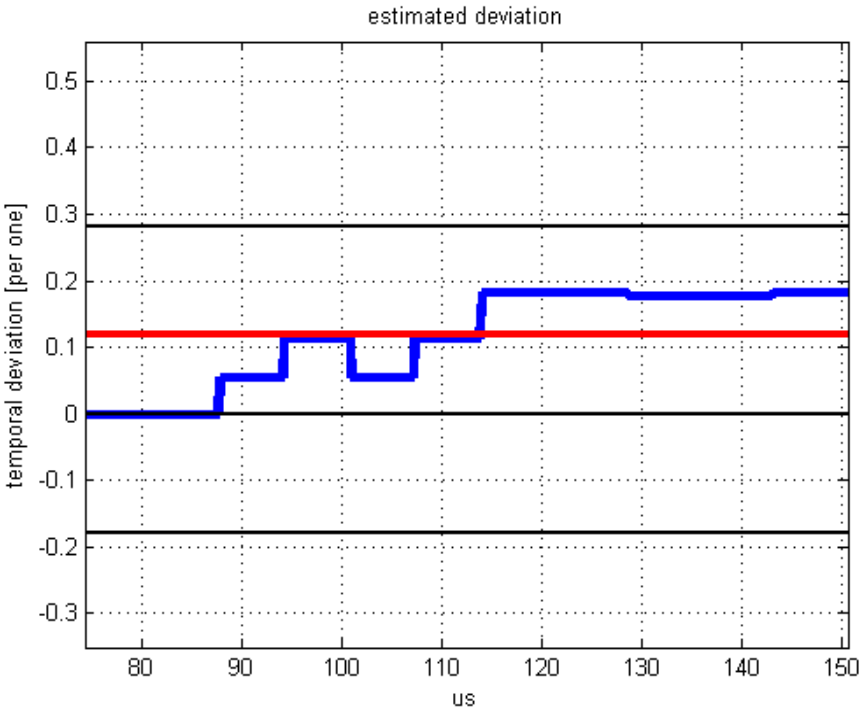


Figure 5.28 Blue: Estimated frequency, red: Average received signal frequency, BLF 380 kHz, SNR 3 dB, short preamble.

5.4.3 Symbol detection

The synchronization controller is in charge for the timing of the various stages regarding the synchronization. This block has been added to the synchronizer model and is shown in Figure 5.29.

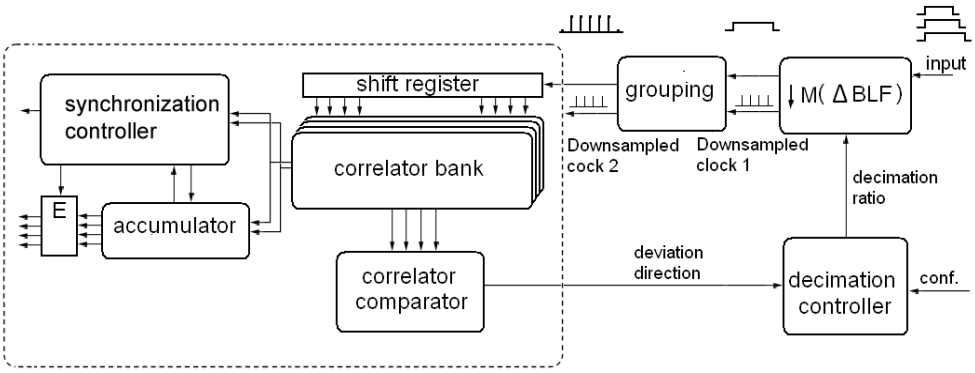


Figure 5.29. Synchronizer model, frequency synchronization

During the pilot sequence the synchronization controller decides the interval change for T1, T2 and T3, indicating when to memorize the correlation maximums, calculates the threshold value for the first S1 symbol, and searches for it.

During the symbol detection stage the synchronization controller decides the start and end of each half symbol interval to look for the symbols. It follows the rules defined in section 5.2 of this chapter. For this purpose, the synchronizer is connected to the main correlators output and to the block denoted as accumulator, which memorizes the maximum values for the main correlation with the four Miller symbols.

Figure 5.29 shows a segment of the main correlation signals in absolute value during the reception of Miller M=2 symbols. The correlation with symbol S1 is represented in continuous blue color, while the correlation with S0 is represented in continuous black. The negative part of these signals correspond to the symbols S3 and S2, they are shown positive because the absolute value representation and in discontinuous blue and black lines respectively.

In the first interval of this example, the synchronizer detects that the symbol S1 has reached a higher value than the other symbol correlations. According to the defined rules the synchronizer readjusts its reference clock to the correlation information. The synchronizer detects that correlation peak for symbol S1 is delayed in comparison to the interval center. The synchronization is performed delaying the end of the next interval to compensate the delay.

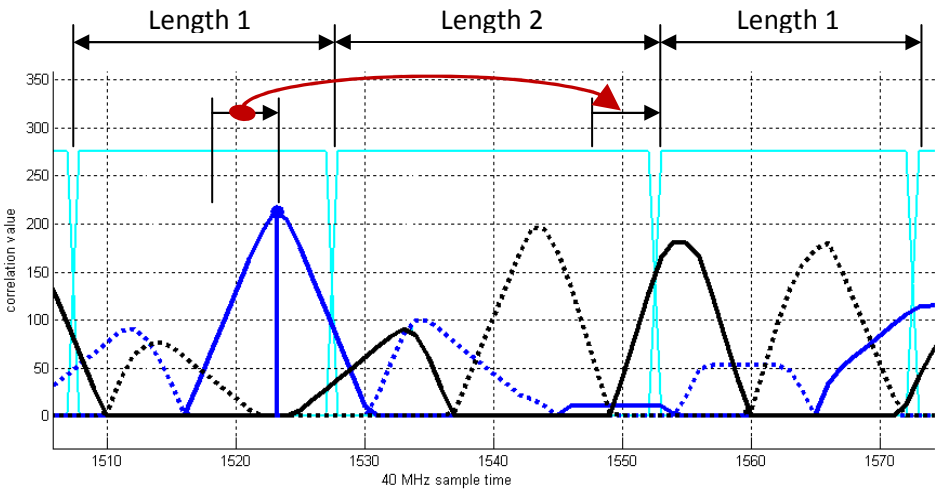


Figure 5.30. Reference clock re-synchronization

At the end of each even interval, which correspond to a real symbol, the four maximum correlation values recorded for each symbol are provided at the synchronization output together with a synchronization signal that is activated to indicate the decoder the validation of the new data.

5.5 Implementation

5.5.1 Algorithm simulation

In the first stage of design, the synchronization unit is programmed in Matlab, since this provides the fastest way to verify its proper functionality and make the appropriate modification as well as the algorithm adjustments. In Matlab many aspects regarding hardware implementation are not taken into account, as for example the way of performing certain calculations, or the resource consumption. The images shown up to now in this chapter correspond to the Matlab design.

Stimulation signals

Generating input signals is necessary to evaluate the synchronizer. First a testbench is created that is able to generate any configuration of the Miller signals defined by the UHF standard, taking into account the BLF tolerances.

Since the received signal passes through many stages in the receiver, the signal utilized to stimulate the synchronizer has been altered to simulate the similar effect. AWGN noise is added to the ideal baseband signal. Then, it is passed through a matched filter that according to the maximum link frequency tolerance, BLF*(1+0.22). According to an ideal threshold the signal is sliced in two levels. The result signal is shown in Figure 5.31 for BLF 350 kHz. Figure shows that the errors concentrate near the pulse edges, specially influencing them by making them thinner and broader.

