An FPGA-Based Architecture for Digital Filtering at Intermediate Frequency Using Undersampling

Patricio Cordeiro, Adalbery R. Castro, Marcel S. Cabral, Lilian C. Freitas and Aldebaro Klautau
Sensors and Embedded Systems Laboratory (LASSE) - Universidade Federal do Para (UFPA)
CP 8619 - Rua Augusto Correa 1 - CEP 66075-110 - Belem - Para - Brazil
Web: www.lasse.ufpa.br - E-mails: {patricio, adalbery, marcelsc, liliancf, aldebaro}@ufpa.br

***Abstract*—Technologies such as cognitive radio have been moving the analog to digital (A/D) conversion closer to the antenna and increasing the sampling rate of the digital signal processing modules. Most A/D chips are not capable of operating at the radio frequency (e.g. at GHz) but they can operate at intermediate frequencies of approximately 100 MHz. This work investigates an architecture based on *undersampling* that uses field programmable gate array (FPGA) to filter signals originally located at an IF above the Nyquist frequency. It is presented a case study of digital filtering for mobile telephony that includes, for example, 78-th order, 14 Bit word length FIR filter implemented in the Altera FPGA Cyclone III.**

1. INTRODUCTION

Wireless systems are very important in modern communication systems. Radio technology is rapidly evolving to cope with new requirements. Software and cognitive radio are good examples of new frameworks [1]. Both share the intention of digitizing the signal as soon as possible, ideally just after the antenna. However, there are several difficulties to achieve this goal, especially because the radio frequencies (RF) can be of the order of GHz.

This paper describes some preliminary results of the first author's master dissertation, which focuses on using *undersampling* and digital filtering to process signals modulated at intermediate frequencies, such as 70 MHz. Undersampling (also called bandpass sampling) is a signal processing technique that can be used when the signal has a bandpass spectrum and allows operation below the Nyquist frequency [2], [3]. Roughly speaking, undersampling uses aliasing in a controlled manner, and allows significant savings in computational cost. For example, the naive application of the sampling theorem [4] to a signal with 2 MHz of bandwidth and centered (modulated) at 100 MHz would use a sampling rate f_s higher than 2×101 MHz while undersampling allows to use $f_s = 4.04$ MHz.

The architecture investigated in this work uses undersampling to reduce the computational cost of digital filtering at IF. The goal is to be able of operating at an IF of approximately 100 MHz, which is a challenge due to the high computational cost. Both finite impulse response (FIR) and infinite impulse response (IIR) filters [5] are of interest.

Digital filters have been most commonly implemented using digital signal processor (DSP) chips. DSPs are microprocessors especialized in performing digital signal processing [6] and can be programmed in the C language. The fixed hardware architecture of a DSP can be a bottleneck for the implementation of systems that require customized and parallel operations. In the last years, field programmable gate arrays (FPGAs) have become an attractive solution for implementing digital signal processing systems [7]. FPGA devices consist of logic elements (LEs) and memory that can be configured to operate in different modes. FPGAs provide a reconfigurable architecture programed via hardware that can achieve a high throughput. This flexibility is the main advantage of FPGA's over the DSP's [8]. On the other hand, it can be more difficult to map applications on FPGA than DSP. For example, when a hardware description language (HDL) is required to program the FPGA, the task is typically considered more difficult than programming a DSP on C. To reduce this disadvantage, FPGA vendors have recently developed development tools such as Altera's DSP builder to targeting FPGA via a software design flow that does not require learning HDL.

Digital filtering at IF can benefit from the parallelism and high sampling rates that a FPGA can provide. Many previous work addressed the implementation of digital filters on FPGAs [9]–[14]. One aspect of implementing digital filters on FPGA is the trade off between area and speed. There are many techniques suggested in literature to reduce the hardware needed to implement digital filters. For example, in [10], [11], [14] a bit-serial structure replaces the bit-parallel structure to reduce area. In contrast, techniques to improve filter performance by using parallel structure that provides higher rates are suggested in [12], [13].

This paper presents the design and implementation of filters at IF. The filters target implementation on an Altera FPGA Cyclone II with a fully parallel structure to provide higher rates. The goal of the work is to evaluate the adoption of undersampling and study the corresponding trade offs. To narrow the scope, the application is digital filtering for digital cellular repeaters.

This paper is organized as follows. Section II presents a brief explanation about undersampling. Section III describes the operation of digital repeaters. Section IV presents the implementation of undersampling with digital filtering on FPGA and discusses the obtained results, while Section V presents the conclusions.

2. USING UNDERSAMPLING AT IF

The sampling theorem [5] states that a signal must be sampled at a rate greater than twice its maximum frequency to ensure non ambiguous information. Periodically sampling an analog signal with spectrum $H(f)$ at a sampling rate f_s produces a discrete-time signal with spectrum $H(e^{j\omega})$. It can be observed that $e^{j\omega} = e^{j(\omega+2\pi)}$. Therefore, for all discrete time signals, $H(e^{j\omega})$ is periodic in 2π. When the discrete-time signal is converted back to continuous time (the process is called reconstruction [5]), the periodicity of $H(e^{j\omega})$ manifests as the repetition of a spectrum $\hat{H}(f)$ with period f_s. When $H(f)$ is low pass, most of the times the goal is to avoid aliasing [5] and have the possibility of recovering $H(f)$ after filtering $\hat{H}(f)$.

