# Low Power System Design of DDPSK Receiver

MSc. Assignment By Victor van Rooij v.a.vanrooij@student.utwente.nl

Supervisors: S. Safapourhajari, MSc. dr. ir. A.B.J. Kokkeler dr. ing. D.M. Ziener dr. ir. M.S. Oude Alink

13th June 2019

University of Twente - CAES Electrical Engineering

## Contents

| Lis | st of | abbreviation and keywords                      | 3  |
|-----|-------|--|----|
| 1   | Intro | oduction                                       | 5  |
|     | 1.1   | Research Questions                             | 6  |
|     | 1.2   | Outline of the Thesis                          | 7  |
| 2   | The   | ory  | 8  |
|     | 2.1   | Autocorrolation Demodulator                    | 9  |
|     |       | 2.1.1 Conventional Autocorrolation Demodulator | 9  |
|     |       | 2.1.2 Proposed Autocorrolation Demodulator     | 13 |
|     | 2.2   | Conversion to baseband                         | 15 |
|     | 2.3   | Decimation                                     | 15 |
|     |       | 2.3.1 Finite Impulse Response                  | 16 |
|     |       | 2.3.2 Infinite Impulse Reponse                 | 17 |
|     |       | 2.3.3 Cascade Integrator-Comb                  | 17 |
|     | 2.4   | Power use in ASIC                              | 19 |
| 3   | Syn   | thesis investigation                           | 21 |
|     | 3.1   | Synthesis properties                           | 21 |
|     | 3.2   | Implementation of arithmetic elements          | 22 |
|     | 3.3   | Result of Power estimation                     | 22 |
|     | 3.4   | Design Considerations                          | 24 |
| 4   | Des   | ign  | 25 |
|     | 4.1   | Demodulator                                    | 25 |
|     | 4.2   | Baseband conversion                            | 28 |
|     | 4.3   | Decimation                                     | 28 |
|     | 4.4   | Quantisation of each part                      | 30 |
|     | 4.5   | Overview of design                             | 32 |
| 5   | Imp   | lementation                                    | 33 |
|     | 5.1   | Downsizing                                     | 33 |

|    | 5.2    | Conve   | rting Design to VHDL                      | 35 |
|----|--------|---------|---|----|
|    |        | 5.2.1   | Control block                             | 35 |
|    |        | 5.2.2   | Baseband conversion                       | 36 |
|    |        | 5.2.3   | CIC filter                                | 36 |
|    |        | 5.2.4   | FIR filter                                | 36 |
|    |        | 5.2.5   | Autocorrolation Demodulator               | 37 |
| 6  | Res    | ults    |   | 39 |
|    | 6.1    | Trade-  | offs in design                            | 39 |
|    | 6.2    | Perfor  | mance of implementation                   | 39 |
|    | 6.3    | Power   | consumption                               | 41 |
| 7  | Con    | clusion |   | 44 |
|    | 7.1    | Discus  | sion                                      | 44 |
|    |        | 7.1.1   | Trade-offs in design                      | 44 |
|    |        | 7.1.2   | Performance                               | 45 |
|    |        | 7.1.3   | Power consumption                         | 45 |
|    | 7.2    | Propos  | sed low power Autocorrolation Demodulator | 45 |
|    | 7.3    | Recom   | nmendations                               | 46 |
|    |        | 7.3.1   | Tuneable design                           | 46 |
|    |        | 7.3.2   | Reducing power consumption                | 46 |
| Li | st of  | Figures | 5   | 48 |
| Li | st of  | Tables  |   | 50 |
| В  | ibliog | raphy   |   | 51 |

# List of abbreviations and keywords

| ACD       | Autocorrolation Demodulator  |
|-----------|--|
| ADC       | Analog to Digital Converter  |
| ASIC      | Application-Specific Integrated Circuit  |
| BER       | Bit Error Rate   |
| BW        | Bandwidth  |
| BPSK      | Binary Phase Shift Keying  |
| Conv. ACD | Conventional Autocorrolation Demodulator<br>Double Differential PSK demodulation as explained in [1] |
| Prop. ACD | Proposed Autocorrolation Demodulator<br>Double Differential PSK demodulation as proposed [2]         |
| CIC       | Cascade Integrator-Comb  |
| DDPSK     | Double Differential Phase Shift Keying   |
| DSP       | Digital Signal Processing  |
| DSC       | Digital Signal Conditioning  |
| $E_b/N_0$ | Energy per Bit to Noise power spectral density ratio   |
| FIR       | Finite Impulse Response  |
| Fs        | sampling frequency   |
| IF        | Intermediate Frequency   |
| IIR       | Infinite Impulse Reponse   |
| LNA       | Low Noise Amplifier  |
| LSB       | Least Significant Bit  |
| MSB       | Most Significant Bit   |
| PR        | Performance Ratio  |
| PSK       | Phase Shift Keying   |
| s/S       | samples per Symbol   |
| SNR       | Signal to Noise Ratio  |

| VNB   | Very Narrowband                     |
|-------|-------------------------------------|
| VHDL  | VHSIC Hardware Description Language |
| VHSIC | Very High-Speed Integrated Circuit  |
| WSN   | Wireless Sensor Networks            |

# Introduction

Within the electromagnetic spectrum for wireless communication, bands are getting more crowded over the last few years. This is because of the evolvement of technology and the ease of connecting devices wirelessly. An example of a system with a lot of wireless devices are Wireless Sensor Networks (WSN). They consist of multiple cheap and low power sensor nodes that try to communicate with each other and/or with a base station [3].

There is only limited spectral space for devices to communicate wirelessly. The crowded wireless bands create a challenge for WSN nodes to communicate. This is because communicating in crowded wireless bands consume significant amount of power. A solution for these WSN nodes is sought within the Slow Wireless project [4]. The main goal of the project is to use Very Narrowband (VNB) signals and transmit data with an data rate in the order of 100 bits/s.

When VNB signals are used, the receiver and transmitter can easily get out of the band. This is due to the carrier frequency offset. The frequency offset can be caused by Doppler shift or mismatch between oscillators. It is possible to overcome the mismatch in oscillators by using expensive components that are more precise, however this is not preferred when using cheap sensor nodes. Conventional methods to overcome frequency offset can be too power hungry for low power sensor nodes. To address this challenge the Autocorrolation Demodulator (ACD) for Double Differential Phase Shift Keying (DDPSK) can be used [1]. It is a frequency offset tolerant demodulator which is able to overcome frequency offset without extra carrier synchronization modules.

The ACD is able to tolerate frequency offset but only to a limited frequency offset The digital demodulator is limited by its sampling frequency and filter before the demodulator as seen in Figure 1.1a. Figure 1.1b shows the frequency spectrum at the input of the demodulator. The signal that is received should be somewhere within the filter Bandwidth (BW). If there is a lot of frequency offset, the received signal might move out of the filter BW. To ensure the data stays within the BW, it would make sense to increase the filter BW. However, by increasing the filter BW the noise BW is also increased.

When using the Conventional Autocorrolation Demodulator (Conv. ACD) as described in [1], the Bit Error Rate (BER) performance will degrade when the filter BW increases.

The Proposed Autocorrolation Demodulator (Prop. ACD) is an extension to the Conv. ACD as proposed by [2]. The Prop. ACD is able to compensate for the increased noise BW using multiple parallel demodulators. With this it has a better BER performance than the Conv. ACD. The theory behind this will be elaborated in section 2.1.

The main focus of this thesis is to make a power efficient design of the digital part of a DDPSK receiver which is using an ACD. The design is evaluated and compared with theory for validation.



Figure 1.1: (b) Signal spectrum between the filter and demodulator. Data BW indicates the BW of the signal of interest.

#### 1.1 Research Questions

A wireless receiver consists of multiple parts, as can be seen in Figure 1.2 [5]. The part that will be designed is indicated with the dashed rectangle. Proceeding from here it will be called 'the system'. The output of the Analog to Digital Converter (ADC) (input for the system) is assumed to consist of two signals, an in-phase component a quadrature-phase component. The received signal is mixed down to a Intermediate Frequency (IF) of 1 MHz during the analog signal conditioning before the ADC. The output of the system will be the symbols detected by the DDPSK demodulator. The system itself consists of two parts, the Digital Signal Conditioning (DSC) and ACD.



Figure 1.2: The focus of the design indicated with the dashed rectangle. (The filter of Figure 1.1a is incorporated into the DSC)

In order to investigate a power-efficient design, the thesis will answer the following research questions:

- 1. Which trade-offs can be found in the design of the different stages of the system? In order to design a power efficient system, the first step is identifying trade-offs. By investigating these trade-offs, optimal design choices can be made for specific requirements.
- 2. How is the performance of the power-efficient design and what is the cost in power consumption?

The performance of the design must be evaluated together with its power consumption to determine if the design can be used in practice.

What is the difference in power consumption and performance between the Prop. ACD and Conv. ACD?

To determine which demodulator is most suitable for the DDPSK receiver, it is important to investigate the characteristics of both ACDs.

To assess the quality of the design, the performance is compared with the theoretical performance as described in [2]. To do so, the design is using the same specifications as in [2]. These specifications are given in Table 1.1.

| Input                          | in-phase & quadrature-phase signal |
|--------------------------------|------------------------------------|
| Intermediate Carrier Frequency | 1 MHz                              |
| Data rate                      | 100 Symbols per second             |
| Packet size                    | 200 Symbols                        |
| Modulation                     | BPSK                               |
| samples per Symbol (s/S)       | 16                                 |
| Maximum frequency offset       | 700 Hz                             |

Table 1.1: Specifications of the system

## 1.2 Outline of the Thesis

The approach that is used to answer the research questions and content of the thesis is described in this section.

The starting point of the thesis is the information stated in [2]. The theory explaining the ACD, together with the theory and possibilities for the necessary signal conditioning components are discussed in chapter 2.

Before designing the system, the properties of synthesized arithmetic functions by a synthesis tool that is used in this thesis are investigated. The result can be found in chapter 3. This chapter is also used to determine some design rules that can be used during the design.

To investigate design options of the system, a MATLAB Simulink[6] model is made of the demodulator of [2]. By adding the necessary DSC to this model, trade-offs are identified and the optimal design choices are determined to optimize power consumption. These steps are discussed in chapter 4.