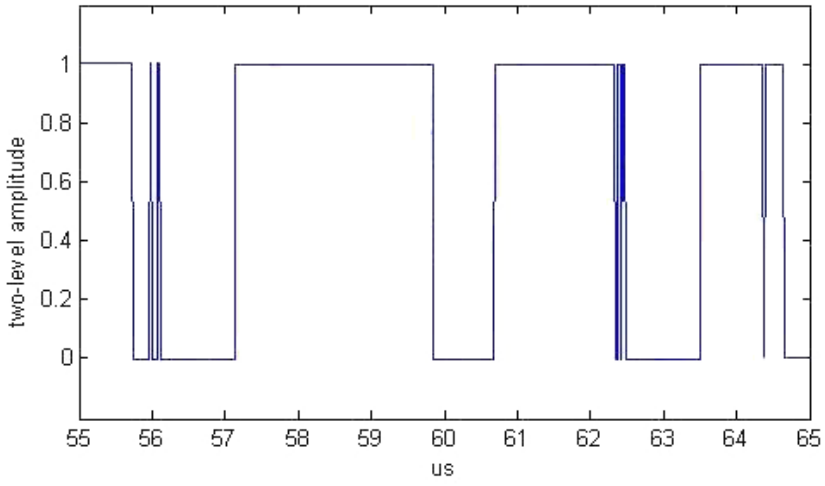


Figure 5.31. Synchronizer stimulation signal over sample cycle

5.5.2 Simulink implementation

The implementation of the final model is carried out in Simulink. Every part of the synchronizer is tested independently and as a whole. Since it is highly

compatible with Matlab, the input signals generated for the Matlab test are imported and used in this model too.

For the Simulink development two additional tools have been utilized. For several tasks as counters or switches, embedded Matlab functions have been included. The synchronization controller has been develop with Stateflow[14], which is a tool specialized for the design of flow diagrams compatible with Simulink and to a certain extend with Simulink HDL Coder. These limitations have to be taken into account in the Stateflow design.

Varying the configuration parameters of the input signal the correct functionality of the synchronizer is tested. For example, Figure 5.32 and Figure 5.33 show some screenshots of a realization for Miller coding M=2 for BLF 100 kHz, -20% frequency deviation and SNR 3 dB. The first one shows the output of the main correlators and the synchronization signal indicating the position of each symbol detected. In the second one the synchronization signal is shown again and the maximum correlation values that correspond to each of the symbols.

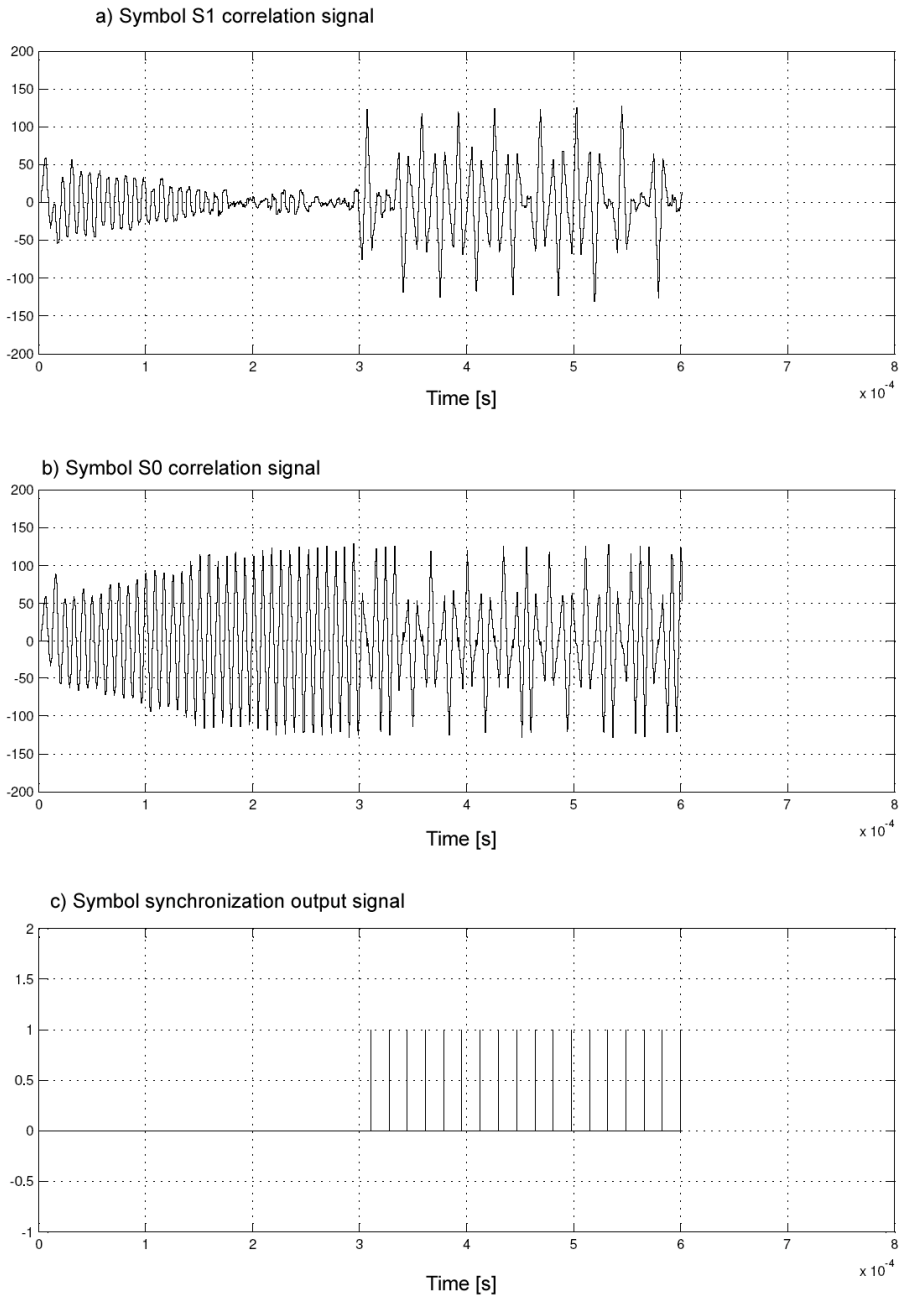


Figure 5.32 Simulink simulation capture: a) Symbol S1 correlation. b) Symbol S0 correlation. c) Synchronization signal.

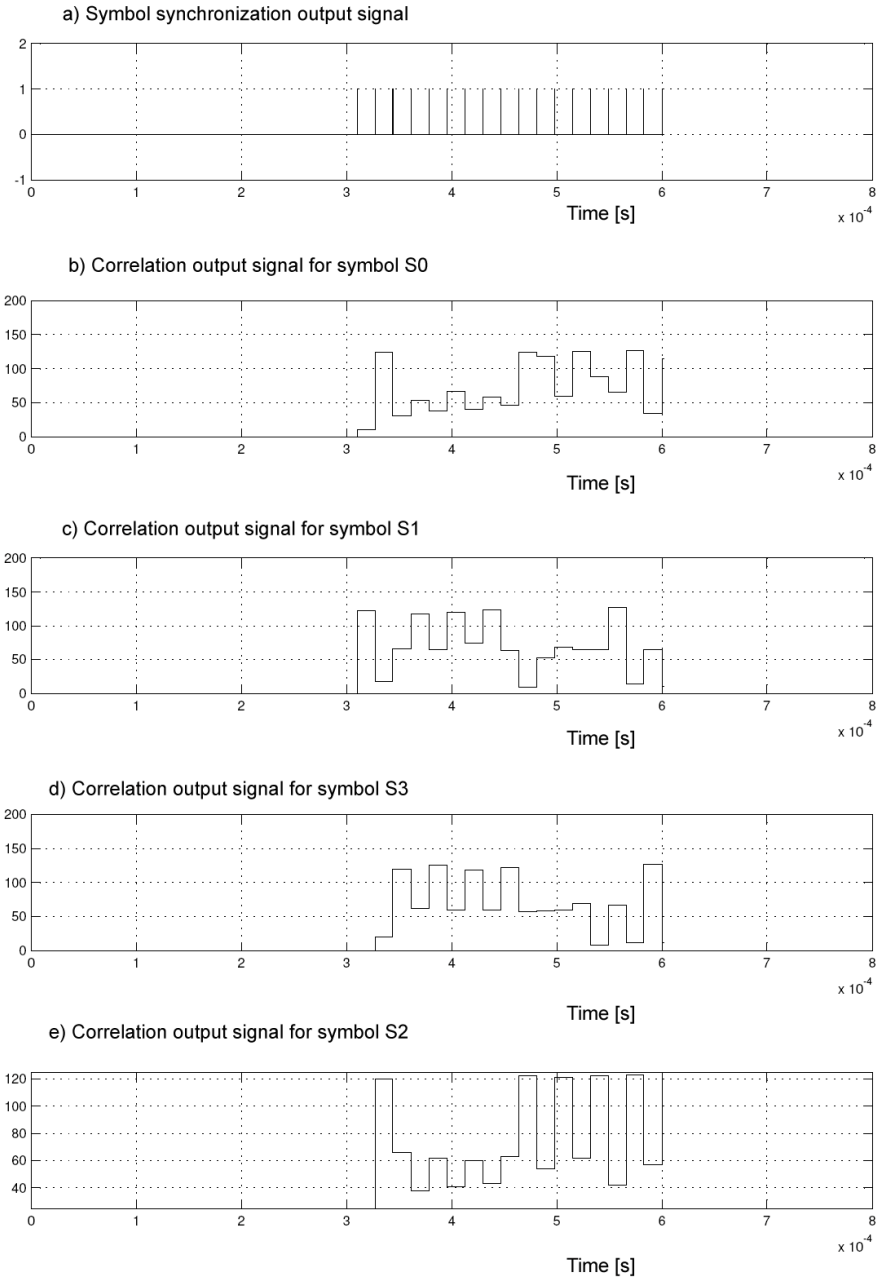


Figure 5.33. Simulink simulation capture: a) Synchronization signal, b) S0 correlation output, c) S1 correlation output, d) S3 correlation output, e) S1 correlation output.

