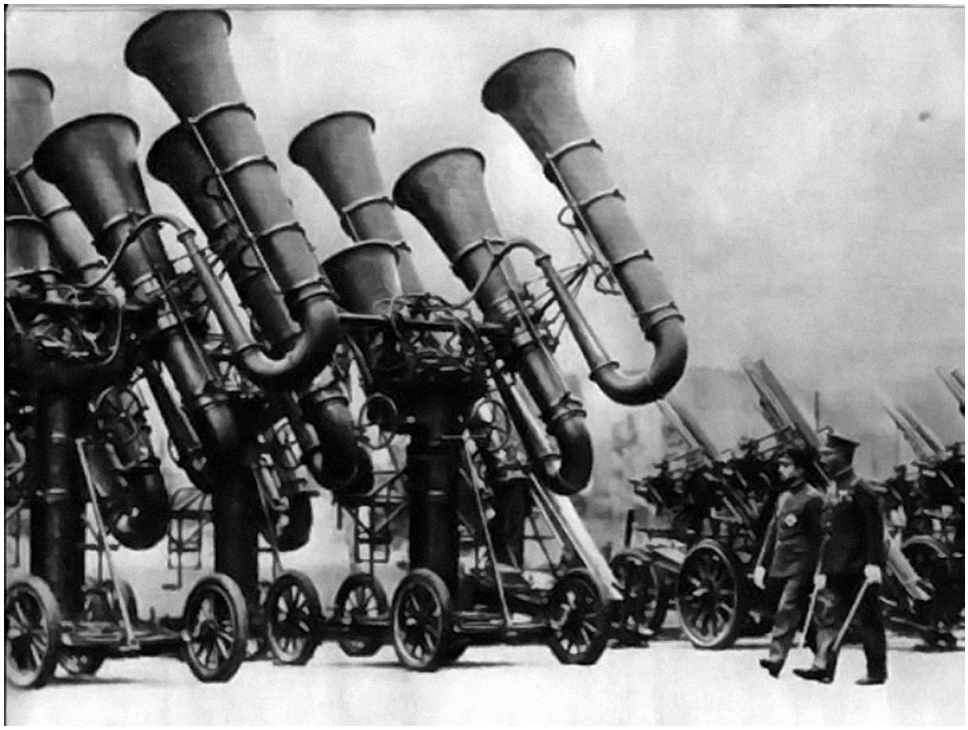
CHALMERS
UNIVERSITY OF TECHNOLOGY



Energy Efficient Digital Signal Processing in Radar Receiver Systems

Degree Project, MSc in Electrical Engineering

JOHAN FRIBERG
JOHAN KALMNER

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014

Energy Efficient Digital Signal Processing in Radar Receiver Systems

Johan Friberg, Johan Kalmner

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Tel: +46 (0)31-772 1000
Fax: +46 (0)31-772 3663

Cover:
The picture on the front page is a Japanese War Tuba and was used as an acoustic airplane detection system before the radar was invented [1]. Even in the predecessors of radar it was desirable with multiple receivers to gain resolution.

Department of Computer Science and Engineering
Gothenburg, 2014

Abstract

The number of digital receiver channels in today's radar systems are increasing rapidly. More receiver channels leads to more hardware which causes the power consumption to rise. Meanwhile more channels offers higher system performance. Therefor it is desirable to find a solution that reduces the power consumption, and allows implementation of more channels. Radar systems are commonly implemented in FPGAs and we have evaluated several different methods to improve the efficiency and reduce the amount of hardware needed, including efficient hardware structures, and increased hardware utilization.

Implementations with direct FIR filters are used to benchmark the improved implementations and are considered a reference system. Based on an extensive literature study several simulations, including performance estimations, are performed. A subset of the simulations of each design are of particular interest and are further evaluated through hardware implementation in a Xilinx Kintex 7 FPGA. The power consumption of each implementation is measured, and based on the results the performance for a single demodulator chain is calculated.

An improved system implementation is presented, where the FIR filter is replaced with a CIC-FIR filter combination. The filter implementation offers more than 80 dB attenuation whilst reducing the power consumption, and using a fraction of the hardware compared to the reference system. With the suggested implementation the system can run at 500 MHz. At the cost of a small hardware penalty with slightly increased power consumption the implementation is capable of more than 100 dB attenuation. Finally, we present some areas that can be further investigated to possibly reduce the power consumption even further, and some actions that can be taken that might further improve the performance of the suggested system.

Acknowledgements

This work has been carried out at Saab Electronic Defence Systems. We would like to thank everyone that have been involved, and we appreciate the possibilities we have been given in terms of access to everyones knowledge, high end hardware, and laboratory equipment. Especially, we would like to thank Charlott who made this master's thesis work happen in the first place, Claes and Lars for supervision at Saab and Chalmers respectively, Anders for help with the implementation and the script environments, Rune for lots of interesting input regarding different solutions and theory, and Bo for the help he provided in the laboratory.

JOHAN FRIBERG
JOHAN KALMNER
Göteborg, June 2014

Contents

1

Introduction

Moore's law [2] states that the amount of transistors in integrated circuits doubles approximately every 24 months. To fulfill Moore's law the industry is continuously pushing towards smaller technology nodes [3]. As smaller technology nodes are more suitable for digital than analog circuitry [4] it is desirable to replace analog designs with digital equivalents.

In radar transceiver systems the data rates are increasing and the trend goes towards an increasing amount of parallel channels [5]. Higher data rates and more parallel channels increases the system requirements with respect to data processing rate. The growing amount of transistors in integrated circuits yields increased computational power, which partially handles the data processing rate demands. Though, the available processing power is limited by the available power budget [4] and thus has to be used wisely.

In today's radar systems there are relatively few digital channels processing data that is arriving at a manageable rate. Thus, power efficiency has been a secondary design parameter with low influence on the maximum possible performance. However, as multi-channel radar gains ground, the amount of received data grows fast. Also, transmitter and receiver sub-systems are moved from the analog domain to the digital domain [6] in rapid succession. Typical sub-systems are signal generation, up and down conversion, filtering and antenna lobe forming [7]. The sub-system migration to the digital domain increases the strain on power and space budgets for the front end systems, which are mainly implemented in FPGAs.

For radar applications the design strains can to a large extent be seen in the receiver systems, as the amount of data to process increases rapidly. As a result of the dataset growth and sub-system migration larger and more powerful digital circuits are needed to coop with the increasing data rates, resulting in higher power consumption. With higher power consumption comes increased demands on power supplies and cooling systems.

Saab Electronic Defence Systems, EDS, is therefor interested in techniques to reduce power consumption in FPGA implementations of radar receiver systems with sustained

or increased performance, as stated in the project specification in Appx. A. Thus, the aim is to investigate energy and power saving techniques for digital circuit implementations of signal processing functions in multi-channel radar systems. Power saving techniques may be improved hardware structures, increased utilization of the available hardware resources, and extended exploitation of the implementation tools. We are going to implement one or more of the solutions on an FPGA and measure the actual power consumption. In addition we will compare the achieved results to those of a Saab EDS standard implementation of a corresponding system. To avoid problems with confidential material an implementation with a direct implemented FIR filter will be considered a standard implementation. If possible we will conclude our findings as general design guidelines for use in future work.

2

Theory

In modern radar systems the fundamental functionality lies within wireless communication and Digital Signal Processing, DSP. This chapter delineates the functionality of radar systems, and fundamental wireless communication of which radar is a special case. Different filter structures appropriate for radar applications are introduced, including polyphase filter structures. Further more, what is causing power consumption in digital circuits is briefly addressed.

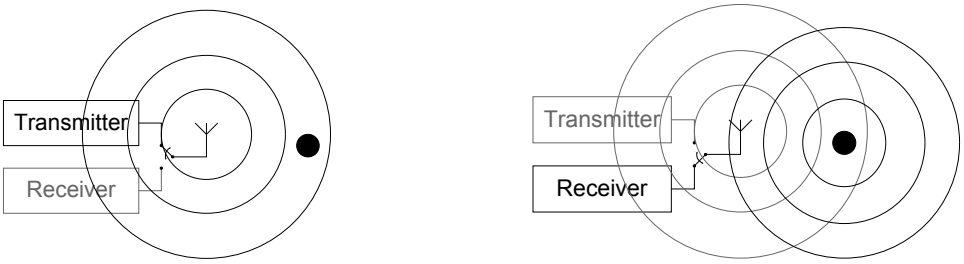
2.1 Radar

Radar is short for radio detection and ranging [8, 9, 10] which indicates what radar systems are used for. Modern radar systems are cable of more than just finding the direction and distance to a target [8, 9, 10]. In contrast to other systems that can perform similar measurements radar can operate in darkness, haze, fog, rain, and snow which is vital in many applications, such as avionics [8, 10].

Radar works by emitting electromagnetic waves from an antenna, as illustrated in Fig. 2.1a. When a wave hits an object it is partially reflected back to the radar receiver [8, 9, 10], as shown in Fig. 2.1b. The distance, R, to the object is found to be the time, Δt, between sending and receiving the pulse multiplied by the speed of light, c, and divided by two because the pulse has to travel twice the distance to the target [8, 9, 10], and can be calculated according to Eqn. 2.1.

$$R = \frac{c\Delta t}{2} \tag{2.1}$$

The most common radar types uses the same antenna for transmitting and receiving, which can be achieved by sending for a short time and thereafter listen for the echoes from distant objects. Hence this type of radar is called pulse radar [10]. A block diagram for a pulse radar system can be seen in Fig. 2.2.



(a) Transmitting. **(b)** Receiving.

Figure 2.1: Conceptual view of a radar system.

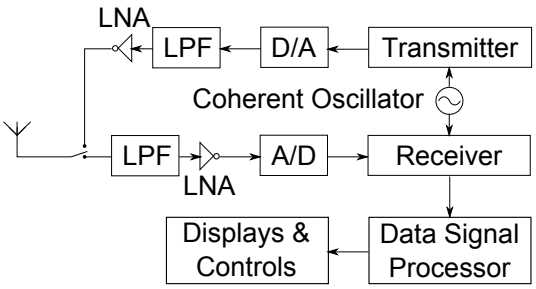


Figure 2.2: Block diagram for a pulse radar system.

When a pulse is to be sent the antenna switch is set to its upper position. The pulse is generated in the transmitter and is converted to an analog signal in the D/A-converter. To smooth the quantized analog converted signal it is low-pass filtered. Before the analog signal is emitted via the antenna it is amplified using a low-noise amplifier. When the pulse is sent the antenna switch is again set to receive signals, echoes of transmitted pulses.

The received signals from the antenna are low-pass filtered and amplified before they are converted into the digital domain. By low-pass filtering the signal folding of high frequency noise into the frequency band of interest in the A/D-converter is avoided. In the receiver the digital signal is processed and converted into raw data for further processing in the data signal processor. The calculated radar image is then displayed in an user interface. Complementary information such as target speed, direction, altitude, information from systems such as Identify Friend or Foe, IFF, and much more [8, 9, 10] is added to the displayed radar image in modern radar systems.

2.1.1 Radar Equation

The radar equation is one of the fundamental design considerations when designing a radar system since it offers the possibility to make an estimation of the system performance. The derivation of the radar equation, as shown in this section, has been done by Skolnik, Toomay, Faulconbridge, and Richards [7, 8, 9, 10] among others.

