VLSI System Design

Design of a demodulator

Supervisor Ondrej Harwot

Mats Larsen

Deadline: d. 14.05 2012

Czech Technical University in Prague - Electrical Faculty.

Contents

Chapter 1 Introduction **2**

Chapter 2 Blockdiagram **3**

Chapter 3 Description of blocks **4**

3-1 CIC Filter 4
3-2 Cordic 4
3-3 Shiftregister 4
3-4 Click Remover 4
3-5 Differential filter 6

Chapter 4 Suppression of noise **7**

Chapter 5 Conclusion **8**

1 Introduction

This project is about implementing a FM demodulator in VHDL, more precisely an incoherent demodulation. Figure 1.1 illustrates the system and process.

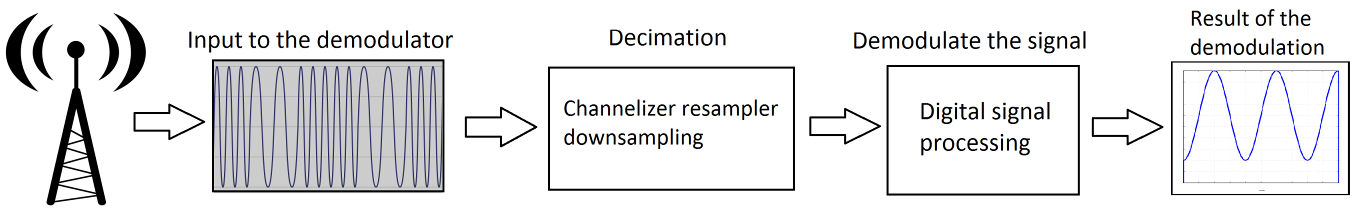
Input to the demodulator
Decimation
Channelizer resampler downsampling
Demodulate the signal
Digital signal processing
Result of the demodulation

Figure 1.1: Demodulator process

The input signal to the demodulator is a generated signal by Matlab. Matlab generates a text file with the coordinates of the input signal, which the simulation is able to read. Then the signal will be demodulated, which is the main goal for this project. The output signal of the demodulation will be write to another text file, so it can be plotted by Matlab for further investigating.

2 Blockdiagram

The blockdiagram of the demodulation is present in figure 2.1.

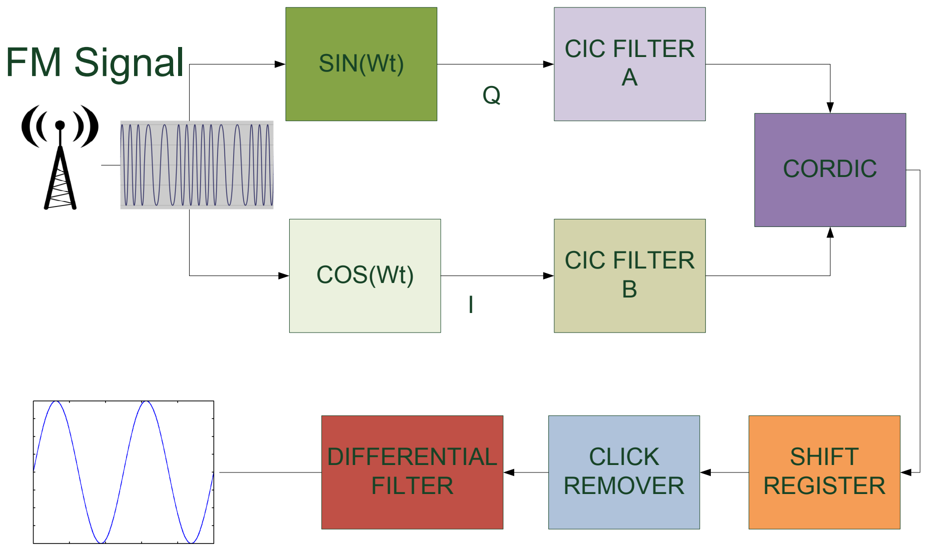

Figure 2.1: Blockdiagram of the demodulation

The project basically consists of five blocks or modules. These blocks will be described in following sections. To obtain I and Q, it is needed to mix sine and cosine separately to the input signal. Then I and Q will be phase 90 degrees between each other. By mixing cosine it will produces a real in-phase(I) and by mixing sine it produces an imaginary quadrature-phase(Q).

3 Description of blocks

3-1 CIC Filter

The CIC filter is implemented by using Xilinx IP core. Filters' purpose in the design is to provide decimation, to reduce number of samples by 10. It means that the sampling rate is 100 MHz from the main clock and after the processing by the filter the sampling will be 10 MHz. The ready signal for all blocks is working in a chain. It means when a block is finish and delivers the result to the next block with a ready signal as control signal. This core has latency on 15.