The Simulink model is build up out of simple arithmetic function. Using the simple functions, the model is converted to VHSIC Hardware Description Language (VHDL) manually. This conversion is described in chapter 5.

The VHDL description is used for synthesis and power estimation of the system. Using the synthesized Application-Specific Integrated Circuit (ASIC) design and Simulink model, the performance of the design is evaluated and discussed in chapter 6.

In chapter 7 research questions are answered and conclusions are drawn. This is discussed together with possible improvements and future research that can be done.



This chapter will discuss the theory of the Conv. ACD and Prop. ACD. It will also introduce theory that is used in the DSC. In Figure 2.1 it can be seen that the considered systems consist of two parts. The ACD for DDPSK and the DSC needed to properly condition the signal from the ADC for the demodulator.

The DSC consist of three steps, as seen in Figure 2.1. First, the IF signal received from the ADC needs to be converted to baseband. Then, the signal is filtered such that aliasing does not occur when the samples are decimated in the third step.



Figure 2.1: Detailed figure containing the steps of the signal conditioning.

Because the system is discrete, it is important to ensure that the requirements of the sampling theorem are met [7]. Figure 2.2 states the single-sided BW and the sampling frequency of the input and output of the different parts of the receiver.

The sampling rates in the system depend on the chosen "samples per Symbol (s/S)" and output data rate of the demodulator. The reasoning behind this is explained in section 2.1. Because the output data rate is chosen to be 100 bit/s and the s/S is 16, the sampling frequency of the demodulator at the input is 1600 Hz. The IF of 1 MHz, data BW of 100 Hz and allowed single sided frequency offset of 700 Hz sets the sampling frequency at the input of the DSC to at least 2001600 Hz.



Figure 2.2: Single sided data BW and sampling frequency (Fs) of each part of the system.

To determine the performance of the system, BER curves in terms of 'Energy per Bit to Noise power spectral density ratio  $(E_b/N_0)$ ' are considered. In this thesis, a symbol consists of one bit.

#### 2.1 Autocorrolation Demodulator

The demodulator is the most important part of the design. This section will discuss the Conv. ACD, introduce the challenge that the Conv. ACD faces and explain the solution for the challenge as proposed by [2].

#### 2.1.1 Conventional Autocorrolation Demodulator

The block diagram of a Conv. ACD is depicted in Figure 2.3 [1]. It consists of delay elements, complex multipliers and an integrator. The received signal  $b_1(t)$  will be demodulated in the following explanation.



Figure 2.3: Double differential demodulator.

Assume that the received signal  $b_1(t)$  is as follows [8]. It consists of the transmitted signal  $a_1(t)$ , an unknown frequency offset component  $e^{j2\pi\Delta f_1t}$  and a random constant phase offset component  $e^{j\theta_1}$  as seen in Equation 2.1. The transmitted signal  $a_1(t)$  stays constant for each symbol.

$$b_1(t) = a_1(t) \cdot e^{j2\pi\Delta f_1 t} \cdot e^{j\theta_1}$$
(2.1)

In the first autocorrelation stage, the received signal is multiplied with the conjugate of its delayed version. This results in  $d_1(t)$  as shown in Equation 2.2. To facilitate the explanation it is assumed that  $T_1$  is equal to  $T_2$ .

According to Equation 2.2, the first stage of autocorrelation has three effects. The phase component  $e^{j\theta_1}$  is removed. The frequency offset component  $e^{j2\pi\Delta f_1 t}$  is converted to a constant phase offset  $e^{j2\pi\Delta f_1 T}$ . And signal  $a_1(t)$  is multiplied with  $a_1^*(t-T)$ .

$$d_{1}(t) = b_{1}(t) \cdot b_{1}^{*}(t-T)$$

$$= (a_{1}(t) \cdot e^{j2\pi\Delta f_{1}t} \cdot e^{j\Delta\theta_{1}}) \cdot (a_{1}^{*}(t-T) \cdot e^{-j2\pi\Delta f_{1}(t-T)} \cdot e^{-j\Delta\theta_{1}})$$

$$= a_{1}(t)a_{1}^{*}(t-T) \cdot e^{j2\pi\Delta f_{1}t} \cdot e^{-j2\pi\Delta f_{1}t} \cdot e^{-j2\pi\Delta f_{1}(-T)} \cdot e^{j\Delta\theta_{1}} \cdot e^{-j\Delta\theta_{1}}$$

$$= a_{1}(t)a_{1}^{*}(t-T) \cdot e^{j2\pi\Delta f_{1}T}$$
(2.2)

Then, to complete autocorrelation  $d_1(t)$  is integrated over one symbol. It acts like a low pass filter that filters noise from the  $d_1(t)$ .

The constant phase component  $e^{j2\pi\Delta f_1T}$  of Equation 2.2 is removed by the second stage of the Conv. ACD. Assume that,  $b_2(t)$  is defined as seen follows.

$$a_{2}(t) = a_{1}(t)a_{1}^{*}(t-T)$$

$$\Delta\theta_{2} = 2\pi\Delta f_{1}T$$

$$b_{2}(t) = a_{2}(t) \cdot e^{j\Delta\theta_{2}}$$
(2.3)

When, another differential decoder is applied to  $b_2$ ,  $d_2$  is obtained as follows.

$$d_{2}(t) = b_{2}(t) \cdot b_{2}^{*}(t-T)$$

$$= (a_{2}(t) \cdot e^{j\Delta\theta_{2}}) \cdot (a_{1}^{*}(t-T) \cdot e^{-j\Delta\theta_{2}})$$

$$= a_{2}(t)a_{2}^{*}(t-T) \cdot e^{j\Delta\theta_{2}} \cdot e^{-j\Delta\theta_{2}}$$

$$= a_{2}(t)a_{2}^{*}(t-T)$$
(2.4)

Substituting  $a_2(t)$  from Equation 2.3 we have:

$$d_2(t) = a_1(t)a_1^*(t-T)a_1^*(t-T)a_1(t-2T)$$
(2.5)

The result of Equation 2.5 shows that signal  $a_1$  is multiplied with its delayed and complexconjugated version. In order for the demodulator to produce the original information, the information must first be encoded properly at the transmitter. For this purpose, a double differential encoder is used.

The block diagram of a double differential encoder is shown in Figure 2.4. The information signal x(t) is shifted in time and multiplied by itself in the first stage to get  $e_1(t)$ . Then in the second stage, another multiplication with its delayed version is done to construct  $e_2(t)$ . Equation 2.6 formulates the encoder and its output.



Figure 2.4: Double differential encoding.

$$e_{1}(t) = x(t)e_{1}(t - T)$$

$$e_{2}(t) = e_{1}(t)e_{2}(t - T)$$

$$e_{2}(t - T) = e_{1}(t - T)e_{2}(t - 2T)$$

$$e_{2}(t) = (x(t)e_{1}(t - T))(e_{1}(t - T)e_{2}(t - 2T))$$
(2.6)

By replacing  $a_1$  of Equation 2.4 with  $e_2$  from Equation 2.6 we have:

$$d_{2}(t) = e_{2}(t) \cdot e_{2}^{*}(t-T) \cdot e_{2}^{*}(t-T) \cdot e_{2}(t-2T)$$

$$d_{2}(t) = x(t)e_{1}(t-T)e_{1}(t-T)e_{2}(t-2T) \cdot e_{1}^{*}(t-T)e_{2}^{*}(t-2T) \cdot e_{1}(t-T)e_{2}^{*}(t-2T) \cdot e_{2}(t-2T)$$

$$d_{2}(t) = x(t) \cdot e_{1}(t-T)e_{1}^{*}(t-T) \cdot e_{1}(t-T)e_{1}^{*}(t-T) \cdot e_{2}(t-2T)e_{2}^{*}(t-2T) \cdot e_{2}(t-2T)e_{2}^{*}(t-2T)$$

$$(2.7)$$

Lets define  $c_1$  and  $c_2$ .

$$c_1 \stackrel{\Delta}{=} e_1(t-T)e_1^*(t-T) \cdot e_1(t-T)e_1^*(t-T)$$

$$c_2 \stackrel{\Delta}{=} e_2(t-2T)e_2^*(t-2T) \cdot e_2^*(t-2T)e_2(t-2T)$$
(2.8)

By substituting  $c_1$  and  $c_2$  in Equation 2.7 it can be shown that the phase information of x(t) is unaffected as denoted by.

$$d_2(t) = x(t) \cdot c_1 \cdot c_2 \tag{2.9}$$

 $e_1(t-T)e_1^*(t-T)$  and  $e_2(t-2T)e_2^*(t-2T)$  will always result in real values. Therefore  $c_1$  and  $c_2$  will also result real values. These real values will multiply with x(t) and only affect the magnitude of the signal. This means that, when x(t) is a phase modulated signal, its information will be unaffected by the ACD.

In this thesis Binary Phase Shift Keying (BPSK) is used to modulate x(t). The only possible values for x(t) are then 1 + j0 and -1 + j0. Therefore  $c_1$  and  $c_2$  will also both result in 1 and  $d_2(t) = x(t) \cdot 1 \cdot 1$ .

A property which was not discussed in this section is the choice for delay  $T_1$  and  $T_2$  in the first stage and second stage differential modulator and demodulator, respectively. Possible choices are discussed in [1]. It states that the BER improves when the delay of T is increased due to a decrease in noise correlation. The values for T are quantized because the delay has to be at least the duration of one symbol  $a_1(t)$ .  $T_2$  can only be the same, or a multiple of  $T_1$ . In this thesis the same delay values are used as in [2],  $T_1 = 2T_2$ . The delay in the second stage is twice the delay of the first stage.

#### Digital ACD and noise

In this section a discrete time version of the ACD is analysed in the presence of white Gaussian noise in the input of the demodulator

Figure 2.5 shows the discrete time version of Figure 2.3 With  $N \in \mathbb{N}$  as the number of "samples per Symbol (s/S)".

To capture the signal in the presence of a large frequency offset a wide filter is used. The sampling frequency of the demodulator will increase which means that N increases.

The summation after the first differential demodulator is placed to average over the samples of the first stage. n is used as the sample index at the first stage of the demodulator. k is used as the index of the second stage of the demodulator. The second stage of the demodulator is running at a different sampling frequency than the first stage. The relation of the sampling frequency in the first and the second stage is such that n/N = k with  $k \in \mathbb{N}$ .