As mentioned in Sec. 2.1 a radar transmitter emits electromagnetic waves that are reflected in the target and returns to the radar antenna. The power density at the target, P_d, at a distance R from an antenna with a specified gain, G_t is determined by the following expression:

$$P_d = \frac{P_t G_t}{4\pi R^2} \tag{2.2}$$

where P_t is the transmitted power from an isotropic antenna. Since radars make use of directive antennas, the power is focused in the direction of the target. The antenna gain, G_t, is also denoted directivity and can be defined according to the following expression:

$$G_t = \frac{\text{maximum power density radiated by directive antenna}}{\text{power density radiated by a lossless isotropic antenna with the same power input}} \tag{2.3}$$

Not all the directed power hits the target and not all power that actually hits the target is reflected back towards the radar. How much of the reflected power that is directed back towards the radar is dependent on the target cross section, σ. The power, P, that is reflected back from the target can be calculated with the following equation:

$$P = \frac{P_t G_t \sigma}{4\pi R^2} \tag{2.4}$$

The reflected power, P, then has to be transmitted back over the distance, R, to the receiving antenna. Depending on the effective receiving-antenna area, A_e, the received power, P_r, is determined by the following equation:

$$P_r = \frac{P_t G_t \sigma A_e}{(4\pi)^2 R^4} \tag{2.5}$$

where the effective area of an antenna can be expanded to the following expression:

$$A_e = \frac{G_r \lambda^2}{4\pi} \tag{2.6}$$

where λ is the carrier frequency wavelength.

Radar systems often use the same antenna for sending and receiving pulses. In such systems G_t, and G_r are equal, and Eqn. 2.5 for received power is reduced to the following equation:

$$P_r = \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 R^4} \tag{2.7}$$

Equation 2.7 is known as the *radar equation*. The maximum rage at which a target can be detected with a given system can be found by solving the radar equation for R. By replacing the received power with the minimum detectable power, S_{min}, the maximum range, R_{max}, can be derived from the following expression:

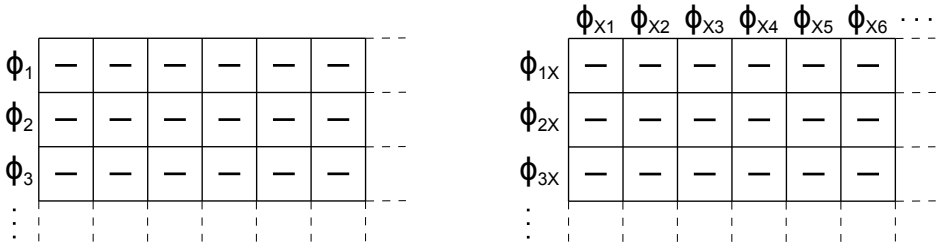
$$R_{max} = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 S_{min}} \tag{2.8}$$

2.1.2 Active Electronic Scanned Array Radar

In the history of radar the most common type of radar system has been using a rotating or pointed reflector with a Transmit/Receive Module, TRM. This type of system has some drawbacks; since the antenna is mechanically steered it is quite slow to switch between targets, if pointed towards them, or has a fixed refresh rate, if the antenna is rotating [8, 10].

A commonly used solution to the problems with mechanically steered antennas is spelled Active Electronic Scanned Array, AESA, radar. The antenna in AESA systems is constructed from arrays of antenna elements which enable the possibility to steer the direction of the antenna lobe by changing the amplitude and phase of the emitted signal among consecutive antenna elements in the array [8].

An AESA antenna can consist of either a linear array of antenna elements, or a planar array, which can be considered a linear array where each array element is a linear array of antenna elements. In a linear array antenna, illustrated in Fig. 2.3a, the phase of the emitted signal from each element in one row is the same. It reduces the number of required TRMs, but limits the lobe steering to the vertical plane. Figure 2.3b is showing a conceptual view of a planar array antenna, where the phase of the emitted signal from each element is individually steerable which makes lobe steering in two dimensions possible. All elements in a planar array ideally consists of one antenna element connected to a TRM, but may consist of several antenna elements connected to one TRM in order to reduce the amount of hardware needed.



(a) Linear array. **(b)** Planar array.

Figure 2.3: Phase shift, ϕ, possibilities in different AESA antenna structures.

Rapid and agile beam forming, multiple target tracking, and lower radar cross section are some of the benefits with AESA antennas, if properly designed [8]. As the technologies for AESA antennas and digital hardware develop in rapid succession, the number of TRMs keeps increasing, which significantly reduces the radio frequency, RF, losses in the system compared to using a centralized TRM [11]. The increasing number of TRMs also increase the reliability of the system since it allows graceful degradation of system performance compared to the old mechanically steered single TRM systems [11, 12]. Higher numbers of TRMs also allows for more tasks to be carried out virtually simultaneously since more spatial beams can be controlled at the same time [12].

2.2 Transceiver

In order to transmit and receive data via some kind of physical channel it has to be processed in several steps. Fig. 2.4 is showing a general transmitter/receiver pair, as proposed by Shannon [13], that forms a communication link. Depending on the functions within the different sub-systems the model is applicable on a wide range of systems, from data links over fiber cables to radar systems [14]. There are several different ways to process data depending on the system requirements; it can be amplitude, phase, or frequency keyed, 1D or 2D modulated, time or frequency divided, and more [14]. In the following sections the functionality of each block in Shannons model is described in further detail.

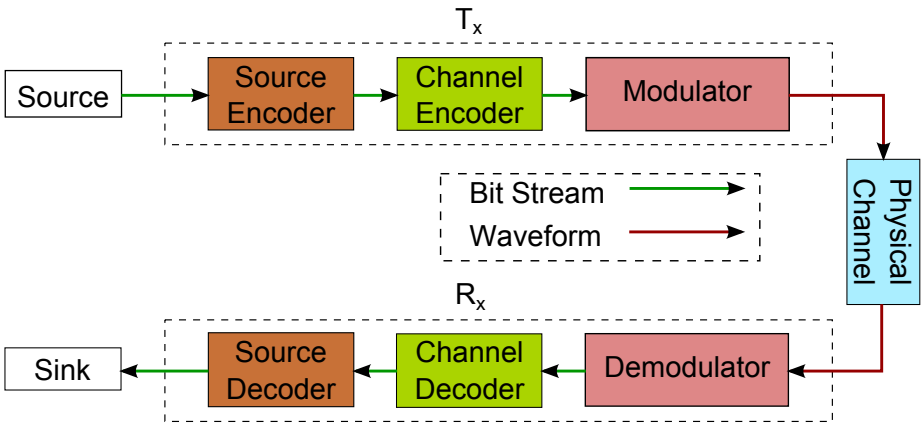


Figure 2.4: Shannon's communication model.

In the source encoder step, the source data is mapped to a constellation point represented by a complex number. The real part and imaginary parts of the complex number forms a pair of inphase, I, and quadrature, Q, signals. Furthermore, the I and Q parts are encoded with a, for the system, suitable pulse in the channel encoder [14].

The signal is upsampled in the modulator. In 2D modulation, orthogonal carriers are used, usually sine and cosine [8], in order to increase the data rate and make best use of the available bandwidth. 1D modulation is a special case of 2D modulation, where the information is carried in only one dimension [14]. In Fig. 2.5 the frequency content of a signal before and after upsampling is shown. Upsampling with a factor R leaves the information unchanged but introduces $R-1$ copies of the original frequency spectrum within the new frequency spectrum $-\pi \leq \omega \leq \pi$, where ω is the normalized frequency with respect to the sampling frequency [15]. In order to remove the introduced copies of the original spectrum the signal is low-pass filtered.

As baseband transmission is impractical the I and Q components are multiplied with a sine and cosine carrier, respectively, of the desired transmission band center frequency, f_c [14]. Before transmission the two signals are combined into a single signal. Since the carriers are orthogonal it is possible to recover the separate signals [15]. Let $s(t)$ denote

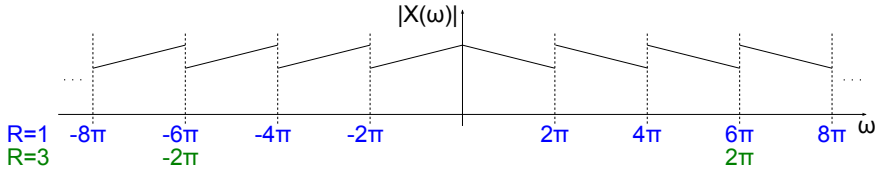


Figure 2.5: Frequency content of a signal before upsampling, $R = 1$, and after upsampling, $R = 3$.

the sent signal, which can be described with the following expression:

$$s(t) = x_I(t)cos(2\pi f_c t) + x_Q(t)sin(2\pi f_c t) \tag{2.9}$$

where $x_I(t)$ is the I component, and $x_Q(t)$ is the Q component.

In the frequency domain the impact on the signal when multiplied with orthogonal carriers is concrete. The frequency representation of the signal is achieved by Fourier transforming it. The Fourier transform of the signal $s(t)$, denoted $S(f)$, is derived according to the following calculations:

$$\begin{aligned} S(f) &= \mathcal{F}\{s(t)\} \\ &= \mathcal{F}\{x_I(t)cos(2\pi f_c t)\} + \mathcal{F}\{x_Q(t)sin(2\pi f_c t)\} \\ &= \mathcal{F}\{x_I(t)\}\mathcal{F}\{cos(2\pi f_c t)\} + \mathcal{F}\{x_Q(t)\}\mathcal{F}\{sin(2\pi f_c t)\} \\ &= X_I(f)\frac{1}{2}\Big[\delta(f+f_c) + \delta(f-f_c)\Big] + X_Q(f)\frac{j}{2}\Big[\delta(f+f_c) - \delta(f-f_c)\Big] \\ &= \frac{1}{2}\Big[X_I(f+f_c) + X_I(f-f_c) + jX_Q(f+f_c) - jX_Q(f-f_c)\Big] \end{aligned} \tag{2.10}$$

which shows that copies of the baseband signal occurs at $\pm f_c$. Figure 2.6 illustrates the shifting of the baseband signal frequency content of bandwidth B to the carrier frequency.

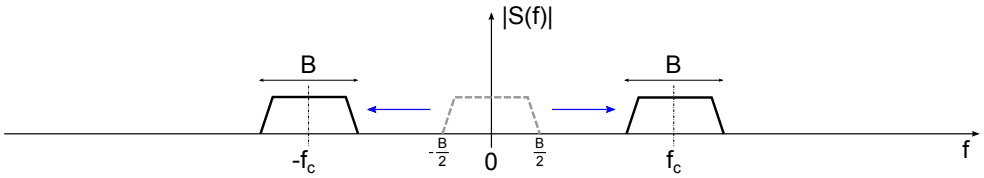


Figure 2.6: Frequency spectrum of a signal moved from baseband to carrier frequency, f_c.

Figure 2.7 is showing a block diagram of a 2D demodulator commonly used in radar applications. The 2D demodulator in Fig. 2.7 corresponds to the demodulator block in Fig. 2.4.