5.5.3 ModelSim verification

The Simulink synchronizer model is converted to VHDL code by Simulink HDL Coder. The resulting code is instantiated in the receiver, where the it is provided with the configuration signals.

The complete system in synthesized to verify complete compatibility with the FPGA device. This procedure has been carried out several times during the synchronizer design to verify the validity of the methods or decide the appropriate modifications.

The stimulation signals generated previously are used as testbench in the ModelSim receiver model. The stimulation signals are connected directly to the synchronizer unit input. Figure 5.34 shows the result of a ModelSim realization for BLF 380 kHz, -22% frequency deviation and SNR 3 dB.

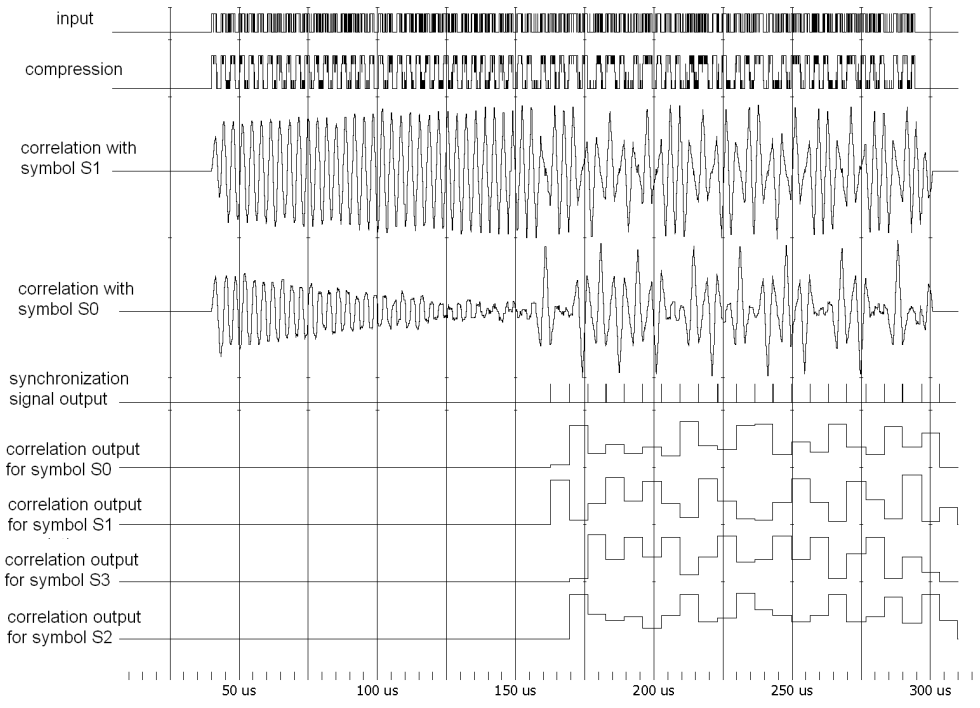


Figure 5.34. ModelSim realization for Miller M=2, BLF = 380 kHz, frequency deviation -22% SNR 3 dB.

6 Results

6.1 Filter Bank Bandwidth

The real bandwidth that the filters present in practice is checked. For this purpose the receiver model is loaded on the FPGA board and provided with white noise at the analog to digital input. The input is connected to the filter bank and its output connected again to the digital to analog converter so that the frequency response can be measured with a spectrum analyzer. Figure 6.1 and Figure 6.2 show as example the bandwidths measured for 40 kHz BLF and 640 kHz BLF, which respective bandwidths are 480 kHz and 7.68 MHz.