Figure 2.5: discrete version of the demodulator.

It is assumed that the Nyquist criterium is met. In order words, the signal is within  $\frac{1}{2}$  of the sampling frequency of the demodulator and the signal and noise are filtered with a proper filter. A filter that has enough attenuations to suppress aliasing. The noise samples are white circularly symmetric Gaussian random variables.

Equation 2.10 is the discrete time version of Equation 2.1.  $\eta[n]$  denotes the samples of noise. Due to the addition of the noise component, multiple terms are created when the signal goes through the first stage of the demodulator. This is seen in Equation 2.11.

$$b_1[n] = a_1[n] \cdot e^{j2\pi\Delta f nT_s} \cdot e^{j\theta} + \eta[n]$$
(2.10)

$$d[n] = b[n] \cdot b^{*}[n-N]$$

$$= (a_{1}[n] \cdot e^{j2\pi\Delta f nT_{s}} \cdot e^{j\theta} + \eta[n]) (a_{1}[n-N] \cdot e^{-j2\pi\Delta f(n-N)T_{s}} \cdot e^{-j\theta} + \eta[n-N])$$

$$= a_{1}[n] \cdot e^{j2\pi\Delta f nT_{s}} \cdot e^{j\theta} a_{1}[n-N] \cdot e^{-j2\pi\Delta f(n-N)T_{s}} \cdot e^{-j\theta}$$

$$+ a_{1}[n] \cdot e^{j2\pi\Delta f nT_{s}} \cdot e^{j\theta} \cdot \eta[n-N]$$

$$+ \eta[n]a_{1}[n-N] \cdot e^{-j2\pi\Delta f(n-N)T_{s}} \cdot e^{-j\theta}$$

$$+ \eta[n]\eta[n-N]$$

$$(2.11)$$

To keep track of the different terms of Equation 2.11, d[n], s[n],  $g_s[n]$  and  $g_n[n]$  are defined as seen in Equation 2.12. s[n] is defined as the signal component with phase offset.  $g_s[n]$  is defined as the two signal components multiplied with noise components. And  $g_n[n]$  consists only of noise components. Equation 2.12 shows that the noise  $\eta[n]$  has a significant influence in the resulting signal of the first stage of the demodulator.

$$s[n] \stackrel{\Delta}{=} a_1[n] \cdot e^{j2\pi\Delta f nT_s} \cdot e^{j\theta} a_1[n-N] \cdot e^{-j2\pi\Delta f(n-N)T_s} \cdot e^{-j\theta}$$

$$= a_1[n]a_1[n-N] \cdot e^{j2\pi\Delta f NT_s}$$

$$g_s[n] \stackrel{\Delta}{=} \eta[n-N]a_1[n] \cdot e^{j2\pi\Delta f nT_s} \cdot e^{j\theta} + \eta[n]a_1[n-NT] \cdot e^{-j2\pi\Delta f(n-N)T_s} \cdot e^{-j\theta} \qquad (2.12)$$

$$g_n[n] \stackrel{\Delta}{=} \eta[n]\eta[n-N]$$

$$d[n] = s[n] + g_s[n] + g_n[n]$$

The summation resulting in  $d_{sN}[k]$ , as seen in Figure 2.5, is used to average the samples of one symbol. Equation 2.13 states that for each increase of N, one s[n] component and two noise components are added.

$$d_{sN}[k] = \sum_{\substack{n=1\\N}}^{N} d_1[n]$$

$$d_{sN}[k] = \sum_{\substack{n=1\\N}}^{N} s_{d1}[n] + g_{d1s}[n] + g_{d1n}[n]$$
(2.13)

Since  $\eta[n]$  are white Gaussian noise samples, the  $g_s[n]$  samples that are added due to the increase of N will be averaged out by the summation of  $d_{sN}[k]$ . However, because  $g_s[n]$  contains two 'signal multiplied with noise' components the power of the noise component is increasing .

The power of the noise components of  $g_n[n]$  is increased when N is increased. This happens because the Gaussian noise samples are multiplied with each other are not Gaussian any more. Therefore they do not average out as effectively when the samples are summed.

To tolerate more frequency offset, N is increased and a wider passband of the filter before the demodulator is used. However, extra noise is included that is not averaged out efficiently, causing a degradation in performance. Because of this, the increase of N has a negative influence on the BER of the demodulator.

#### 2.1.2 Proposed Autocorrolation Demodulator

In [2] a method has been proposed to increase the Signal to Noise Ratio (SNR) of s[n] and solve the problem of a wide filter.

Multiple parallel paths of ACD are used. In these paths the symbols are shifted and then correlated. In this way duplicates of  $d_2$  are generated with similar signal components and uncorrelated  $g_n[n]$  components. When they are added, the signal components are added coherently while noise components are added incoherently leading to increased SNR.

The complete Prop. ACD is shown in Figure 2.6. The input of b[n] is shared with the multiple ACD each having their specific delay in  $b^*[n-NT]$ . At  $d_{2tot}$  all the paths are summed together.



Figure 2.6: Figure of Prop. ACD with the conventional path, L paths and R paths.

Figure 2.7 explains what happens with the sample of each symbol in the Conv. ACD and Prop. ACD. Figure 2.7a represents the first stage of the Conv. ACD. N samples of symbol  $k_2$  are multiplied with the corresponding samples of  $k_1$ . This is a representation of  $d_1[n] = b[n] \cdot b^*[n - NT]$ . After the multiplications the samples are summed.

By shifting the delay of  $b^*[n-N]$  it is possible to create paths with uncorrelated noise. A reduction in delay creates a so-called 'Left'(L) path as seen in Figure 2.7b. A addition in delay creates a so-called 'Right'(R) path as seen in Figure 2.7c. When symbols besides  $k_1$  and  $k_2$  are received sequentially, it is important to only use the multiplied samples that belong to  $k_1$  and  $k_2$ . To do so the sum after multiplication is changed accordingly. The number of extra paths p can vary from 0 up to N-1 for the L and R path.



Figure 2.7: Explanation on how the samples are multiplied and summed with N = 4. The conventional method can be seen at (a). (b) and (c) shows the steps that are taken in the L and R path of the proposed method when P = 1.

Figure 2.8 shows that the theoretical performance of the Conv. ACD and Prop. ACD. The BER of the different demodulators is illustrated at different  $E_b/N_0$ . The Prop. ACD performs better than the Conv. ACD, at the cost of additional complexity. The Prop. ACD has multiple ACD that needs to be implemented compared to the single ACD that the Conv. ACD has. This is gives trade-off between the increase in performance and increase of complexity is investigated in chapter 4.



Figure 2.8: Theoretical performance of the ACD [2]. The C curves are the performance of the Conv. ACD and the P curve belongs to Prop. ACD. The number behind the 'C' or 'P' indicates the value of N.

#### 2.2 Conversion to baseband

Baseband conversion is one of the steps in the system where the number of operations per second is the highest. Therefore, it is important to optimize this part if possible.

Baseband conversion is done by multiplying the signal on an IF carrier with the IF signal, as seen in Equation 2.14.

$$s[n]e^{j\omega_{IF}nT} \cdot e^{-j\omega_{IF}nT} = s[n]e^{j(\omega_{IF}-\omega_{IF})nT} = s[n]e^0$$
(2.14)

Where  $\omega_{IF} = 2\pi \cdot f_{IF}$  and  $s[n]e^{j\omega_{IF}nT}$  the signal that is on the IF. In discrete time the IF carrier is sampled. The number of samples to multiply the incoming signal with is based on the 'sampling ratio'  $= \frac{Fs}{F_{IF}}$ . The optimal choice is investigated in section 4.2. By selecting an optimal choice for the sampling rate, unnecessary complexity can be reduced.

#### 2.3 Decimation

The second step of the DSC of the system is decimation. Decimation consists of two steps. First, it must be ensured that there is no signal outside the spectrum of the new sampling frequency. This is done by filtering. The second step is to downsample and with that archive the new sampling rate.

The minimum sampling ratio is twice the IF divided by the sampling frequency of the ACD,  $\frac{2 \cdot 10^6}{1600} = 1250$ . It is necessary to have a digital filter which has a sampling rate of > 2 MHz and a passband of 800 Hz.

Figure 2.9 displays a figure of a filter that is needed if the decimation were to be done using a single Finite Impulse Response (FIR) filter. The specifications for the filter were chosen such that the filter would have an attenuation that is less than needed. This is such that an indication of the number of coefficients could be obtained. The number of coefficients gives an indication of the complexity of the filter. While it is possible to create a filter with 7289 coefficients it is preferred to have a less complex system. In the project three types of filters are considered; the FIR filter the Infinite Impulse Reponse (IIR) filter and the Cascade Integrator-Comb (CIC) filter.



Figure 2.9: FIR filter with 7289 coefficients.

#### 2.3.1 Finite Impulse Response

The structure of an FIR filter can be seen at Figure 2.10. The FIR filter has a relatively simple structure. It consists of delay elements ' $z^{-1}$ ', addition elements, and multipliers that multiply with a constant value; b(1),...,b(10). A sample enters as input and goes through the delay elements before it can reach multiplier b(10). In between the delays, the input sample is multiplied with 'b(1) to b(9)' and added with other samples that are multiplied with b(1) to b(10), respectively.



Figure 2.10: FIR filter with 10 coefficients.

The number of coefficients governs the number of delay elements, multipliers and additional elements in the filter. For each coefficient, all three elements are needed. In this project, the coefficients are determined using the filter designer of MATLAB [9]. The number of coefficients increases depending on the steepness of the transition band, ripple in passband and attenuation in the stopband. The FIR filter is stable, meaning that the output will always stay within bounds. When quantizing the FIR filter it will stay stable but its behaviour could change depending on the precision of the coefficients.

#### 2.3.2 Infinite Impulse Reponse

Figure 2.11 shows an IIR filter with 3 sections that are connected sequentially. The structure is more complex than for FIR filters. The structure consists of the same elements as the FIR filter but in a feedback configuration. An input sample starts at the input and goes through several loops within the filter.