In an ideal system the signal arriving at the receiver, $y(t)$, and the signal sent from the transmitter, $s(t)$, are assumed to be equal. Under that assumption the source data can be perfectly reconstructed. The first step in the reconstruction process is to relocate

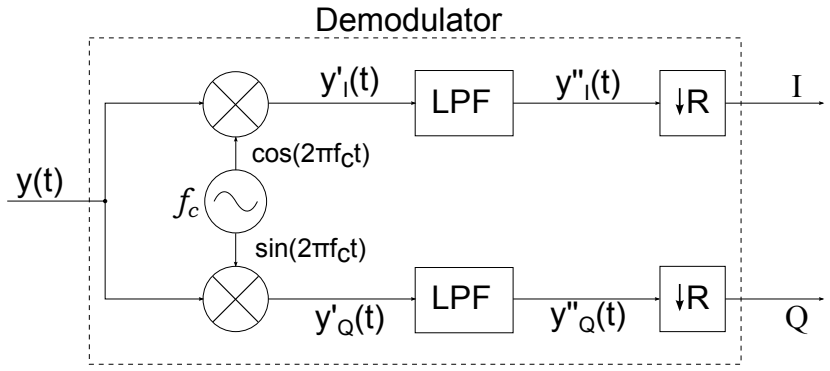


Figure 2.7: The receiver signal processing blocks in question.

the frequency content of the received signal to baseband. It is done in the demodulator stage by multiplying it with the same orthogonal carriers as in the modulator stage [8]. Thus it follows that the I component baseband signal, $y'_I(t)$, can be derived from the following calculations:

$$\begin{aligned} y'_I(t) &= y(t)cos(2\pi f_c t) \\ &\approx s(t)cos(2\pi f_c t) \\ &= (x_I(t)cos(2\pi f_c t) + x_Q(t)sin(2\pi f_c t))cos(2\pi f_c t) \\ &= x_I(t)cos^2(2\pi f_c t) + x_Q(t)sin(2\pi f_c t)cos(2\pi f_c t) \\ &= \frac{1}{2}\Big(x_I(t) + x_I(t)cos(2\pi 2 f_c t) + x_Q(t)sin(2\pi 2 f_c t)\Big) \end{aligned} \tag{2.11}$$

where the factor $\frac{1}{2}$ origins from sine and cosine not being orthonormal, but orthogonal. An orthonormal basis is easily achieved if both sine and cosine are multiplied with a factor $\sqrt{2}$ in the modulator and demodulator. Similarly it can be shown that the Q component baseband signal, $y'_Q(t)$, is determined by the following equation:

$$y'_Q(t) = \frac{1}{2}\Big(x_Q(t) + x_I(t)sin(2\pi 2 f_c t) - x_Q(t)cos(2\pi 2 f_c t)\Big) \tag{2.12}$$

As the signal is mixed down to baseband copies of the signal frequency content are introduced at $\pm 2f_c$ [15]. The Fourier transform of $y'_I(t)$ gives the signals frequency representation, $Y'_I(f)$, which is determined by the following equation:

$$Y'_I(f) = \frac{1}{2}\left(X_I + X_I\frac{1}{2}\Big(\delta(f-2f_c) + \delta(f+2f_c)\Big) + X_Q\frac{1}{2j}\Big(\delta(f-2f_c) + \delta(f+2f_c)\Big)\right) \tag{2.13}$$

and similarly the frequency representation of $y'_Q(t)$, $Y'_Q(f)$, is determined by the following expression:

$$Y'_Q(f) = \frac{1}{2}\left(X_Q + X_I\frac{1}{2j}\Big(\delta(f-2f_c) + \delta(f+2f_c)\Big) + X_Q\frac{1}{2}\Big(\delta(f-2f_c) + \delta(f+2f_c)\Big)\right) \tag{2.14}$$

Figure 2.8 is showing the frequency content of $y'_I(t)$ and $y'_Q(t)$. The frequency content centered around $\pm 2f_c$ from Eqn. 2.13 and Eqn. 2.14 is clearly seen.

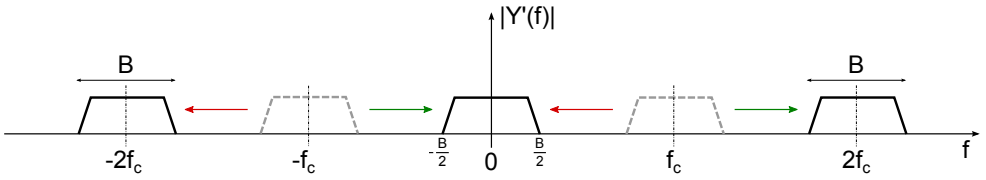


Figure 2.8: Frequency spectrum of a signal mixed down from carrier frequency, f_c, to baseband.

Generally the signal is low-pass filtered after the mixer to attenuate the copies at $\pm 2f_c$ and reduce aliasing [15]. Aliasing occurs when the sampling frequency is decreased. Frequency content located below the Nyquist frequency before downsampling occurs above the new Nyquist frequency, which causes distortion in the band of interest. Fig. 2.9 illustrates the frequency content of a a distorted signal, $|Y_D(f)|$, caused by aliasing. The frequency content of the signal before decimation, $y''(t)$, as a result of ideal low-pass filtering of $y'(t)$ is shown in Fig. 2.10.

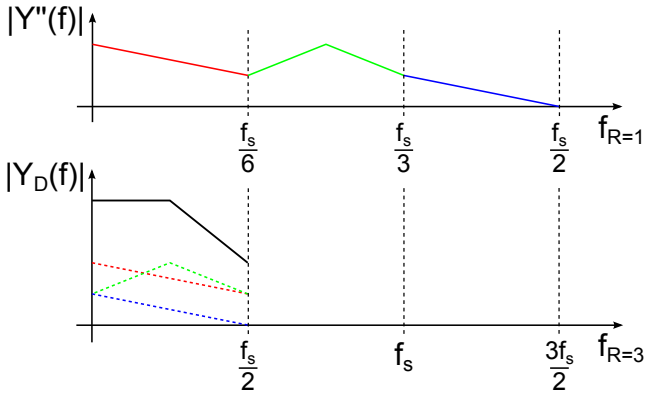


Figure 2.9: Distortion caused by aliasing when downsampling with a factor R=3.

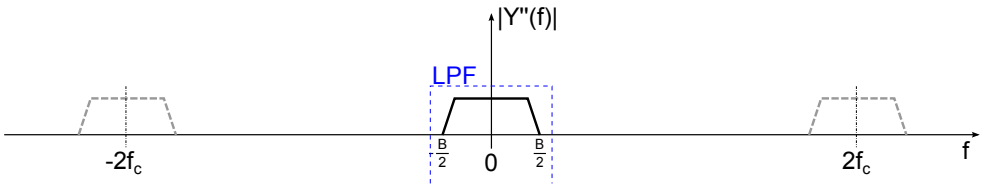


Figure 2.10: Frequency spectrum after ideal low-pass filtering of a signal moved from carrier frequency, f_c, to baseband.

The channel decoder is receiving the output from the demodulator and handles pulse

compression, transforms the pulses into IQ pairs, and accounts for channel equalization. In the source decoder the received IQ pairs are mapped to the constellation and the source data is reconstructed [14].

2.3 Filters

Digital filters play a significant role in signal processing systems, which can be seen in Sec. 2.2. In radar systems the filters need to meet steep requirements whilst being very power efficient. Thus, it is promising to regard different filter structures and the different properties of those.

2.3.1 FIR Filters

Finite Impulse Response, FIR, filters are commonly used in digital signal processing applications where stability and phase characteristics are of great importance. The filter output is produced by delaying the input one or more times, multiply each delayed sample with the filter coefficient corresponding to the delay, and sum the results. In Fig. 2.11 the structure of a basic FIR filter is shown; where h_i corresponds to the i^{th} filter coefficient. A filter of order M has $M+1$ filter coefficients and M delay steps.

Due to the structure the FIR filters are unconditionally stable since they are non-recursive [15]; when a signal has rippled through the filter it no longer has any effect on the output [16]. Also, for a limited input signal the output will be limited since it depends on a limited sum of weighted input values.

An additional property directly connected to the non-recursiveness is that FIR filters have a group delay which is independent of signal frequency [16]. By choosing the filter coefficients in such way that they are symmetric around the middle coefficient the phase response of an FIR filter will be linear [15], a desirable property for radar applications.

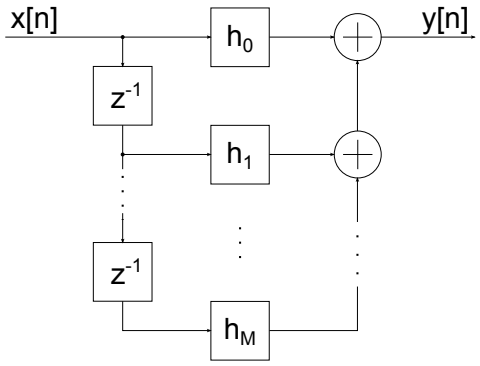


Figure 2.11: Basic FIR filter structure.

From Fig. 2.11 the transfer function of an order M FIR filter can be derived. The filter output, $y[n]$, for an input $x[n]$ is a weighted sum of the current and previous

samples; $x[n]$,$x[n-1]$,$\cdots$,$x[n-M]$. The FIR filter transfer function is given by the following expression:

$$y = \sum_{k=0}^{M} b_k z^{-k} \tag{2.15}$$

To meet higher demands on attenuation, transition band width, and passband ripple the required filter order increases [16]. Higher filter order equals longer delay and more calculations, i.e. multiplications, as the number of filter coefficients grows [16].

A drawback of FIR filters compared to, for example, Infinite Impulse Response, IIR, filters is that for a given specification a greater amount of hardware is required to realize an FIR filter than an IIR filter [15]. IIR filters, though, are not unconditionally stable as there are poles in the transfer function that might be located outside the unit circle [15] in the z-plane.

For a symmetric FIR filter it is possible to reduce the amount of hardware with unchanged functionality by folding the filter and take advantage of the fact that two samples delayed differently will be multiplied with the same filter coefficient [16]. Folding increases the amount of adders, as two samples must be added before multiplication, but saves a severe amount of multiplications, which is desirable [16]. A folded FIR filter structure with even number of taps can be seen in Fig. 2.12.

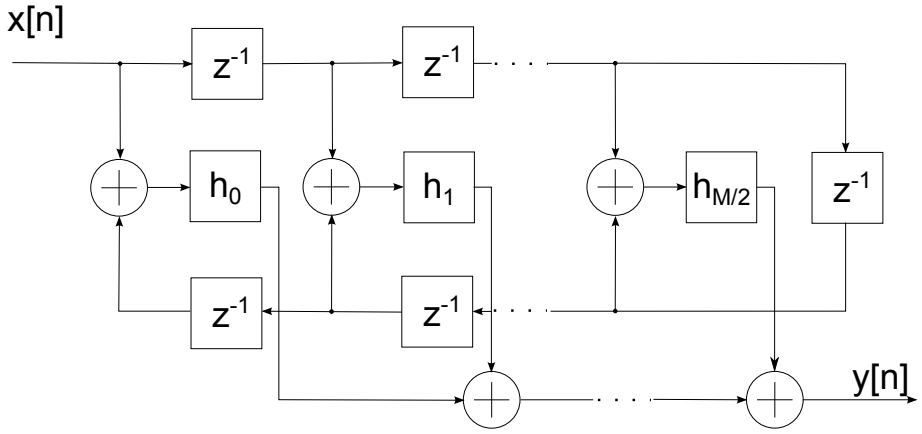


Figure 2.12: Folded FIR filter structure with even number of taps.

2.3.2 CIC Filters

Cascaded Integrator-Comb, CIC, filters is a category of filters that was developed for high-speed DSP applications [17]. The filter type was developed by Hogenauer [18] to have a structure that is flexible and more suitable for hardware implementations. Since a CIC-filter is based on adders rather than large multipliers it can be faster and more efficient than a conventional FIR filter [17]. The possibility to decimate or interpolate

at high rates using the filter makes it suitable for front end applications such as data converters [17].

A CIC decimation filter consists of a number of integration stages working at high sampling rate and an equal number of comb stages that is working on a decimated frequency. Each pair of integrator and comb in the chain produces a linear phase FIR filter [18], which is stable in its nature. The structure for a decimating filter can be seen in Fig. 2.13, where R denotes the decimation rate between the input and output sample rate, and M denotes the differential delay of the comb filters. The differential delay is a design parameter, usually limited to 1 or 2 [17].