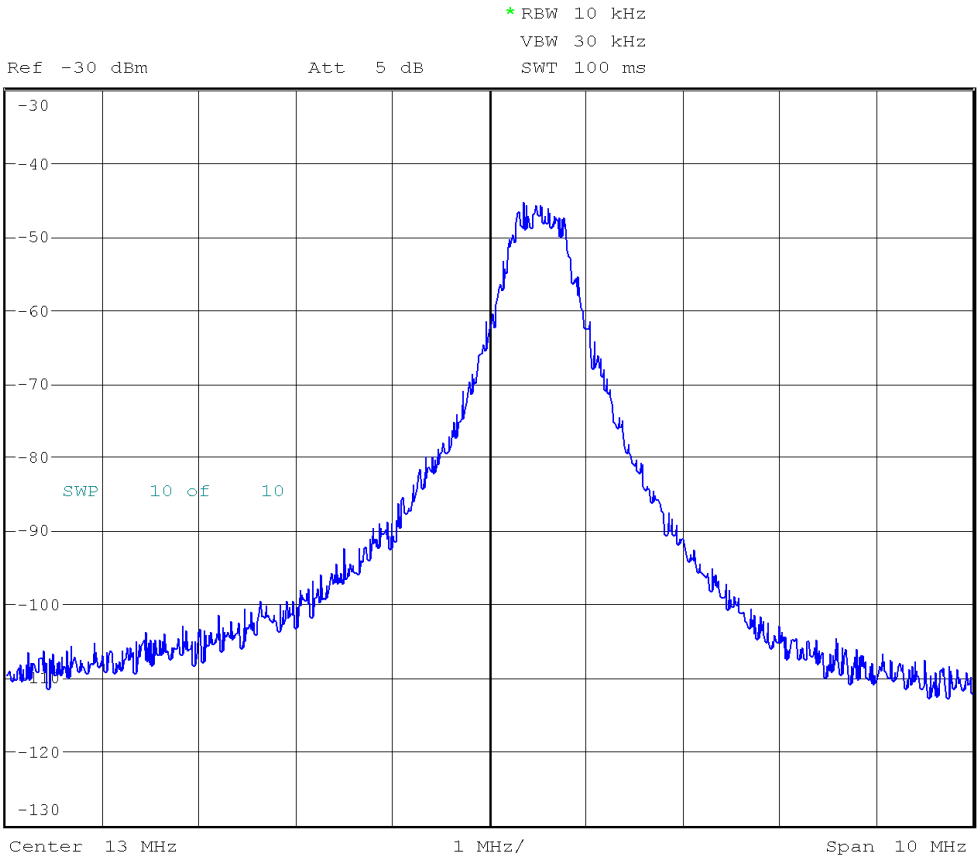


Figure 6.1. Frequency response of filter for BLF= 40 kHz, BW=40 kHz

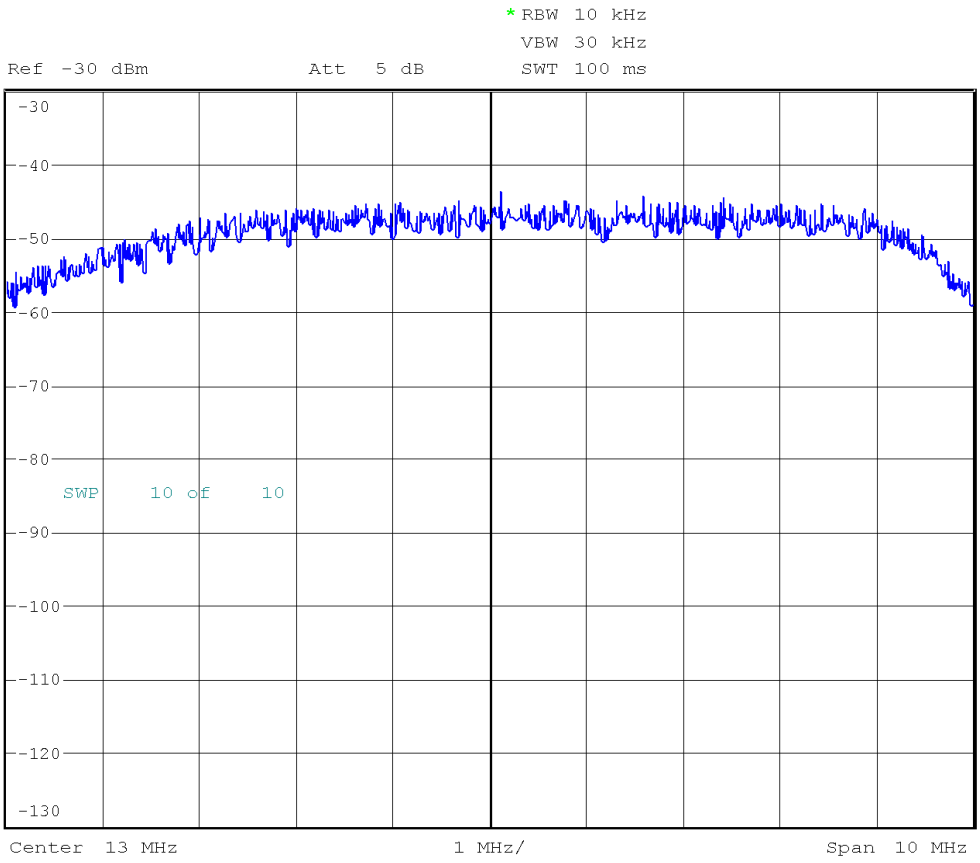


Figure 6.2. Frequency response of filter of BW=7.68 MHz

6.2 Time Domain Measurements.

Figure 6.3 shows a time based representation of the input signal to the receiver. The upper part shows the signal provided externally to the system. The lower part is the signal measured when the input signal is directly output without any influence of the receiver units. This shows the Digital to analog effect on the measurements isolated. The arrows in the figure indicate the signal parts corresponding to the transmit and received signal.

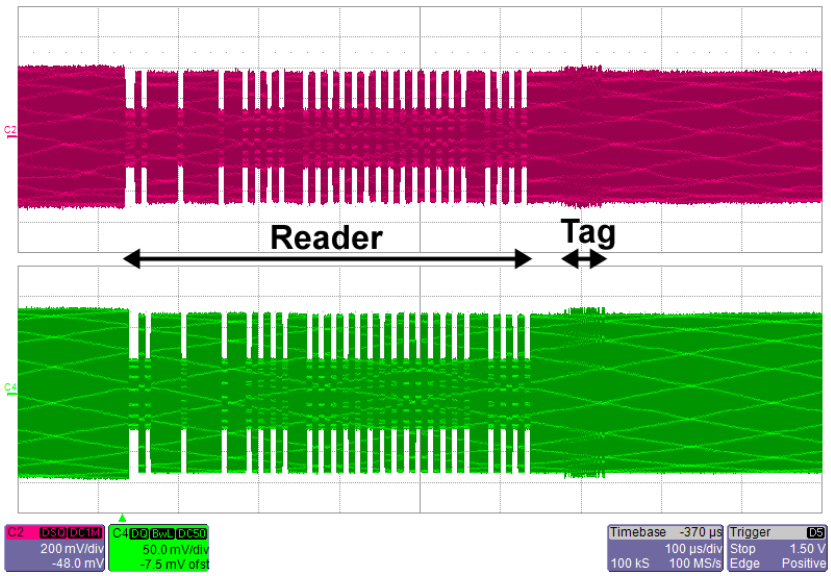


Figure 6.3. ADC and DAC effect

Figure 6.4 focuses on the part that contains the receive signal. This is the one of interest regarding the receiver evaluation.

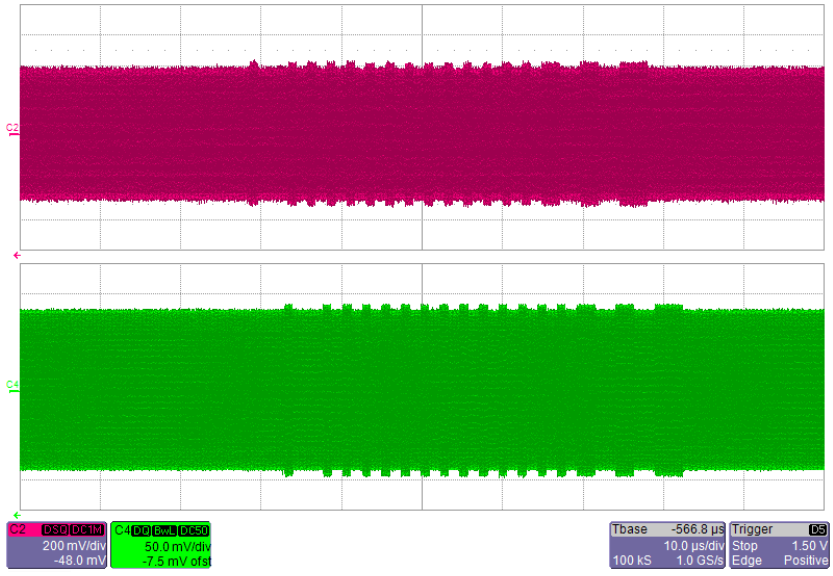


Figure 6.4. ADC and DAC effect, zoom to tag response.

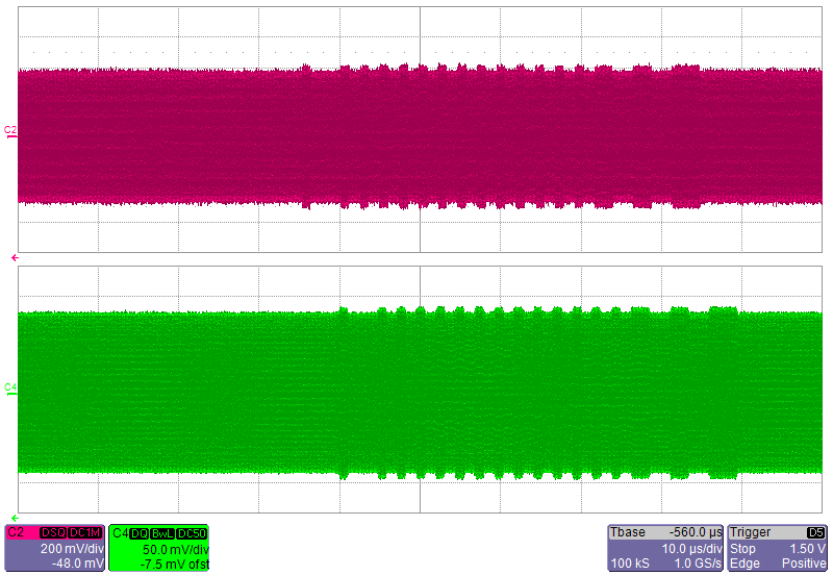


Figure 6.5. Input versus signal after 10 MHz band pass filer.

Figure 6.5 shows the measurement after the bandpassfilter bank. In the time domain no meticulous details can be observed.

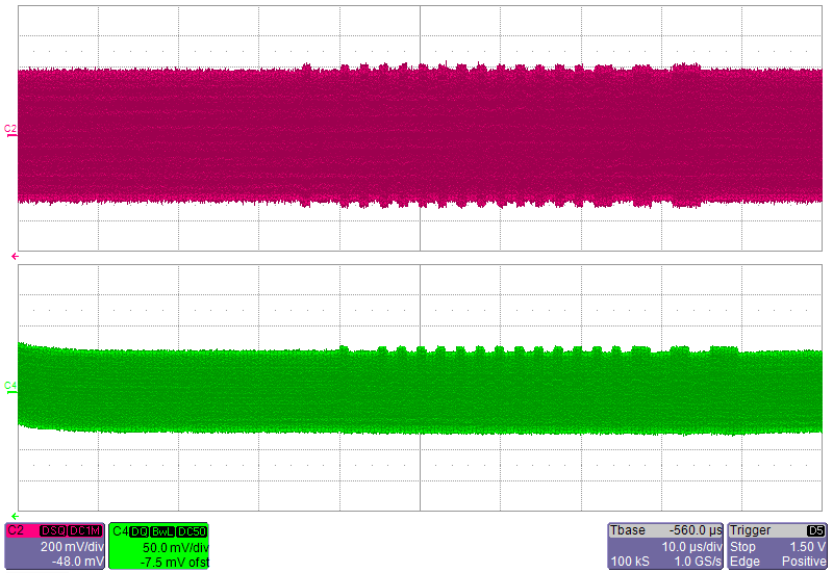


Figure 6.6. Input versus in-phase multiplier output.