The number of coefficients increases depending on the steepness of the transition band, ripple in passband and attenuation in the stopband. The IIR filter needs significantly fewer coefficients compared to the FIR filter for the same specified behaviour. However, the word length of the accumulators needs to be larger compared to those of the FIR filter.



Figure 2.11: IIR filter consisting of 3 sections.

It is important to notice that the impulse response of the filter is infinite. This is caused by the loops in the structure of the filter. The word length of the coefficients and accumulators need to be sufficient in order for the filter to be stable.

#### 2.3.3 Cascade Integrator-Comb

The CIC filter is an efficient boxcar filter [10]. It is an FIR filter where the coefficients have the value 1. This creates a moving average structure as seen at Figure 2.12a. The structure can be simplified by using a single accumulator, and adding the difference of the sample that is entering the moving average with the value that is exiting the moving average, as seen in Figure 2.12b. The components of Figure 2.12b can be identified as a comb filter preceding an integrator. In [10] it is shown that the order of the comb filter and the integrator can be changed because of the noble identity [11].



Figure 2.12: Left the moving average filter, Right another implementation of the same filter [10].

When the CIC filter is used as a filter before decimation, it is possible to convert the filter that is seen in Figure 2.13, to the filter as seen in Figure 2.14. In Figure 2.14 the comb filter is placed behind the integrator, using noble identity. It is also placed behind the down sampler. This is allowed because the delay in the comb filter is changed accordingly. The delay is changed to compensate for the samples that are removed during decimation.



Figure 2.13: CIC filter before decimation.



Figure 2.14: CIC filter combined with decimation, also called the Hogenauer filter.

The main advantage of decimation using the CIC filter, is that it consists of three elements. An accumulator, a decimator and delay element. It is able to downsample a signal without the use of a multiplier. The filter does require an accumulator with a sufficient word length. The downside of the CIC is that its frequency response is not flat at the passband as seen in Figure 2.15.



Figure 2.15: Magnitude response of a CIC filter of Figure 2.14.

Only a few parameters of the CIC filter can be changed. The first one is the decimation factor. This will affect the frequency response by scaling the response over the frequency axis . The second one the amount of delay in the integrator and comb part. This will scale the attenuation of the frequency response. And the last one is the number of integrator and comb stages. Increasing the integrator and comb stages is equivalent to cascading multiple of the same CIC filters without decimation .

#### 2.4 Power use in ASIC

In this thesis, the power use of the system is determined using a synthesis tool. The tool categorized three types of power[12]. Internal power and switching power which together is called dynamic power. And leakage power which is a static power. A schematic current flow of the three types of power is shown in Figure 2.16. It is important to mention that the voltage of the system will influence all the types of power consumption.

Any power dissipation due to internal switching of a cell is called the internal power. It consists of the momentary short-circuit current due to the CMOS transistors of a cell and other elements that need to be powered within the cell.

The Switching power is the power that is used for driving the output capacitance of a cell. The amount of power depends on the switching activity and the current that is needed to drive the load capacitance of the next cell.

Leakage power is caused by the leakage current of the circuit. When the system is powered the leakage current is always flowing. It is possible that the amount of leakage current could be different depending on the output state of the implemented cell.



Figure 2.16: Components of power Dissipation as explained in  $\left[ 12\right] .$ 

# Synthesis investigation

Before designing the system an investigation is done on the power consumption of arithmetic elements that are implemented by the synthesis tool. The goal is to create a design guide. This is done using the results of this investigation.

#### 3.1 Synthesis properties

The technology that is synthesised, is from tsmc[13]. It is a 40 nm technology with the product name: tcbn40lpbwp12t40m1p.

Design Compiler Graphical (version o-2018.06) by Synopsys [14] is used as the synthesis tool. It is located on several 'CAES' servers. The settings that were used can be found at '/opt/Synopsys/local/setenv \_syn\_O-2018.06.sh'[15].

All synthesis is done using the same method and the same script. This such that the only variable is the VHDL description. The used script is based on the script 'generate-design-core03' by Sabih Gerez made on 14-9-2017. The two most important settings are; 'uniquify', which allows two instances of the same entity to be optimized separately. And 'compile -map\_effort high' which tries to minimize the number of gates in order to optimize on the area.

Power consumption reports are also determined using the synthesis tool. There are two methods with which the synthesis tool is used to generate a power report. For both methods, the syntheses tool is using a global operating voltage of 1.1 V

Using the first method a power report is generated by using the command 'report\_power' immediately after synthesis of the system. The synthesis tool will determine the power consumption based on the switching activity of the primary input of the design. For each primary input, the state will be '1' for 10 percent of the time and the signal will switch once's in every 10 clock cycles [16]. The advantage of this method is that the power can be estimated without specifying the input.

The second method determines the power by using a post-synthesis simulation. With this method, the input of the system can be given. The power report is generated using the same command, 'report\_power'. The report will depend on the switching activity of the post-synthesis simulation. The advantage of this method that the power consumption of the two systems can be compared with the same behaviour and input.

## 3.2 Implementation of arithmetic elements

For the investigation, arithmetic operations that will be used in the system are implemented. These are:

- Addition with two unknowns (e.g. used in the ACD)
- multiplication with two unknowns (e.g. used in the ACD)
- multiplication with one unknown and one constant (e.g. used in the digital filters)

The investigation will parameterize the word length and clock period of the implementations. Word lengths from 3 bits to 25 bits are implemented and clock periods from 10 ns to  $10\,\mu$ s.

The input of the implementation is sampled using the clock of the system. This is to ensure that the system will not only consist of combinational logic and the power consumption is depended on the clock speed.

It is assumed that the synthesis tool will optimize the implementation as per design constraints. Therefore the VHDL description of the arithmetic operations is described as '\*' and '+'. In this way, a specific implementation is not forced since only the function is defined.

## 3.3 Result of Power estimation

The result of the investigation is shown in Figure 3.1. The figure plots the power consumption for the three arithmetic operations using different clock periods and different word lengths.

It can be seen that when the word length is increased the power consumption of the arithmetic functions also increases. When the word length increases the complexity of the implementation is increased. The power increase of a larger word length is a result of more complexity. This causes an increase in power consumption. An decrease in clock period will also increase the power consumption. This is expected since there is more switching activity per time unit when using a higher clock frequency.



Power Consumption of Implementation

Figure 3.1: Power consumption of different arithmetic operations at different word length and different clock periods. 'Multiply Static 7' is a implementation of a multiplication with an unknown that is multiplied with 7.

Figure 3.1 indicates a significant difference in power consumption between the different arithmetic functions. A static multiplication consumes less power compared to addition with two unknowns. The power consumption of both is increasing similarly when the word length is increased or the clock period is decreased. The power of a multiplier with two unknowns uses an amount of power which can be up to an order of magnitude higher than the other arithmetic functions when the word length increases.

Figure 3.2 shows different power components at different clock periods for each arithmetic function. All implementations have the same behaviour for their power consumption.

The leakage power of the implementation is not affected by the decrease in clock period. This is not unexpected since the leakage power is connected to the size of the implementation. The decrease in clock frequency affects the dynamic power. Therefore there is a certain point where the dynamic power becomes less significant than leakage power in the total power use.



Figure 3.2: Power consumption of different arithmetic operations when using different clock periods.

When comparing the implementation of the static multiplier with the adder, it is seen that the leakage power of the adder is slightly lower. It happens because the implementation of the adder is smaller than the implementation of the static multiplier. However, the total power use of the adder is higher than the total power use of a static multiplier.

## 3.4 Design Considerations

The following design consideration are used in the thesis after considering the result of section 3.3.

- 1. Since the increase in word length will cause an increase in power, the worth length of each part of the system should be decreased as much as possible.
- 2. It is preferred to reduce the number of multipliers with two unknowns as they consume a significant amount of power compared to the other arithmetic functions.
- 3. The most effective way to optimize power consumption of the system is to optimize the block that is consuming the most power. This block can be identified by considering the arithmetic functions per block. When assuming the same word length, the function with the highest power is; Multiplications, followed by addition, followed by multiplications with an implemented constant.
- 4. The energy efficiency of the system is increased when the system is running at a higher clock frequency(more operations per watt). This is because the leakage power plays a more significant role when the clock frequency decreases.



In this chapter, the method that is used to design the system will be discussed. The choices are made based on simulations of the system. At this point, the focus is the design on the algorithmic level. The design considerations of section 3.4 are taken into account. The main goal is to identify trade-offs in the system and to choose the most efficient option. The complete system is designed in separate steps starting with the demodulator. While there are two ACDs to be designed, the DSC for both demodulators are similar.

#### 4.1 Demodulator

The Prop. ACD has a trade-off that can be optimized at this stage. As seen in Figure 2.6, the Prop. ACD consist of multiple Conv. ACD, with a small modification in their delay and summation. It is possible to use the Prop. ACD without using all available paths at the left and right side. This would improve the performance of the demodulator compared to the Conv. ACD while it also reduces the number of multipliers compared to the complete Prop. ACD. For each increase in s/S the Prop. ACD will add an modified Conv. ACD on the left, and right path. So each increase of s/S will add two extra paths causing the increase of 2 complex multipliers. One complex multiplier consist of 2 multipliers with two unknowns. So the increase of s/S will cause a significant increase in power consumption.

#### L and R paths selection based on performance improvement

The system is optimized for the specific BER of  $10^{-3}$ . Simulations of the Conv. ACD, Prop. ACD and Prop. ACD with different numbers of paths are done. Based on simulations, the  $E_b/N_0$  of the incoming signal that is needed to reach the specific BER is estimated.

The result is shown in Table 4.1. C16 stands for the Conv. ACD with N = 16 s/S. P16 stand for the Prop. ACD with N = 16 s/S. And U# in U1,...,U15, stands for the Prop. ACD with # as the number of paths at the left and right side.