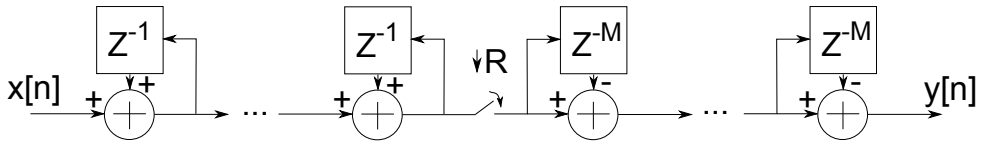


Figure 2.13: Basic CIC filter structure.

It is possible to derive the transfer function for a CIC decimator of order N in a few steps. As a first step recall the transfer function for a single integrator stage [18], $H_I(z)$, and a single comb stage [18], $H_C(z)$, shown in the following equations:

$$H_I(z) = \frac{1}{1 - z^{-1}} \tag{2.16}$$

$$H_C(z) = 1 - z^{-MR} \tag{2.17}$$

The delay factor, R, in the comb stage transfer function is a result of the decimation, only every R^{th} sample from the integrator chain is processed, the other $R-1$ samples are discarded. As serial filtering in the frequency domain corresponds to multiplication of the filter transfer functions the transfer function for a N^{th} order CIC filter [18], $H_{CIC}^N(z)$, is derived according to the following calculations:

$$\begin{aligned} H_{CIC}^N(z) &= H_I(z)^N \cdot H_C(z)^N \\ &= \frac{\left(1 - z^{-MR}\right)^N}{(1 - z^{-1})^N} \\ &= \left[\sum_{k=0}^{MR-1} z^{-k}\right]^N \end{aligned} \tag{2.18}$$

To achieve the frequency response Eqn. 2.18 is evaluated at values given by the following equation:

$$z = e^{j(2\pi f/R)} \tag{2.19}$$

for frequencies, f, relative to the sampling frequency at the low rate side, f_s/R. The evaluation of Eqn. 2.18 gives the power response, $P(f)$, which is expressed in the following

equation:

$$P(f) = \left[\frac{sin(\pi M f)}{sin(\frac{\pi f}{R})} \right]^{2N} \tag{2.20}$$

The CIC decimation filter is easily converted into an interpolating filter by reordering the filter parts so that the comb stages are at the high rate side in the filter and the integrators are at the low rate side. Hence every R^{th} sample in the integrator stage comes from the comb stage and the $R-1$ samples in between are filled with zeros [17, 18].

One of the drawbacks of CIC filters is that they have a substantial amplitude droop in the passband [19], especially for higher order filters. The amplitude droop problem can be solved either by using a compensating FIR filter [19] in series with a CIC filter, or using a more complex CIC filter structure [20]. In addition to compensating for the passband droop it is desirable that the attenuation in the stop bands increases or remains unchanged.

2.3.3 Polyphase Filter Representation

In a decimating FIR filter with decimation rate R only every R^{th} calculated output contributes to the actual output. The transfer function for such filter of length M can be seen in Eqn. 2.15. Assume that the filter length, M, is such that $M = mR - 1$, and let $k = lR + n$. Then Eqn. 2.15 can be rewritten according to the following equation:

$$H(z) = \sum_{n=0}^{R-1} z^{-n} E_n\left(z^R\right) \tag{2.21}$$

where $E_n(z)$ is determined by the following expression:

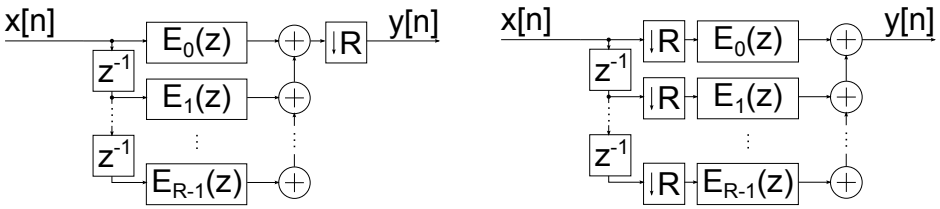
$$E_n(z) = \sum_{l=0}^{M/R-1} h(lR+n) z^{-l} \tag{2.22}$$

Equation 2.21 and Eqn. 2.22 states that the original direct implemented FIR filter can be divided into R filters $E_n(z)$ consisting of every R^{th} filter coefficient from the original filter, with the initial offset n in the n^{th} filter. The new filter structure is shown in Fig. 2.14a.

The parallelism in the structure shown in Fig. 2.14a offers no relief in computational power. Still M multiplications are done in the filter and $R-1$ out of R outputs are discarded. The Noble identity [21] permits the filtering and decimation to switch place, as shown in Fig. 2.14b. That removes the unnecessary computations and the calculations are done at the low rate side, but still all computations are done simultaneously.

Introducing a multiplexed input, as shown in Fig. 2.15, lowers the peak processing load. Instead of calculating all multiplications in one high rate clock cycle M/R calculations are done every R^{th} high rate clock cycle and the results are accumulated.

A polyphase CIC filter structure was proposed by Yang and Snelgrove [22]. A regular CIC filter structure, described in Sec. 2.3.2, is decomposed into the polyphase structure



(a) Filtering performed before decimation.

(b) Filtering performed after decimation.

Figure 2.14: Polyphase FIR filters with decimation factor R.

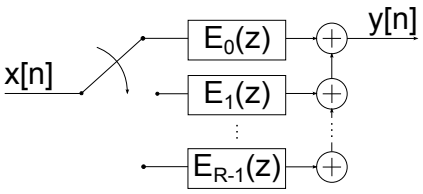


Figure 2.15: A polyphase FIR filter with a decimation factor R without delays.

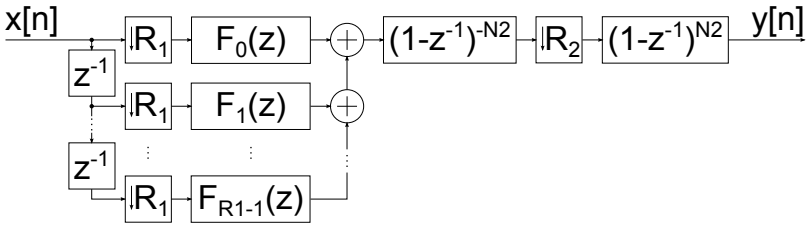


Figure 2.16: Polyphase CIC filter structure proposed by Yang and Snelgrove [22].

shown in Fig. 2.16. It is assumed that the downsampling factor, R, can be factored as $R = R_1 \cdot R_2$, which gives the factored transfer function given by the following equation:

$$\begin{aligned} H(z) &= \left(\frac{1 - z^{-R1}}{1 - z^{-1}}\right)^{N1} \left(\frac{1 - z^{-R}}{1 - z^{-R1}}\right)^{N2} \\ &= H_1(z) H_2(z^{R1}) \end{aligned} \tag{2.23}$$

where the factors $H_1(z)$, and $H_2(z)$ are determined by the following expressions:

$$H_1(z) = \left(1 + z^{-1} + \cdots + z^{-(R1-1)}\right)^{N1} \tag{2.24}$$

$$H_2(z) = \left(\frac{1 - z^{-R2}}{1 - z{-}1}\right)^{N2} \tag{2.25}$$

Polyphase decomposition of Eqn. 2.24 yields the following expression:

$$H_1(z) = \sum_{i=0}^{R1-1} z^{-i} F_i(z^{R1}) \tag{2.26}$$

The polyphase components of $H_1(z)$, $F_i(z)$, are operating at lower rate, f_s/R_1, than the original filter. The complexity of $H_1(z)$ depends on the choice of N_1, and R_1. It can be shown that the polyphase components of $H_1(z)$ can be realized with only a small number of adders and shift registers [22]. $H_2(z)$ is simply a decimating CIC filter.

2.4 Power Consumption

The power dissipation in an integrated circuit can be divided into two categories: dynamic power dissipation, $P_{dynamic}$, and static power dissipation, P_{static}. Dynamic power dissipation is caused by switching of the circuit, as it charges and discharges capacitances [23]. Static power dissipation, on the other hand, is caused by leakage currents such as subthreshold leakage through off-state transistors [24], I_{sub}, leakage from the drain/source diffusion junction, I_{junct}, and leakage through the gate dielectric, I_{gate}. It is also dependent on the supply voltage, V_{dd}. Thus, the static power dissipation is determined by the following equation:

$$P_{static} = (I_{sub} + I_{gate} + I_{junct})\, V_{dd} \tag{2.27}$$

There are several methods that address the static power consumption in order to reduce it. Most of those methods are part of the manufacturing of the chip, though there are methods that can be used at a higher level [4]. From Eqn. 2.27 one obvious way to lower the static power consumption is to lower the supply voltage. Though, lowering the supply voltage might cause impact on system performance [4]. Another, more complex method, is power gating [4]. The supply voltage to parts of the chip that are not in use is lowered or turned off. Power gating though need some built-in hardware support, and careful scheduling.

Switching the transistors in a circuit causes two types of power consumption. Mainly power is consumed when the capacitances connected to the output of the transistors are charged and discharged [24]. A small amount of power is consumed when both the NMOS and PMOS transistors are leading simultaneously as it causes a current to flow through the transistors [4]. Summation of the capacitance switching power, $P_{switching}$, and the short circuit power, $P_{\text{short circuit}}$, gives the following expression for the total dynamic power:

$$P_{dynamic} = P_{switching} + P_{\text{short circuit}} \tag{2.28}$$

In comparison to the power consumed when the capacitances are switched the short circuit power is almost negligible [4]. Thus the dynamic power can be estimated as the capacitance switching power. The capacitance switching power is dependent on the switching activity in the circuit, α, the switching frequency of the transistors, f, the

switched capacitance, C, and the supply voltage squared [25] and is determined by the following equation:

$$P_{switching} = \alpha f C V_{dd}^2 \tag{2.29}$$

Clock gating is a method used to tackle dynamic power consumption. Clock gating is similar to power gating, but instead of turning off the power to a certain part of the chip that part is not clocked. Turning off the clock to circuitry which is not currently in use reduces the switched capacitance. As for power gating clock gating requires hardware support, and scheduling of when the clock to selected parts of the chip can be turned off without affecting the functionality of the circuit [4].

Combining Eqn. 2.27 and Eqn. 2.28 gives the following expression for the total power consumption:

$$P_{total} = P_{static} + P_{dynamic} \tag{2.30}$$

which shows that in terms of power consumption low clock frequency, supply voltage, and amount of switched capacitance is beneficial. Translated to modern FPGAs it means that a smaller implementation footprint reduces the static power consumption whilst a lowered switching frequency reduces the dynamic power consumption. A lower supply voltage affects both static and dynamic power consumption.

3

Simulations

THROUGHOUT the last decade severe amounts of research regarding energy awareness in digital circuits has been performed. As the radar signal processing field is closely coupled with the defence industry there are relatively few resent scientific articles presented on the subject. Based on the available material, and the limitations given by the system several simulations have been performed. Main focus has been on the filters in the receiver, as they stand for a major fraction of the hardware and therefore the largest potential gain, and several alternatives have been evaluated.