Figure 6.6 shows the effect of the in-phase demodulator multiplier. In this point the signal contains both the baseband component and the image component. Figure 6.7 shows that the baseband signal clearly remains after the low-pass filter.

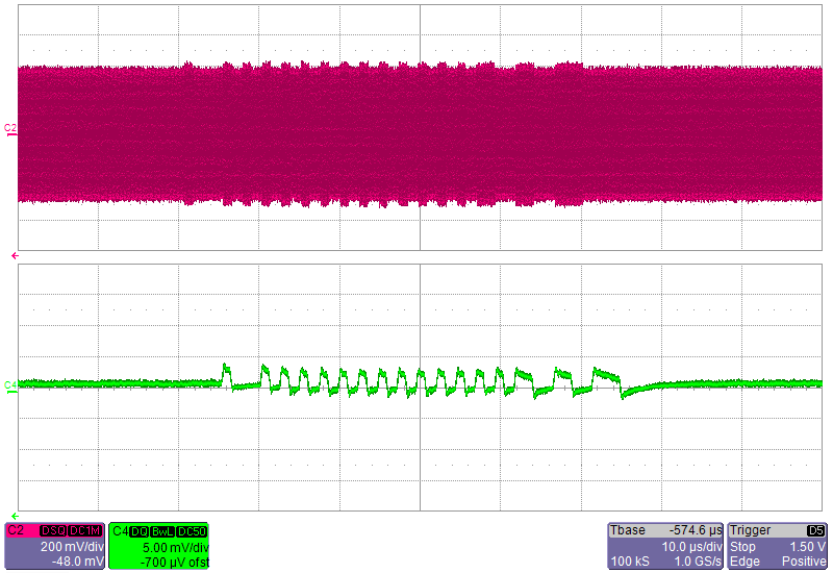


Figure 6.7. Input and low-pass filter output at the demodulator in-phase branch.

The two levels can be clearly differentiated in the baseband signal. Because of a high pass filter effect of the connections the signal does not keep in a constant levels.

Figure 6.8 shows the result of the matched filter. Here again the signal does not maintain a constant level because of the high pass filter external to digital design of the receiver.

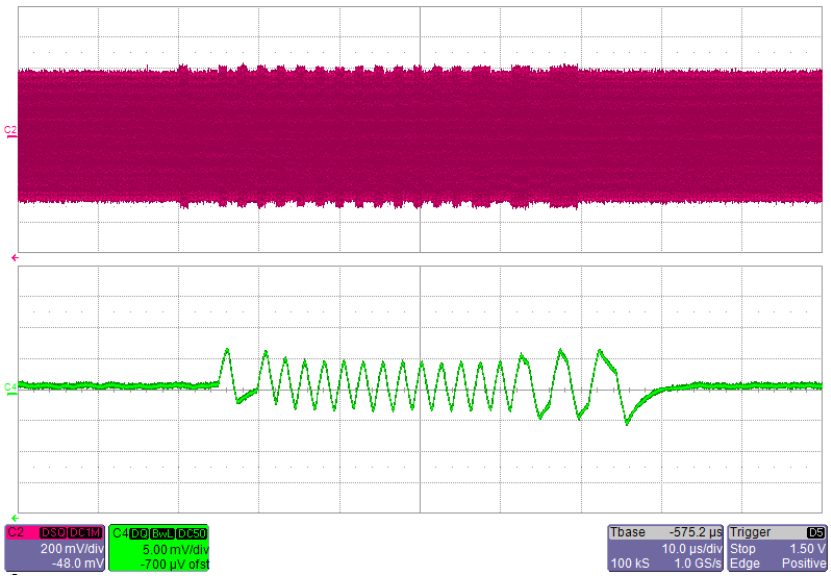


Figure 6.8. Input and matched filter output, integration length 24 samples.

6.3 Synchronizer BER

Bit error rate evaluations have been calculated using the Matlab implementation of the algorithm to evaluate the synchronization unit performance.

Figure 6.9 shows simulations for link frequencies of 645 kHz, 137 kHz and 76 kHz, in blue, green and red color respectively. For these realizations 10,000 repetitions of 16 bits sequences have been performed. The discontinuous line indicates the probability that the first symbol of each transmission is detected in the right time. If this doesn't happen the whole symbol sequence will be detected erroneously. The remaining realizations have been utilized to calculate the BER, which is represented by the continuous lines.

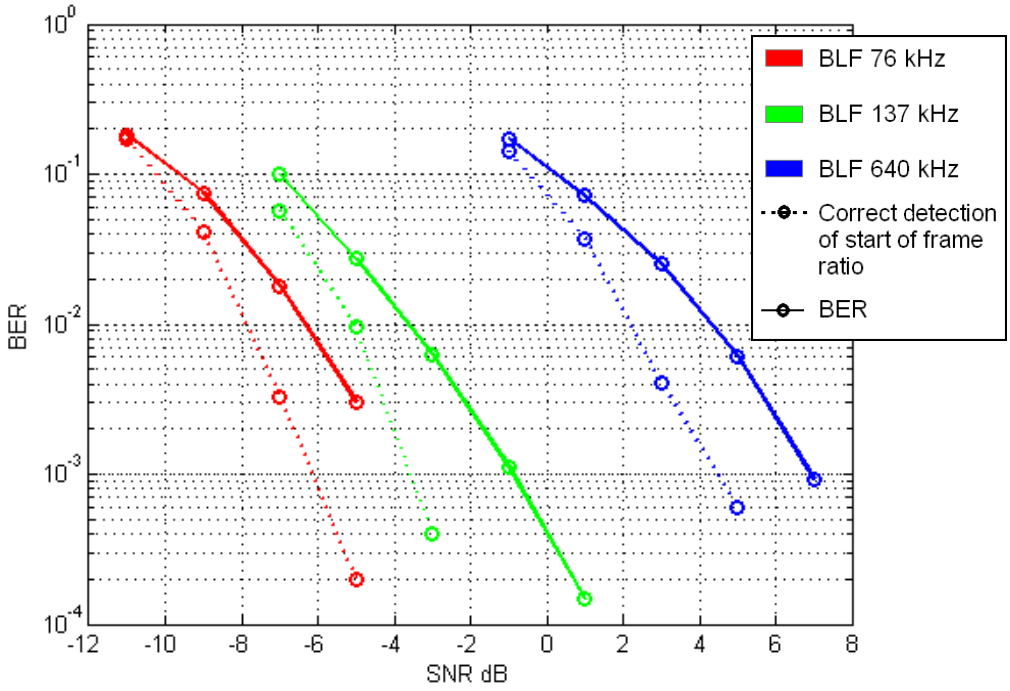


Figure 6.9. Detection probabilities, long pilot.

In Figure 6.10 the same simulation has been made for short pilot transmissions.

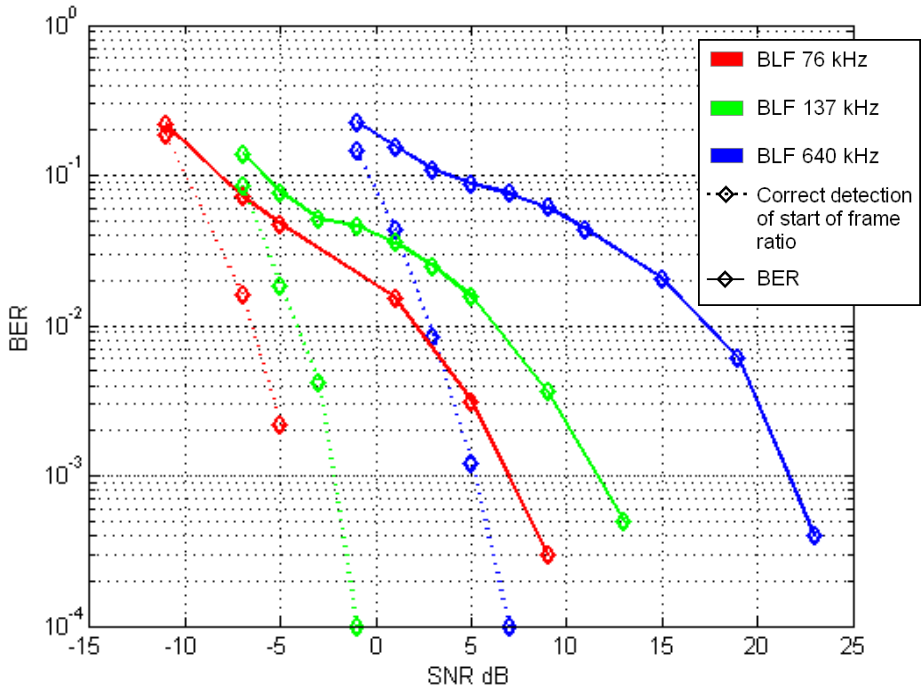


Figure 6.10. Detection probabilities, short pilot.