In order to find a suitable number of paths a new measure is used, called the Performance Ratio (PR). To determine the PR for each U#, the decrease of  $E_b/N_0$  required for a BER of  $10^{-3}$  is compared to the  $E_b/N_0$  required by C16. This value is divided by the number of multiplications that is needed to determine one symbol in the first stage of the ACD. For each path at the left and right side, the number of multiplications is different. The total number of multiplications for U# is the sum of the multiplications of each path.

|           | E. /N. | $E_b/N_0$ distance | Multiplications | Performance |  |
|-----------|--------|--------------------|-----------------|-------------|--|
| ACD type  |        | to C16             | in first stage  | ratio       |  |
| C16       | 12.1   | 0                  | 16              |             |  |
| U1        | 11     | 1.1                | 46              | 0.0239      |  |
| U2        | 10.7   | 1.4                | 74              | 0.0189      |  |
| U3        | 10.5   | 1.6                | 100             | 0.0160      |  |
| U4        | 10.4   | 1.7                | 124             | 0.0137      |  |
| U5        | 10.35  | 1.75               | 146             | 0.0120      |  |
| U6        | 10.3   | 1.8                | 166             | 0.0108      |  |
| U7        | 10.25  | 1.85               | 184             | 0.0101      |  |
| U8        | 10.2   | 1.9                | 200             | 0.0095      |  |
| U9        | 10.2   | 1.9                | 214             | 0.0089      |  |
| U10       | 10.15  | 1.95               | 226             | 0.0086      |  |
| U11       | 10.15  | 1.95               | 236             | 0.0083      |  |
| U12       | 10.15  | 1.95               | 244             | 0.0080      |  |
| U13       | 10.1   | 2                  | 250             | 0.0080      |  |
| U14       | 10.1   | 2                  | 254             | 0.0079      |  |
| U15 (P16) | 10.1   | 2                  | 256             | 0.0078      |  |

Table 4.1: Performance of different version of ACD at a BER of  $10^{-3}$ 

As seen in Table 4.1, for U13 to U15, the same  $E_b/N_0$  is needed to achieve a BER of  $10^{-3}$ . Using the U13 ACD would thus reduce the power consumption compared to the U15 ACD while keeping the same performance. In this thesis, the choice for U# did not only depend on having the same performance. Figure 4.1 shows that for an efficient design U8 is the best choice. After U8 the PR curve is almost constant. Thus, this means that when choosing a higher U# it would not be improving the performance as much as the complexity of the ACD.

While U8 is chosen for the final system, U1 is used in simulations. U1 is less complex than U8 which will speed up the needed simulations for the design.



Figure 4.1: Performance ratio of Prop. ACD with different number of paths at the Left and Right side.

#### U# selection based on constraints

Another way of selecting U# is possible when the system is implemented for a real-life scenario. During a real-life scenario there will be constraints on the transmission power or power consumption of the receiver. Determining the number of U# can then be done by considering the trade-off in power consumption between the transmitter and receiver. In the same environment, the increase in  $E_b/N_0$  will mean the increase in transmission power. Selecting a U# with more paths, will increase the digital power consumption of the receiver but does not require an increase in transmission power.

#### Quantisation

Before designing the other parts of the system, the effect of the quantization on the input of the demodulator investigated. By choosing a suitable word length, the complexity of the rest of the system might be reduced.

Choosing the quantization of the input of both the Conv. ACD and Prop. ACD is done in the same way. Simulations are done using Simulink. The input is quantized using the method as seen in Figure 4.2. Simulations using an  $E_b/N_0$  of only 8,9 and 10 were made to reduce simulation time. First, the input is resized to fit the word length. Because the input consists of the modulated signal and noise, the values are stochastic. The stochastic values are normalized after first determining the range of values using simulation. Then the values are floored such that only integers are used. To ensure that there are no integers out of bound, a saturation block is used.



Figure 4.2: The Simulink model for quantisation of input values.



Figure 4.3: Influence of input word length on the performance of the demodulator U1 and C16.

As seen in Figure 4.3, the performance of both ACD's remains constant after the input word length has reached 7 bit. Therefore, a 7 bit input word length seems the best choice for the demodulator as it does not limit the performance.

## 4.2 Baseband conversion

The main design choice for the baseband conversion is the sampling ratio as introduced in section section 2.2. Baseband conversion consists of multiplying the received signal with the IF, it is implemented using a complex multiplier. Furthermore, baseband conversion is one of the parts with the highest clock frequency. Therefore this part consumes a considerable amount of power.

First, the minimum sampling ratio for baseband conversion is determined. The incoming signal has a BW of 1 MHz + 800 Hz. 1 MHz for the IF and 800 Hz for the signal with possibly frequency offset. The signal needs to be down converted with a signal of 1 MHz to remove the IF. A sampling ratio of 2 would be insufficient in this case because then the BW of the signal cannot be wider than 2 MHz. It is preferred to keep the sampling ratio as low as possible to decrease the sampling frequency and with that the number of calculations that need to be done per second.

| Sampling ratio | Unique values to multiply with                         |
|----------------|--|
| 1              | 01   |
| 2              | -1 0 1   |
| 3              | -0.86603 -0.5 0 0.86603 1                              |
| 4              | -1 0 1   |
| 5              | -0.95106 -0.80902 -0.58779 0 0.30902 0.58779 0.95106 1 |
| 6              | -1 -0.86603 -0.5 0 0.5 0.86603 1                       |

Table 4.2: Unique values to multiply with for baseband conversion based on sampling ratio

Table 4.2 shows the unique values with which the incoming signals samples should be multiplied with, for baseband conversion. The minimum sampling ratio is 3, which would mean that there is a need to multiply with 5 different values. It is, however, better to consider a sampling ratio of 4. When sampling using a sampling ratio of 4 (Fs = 4IF), the multiplicands are; 1, -1 and 0. In this case, the implementation would not need multipliers.

## 4.3 Decimation

Now that the sampling ratio of the baseband conversion is known, it is possible to design the necessary down sampling. The sampling ratio of 4, will cause the sampling frequency of the baseband conversion to be 4 MHz. The sampling frequency of the ACD at the input is 1600 Hz as stated in chapter 2. This results in a down sampling factor of  $\frac{4000000}{1600} = 2500$ .

Due to simplicity compared to the FIR and IIR filter, the CIC filter is preferred for decimation. It is not possible to implement a CIC that decimates 2500 times. This is because the internal word length of the accumulator of the CIC would be wider than 128 bit. Instead, a CIC of 1250 is used. This results in the need for a second stage. It is not possible to use another CIC filter in the second step. If this was possible, both CIC filters could be combined to a singular filter using noble identity.

#### Filter choice for second stage decimation

For the second stage in decimation, there is the need for a filter which will attenuate half of its bandwidth. This gives the specification regarding the passband and stopband of the filter. By increasing the transition band the complexity of the FIR and IIR filters can be

modified. In this thesis a passband which is 0.8 times the stopband is used for the FIR filter and a passband which is 0.9 times the stopband is used for the IIR filter. These values were chosen based on the number of coefficients they produce. This ensures a transition band which will avoid a complex filter while the ACD is still able to function properly. The IIR filter has a naturally more steep transition band than the FIR filter. Due to this, the transition band of the IIR filter can be reduced without becoming more complex than the FIR filter.

The attenuation of the filters is chosen to be  $36 \, \text{dB}$ . Because the signal that needs to be filtered consists of noise, it is not possible to be certain what the maximum value is that needs to be attenuated. To have a starting point for the attenuation the following idea was used.

For the filter, all frequencies outside of the passband should be attenuated such that it is removed from the signal. It is determined that the demodulator uses an input of 7 bit. Considering 2's complement values it would mean that the maximum magnitude of a value entering the demodulator is 64. When attenuating the maximum value, it should be attenuated such that it will be quantized to 0. An attenuation of 65 converted to decibel is  $20\log_{10}(65) \approx 36dB$ .

With the given specification three filters were designed using MATLAB filterDesigner [9]. An FIR filter called 'FIR1' which has 27 coefficients. An IIR filter called 'IIR1' with 13 coefficients. And an FIR filter, 'FIR2' where the attenuation was reduced to decrease the complexity of the filter. The properties of the filters are stated in Table 4.3.

The ACD has a natural method to compensate for noise in signals. It is assumed that having some folding noise in the signal when using FIR2 due to non-strict filtering might not affect the performance of the ACD.

| Property             | FIR1 (f2)  | FIR2 (f1)  | IIR1(f3) |
|----------------------|------------|------------|----------|
| Fs                   | 3200 Hz    | 3200 Hz    | 3200 Hz  |
| Passband             | 640 Hz     | 720 Hz     | 720 Hz   |
| Stopband             | 800 Hz     | 1000 Hz    | 800 Hz   |
| Passband attenuation | 1dB        | 1dB        | 1 dB     |
| Stopband attenuation | 36 dB      | 20 dB      | 36 dB    |
| Design method        | Equiripple | Equiripple | Elliptic |
| Multiplications      | 27         | 10         | 13       |

Table 4.3: Properties of considered filters

The attenuation behaviour of the frequency responses of the three filters can be seen in Figure 4.4. There is a clear trade-off between the transition band of the filter and the complexity of the filter. The FIR2 filter is the least complex with 10 coefficients. However, its frequency response is the worst among the three, the attenuation is small, and the transition band is wider than the other filters. The IIR1 filter has only 13 coefficients but 3 sections. This means that there are three accumulators which need to be implemented and a sufficient word length. The IIR1 filter does have the best frequency response. The FIR1 filter has 27 coefficients, and better frequency response compared to the FIR2 filter and reasonable frequency response compared with IIR1.



Figure 4.4: Frequency response of the designed filters. The frequency axis is normalized to the BW of the sampling frequency of the filter.

To choose the most applicable filter, the BER performance of the ACD is considered. The BER curves of the receiver with the mentioned filters are shown in Figure 4.5. Simulations are done for C16 and U1. All filters except 'FIR2 - C16' show on average the same performance. It is expected that FIR2 - C16 has a performance which is deviating from the other C16 curves. As discussed in section 2.1, the performance of the Conv. ACD is decreasing when the noise samples, in this case, noise due to folding, are entering the demodulator. The reason is that noise has a larger impact on the Conv. ACD than the Prop. ACD.

The performance of FIR2 is the same as the other filters in U1. In C16 the performance is marginally close to the other filters. Since FIR2 is by far the least complex filter, it is chosen as the most optimal choice of the three filters.



Performance of the different filters

Figure 4.5: Performane of the demodulator with the filters infront of the ACD.