In Fig. 3.1 the desired system performance has been summarized. The passband frequency, f_{PB}, was chosen to include the specified signal bandwidth, BW. Stopband frequency, f_{ST}, were set below the Nyquist frequency at the low sampling frequency, $f_{s,LO}$, side to avoid distortion due to aliasing effects, as described in Sec. 2.2. Input sampling frequency, $f_{s,HI}$, were chosen to 500 MHz and the wanted $f_{s,LO}$ was about 10 MHz which led to a decimation rate, R, of 48. The decimation rate were possible to fraction in several ways, which offered more possible filter combinations than i.e. a decimation rate of 50. The attenuation in the stopband, A_{ST}, should exceed 80 dB with passband ripple, A_{PB}, not exceeding 0.1 dB. In addition to the system requirements stated in Fig. 3.1 it was essential that the phase response of the system was linear in the passband.

Figure 3.1: Design target properties.

The system requirements combined with an extensive literature study forged a set of plausible designs. Four of the filter designs were simulated: direct form FIR, polyphase FIR, CIC, and CIC with compensating FIR. Impact on the result for each filter type was explored by alternating the design parameters; desired attenuation, number of filter stages, and decimation rate in each stage. Out of the simulations performed a subset will be presented in the following two sections.

Simulations of the different filters have been carried out using Matlab R2013b [26], and for filter generation the built-in DSP System Toolbox [27] were used. In an initial step filters with full precision floating-point coefficients were generated. As a second step the coefficients were transformed into 18-bit integer coefficients, which is the format used in the target system. The filter response was recalculated using the integer coefficients to verify that the filters still met the specification since integers has lower resolution than floating point numbers.

To the filter generating program a function estimating the total number of Multiplications Per Input Sample, MPIS, and Additions Per Input Sample, APIS, were added. The arithmetic operation estimation extended with compensation for the decreased sample rate provided an initial, weighted, measurement on efficiency. Compensation for the decreased sample rate was done by multiplying the number of MPIS in each filter stage with the inverse of the accumulated decimation it the particular filter before summating the total number of MPIS.

3.1 FIR Filter Simulations

Three different FIR filter configurations out of those simulated were of particular interest; two FIR filters divided into two steps, and one FIR filter divided into three steps. The two first divisions represented the extremes in terms of decimation; in one implementation the signal initially was decimated a factor two, followed by decimation with a factor 24, and in the other implementation the decimation were performed in reversed order. A more optimized division was represented by the FIR filter divided into three steps. Matlab simulations showed that decimation with a factor two, then four, and finally six offered a good trade-off between arithmetic operations per input sample and required number of filter coefficients.

Figure 3.2 shows a cascaded filter setup decimating the signal a factor 2, and then a factor 24. The minimum attenuation was 79.44 dB, and the passband ripple was 0.054 dB. To achieve the required attenuation 379 filter coefficients were needed, which made it the longest filter of the three presented. In the direct implementation of the filter cascade the number of multiplications and additions per input sample both were 196, whereas for polyphase implementation the corresponding numbers were 15 multiplications and 21 additions. Weighting the number of multiplications with the inverse of the decimation factor resulted in a factor 104.5 for the direct implementation and 7.17 for the polyphase implementation. The polyphase result were the lowest among the three simulations.

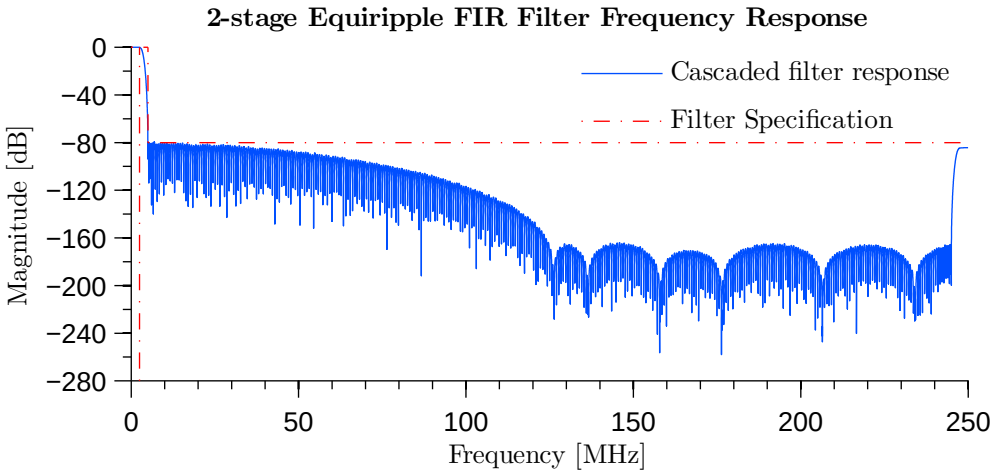


Figure 3.2: Frequency response of two cascaded FIR filters with decimation rate 2, and 24 respectively.

The second setup, where the signal was decimated a factor 24, and then decimated a factor 2, is shown in Fig. 3.3. The filter cascade consisted of 278 coefficients and achieved 79.80 dB minimum attenuation, and 0.072 dB passband ripple. The number of multiplications per input sample were 248.29 or 12.00, and the number of additions per input sample were 248.29 or 18.00 for direct and polyphase implementations respectively. With weighting the numbers were slightly reduced; 247.05 for direct implementation, and 11.02 for polyphase implementation.

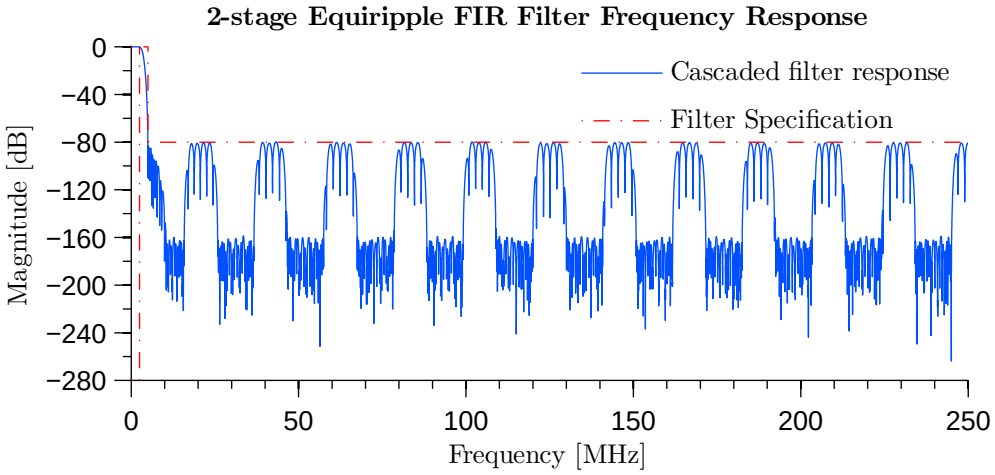


Figure 3.3: Frequency response of two cascaded FIR filters with decimation rate 24, and 2 respectively.

Filter setup three, where the signal were decimated a factor 2, then a factor 4, and finally a factor 6, provided 79.92 dB attenuation and 0.068 dB passband ripple. In

Fig. 3.4 the system frequency response is shown. 139 filter coefficients divided on the three steps translates to 41.5 or 14 multiplications, and 41.5 or 20 additions per input sample for direct implemented and polyphase filters respectively. When the multiplications in each step were weighted the outcome was 22.94 and 7.67 for direct implemented and polyphase implementation respectively.

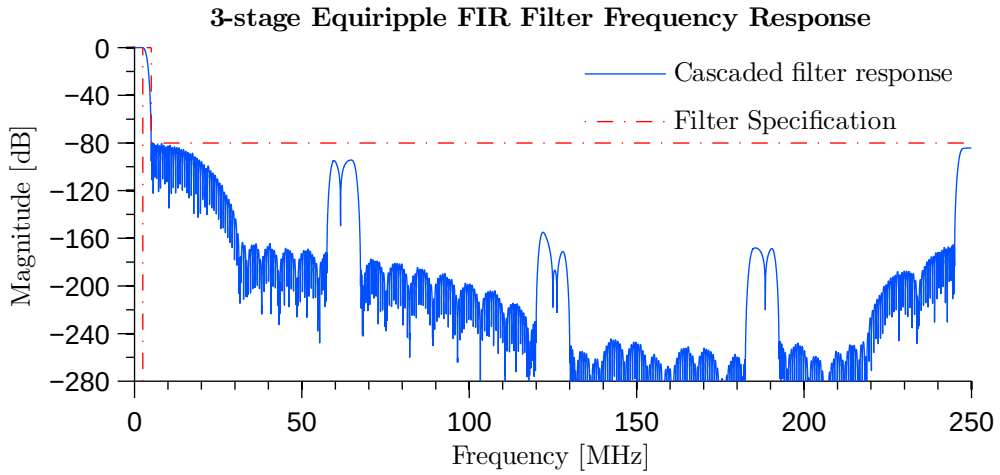


Figure 3.4: Frequency response of three cascaded FIR filters with decimation rate 2, 4, and 6 respectively.

In Tab. 3.1 a summary of the simulation results is presented.

Table 3.1: Summary of the FIR filter simulation outcomes.

	FIR 2-24		FIR 24-2		FIR 2-4-6	
	Direct	Poly.	Direct	Poly.	Direct	Poly.
Attenutation	79.44 dB		79.80 dB		79.92 dB	
PB Ripple	0.054 dB		0.072 dB		0.068 dB	
Coefficients	379		278		139	
MPIS	196.00	15.00	248.29	12.00	41.50	14.00
MPIS (weighted)	104.50	7.17	247.05	11.02	22.94	7.67
APIS	196.00	21.00	248.29	18.00	41.50	20.00

3.2 CIC Filter Simulations

A multitude of CIC-filters were simulated to evaluate the theories from the literature study on how the frequency response reacted on different sets of parameters. The simulations resulted in a range of filters that matched the filter specification or exceeded it. Simulations showed that a CIC-filter with a decimation factor 24 and differential delay of two was the best mach for the filter specification, shown in Fig. 3.1. Since increased differential delay introduces notches at suitable frequencies the attenuation in the stopband were increased even further. An example of two CIC-filters with the same parameters except the differential delay can be seen in Fig. 3.5.

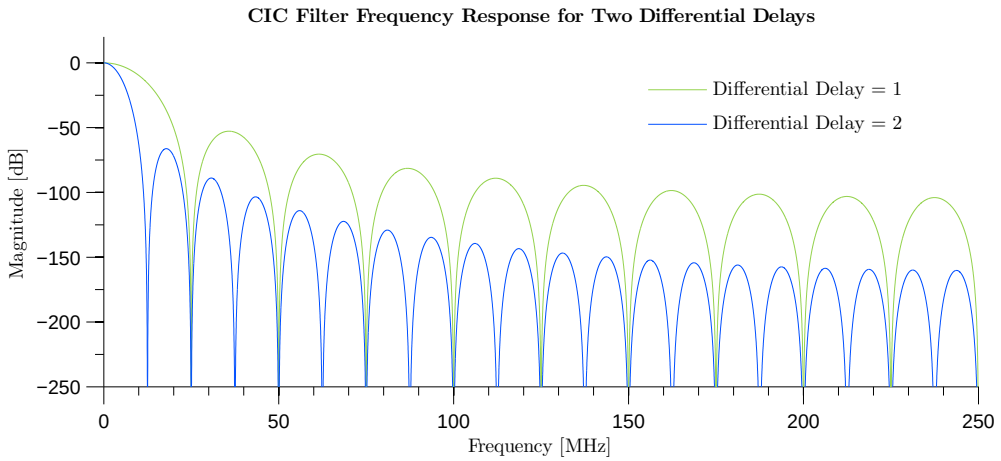


Figure 3.5: Frequency response of two CIC filters with differential delay 1 and 2.