In Figure 6.11 and

Figure 6.12 short and long pilot simulations are compared. Long preamble simulations achieve a better link frequency synchronization accuracy. The longer pilot time enables a slower and more precise estimation, and therefore shows lower bit error ratio.

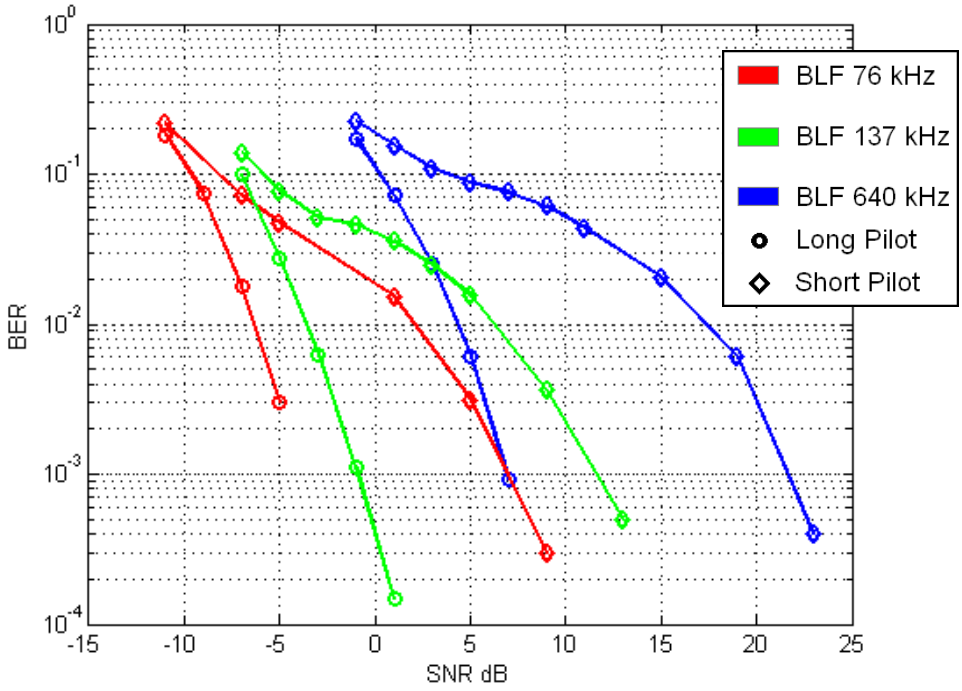


Figure 6.11. Short pilot and long Pilot BER.

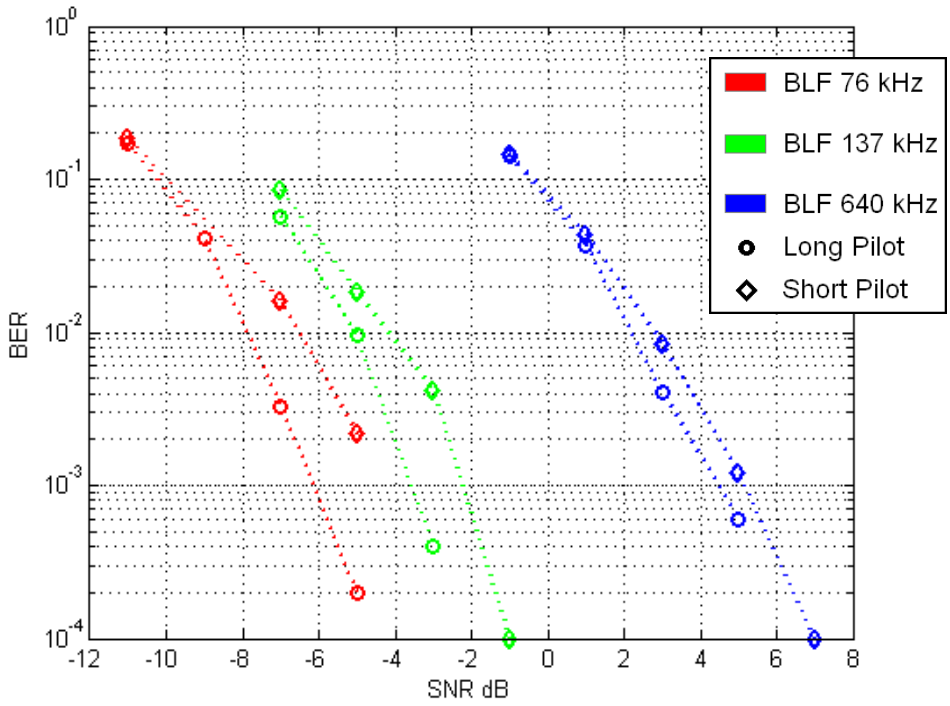


Figure 6.12. Short pilot and long pilot probability of correct detection of start of frame

Figure 6.13 shows the variation of the BER in the case for BLF 640 kHz and SNR of 3 dB over the deviation of the received signal frequency.

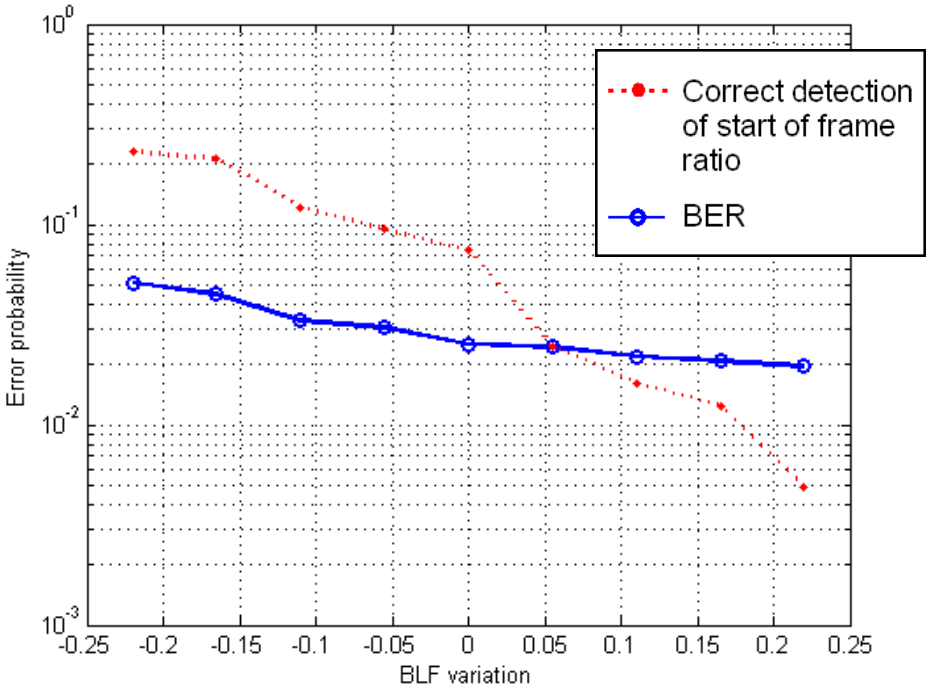


Figure 6.13. Start of sequence detection and BER over BLF variation. Requested BLF 640 kHz, SNR 3 dB.

7 Conclusions

This project carries out various design, implementation and testing activities required in the development of the RFID reader prototype to meet the specifications of the ECP Global standard for UHF RFID [1] maintaining the HF domain operation capability of the prototype.

This project contributes to the RFID reader by integrating the digital implementations of an entrance filter bank, an image removal filter and a matched filter in the receiver signal processing section, providing the flexibility required by the mentioned standard for UHF.