3-2 Cordic

This block is implemented by using Xilinx IP core. The Cordic get two inputs respectively I and Q from CIC Filters. The Cordic performs Vector Translation on the inputs to transform rectangular to polar which is amplitude and phase. In this design the phase output is only useful, because the amplitude only concerning AM modulation. This core has latency on 20. The Cordic phase output can be seen in figure 3.1. In the figure we can see some transitions from positive to negative and vice versa. The reason is that the Cordic only can provides output from $-\pi$ to π. It will show later, that these transitions will cause problems.

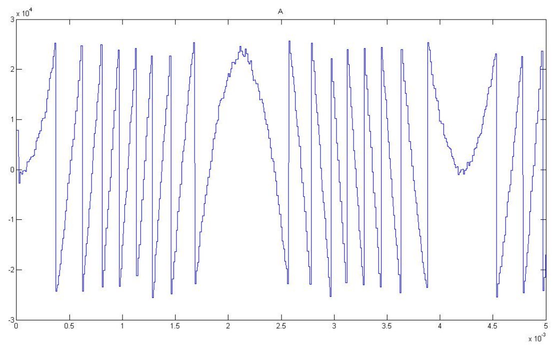

Figure 3.1: Phase output of the Coridc

3-3 Shiftregister

The differential filter requires previous sample, x(n), x(n-2),x(n-4),x(n-6). The shiftregister will shift all the samples, when a new sample is read. The process is illustrated in figure 3.2.

3-4 Click Remover

The purpose of this block is to avoid clicks in the final output. The output when the click remover is disabled can be seen in figure 3.3. This part of the design was one the more difficult to fine-tune the

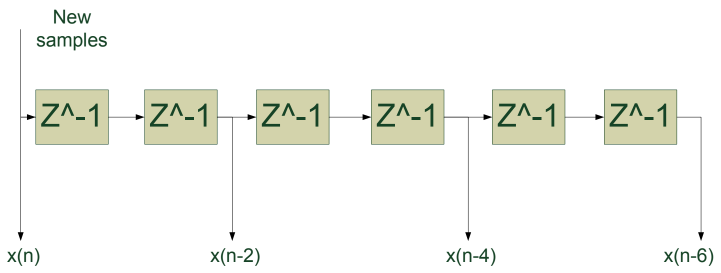

Figure 3.2: Shiftregiser process

click remover so it removes all clicks. These clicks occur when the differential filter uses the sequence of samples which contains a transition from positive to negative or versa visa. First the clicks will be detected by comparing $x(n) - x(n-6) > \pi$ or $x(n) - x(n-6) < -\pi$. If a click is detected, the algorithm will add two π to the four samples, if the samples are less than zero. After this operation the sample interval will be continuous(It is continuous when this specific transition isn't present). The illustration can be seen in figure 3.4.

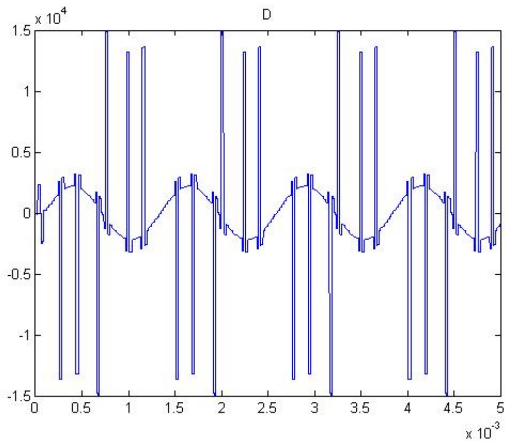

Figure 3.3: The final output with clicks

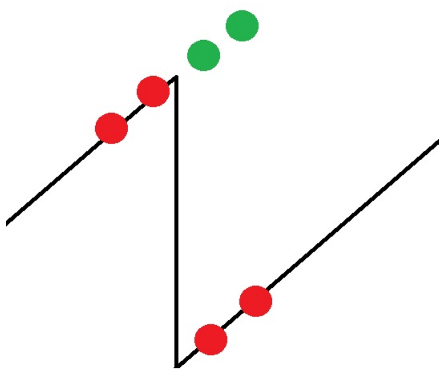

Figure 3.4: The red points are the original samples, it makes this discontinuity. The green points are the result of the click remover.