As shown in Sec. 2.3.2 CIC-filters have a droop in the passband which increases with increased differential delay and is quite substantial for higher order CIC-filters. The droop can, as mentioned in Sec. 2.3.2, be compensated by a FIR-filter that has a frequency response which in the passband is a horizontally mirrored version of the CIC-filter frequency response. The frequency response of a compensating FIR-filter for the CIC-filter with differential delay 2 shown in Fig 3.5 can be seen in Fig 3.6. However the compensation filter introduces some ripple in the passband, which reduces with increasing number of coefficients in the compensation filter.

When the above mentioned filters are cascaded the resulting frequency response should be within the filter specification. In Fig. 3.7 the passband and parts of the transition band of the CIC-filter with differential delay 2, shown in Fig. 3.5, and the FIR-filter, shown in Fig. 3.6, is plotted with the filter specification for reference. As can be seen in Fig. 3.7 the FIR-filter frequency response is the inverse of the CIC-filter in the passband and has quite a large gain in the transition band before dropping of. A small amount of ripple can also be seen in the passband of the cascaded filter.

Through the simulations three suitable filter configurations were found that all had a decimation rate of 24 and 2 in the CIC-filter and FIR-filter respectively. The CIC-filters

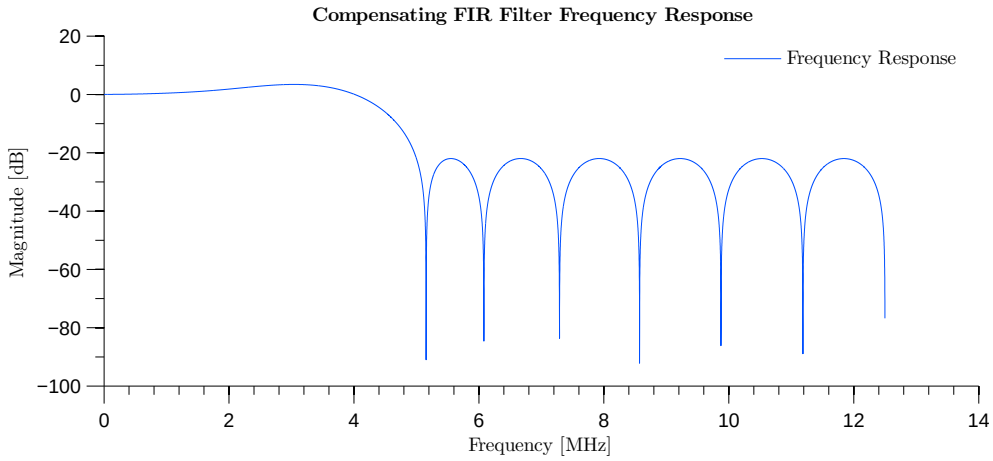


Figure 3.6: Frequency response of a compensating FIR-filter.

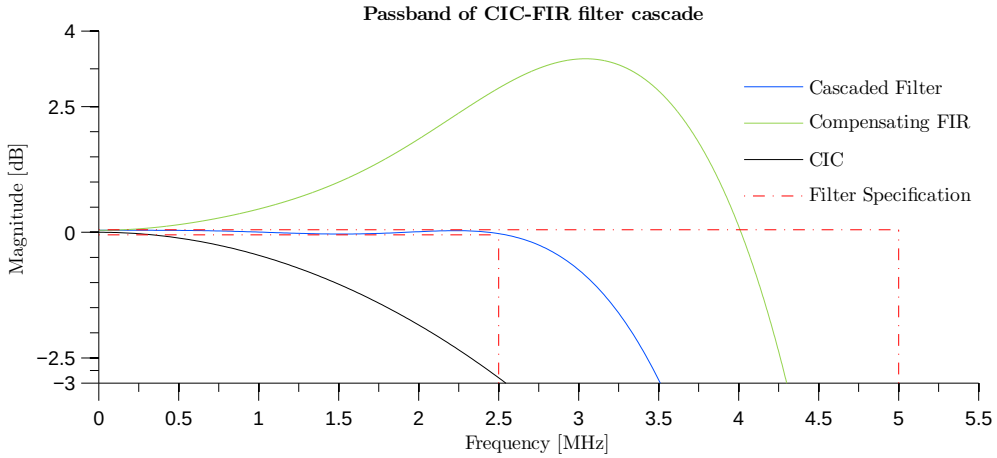


Figure 3.7: Frequency response of CIC-filter, compensating FIR-filter and cascade of the two filters.

had a differential delay of 2 to get good noise suppression at the FIR-filter sampling frequency since that part of the frequency response will fold into the passband when the signal is decimated. As can be seen in Fig. 3.8 the minimum attenuation of the cascaded filter response will be dependent on the fist two notches since the attenuation is increasing with frequency.

In Fig. 3.9 the lower part of the frequency response shown in Fig. 3.8 has been enlarged and the areas that will be folded down into the passband have been marked in white. In this plot it is easier to see how the filters contribute to the total attenuation of the of the filter cascade, the frequency response of the CIC-filter is still distinguishable, with notches at 10.4 MHz and 20.8 MHz, and the effects of the FIR-filter are also clearly distinguishable with the filter response mirrored around 10.4 MHz. Hence, the FIR-

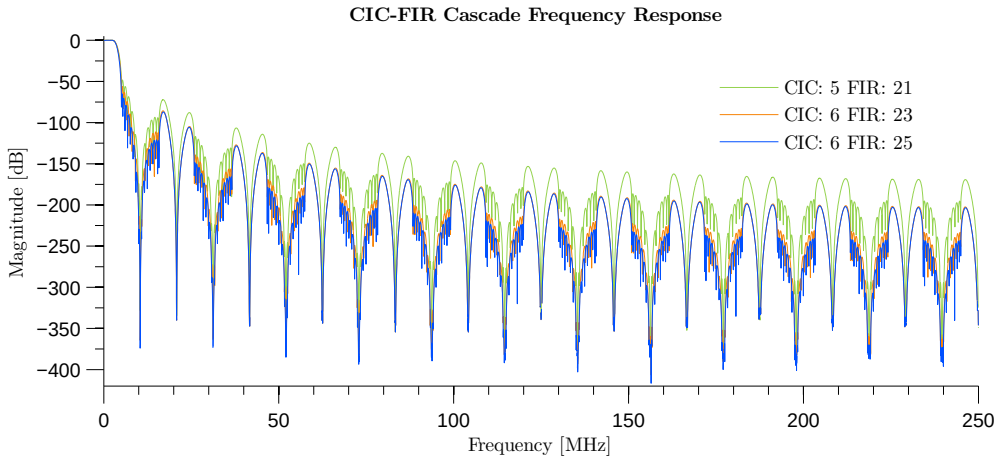


Figure 3.8: Frequency response of the three suitable CIC and FIR cascaded filters.

filter can only increase the attenuation in the first notch and therefore only need a small attenuation to increase the attenuation to the level of the second notch, where the CIC-filter causes all attenuation. Hence the minimum attenuation for the three filter cascades can be measured as difference in magnitude between the passband and the first or the second stopband depending on the filter.

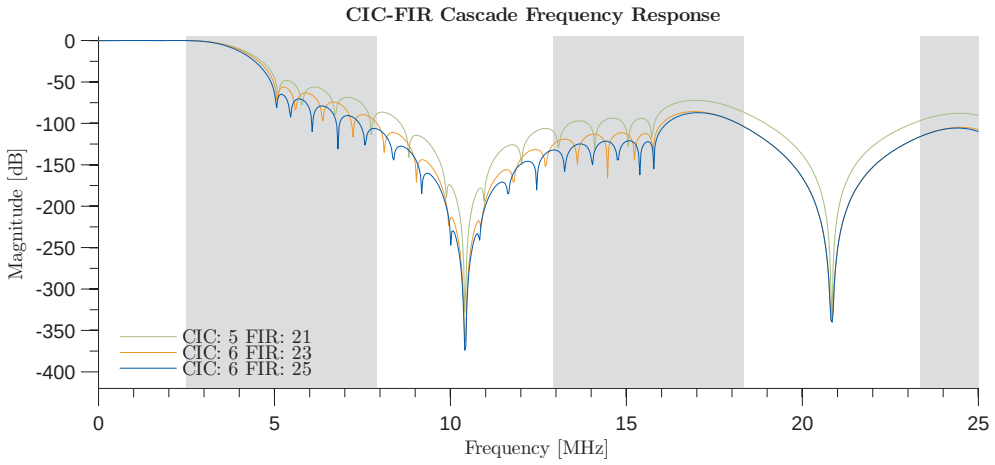


Figure 3.9: Detailed view of the frequency response shown in figure 3.8. The frequency bands marked in white will fold into the passband when the signal is decimated.

In Tab. 3.2 the results from the simulations has been summarized showing the attenuation and hardware utilization for the filter types. As can be seen the MPIS is very low compared to the pure FIR-filters as a result of a low number of multipliers, operating in the low-frequency domain of the filter chain.

Table 3.2: Summary of the CIC-FIR filter simulation outcomes.

	Simulation 1	Simulation 2	Simulation 3
CIC stages	5	6	6
FIR Coefficients	21	23	25
Attenutation	86 dB	100 dB	103 dB
PB Ripple	0.084 dB	0.076 dB	0.075 dB
MPIS	1.46	1.50	1.54
MPIS (weighted)	1.019	1.021	1.023
APIS	6.08	7.20	7.79

4

Implementation

The system on which the implementations were evaluated consisted of several modules. The main module in the system was a Xilinx KC705 evaluation board [28]. A Texas Instruments, TI, ADS5402EVM ADC evaluation module [29] was available to provide real signal input to the KC705 evaluation board. For voltage and current monitoring a TI USB Interface Adapter evaluation module [30] was used. Output data was monitored using a HP 1670E [31] logic analyzer.

Featuring a Kintex 7 XC7K325T-2FFG900C FPGA [32] the KC 705 evaluation board offered both extensive amounts of hardware resources, and multiple I/O interfaces. There were 840 DSP48E1 [33] slices optimized for digital signal processing applications, 16 kB block RAM, 50950 Configurable Logical Cells, CLBs, and 16 GTX transceivers [34]. Each CLB was consisting of four Lookup Tables, LUTs, and eight flipflops each, and each GTX tranceiver was capable of data rates up to 12.5 Gb/s.

The hardware implementation of the demodulator chain was written i VHDL for the Kintex 7 device and was partitioned into blocks to allow several copies of the sub designs to be instantiated. The partition was a conscious choice since the implementation of all components except the filters was identical. Figure 4.1 is showing the system block schematic. Setting generics from the top design file allowed changing the number of demodulator chains and the type of filter used; an important feature when the performance of one chain were evaluated. Several implementations of the system with increasing number of chains were made with each type of filter. Based on the output of the different implementations linear regression were used to account for overhead logic, such as the signal generation block.