#### 4.4 Quantisation of each part

With the design of the system completed, all the parts of the DSC (baseband conversion, CIC filter, FIR filter) are quantized. The minimal word length that does not affect the

performance of the demodulator is determined. This is done with the same method as the quantization of the demodulator. Multiple simulations are done with different word lengths for the CIC and FIR filter. The baseband conversion does not have an arithmetic operation in its implementation, as such it is using the same word length as the CIC.

The result is seen in Figure 4.6. At a certain number of bits, the performance remains constant. This is the point that is chosen to be the word length. For the FIR filter, an 8-bit word length is chosen and for the CIC a word length of 5 bit is chosen.



Figure 4.6: BER performance of system where the FIR and CIC are quantized with different word lengths.

## 4.5 Overview of design

The complete design can be seen in Figure 4.7. The demodulator is still shown as a global block because the U8 and C16 demodulator will be implemented. Baseband conversion can be done without multiplications. The CIC is designed with a decimation factor of 1250. It consists of one integrator and one comb stage. The FIR filter consists of 10 coefficients and can be implemented using multipliers where one of the two values is constant. Table 4.4 gives an overview of the word lengths of the different parts.



Figure 4.7: Designed system.

| Table 4.4: | Word | length | of  | different | parts | of  | the | design |
|------------|------|--------|-----|-----------|-------|-----|-----|--------|
|            |      |        | • • |           | p     | • • |     | ~~o.o. |

| Part                | Word length | Remark                    |
|---------------------|-------------|---------------------------|
| Baseband conversion | 5 bit       | following CIC word length |
| CIC decimator       | 5 bit       |                           |
| FIR filter          | 8 bit       |                           |
| Demodulator         | 7 bit       |                           |

# Implementation

The implemented system is seen in Figure 5.1. The system looks similar to the designed system as discussed in section 4.5. However, in the implementation 'Downsize' entities are added to connect between blocks that are using different word lengths. The down sampling block is seen in Figure 4.7 has been removed. When the system is implemented in VHDL re-sampling is done by using a different clock frequency.

Another difference between the implementation and the designed system of Figure 4.7 is the separate track for the in-phase part of the signal and quadrature phase of the signal. It is assumed that the signal received by the receiver is first converted to IF and with that, the in-phase and quadrature phase of the signal is split. Two ADC's will be used to convert the IF signal to the digital domain. The in-phase and quadrature phase tracks are called the Real and Imaginary track respectively.



Figure 5.1: System to that is implemented in VHDL.

It is tried to implement the system with entities such that the designed blocks can be modified without affecting the rest of the system. A control entity is added to generate the necessary clock signals of each entity. It also takes care of the synchronization needed for the samples.

## 5.1 Downsizing

Each entity is implemented such that at an internal level, the calculations are done in full precision. In chapter 4 the minimum input word length of each block is investigated. For each entity, it is necessary to match the output word length, such that it can fit the input word length of the following entity.

When resizing a word length using the conventional method, the Least Significant Bits (LSBs) are removed until the required word length is reached. This makes sense since the LSB only represents a detail of the represented value.

In this thesis, a non-conventional method for downsizing is used. The method consists of

two steps. Due to operations with a full adder, a multiplier or accumulator, it is possible that the number representation is increased to ensure the operations are done with enough precision. However, this might cause an increase in word length where some of the Most Significant Bit (MSB) are static.

To reduce the word length and optimize on the bits that are never changed, a method is applied as seen in Figure 5.2. First, a simulation is done to determine which MSBs are never. After identifying these bits, the MSB for the new word length is identified. Then starting from the MSB the number of bits for the new word length are selected and the LSB that are not part of the new word length removed.



Figure 5.2: Decreasing the word length by selecting relevant bits using the method that is used in this thesis.

Figure 5.3 shows the effect of the method on a signal. There is three axes in the figure called the 'quantization axis'. Each quantization axis shows where the quantization points of the signals are. When the conventional method of downsizing is used, it would simply redefine the number of quantization levels according to the new word length, and distribute them evenly over the new quantization axis.

The method in this thesis first defines a new minimum and maximum for the signal. This is done by simulating the system and identifying the minimum and maximum value in the signal. With the newly determined minimum and maximum, the new quantization axis is first scaled before the quantization levels are distributed accordingly. It is possible to choose the new minimum and maximum closer to zero, than the minimum and maximum value of the signal. Doing so can add precision to the values closer to zero. Because of the value representation in 2's complement, the minimum and maximum value that are chosen are linked to the word length.



Figure 5.3: Effect of decreasing a word length on a signal.

When a sample is represented by the MSB that is removed for the next entity, the value should is saturated at the maximum (or minimum) value that can be represented by the new word length. By doing so, the information the value is representing is not lost. It is however altered, which could be seen as a type of clipping noise that is added to the signal.

The key requirement for this method to work is that the entities need to be 'gain independent'. An output of one entity might change in gain before entering the following entity. The trade-off in this downsizing method is the increase of noise of the signal due to clipping the maximum swing, against the decrease in quantization noise since LSBs are left for the smaller values. for this specific system, it is assumed that the Low Noise Amplifier (LNA) of the receiver is also trying to adjust the gain of the received signal. Because of this, the signal that is converted by the ACD will effectively use all of the bits to quantize the signal the whole time.

## 5.2 Converting Design to VHDL

The implementation of the system in VHDL is made to resemble the MATLAB Simulink model as used during the design. The Simulink simulations made use of simple arithmetic blocks. This makes it easier to convert the model to VHDL.

An important difference in Simulink and the implementation is the delay of the samples that are going through the system. Each block in Simulink waits for the previous block to finish its computation of the first sample. The control entity made in the implementation ensures that the same thing happens in the implementation.

It is tried to use as many 'generics' in VHDL as possible. This is such that the implementation can be adapted and changed according to changes in the design.

#### 5.2.1 Control block

As seen in Figure 5.1, the system consists of multiple sampling frequencies. Clock signals from the control block are used to create multiple sampling frequencies.

Generating the correct clock signal is done using counters. The primary clock frequency of the system is a multiple of the secondary signals. There are 4 clock signals in the system, listed as follows:

- Primary clock: 4 MHz used for the baseband conversion and first halve of the CIC
- Secondary clock 1:  $\frac{4x40^6}{1250}$  =3.2 kHz used for the second halve of the CIC filter and FIR filter.
- Secondary clock 2:  $\frac{4x10^6}{1250x^2} = 1.6 \text{ kHz}$ used for the first part of the demodulator
- Secondary clock 3:  $\frac{4x10^6}{1250x2x16} = 100 \text{ Hz}$  used for the second part of the demodulator

Using the appropriate minimum (negative) values of the clock counters, the control block is able to start each entity at the correct input sample. when the counter is negative the reset signal for the respective entity is enabled. After reaching the maximum value of the counter. The counter is set to 0 instead of the minimum value. It is crucial that the accumulator of the ACD starts at the correct sample. When this is not the case the performance of the ACD will decrease significantly. If the accumulator of ACD starts at an

incorrect sample, the accumulator will average over two different *symbols* instead of one. Then the information of two symbols will be mixed.

#### 5.2.2 Baseband conversion

As seen in Figure 5.4 and discussed in section 4.2 the implementation of the baseband conversion is without multipliers. A case statement is used to connect the correct values for the real and imaginary output of the entity to the input or inverted input of the entity. At each clock period the entity cycles to the next step. This emulates the multiplications with sine and cosine needed for the baseband conversion.



Figure 5.4: Schematic implementation of baseband conversion.

#### 5.2.3 CIC filter

A schematic overview of the CIC implementation is seen in Figure 5.5. M1, M2 and M3 are register, that are used as a delay element. Register M2 could be optimized in future implementations. It was left in the implementation because it is useful for debugging purposes.

The implementation has two clock inputs since different parts have a different sampling rate respectively. With the two different clock inputs, it is possible to change the down sampling ratio of the CIC filter without modifying the CIC.



Figure 5.5: Schematic of CIC filter implementation.

#### 5.2.4 FIR filter

The so called 'transposed direct form' of the FIR filter [17] is implemented as seen in Figure 5.6. The advantage of the direct form is that there is no need to synthesize a summation with a significant number of inputs. It also reduces the complexity of describing

the filter in VHDL. There are a lot of repetitive elements as seen in Figure 5.6. Because of this, a loop can be used for its implementation. This makes it easy to modify the filter coefficients in the VHDL description



Figure 5.6: Schematic of FIR filter implementation.

#### 5.2.5 Autocorrolation Demodulator

The ACD and the difference between the Conv. ACD and the Prop. ACD are discussed in section 2.1. The first and the second stage of Conv. ACD and Prop. ACD are implemented using the same VHDL description. The schematic description of the implementation can be seen in Figure 5.7 and Figure 5.8. Only the summation between the first and second stage is different for the ACDs as seen in Figure 5.9.



Figure 5.7: Schematic of first stage of ACD.

The first stage of the ACD is using a delay of 32 samples. This is implemented using a loop of registers that are described using VHDL. It is shown in the schematic as register M1 to M32



Figure 5.8: Schematic of second stage of ACD.

In the second stage of both ACDs, optimization has been made on the complex multiplication. The information about the imaginary value is not used. Because of this, the imaginary output of the multiplications is removed. By determining the sign of Real-out the symbol that was transmitted can be detected.



Figure 5.9: (a) Moving average used as a sum for the Conv. ACD. (b) Moving accumulator used as a sum for the Prop. ACD. i is the number of samples to accumulate.

The Prop. ACD incorporates multiple ACDs, but there is only one ACD when using the Conv. ACD. Therefore it is possible to use a moving average for the integrator between the first and second stage of the ACD. In the Prop. ACD is only able to use an accumulator between the first and second stage of the ACD. This is because each path has a different amount of samples to contribute to the detection of the symbol.



The goal of the thesis is to make a low power system design. These results will show the performance and power consumption of the designed low power system. The thesis showed which steps were taken to archive the design. The trade-offs of the design are also discussed in this chapter

## 6.1 Trade-offs in design

The following design options that were identified during the design of the system, have an influence on an important trade-off. They are as follows:

- Number of path when using the Prop. ACD (performance vs complexity) By decreasing the number of paths, for example from U15 Prop. ACD to U8 Prop. ACD, the complexity of the implementation is decreased. However, when the number of paths is decreased too much, the performance will be affected
- Word length (quantisation noise vs complexity)
   Decreasing the word length for each part of the system is an important design step.
   Although it significantly reduces the complexity of the system, it will increase the quantization noise. This is the trade-off imposed by quantization.
- 3. Down sampling design (noise folding vs complexity) It is tried to keep the down sampling as simple as possible. While it is preferred to use only one CIC, this is too complex due to the increased word length. Therefore, an extra filter was used for the last step of down sampling. In this last step, a trade-off between complexity and noise folding is identified. The frequency response of the FIR filter is chosen such that the complexity is kept as low as possible without affecting the performance of the ACD. This is done by using a frequency response that allows, the noise power of the signal to increase due to noise folding.