A reconfigurable correlation based synchronizer algorithm for Miller modulation has been designed and evaluated. Among the implemented components, the synchronizer design is the most challenging one, since it has to meet the demanding requirements of the standard and since no documentation was found during the development of this project describing a digital design solution for this encoding type. Hence this work sets a first approach to Miller coding correlation based synchronization.

The implementations of this project are developed by means of automated design flow tools. Measurements are taken to verify compatibility and functionality of the designed algorithms. These measurements also show the benefit of the rapid prototyping concept. The designs, delivered in this highly efficient environment, provide the benefit of a fast verification capability and an efficient testing way of parameter variations or new improvements.

Finally, besides the technical results, these developments permitted the author to face real design tasks in a challenging environment providing a complementary education and further development of skills to undertake more complex forthcoming designs.

8 References

[1] EPC Global, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID," http://www.epcglobalinc.org.

[2] C. Angerer, B. Knerr, M. Holzer, A. Adalan, and M. Rupp, "Flexible Simulation and Prototyping for RFID Designs," *in First International EURASIP Workshop on RFID Technology*, Vienna, Austria, Sept. 2007, pp. 51–54.

[3] C. Angerer, "A digital receiver architecture for RFID readers," in *IEEE 3rd International Symposium on Industrial Embedded Systems*, Montpellier, France, June 2008, pp. 97–102.

[4] C. Angerer, M. Holzer, B. Knerr, and M. Rupp, "A flexible dual frequency testbed for RFID," in *Trident*Com *'08: Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, Innsbruck, Austria, 2008.

[5] Liu, C. Huang, H. Min, G. Li, and Y. Han, "Digital correlation demodulator design for RFID reader receiver," *in Wireless Communications and Networking Conference (WCNC 2007)*, Kowloon, China, March 2007, pp. 1664–1668.

[6] C. Mutti and A. Wittneben, "Robust Signal Detection in Passive RFID Systems," *in First International EURASIP Workshop on RFID Technology*, Vienna, Austria, Sept. 2007, pp. 39–42.

[7] Xilinx LogiCORE™ FIFO Generator v 3.2. User guide: UG175 September 21, 2006 at: http://www.xilinx.com/support/documentation/ip_documentation/fifo_generator_ug175.pdf

[8] FDATool User Guide in the Signal Processing Toolbox of MATLAB www.mathworks.com

[9] Virtex-II Platform FPGAs User Guide at www.xilinx.com.

[10] EPC Global organization www.epcglobalinc.org

[11] Simulink: Simulation and Model-Based Design tool for dynamic and embedded systems. http://www.mathworks.com/products/simulink/ User Guide at http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/

[12] Simulink HDL Coder: Simulink to HDL code model conversor www.mathworks.com/products/slhdlcoder

[13] Xilinx ISE and System Generator are EDA tools especially for FPGA design and automatic conversion from Simulink to VHDL, http://www.xilinx.com.

[14] Stateflow version 7.3: State Design and simulate state machines and control logic www.mathworks.com/products/slhdlcoder

ATTACHEMENT:

Frequency Response Measurements of the Entrance Filter Bank.

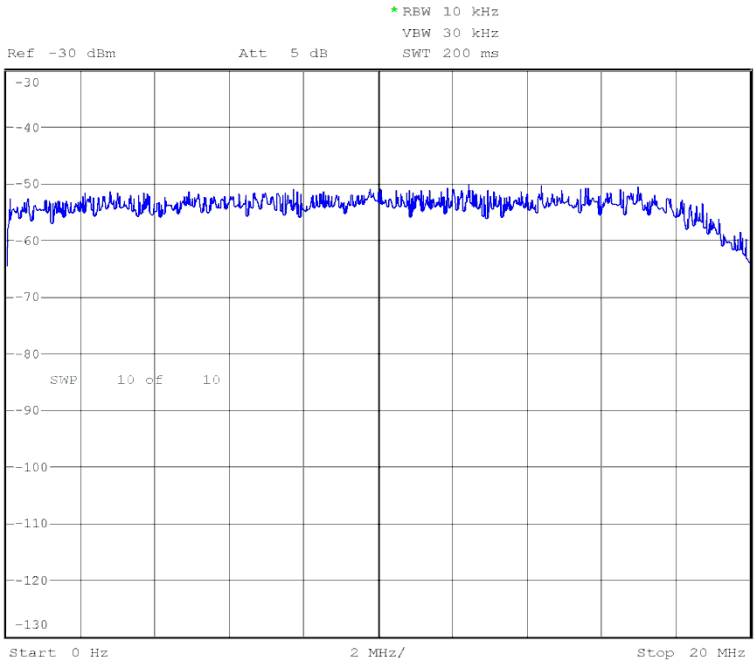


Figure 14. Influence on AWGN measurements of the ADC interface.