The test-signal generator block created the internal test signals and consisted of I and Q signal generation, and a Digital Up Converter, DUC. Virtually the test signal generator is comparable with the modulator described in Sec. 2.2. Each of the I and Q test signals was one or two superimposed sinusoidal signals with or without additive noise. To generate the sinusoidal signals in the FPGA equally distributed amplitude

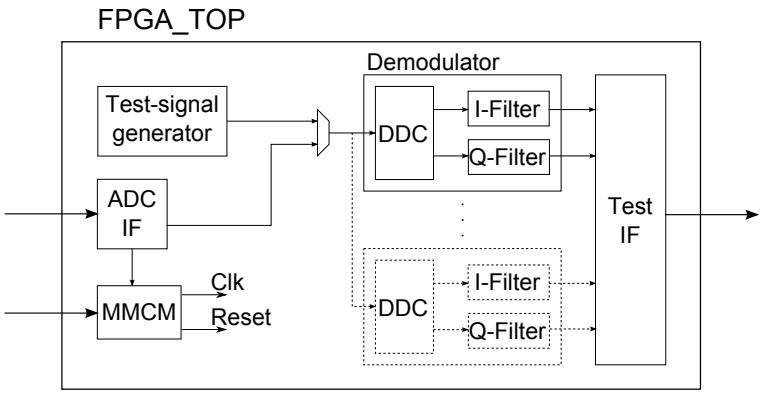


Figure 4.1: Block schematic of FPGA top.

samples of a sinusoidal period were stored in a LUT. Generation of the actual sinusoidal signal from the LUT the address was stepped between consecutive clock pulses. The address step length defined the frequency of the generated signal as a multiple of the lowest representable frequency, which was achieved with step size one. Generation of correct signals address step size had to be shorter than half the amount of entries in the LUT, according to the Nyquist-Shannon sampling theorem [15]. The cosine signals used the same LUT but the address was incremented with a forth of the maximum address which corresponds to the phase shift between a sine and cosine signal.

To allow for realistic test conditions noise could be added to the generated signals by setting a generic. The generation of random noise in the system has implemented as a Linear Feedback Shift Register, LFSR, with variable noise amplitude. The output from the LFSR followed a pseudo-random function with enough states to appear random [35, 36]. I and Q signals were modulated into the transition band and then combined into one signal, as described in Sec. 2.2, forming the test signal to the system. The test signal emulated the radar antenna, DAC, the transmission impairments, and conversion into the digital domain in a real system described in Sec. 2.1.

The signal from the ADC, emulated by the test signal generator, was connected to the digital receiver, corresponding to the receiver in Sec. 2.2. The implemented receiver was consistent with a standard radar or communication receiver, shown in Fig. 2.7, containing a Digital Down Converter with I/Q separation and decimating filter chains.

Several different filters were implemented in order to compare the overall system performance. Xilinx LogiCORE [37, 38], which is Xilinx IP core generation tool, were used to generate the filters to achieve implementations that were capable of operating at the relatively high frequencies targeted whilst being optimized for the Kintex 7 FPGA. Three sets of filters were generated. The first set contained direct implemented FIR-filters, the second set contained polyphase versions of the direct implemented FIR-filters, and the last set contained CIC-filters with compensating FIR-filters. Each filter in the first two sets consisted of several steps, where each filter step were generated as an individual IP-block. Similarily the CIC-filters and FIR-filters in the third setup were

generated as individual IP-blocks. Each filter configuration was connected using a VHDL wrapper which ensured that all filters had the same entity towards the system. Hence, ensuring filter interchangeability without altering the adjacent system blocks.

Two different clocks were used in the system; the system clock which was running at 500 MHz and the test output which was driven at 100 MHz. A Multi-Mode Clock Module, MMCM, were used to derive the two clocks from a fixed rate oscillator operating at 200 MHz located at the KC705 evaluation board. LogiCORE [39] were used to generate a MMCM IP-block with the desired clock outputs, and a global reset signal output.

A subset of the demodulator chain output signals were routed to output pins at the KC705 evaluation board through a test interface block. Routing the signals to the output ensured that the demodulator chains were kept when the design were optimized. Both the inputs and outputs were located in one corner at the FPGA, which causes the signal path to be suboptimal. The KC705 output were connected to the HP 1670E logic analyzer through a FMC XM105 Debug Card [40].

5

Results

Throughout the project we have managed to implement and evaluate a range of filters for a specific demodulator chain, with can be seen in Fig. 3.1. The target technology for the project has been the Kintex 7 FPGA from Xilinx with the system clock frequency set to 500 MHz. For each type of filter we made several implementations of the system with increasing number of demodulator chains to be able to calculate the the power consumption per chain. We measured the power consumption when the filters were processing data, and when the system were reset and the outcome of the measurements are shown in Appx. B. Linear regression gave an good approximation of the power consumption in each demodulator chain, and the overhead power consumption caused by other parts of the system. The outcomes of the measurements are summarized in Tab. 5.1.

One should be aware that the power consumption for implemented FIR filter with decimation rate 24 then 2 is estimated based on the results we achieved from the other two direct implemented FIR filters. As two demodulator chains did not fit into the FPGA we subtracted the mean overhead power consumption of the two other direct implemented FIR filters from the one measurement we were able to make on the FIR 24-2 implementation. In addition some of the implementations did not meet the timing requirements but did fit into the FPGA. The results of the measurements on those implementations are printed in italic text in the tables in Appx. B and Appx. C.

Compared to the weighted estimations presented in Tab. 3.1, and Tab. 3.2 the measured power consumption, presented in Tab. 5.1, differs at some points, particularly for the FIR filters. Simulations in Sec. 3 gave an efficiency estimation based on the number of multiplications and the decimation rate. Among the direct implementations FIR 24-2 would consume most power, FIR 2-24 second most, and FIR 2-4-6 would consume least power. The corresponding order among the polyphase implementations were FIR 24-2, FIR 2-4-6, and FIR 2-24. For the direct implementations the order were correct in terms of DSP slices, but not in terms of power consumption; FIR 2-24 were using more

power than FIR 24-2 which can be explained when taking the number of LUTs and Flipflops into acount. Regarding the polyphase implementation we found that the order was reversed when comparing the simulations and power measurements. The unforeseen outcome can be explained with the optimisations done by LogiCORE. Less hardware could be used as each DSP slice could be reused with reloaded coefficients when the decimation rate were high, which we did not account for in the simulations. Finally, the CIC-FIR filter measurements were in line with the simulations; CIC-FIR 5-21 consumed least power, followed by CIC-FIR 6-23, and CIC-FIR 6-25.

Relevant data from the place and route process can be found in Appx. C for all the filter implementations. However the results that could be calculated from these values are presented in Tab. 5.1 below in the form of hardware resources per demodulator chain. Some major differences can be seen compared to the estimated hardware utilization in Tab. 3.1 and Tab. 3.2 from the simulations. The differences are results of that Vivado optimizes for area and therefore pushes as many filter taps into a single DSP block as possible by shifting in the coefficients to the same multiplier. The numbers for CIC-FIR 6-23 and 6-25 might seem odd since the smaller filter use more hardware but is more energy efficient, this is a result of that that the 6-23 filter uses more LUTs but of smaller size. The number of RAM blocks is consistent for all filters since filter coefficients are stored locally in LUTs and the only lager LUT in the design is the sine lookup table for the demodulator.

Maximum number of chains for each filter type was calculated solely on the required number of DSP-block for the implementation since this was the constraining hardware resource for the implementation. The Maximum number of filter chains is a theoretical calculated number and to achieve something even close to this we would have to implement the designs in an FPGA mounted on a chip routed for the projects purpose, so that strait signal path through the FPGA and not centred around one corner as in this case.

Table 5.1: Summary of the hardware utilization, and power consumption per demodulator chain in the different implementations.

				HW resources				Max	Power
		Decim.	Coeffs.	DSP	LUT	FF	RAM	Chains	[W]
Direct	FIR	2-24	13-366	396	5517	35178	1	2	**4.42**
		24-2	247-31	502	1484	18505	1	1	***3.40***
		2-4-6	13-34-92	90	1670	7544	1	9	**0.72**
Polyphase	FIR	2-24	13-366	34	2262	3288	1	24	**0.61**
		24-2	247-31	22	1301	2562	1	38	**0.39**
		2-4-6	13-34-92	36	1788	3779	1	23	**0.57**
CIC -	FIR	24-2	5-21	20	674	1611	1	41	**0.23**
		24-2	6-23	18	672	2071	1	46	**0.24**
		24-2	6-25	18	666	2071	1	46	**0.25**

6

Conclusions

The conclusions that can be drawn from our work is that huge amounts of power and hardware can be saved by implementing more efficient filters in the demodulator chains of radar receivers when the number of chains increases. With more efficient filters that are capable of sustaining high sample rates the receivers can be moved closer to the antenna and less signal processing has to be done in the analog domain.

A conclusion that can be drawn from the implementation results, summarized in Tab. 5.1, is that it is wise to have a high decimation rate in the initial filter when designing multi rate filters paths using LogiCORE, if the goal is to save hardware and power. At least if the entire filter chain is running on the input sampling frequency since this allow LogiCORE to locate multiple filter taps in the same DSP block by shifting in coefficients. This reduction in hardware make it possible to fit more demodulator chains into a single FPGA enabling the possibility to have even more parallel channels than what the hardware growth predicted in Moore's law enables.

There is still a lot to investigate in the area to further improve on the designs, a natural fist step to continue the project wold be to connect an ADC and measure the power consumption with a live signal to see if these result differs from the artificial signal. To take this another step towards the future an ADC with GTX interface could be used to allow for higher sample rates by using the possibility of parallel filters for processing the samples from one channel, and to optimize the data path. Another possibility to reduce the power consumption is to investigate the new power optimization options offered in Vivado. Another optimization to explore is to put the sub filters in different clock domains to investigate if the reduced frequency compared to the increased capacitance pays of on the power consumption.

Over and out!

A

Project Specification

Datum	Utgåva	Dokumentidentitet
2013-12-20		

Utfärdad av	Informationsklass företagssekretess
OEDEF Claes Claesson	EJ KLASSAD
Informationsklass exportkontroll	**Informationsklass försvarssekretess**
NOT EXPORT CONTROLLED	EJ FÖRSVARSSEKRETESS

Examensarbete
Energieffektiv digital signalbehandling

1 Bakgrund

I dag implementeras en allt större del av signalgenererings- och mottagarsystemen digitalt i FPGAer. Samplingstakterna, dataflödena och antalet mottagarkanaler ökar i systemen vilket gör att energiförbrukningen och behovet av kylning av de digitala delarna också ökar.

2 Mål

Målet med examensarbetet är att demonstrera någon eller några metoder som, jämfört med en standardimplementation, ger reducerad energiförbrukning i ett mångkanaligt digitalt implementerat mottagarsystem.

3 Ramar

Förslag: Den signalbehandlingskedja som skall studeras är en klassisk nerblandningsenhet med IQ-uppdelning. Kedjan innehåller lokaloscillator, mixer, lågpassfiltrer och decimering av samplingstakt.

Arbetet skall utföras för FPGAer från Xilinx, t ex Xilinx Kintex-7. Programspråk vid implementation är VHDL.

Datum Utgåva Dokumentidentitet
2013-12-20
Utfärdad av
Claes Claesson
Informationsklass exportkontroll
NOT EXPORT CONTROLLED
Informationsklass företagssekretess
EJ KLASSAD
Informationsklass försvarssekretess
EJ FÖRSVARSSEKRETESS

4 Genomförande

Arbetet inleds med ett ***startmöte*** där studenter, examinator och handledare från Saab och Chalmers träffas för avstämning.

Därefter följer ***litteraturstudier***. Studenterna läser in sig på området och tar reda på vad som gjorts tidigare. (3 veckor)

Efter studiefasen genomförs en ***brainstorming*** där eventuellt nya idéer till energireduceringsmetoder kläcks. (2 veckor)

Reduceringsmetoderna ställs mot varandra i en ***teoretisk utvärdering***. Någon eller några av de metoder som verkar mest lovande väljs ut för det fortsatta arbetet. (5 veckor)

Efter cirka sju veckor görs en ***tredjedelsavstämning*** med examinator och handledare där inriktningen på det fortsatta arbetet bestäms.