## 6.2 Performance of implementation

In Figure 6.1 the performance of the implementations is plotted together with the theoretical performance as stated in [2]. The performance of C16 and the theoretical performance of C16 are practically the same. There is a difference between the performance of U15 and the theoretical performance of U15. While it is not entirely certain what caused the drop in performance. It occurred because, in order to save simulation time, U1 was used to estimate the necessary quantization of the ACD instead of U8 or U15. U8 and U15 have the same performance. This means that having more paths than U8 does not significantly improve performance.



Figure 6.1: Performance of the implementation of C16, U8 and U15 compared with the performance in theory as stated in [2].

The system is evaluated with a frequency offset present in the received signal. The performance of the system can be seen in Figure 6.2. The BER is plotted for different frequency offsets. Multiple  $E_{\rm b}/N_0$  lines are plotted, for each demodulator. Having a completely horizontal line would indicate that the system is performing completely frequency offset independent. The system was designed to work properly with a frequency offset of 700 Hz. However, this is not the case due to the FIR filter.

When there is no frequency offset in the signal, only the noise will be folded due to the wide transition band of the FIR filter. But when the signal has frequency offset and is situated around the transition band of the FIR filter, the signal will be folded, decreasing the performance of the demodulator.

Figure 6.2 shows that the performance of the demodulator does not decrease linearly with frequency offset. Due to the folding of noise, the noise power is increasing. With more paths, the Prop. ACD is able to reduce more noise power. The Conv. ACD only has one path, this causes the Conv. ACD to perform worse than the Prop. ACD with respect to frequency offset independence. It also explains why the performance of U8 and U15 is better than the performance of U1 in the presence of frequency offset.



Figure 6.2: Performance of implementations with the received signal containing a frequency offset.

Figure 6.3 shows the BER curve for frequency offset equal to 0 Hz and 684 Hz (the best and almost worst case scenario). Here the system using U8 and U15 have the same performance, as expected. The decrease in performance due to frequency offset is larger than preferred.



Figure 6.3: BER curve of the system when there is frequency offset in the signal.

#### 6.3 Power consumption

Figure 6.4 and Table 6.1 show the power consumption during the processing of 1 packet (200 symbols, 2 s). The different power components are explained in chapter 3. The demodulator is implemented with different number of paths to see how the power consumption is affected.

From Figure 6.4a it can be seen that except, at C16 (U0), the leakage power is consuming the most power in every system (It consumes more power than the Switching or Internal power). The power consumption scales almost linearly with the number of paths. When only considering the internal power of the system, as seen in Figure 6.4b, the power consumption is around 8 to  $9\,\mu$ W.



Figure 6.4: Power consumption of the implementations of different multipath. C16 is plotted at the position of U0.

Table 6.1 shows that the amount of energy of the system that is needed to receive 1 package are around  $16 \,\mu$ J to  $17.3 \,\mu$ J, when considering the internal energy. When not considering the leakage power, all systems require almost the same amount of energy while the performance is significantly different.

| ACD type                                | C16   | U1    | U8    | U15   |
|---|-------|-------|-------|-------|
| Internal power [uW]                     | 8.0   | 8.1   | 8.4   | 8.6   |
| Switching power [uW]                    | 0.6   | 0.6   | 0.7   | 0.7   |
| Leakage power [uW]                      | 4.6   | 9.8   | 42.2  | 72.2  |
| Total power [uW]                        | 13.2  | 18.5  | 51.2  | 81.5  |
| Total Internal energy for a packet [uJ] | 16.01 | 16.11 | 16.73 | 17.26 |
| Total Leakage energy for a packet [uJ]  | 9.2   | 19.6  | 84.4  | 144.4 |

Table 6.1: Power consumption of different systems

Table 6.2 shows the power consumption of individual entities. It can be seen that the Prop. ACDs are consuming the most power. The multipliers of the ACD consume the most leakage power. When the Conv. ACD is used, the internal power of the CIC is consuming the most. In the implementation there are two FIR and two CIC entities, for the Real and Imaginary path respectively.

Table 6.2: Power consumption of individual entities for a 2s system run

|                      | FIR   | CIC   | C16   | U1    | U8     | U15    |
|----------------------|-------|-------|-------|-------|--------|--------|
| Internal power [uW]  | 0.034 | 2.639 | 0.024 | 0.074 | 0.390  | 0.662  |
| Switching power [uW] | 0.005 | 0.117 | 0.002 | 0.006 | 0.033  | 0.058  |
| Leakage power [uW]   | 1.005 | 0.161 | 1.953 | 7.139 | 39.270 | 69.159 |
| Total power [uW]     | 1.044 | 2.917 | 1.979 | 7.219 | 39.693 | 69.880 |

Each implementation; C16, U1,...,U15 uses a different amount of power when a packet of 100 symbols is received. Figure 6.5 shows the performance of the implementations versus the power consumption of the different implementations at different values of  $E_b/N_0$ . This

graph gives an indication of the power that is needed for a certain BER. Its behaviour is similar to Figure 4.1. The figure shows that after investing a certain amount of power, the increase in performance becomes negligible.



Figure 6.5: BER performance versus power consumption of the digital system at different  $E_{\rm b}/N_0$  values.



A power efficient design of the digital part of the DDPSK receiver is proposed in this thesis. This design was made by analysing design trade-offs to archive the best design. The performance of the implementations was close to the theoretical baseline performance of [2]. The Prop. ACD has a significant better BER performance in exchange for a power consumption that is almost 12 times higher than the Conv. ACD

### 7.1 Discussion

The used design methodology considered the technology that was used for the ASIC design. By first investigating the properties of the implemented arithmetic function, the design took the following design considerations into account;

- Multipliers that multiply with a constant were preferred over adders and normal multipliers that multiply with two unknown because of their power consumption
- The word length for part of the system was reduced as much as possible to reduce power consumption
- The number of multipliers were reduced as much as possible as these were the largest power consumers in the system

#### 7.1.1 Trade-offs in design

The first identified trade-off was the *number of internal demodulators* of the Prop. ACD versus the *performance* of the Prop. ACD. By modifying the Prop. ACD and reducing some of the demodulators that are in parallel, 50% of the multipliers were reduced to save power. In this thesis a new ratio, PR, was used to investigate the optimal number of internal demodulators. It was found that the optimal number of internal demodulators was 17. This modified Prop. ACD is called U8. It has 8 internal paths to the left and right side of a conventional path of the ACD as seen in Figure 2.6.

The second trade-off was the *word length* of different parts of the system versus the *quantization noise*. It is a choice for the designer how much power consumption must be saved in exchange for quantization noise. Optimal word lengths are different for various blocks. In this thesis, the performance of the implementation was to be as close as possible to the theoretical BER performance. Considering that, the input word length of the DSC was chosen to be 5 bit and input word length of the ACD was 7 bit.

A non-conventional method was used to connect parts of the system with a different word lengths. By using this method it was possible to reduce the word length of separate parts of the system to save power.

The last trade-off in the design is the filter before the final decimation. The filter has lower attenuation and transition band properties than required to *prevent aliasing*. However, it was chosen to do so, in order to *save power*.

Two ACD systems (the Conv. ACD and the Prop. ACD) were designed with the same DSC to compare between the Conv. ACD and Prop. ACD.

#### 7.1.2 Performance

The systems that are made in this thesis were to perform as well as described in theory while focusing on reducing power consumption.

According to simulations of the ASIC design, the Conv. ACD is performing the same as the theoretical BER. It needs a  $E_b/N_0$  of around 12 to archive a BER of  $10^{-3}$ .

The performance of the U8 Prop. ACD and the normal Prop. ACD are bit different than the expected performance from theory. The theoretical performance of the ACD needs an  $E_b/N_0$  of 10 to achieve a BER of  $10^{-3}$ , while the implementations need a  $E_b/N_0$  of around 10.5.

The ACDs should be insensitive for signals with frequency offset, this was not the case in the designed system. This is caused by aliasing after down sampling at the implemented FIR filter. The FIR filter is reduced in complexity causing it to have a wide transition band. An extra margin of transition band should have been taken to account for aliasing.

#### 7.1.3 Power consumption

The total power consumption of the normal Prop. ACD is  $81.5\,\mu$ W, the U8 version is consuming  $51.2\,\mu$ W, while the power consumption of the implemented Conv. ACD is only  $13.2\,\mu$ W. While the difference in power consumption seems significant, they are in the same order of magnitude.

The dominant power component is leakage power. This is because the sample rate of the system is relatively low. The leakage power is affected by the size of the implementation of an individual entity. The ACD is the largest component in the system due to the number of multipliers. The reason that the Prop. ACD is consuming much more power than the Conv. ACD is due to the difference in number of internal demodulators and therefore number of multipliers.

One of the biggest reduction in power while keeping the same performance, is archived when the number of internal demodulators is decreased. The performance of the U8 Prop. ACD and a normal Prop. ACD is practically the same, while the reduction in power consumption is about 40%.

## 7.2 Proposed low power Autocorrolation Demodulator

Three ACDs have been designed and simulated in this thesis. It has been shown that it is possible to make an implementation of the Conv. ACD, Prop. ACD and an adapted version of the Prop. ACD that perform according to theory. For all the ACDs the same DSC is used. The implementations have a power consumption that ranges between  $13.2\,\mu\text{W}$  and  $81.5\,\mu\text{W}.$ 

The Prop. ACD had the best performance and the Conv. ACD is consuming the least amount of power. However, The U8 Prop. ACD is best choice since it has the performance of a normal Prop. ACD while decreasing power consumption.

The thesis has shown that the design for optimal low power ACD depends on the real-life specifications. There are multiple trade-offs to consider. The designer needs to make choices considering the *performance* that is required and *power consumption* of the *transmitter* and *receiver*.