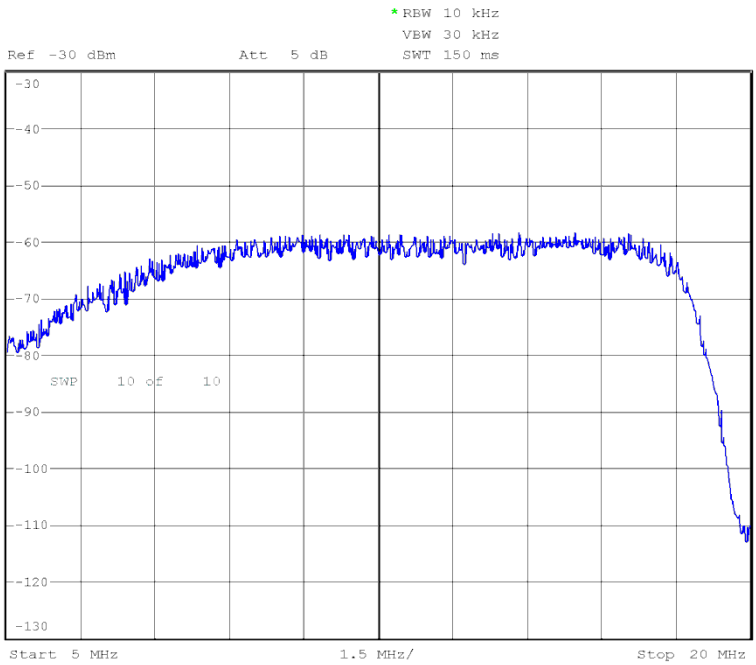


Figure.15. Frequency response of the HF filter. BW=10.16 MHz

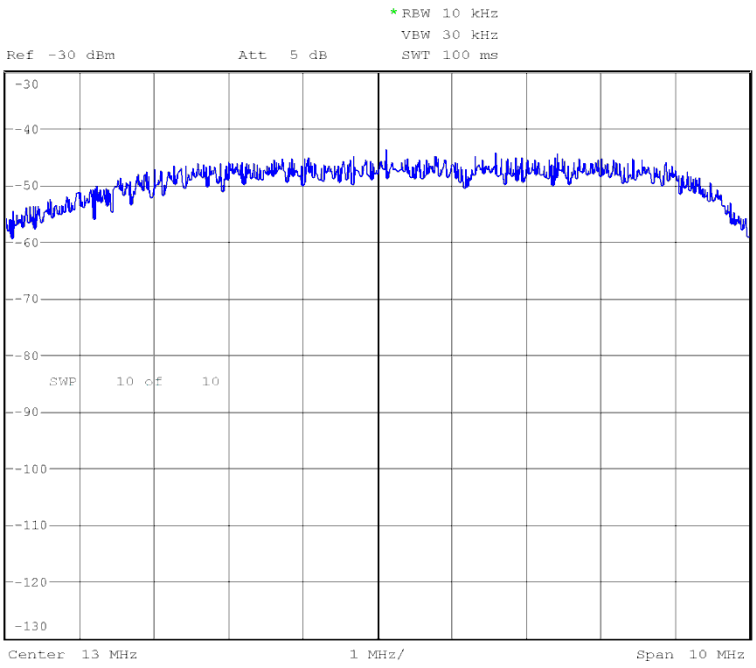


Figure 16. Frequency response of filter for BLF= 640 kHz. BW =7.68

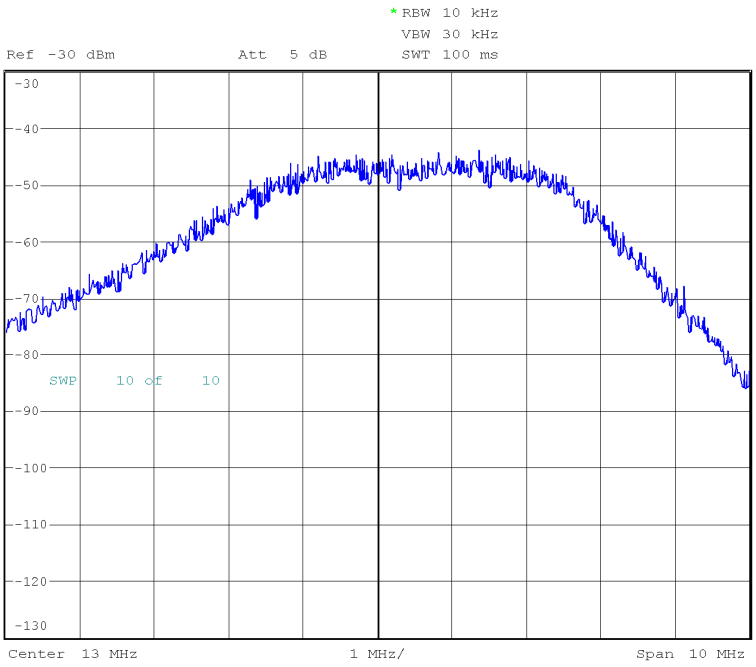


Figure 17. Frequency response of filter for BLF=320 KHZ. BW =3.84 MHz

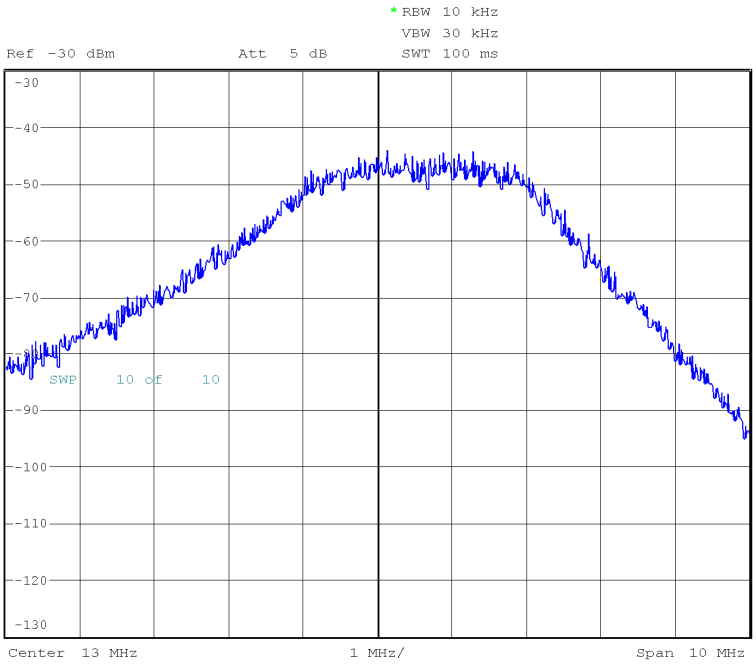


Figure 18. Frequency response of filter for BLF=256 KHZ. BW =3.07 MHz

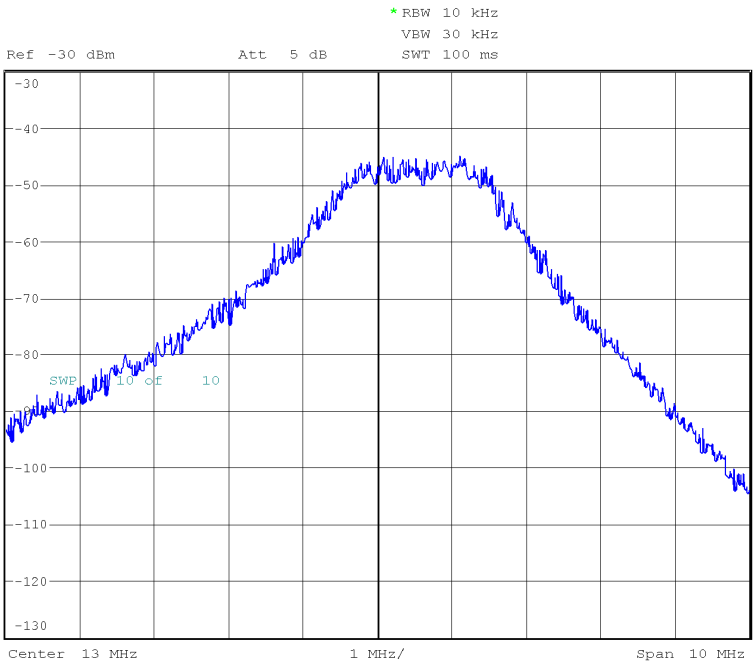


Figure 19. Frequency response of filter for BLF=160 kHz. BW =1.92 MHz

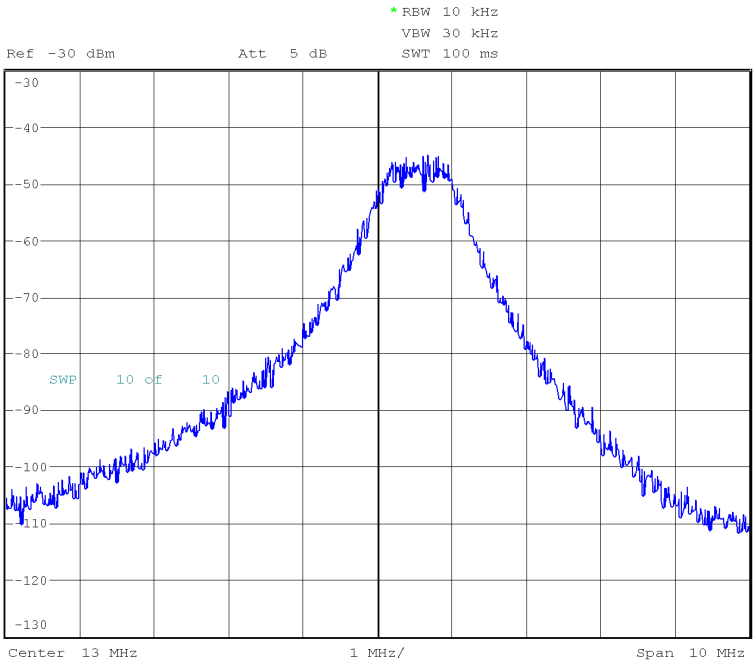


Figure 20. Frequency response of filter for BLF= 107 kHz. BW =1.28 MHz

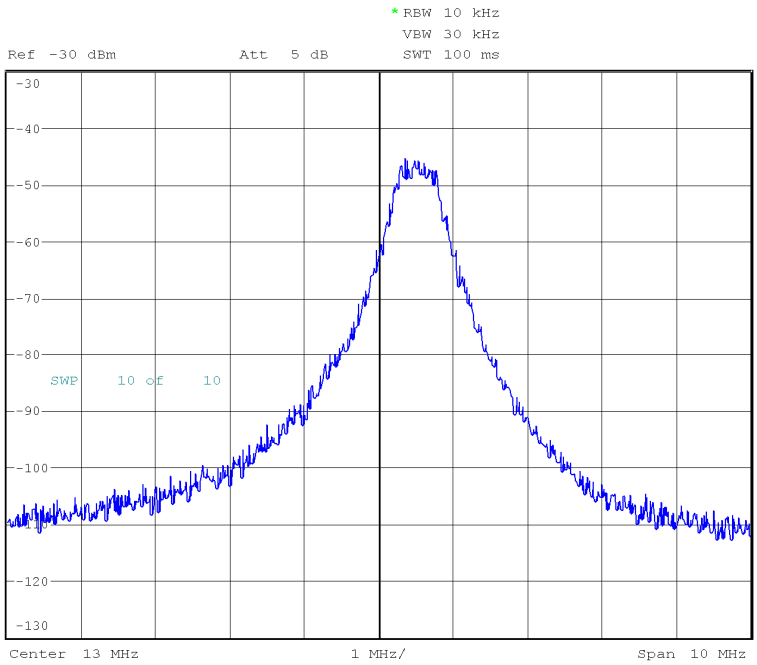


Figure 21. Frequency response of filter for BLF=40 kHz. BW =480 kHz