Efter cirka tio veckor görs en ***halvtidspresentation*** på företaget för att informera medarbetare om arbetet. På så sätt kan erfarenheter från andra medarbetare fångas in. Presentationen är också ett bra tillfälle att få återmatning på arbetet.

Signalbehandlingskedjan ***implementeras*** i FPGA i varianter med de reduceringsmetoder som valts. (5 veckor)

Energireduktionerna från den teoretiska utvärderingen ***verifieras*** genom mätningar i lab. (2 veckor)

I den avslutande fasen görs en reflektion där ***slutsatser*** dras. (2 veckor)

Exjobbsrapporten skrivs med fördel löpande under arbetets gång.

Arbetet avslutas med en väl förberedd ***muntlig presentation***. (1 vecka)

5 Handledare

Handledare på Saab AB är Claes Claesson M.Sc. EE, CTH 97.

False

B

Power Consumption

B.1 Runtime Power Consumption

Table B.1: Runtime power consumption of filter implementations. Implementations marked with X did not fit into the FPGA, and numbers in italic did not meet timing requirements.

				Demodulator Chains					
		Decim.	Coeffs.	1	2	5	9	Per Chain	Overhead
Direct	FIR	2-24	13-366	4.98	*9.70*	X	X	4.42	0.86
		24-2	247-31	*4.25*	X	X	X	3.40	0.85
		2-4-6	13-34-92	1.03	1.74	*5.13*	*6.95*	0.72	0.84
Polyphase	FIR	2-24	13-366	1.00	1.71	3.39	*5.91*	0.61	0.42
		24-2	247-31	0.69	1.04	2.25	3.80	0.39	0.28
		2-4-6	13-34-92	0.85	1.44	3.13	*5.45*	0.57	0.28
CIC -	FIR	5-21	5-21	0.53	0.72	*1.40*	2.32	0.23	0.29
		6-23	6-23	0.55	0.83	1.52	2.52	0.24	0.32
		6-25	6-25	0.55	0.80	1.49	2.55	0.25	0.29

B.2 Reset Power Consumption

Table B.2: Reset power consumption of filter implementations. Implementations marked with X did not fit into the FPGA, and numbers in italic did not meet timing requirements.

				Demodulator Chains				
		Decim	Coeffs.	1	2	5	9	Overhead
Direct	FIR	2-24	13-366	1.35	*2.37*	X	X	0.33
		24-2	247-31	*1.13*	X	X	X	0.28
		2-4-6	13-34-92	0.52	0.80	*1.58*	*2.73*	0.24
Polyphase	FIR	2-24	13-366	0.42	0.62	1.04	*1.66*	0.29
		24-2	247-31	0.38	0.47	0.85	1.30	0.25
		2-4-6	13-34-92	0.41	0.61	1.05	*1.71*	0.26
CIC -	FIR	5-21	5-21	0.41	0.42	*0.69*	1.08	0.28
		6-23	6-23	0.35	0.49	0.74	1.10	0.28
		6-25	6-25	0.35	0.44	0.72	1.13	0.25

C

Hardware Utilization

C.1 DSP48E1 blocks

Table C.1: DSP block utilization for the filter implementations. Implementations marked with X did not fit into the FPGA, and numbers in italic did not meet timing requirements.

				Demodulator Chains			
		Decim.	Coeffs.	1	2	5	9
Direct	FIR	2-24	13-366	398	*794*	X	X
		24-2	247-31	*504*	X	X	X
		2-4-6	13-34-92	92	182	*452*	*812*
Polyphase	FIR	2-24	13-366	36	70	112	*200*
		24-2	247-31	24	46	112	200
		2-4-6	13-34-92	38	74	182	*326*
CIC -	FIR	5-21	5-21	22	42	*102*	182
		6-23	6-23	20	38	92	164
		6-25	6-25	20	38	92	164

C.2 Flip-Flops

Table C.2: Flip-Flop utilization for the filter implementations. Implementations marked with X did not fit into the FPGA, and numbers in italic did not meet timing requirements.

				Demodulator Chains			
		Decim.	Coeffs.	1	2	5	9
Direct	FIR	2-24	13-366	36187	*71365*	X	X
		24-2	247-31	*19537*	X	X	X
		2-4-6	13-34-92	8597	16185	*38700*	*68978*
Polyphase	FIR	2-24	13-366	4354	7789	17450	*30737*
		24-2	247-31	3504	6089	13720	24023
		2-4-6	13-34-92	4720	8521	19800	*34967*
CIC -	FIR	5-21	5-21	2534	4149	*8870*	15437
		6-23	6-23	2992	5065	11280	19559
		6-25	6-25	2992	5065	11280	19559

C.3 LUT

Table C.3: LUT utilization for the filter implementations. Implementations marked with X did not fit into the FPGA, and numbers in italic did not meet timing requirements.

				Demodulator Chains			
		Decim.	Coeffs.	1	2	5	9
Direct	FIR	2-24	13-366	5864	*11381*	X	X
		24-2	247-31	*1848*	X	X	X
		2-4-6	13-34-92	2068	3709	*8713*	*15418*
Polyphase	FIR	2-24	13-366	2807	5188	11800	*20967*
		24-2	247-31	1700	2971	6873	12097
		2-4-6	13-34-92	2189	3944	9298	*16481*
CIC -	FIR	5-21	5-21	1071	1714	*3733*	6454
		6-23	6-23	1069	1710	3724	6436
		6-25	6-25	1063	1697	3692	6382

C.4 Block RAM

Table C.4: Block RAM utilization for the filter implementations. Implementations marked with X did not fit into the FPGA, and numbers in italic did not meet timing requirements.

				Demodulator Chains			
		Decim.	Coeffs.	1	2	5	9
Direct	FIR	2-24	13-366	3	*4*	X	X
		24-2	247-31	*3*	X	X	X
		2-4-6	13-34-92	3	4	*7*	*11*
Polyphase	FIR	2-24	13-366	3	4	7	*11*
		24-2	247-31	3	4	7	11
		2-4-6	13-34-92	3	4	7	*11*
CIC -	FIR	5-21	5-21	3	4	*7*	11
		6-23	6-23	3	4	7	11
		6-25	6-25	3	4	7	11

Bibliography

[1] (2013) Japanese War Tuba. [Online]. Available: http://en.wikipedia.org/wiki/War_tuba

[2] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, pp. 114–117, Apr. 1965.

[3] H. Hassan and M. Anis, *Low-Power Design of Nanometer FPGAs*, 1st ed. Burlington, MA: Morgan Kaufmann, 2009.

[4] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed., M. Hirsch, Ed. Boston, MA: Addison-Wesley Publishing Company, 2010.

[5] M. Kleja. (2014, May) Saab avslöjar nya radarprodukter. Talentum Media AB. Stockholm, Sweden. [Online]. Available: http://www.nyteknik.se/nyheter/it_telekom/radio/article3826110.ece

[6] O. Adrian, "From AESA radar to digital radar for surface-based applications," in *IEEE Radar Conference*, Pasadena, CA, May 2009, pp. 1–5.

[7] M. A. Richards, *Fundamentals of Radar Signal Processing.* New York, NY: McGraw-Hill, 2005.

[8] M. I. Skolnik, *Introduction to Radar Systems*, 3rd ed. New York, NY: McGraw-Hill, 2001.

[9] J. C. Toomay and P. J. Hannen, *Radar Principles for the Non-specialist.* Raleigh, NC: Scitech Publishing Inc., 2004.

[10] I. Faulconbridge, *Radar Fundamentals.* Canberra, Australia: Argos Press, 2002.

[11] S. Kemkemian, A. Larroque, and C. Enderli, "The industrial challenges of airborne AESA radars," in *IET International Radar Conference*, Xi'an, China, Apr. 2013, pp. 1–5.

[12] H. van Bezouwen, H.-P. Feldle, and W. Holpp, "Status and trends in AESA-based radar," in *IEEE MTT-S International Microwave Symposium Digest (MTT), 2010*, Anaheim, CA, May 2010, pp. 526–529.

[13] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Jul. 1948.

[14] J. B. Anderson, *Digital Transmission Engineering*, 2nd ed., S. V. Kartalopoulos, Ed. Piscataway, NJ: Wiley, 2005.

[15] B. Mulgrew, P. Grant, and J. Thompson, *Digital Signal Processing.* New York, NY: Palgrave Macmillan, 2003.

[16] S. Winder, *Analog and Digital Filter Design*, 2nd ed., S. Winder, Ed. Woburn: Newnes, 2002.

[17] M. P. Donadio. (2000, Jul.) CIC filter introduction. Iowegian International. Overland Park, KS. [Online]. Available: www.dspguru.com/sites/dspguru/files/cic.pdf

[18] E. B. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, pp. 155–162, Apr. 1981.

[19] M. Li, S. Zhang, and X. Diao, "Compensation method for the CIC filter in digital down converter," in *International Conference on Electrical and Control Engineering (ICECE), 2011*, Yichang, Sep. 2011, pp. 1024–1027.

[20] C. Candan, "Optimal sharpening of CIC filters and an efficient implementation through Saramäki-Ritoniemi decimation filter structure (extended version)," unpublished.

[21] P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[22] H.-K. Yang and W. M. Snelgrove, "High speed polyphase CIC decimation filters," in *IEEE International Symposium on Circuits and Systems. (ISCCAS), 1996*, Atlanta, GA, May 1996, pp. 229–232.

[23] T. D. Burd and R. W. Brodersen, "Energy efficient CMOS microprocessor design," in *Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences*, Wailea, HI, Jan. 1995, pp. 288–297.

[24] L.-Y. Loy, W. Zhang, Z.-H. Kong, W.-L. Goh, and K.-S. Yeo, "Body-bootstrapped-buffer circuit for CMOS static power reduction," in *IEEE Asia Pacific Conference on Circuits and Systems. (APCCAS), 2008*, Macao, Nov. 2008, pp. 842–845.

[25] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, 3rd ed., S. W. Director, Ed. New York, NY: McGraw-Hill, 2009.

[26] Matlab r2013b. [Online]. Available: http://www.mathworks.se/products/matlab/

[27] Matlab r2013b dsp system toolbox. [Online]. Available: http://www.mathworks.se/products/signal/

[28] *KC705 Evaluation Board for the Kintex-7 FPGA*, Xilinx, 2013.

[29] *ADS540x and ADS54T0x Evaluation Module (EVM)*, Texas Instruments, 2012.

[30] *USB Interface Adapter Evaluation Module*, Texas Instruments, 2006.

[31] *HP 1660E/ES/EP and 1670E Series Logic Analyzers*, Hewlett-Packard, 1998.

[32] *7 Series FPGAs Overview*, Xilinx, 2014.

[33] *7 Series DSP48E1 Slice*, Xilinx, 2013.

[34] *7 Series FPGAs GTX/GTH Transceivers*, Xilinx, 2014.

[35] D. Healy. (2014, May) Understanding Linear Feedback Shift Registers – The Easy Way. University of California Santa Cruz. Santa Cruz, CA. [Online]. Available: http://www.yikes.com/~ptolemy/lfsr_web/index.htm

[36] M. George and P. Alfke. (2007, Apr.) Linear Feedback Shift Registers in Virtex Devices. Xilinx. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp210.pdf

[37] *LogiCORE IP CIC Compiler v4.0*, Xilinx, 2014.

[38] *LogiCORE IP FIR Compiler v7.1*, Xilinx, 2014.

[39] *LogiCORE IP Clocking Wizard v5.1*, Xilinx, 2014.

[40] *FMC XM105 Debug Card User Guide*, Xilinx, 2011.