## 7.3 Recommendations

The main challenge of the thesis were the undefined specifications. Most of the specifications were taken from by [2]. However, for a real-life implementation, the system can be optimized even further using the design method that was used in this thesis.

There are still several other methods that can be used to improve the design. Two possible optimizations are discussed in the following sections.

#### 7.3.1 Tuneable design

When the wireless sensor nodes are used in different places, they will meet different communication channels. The different places will not always have the same  $E_b/N_0$ . It is possible to tune the system to these different circumstances. By doing so a general design can be made that can be used for multiple situations. When tuning the general design, it is also possible to save power when the system is running.

If the frequency offset is less than the system is designed for, e.g. the system is designed for a maximum frequency offset of 700 Hz, but the frequency offset is only 100 Hz. It is possible to reduce the filter BW of the demodulator. Since the system is discrete, all sampling frequencies are related to each other. The CIC is able to change its down sampling rate by simply choosing another decimation factor. By doing so, the number of samples that need to go through calculations can be reduced.

Another method to reduce power is only applicable to the Prop. ACD. The Prop. ACD consists of multiple paths with each an individual ACD. Some of the ACD might be redundant when the  $E_{\rm b}/N_0$  of the received signal is high enough. It is possible to deactivate some of the extra paths that are unnecessary and with that reduce the power consumption

With the current implementation, it is possible to apply both of the mentioned power reduction methods. However, the control entity has to be adapted accordingly.

#### 7.3.2 Reducing power consumption

In chapter 3, it has been shown that static power is a significant part of the power consumption at a low clock frequency. To optimize static power consumption, each entity should perform its calculations with the highest clock frequency as possible. The entity should be cut from power when not used. Because the system in this thesis is designed as a streaming application, this is not possible.

To make it possible for each block to work at its highest possible clock frequency, there is the need for a memory block. The design for such a system is as seen in Figure 7.1. Each block of the system should execute sequentially. First, the signal is received from the ACD and saved as a list of values (in a memory block).

When the memory is filled the CIC block will execute with the highest possible clock frequency . It will read the values from the memory, and write back to the memory, overwriting the original values. Because of the decimation in sampling performed by the CIC there will be no conflict in reading and writing to the list in the memory.

In the following step, the FIR filter will read from the list and write back, behaving similar to the CIC. In the last step, the samples from the memory will pass through the demodulator and the transmitted symbols are detected.

It is not certain that when using a memory block, the system will consume less power than the streaming application. The memory block should consume less power than what can be saved compared to the streaming application. This will need further investigation.



Figure 7.1: Proposed system structure when using a memory block.

# **List of Figures**

| 1.1  | (b) Signal spectrum between the filter and demodulator. Data BW indicates the BW of the signal of interest.   | 6  |
|------|---|----|
| 1.2  | The focus of the design indicated with the dashed rectangle. (The filter of Figure 1.1a is incorporated into the DSC)   | 6  |
| 2.1  | Detailed figure containing the steps of the signal conditioning   | 8  |
| 2.2  | Single sided data BW and sampling frequency (Fs) of each part of the system.  | 8  |
| 2.3  | Double differential demodulator.  | 9  |
| 2.4  | Double differential encoding.   | 10 |
| 2.5  | discrete version of the demodulator.  | 12 |
| 2.6  | Figure of Prop. ACD with the conventional path, L paths and R paths. $\ . \ .$  | 13 |
| 2.7  | Explanation on how the samples are multiplied and summed with $N = 4$ .<br>The conventional method can be seen at (a). (b) and (c) shows the steps that are taken in the L and R path of the proposed method when $P = 1$ . | 14 |
| 2.8  | Theoretical performance of the ACD [2]. The C curves are the performance of the Conv. ACD and the P curve belongs to Prop. ACD. The number behind the 'C' or 'P' indicates the value of $N$ .                               | 15 |
| 2.9  | FIR filter with 7289 coefficients.  | 16 |
| 2.10 | FIR filter with 10 coefficients.  | 16 |
| 2.11 | IIR filter consisting of 3 sections.  | 17 |
| 2.12 | Left the moving average filter, Right another implementation of the same  |    |
|      | filter [10]   | 18 |
| 2.13 | CIC filter before decimation.   | 18 |
| 2.14 | CIC filter combined with decimation, also called the Hogenauer filter   | 18 |
| 2.15 | Magnitude response of a CIC filter of Figure 2.14   | 19 |
| 2.16 | Components of power Dissipation as explained in [12].   | 20 |
| 3.1  | Power consumption of different arithmetic operations at different word length<br>and different clock periods. 'Multiply Static 7' is a implementation of a mul-<br>tiplication with an unknown that is multiplied with 7    | 23 |
| 3.2  | Power consumption of different arithmetic operations when using different clock periods.  | 24 |

| 4.1        | Performance ratio of Prop. ACD with different number of paths at the Left and Right side   | 26 |
|------------|--|----|
| 4.2        | The Simulink model for quantisation of input values.   | 27 |
| 4.3        | Influence of input word length on the performance of the demodulator U1 and C16  | 27 |
| 4.4        | Frequency response of the designed filters. The frequency axis is normalized to the BW of the sampling frequency of the filter                               | 30 |
| 4.5        | Performane of the demodulator with the filters infront of the ACD  | 30 |
| 4.6        | BER performance of system where the FIR and CIC are quantized with different word lengths  | 31 |
| 4.7        | Designed system.   | 32 |
| 5.1        | System to that is implemented in VHDL  | 33 |
| 5.2        | Decreasing the word length by selecting relevant bits using the method that is used in this thesis.  | 34 |
| 5.3        | Effect of decreasing a word length on a signal   | 34 |
| 5.4        | Schematic implementation of baseband conversion.   | 36 |
| 5.5        | Schematic of CIC filter implementation.  | 36 |
| 5.6        | Schematic of FIR filter implementation.  | 37 |
| 5.7        | Schematic of first stage of ACD.   | 37 |
| 5.8        | Schematic of second stage of ACD   | 38 |
| 5.9        | (a) Moving average used as a sum for the Conv. ACD. (b) Moving accumulator used as a sum for the Prop. ACD. <i>i</i> is the number of samples to accumulate. | 38 |
| <b>C</b> 1 |  |    |
| 6.1        | Performance of the implementation of C16, U8 and U15 compared with the performance in theory as stated in [2].   | 40 |
| 6.2        | Performance of implementations with the received signal containing a fre-<br>quency offset.  | 41 |
| 6.3        | BER curve of the system when there is frequency offset in the signal. $\ldots$   | 41 |
| 6.4        | Power consumption of the implementations of different multipath. C16 is plotted at the position of U0.   | 42 |
| 6.5        | BER performance versus power consumption of the digital system at different $E_b/N_0$ values   | 43 |
| 7.1        | Proposed system structure when using a memory block  | 47 |

## List of Tables

| 1.1 | Specifications of the system   | 7  |
|-----|--|----|
| 4.1 | Performance of different version of ACD at a BER of $10^{-3}$ $\ldots$ $\ldots$ $\ldots$ | 26 |
| 4.2 | Unique values to multiply with for baseband conversion based on sampling                 |    |
|     | ratio  | 28 |
| 4.3 | Properties of considered filters   | 29 |
| 4.4 | Word length of different parts of the design   | 32 |
| 6.1 | Power consumption of different systems   | 42 |
| 6.2 | Power consumption of individual entities for a 2s system run                             | 42 |

## Bibliography

- M. K. Simon and D. Divsalar, "On the implementation and performance of single and double differential detection schemes," *IEEE Trans. Commun.,e*, vol. 40, no. 2, pp. 278–291, 1992.
- [2] A. K. Siavash Safapourhajari, "Demodulation of double differential psk in presence of large frequency offset and wide filter," *IEEE 87th Vehicular Technology Conference*, 2018.
- [3] T. S. Changle Li, Liran Wang, "Topology analysis of wireless sensor networks based on nodes' spatial distribution," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2454–2453, 2014.
- [4] a University of Twente | EEMCS Faculty | Telecommunication Engineering, "Slow wireless - project summary," 11-7-2017. https://www.utwente.nl/en/eemcs/te/ projects/SRR/slow\_wireless/.
- [5] A. F. Molisch, Wireless Communications, ch. 10: Structure of a Wireless Communicaiton Link, p. 182. 2011.
- [6] M. Works, "Simulink," 28-1-2019. https://nl.mathworks.com/products/ simulink.html.
- [7] H. J. Landau, "Sampling, data transmission and nyquist rate," Proceedings of the IEEE, vol. 55, no. 10, pp. 1701–1706, 1967.
- [8] Z. Wang, "Slow wireless communication testbed based on software-defined radio," Master's thesis, University of Twente, 1 2017. Chapter 4.4.
- [9] M. Works, "Using filter designer," 28-1-2019. https://nl.mathworks.com/help/ dsp/ug/using-filter-designer.html.
- [10] F. j Harris, Multirate Signal Processing For Communication Systems, ch. 11: Cascade Integrator Comb filters, p. 325. 2004.
- [11] C. S. B. S. Chu, "Multirate filter design using comb filters," IEEE Trans. on Circuit and Systems, vol. 31, no. 11, pp. 913–924, 1984.
- [12] Synopsys, Power Compiler User Guide, ch. 3: Performing Power Analysis, pp. 3–1. 2010.
- [13] tsmc, "40nm technology," 26-2-2019. https://www.tsmc.com/english/ dedicatedFoundry/technology/40nm.htm.
- [14] Synopsys, "Design compiler graphical," 9-1-2019. https://www. synopsys.com/implementation-and-signoff/rtl-synthesis-test/ design-compiler-graphical.html.

- [15] CAES, "Synopsys documentation," 9-1-2019. https://caesdoc.ewi.utwente.nl/ documentation/eda/Synopsys/.
- [16] Synopsys, *Power Compiler User Guide*, ch. 6: Performing Power Analysis, pp. 6–8.2010. Analyzing Power With Partially Annotated Designs.
- [17] S. Arar, "Pipelined direct form fir versus the transposed structure," 21-5-2018. https://www.allaboutcircuits.com/technical-articles/ pipelined-direct-form-fir-versus-the-transposed-structure/.