3-5 Differential filter

This differentiator is the most usable for this design. It has high frequency attenuation and a reasonably linear operation up to $0,17 f_s Hz$. The differentiator is defined by

$$y_{diff} = \frac{-x(n)}{16} + x(n-2) - x(n-4) + \frac{x(n-6)}{16} \tag{3.1}$$

The output is the final result of the demodulation for 500 μs, it is present in figure 4.1.

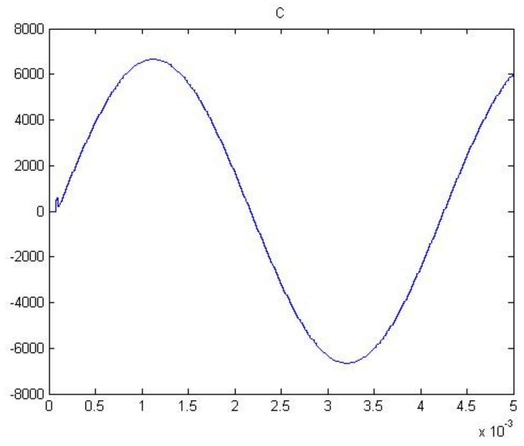

Figure 3.5: Output of the demodulation

It can be noticed that there is some setting time in start of the process.

4 Suppression of noise

An important fact is the property to suppress noise. In this section I will try to add some white Gaussian noise to the FM signal, generated by Matlab. The tolerated Signal-to-Noise-Ratio is specified to 40 dB. Below are some examples with different added noise.

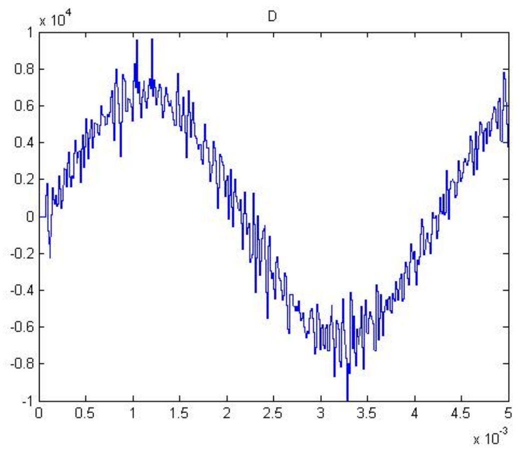

Figure 4.1: Add white Gaussian noise to the input signal which is SNR = 40 dB, it yields 13,6 dB in SNR on output

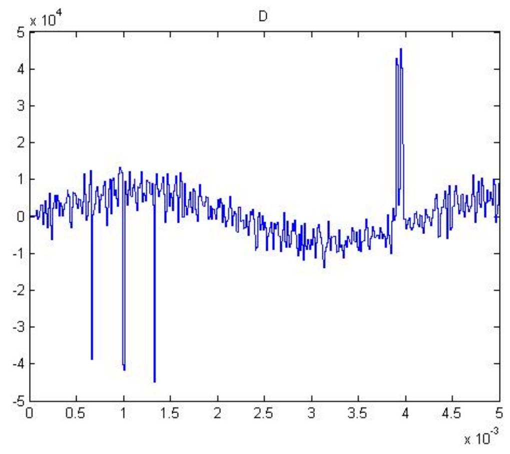

Figure 4.2: Add white Gaussian noise to the input signal which is SNR = 30 dB, it yields 4 dB in SNR on output

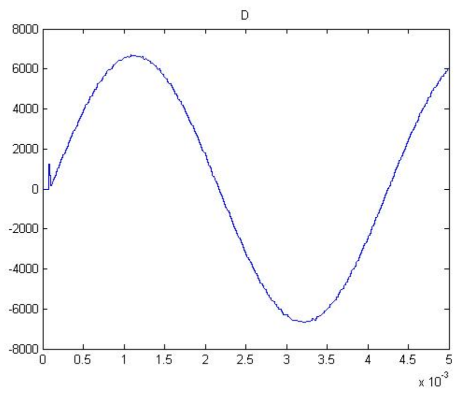

Figure 4.3: Add white Gaussian noise to the input signal correspond to SNR = 70 dB, it yields 40 dB in SNR at output

It shows that the design not suppress the noise very well. It can be seen in figure 4.2 that a signal with SNR = 30 dB will enforce clicks back to the output again. The output will first be satisfied with a signal with SNR higher than 70 dB.

5 Conclusion

I can conclude that I have designed and implemented a demodulation in VHDL. The test shows that implementation works, that the demodulator can interpret the FM signal. I have some problems about removing all clicks, but the result shows that the design can handle to remove all clicks until the SNR be less than 30 dB. The negative aspect is that the design doesn't perform very well respect to the noise. It means that the design needs some improvements to minimize the influence of noise.