When a band pass signal has bandwidth B, to prevent overlapping between aliased images, the signal must be sampled at a rate greater than $2B$ and its spectrum should not cross integer multiples of $f_s/2$. In many applications, this technique greatly reduces the demand required for the A/D converter.

Note that the aliasing effect can be used in favor of the digital processing module, since from an adequate choice of the A/D converter frequency, the designer can choose the position of the band of interest, therefore executing a frequency shift of the signal in the frequency domain. This can be done when the images created by aliasing are identical copies of the original signal, which means that the designer can choose any of them as the signal of interest, keeping in mind that these images should not overlap. One concern that the designer should have in mind relates to the spectral inversion, but it can be easily removed via software.

The valid frequency intervals to sample a continuous signal centered at f_c and with bandwidth of B is given by

$$\frac{2f_c - B}{m} \geq f_s \geq \frac{2f_c + B}{m+1}, \tag{1}$$

where m is an arbitrary and positive integer that represents the interval's number [3].

3. CASE STUDY: DIGITAL REPEATERS FOR CELLULAR SYSTEMS

This section focuses on digital repeaters, which are used as the application to investigate digital filtering at IF. Repeaters are equipments commonly used in cellular networks in areas where the signal propagation is poor or in areas where traffic is low and there is no demand for the installation of a base station.

Some repeaters use the so called heterodynes, which are responsible for translating the RF signals to an IF. Later, the undesired bands are filtered out. A repeater should not amplify bands different from those predetermined. In most it is not possible to know the band of interest before hand or pre-establish it. If that was the case, it would be enough, conceptually, to simply create one or more bandpass filters and amplify the channels or bands in RF frequency.

In practice, the heterodyne allows the band center frequency of the intermediate frequency (IF) to be adjustable, by changing the frequency of the local oscillator, simplifying the bandpass filtering, typically performed with a SAW (surface acoustic wave) filter [15], [16].

A basic scheme of a digital repeater is shown in Figure 1. A repeater uses at least two antennas: one connected to a donor base station, the donor antenna, and another to retransmit the signal or receive the cell phones signals in the repeater's coverage area, the server antenna. As the channels are repeated without changing its frequencies, to prevent instability due to a poor isolation between the antennas, the repeaters usually have a supervisory system that automatically reduces the gain when a poor isolation is detected.

The input signal received by the donor antenna is delivered to a duplexer, responsible for separating the transmission and reception powers belonging to the downlink and uplink paths. A LNA (low noise amplifier) receives the signal and sends it to a variable attenuator used to adjust the total repeater's gain. After that, the resulting signal is passed through a heterodyne module that performs the translation from RF to the IF band. For that, it uses a carrier generated by a local oscillator, normally a NCO (numerically-controlled oscillator), controlled via a PLL (phase locked loop) circuit.

In the architecture that is investigated in this work, the IF analog signal is digitalized by a digital module that amplifies and filters the signal to eliminate the undesired images from modulation. Subsequently, the signal is resampled in order to increase the valid signal range, preparing it to D/A conversion.

Afterwards the digital-analog conversion, using the same carrier of the previous modulator, one another is applied in order to translate the signal to the original RF frequency. Finally, the signal is amplified again to achieve the desired power, limited in band by a SAW filter and retransmitted to the area of poor coverage

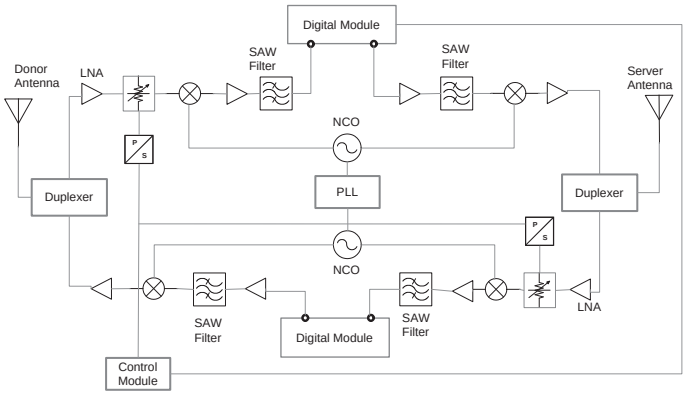
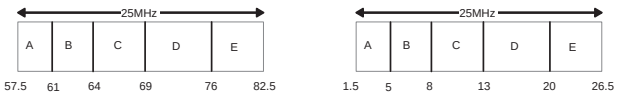


Fig. 1. Block diagram of a digital repeater.

4. DIGITAL FILTERING WITH FPGA

This section discusses the design and implementation of a digital FIR filter for cellular repeaters. As an example, it was considered a GSM signal with bandwidth of 25 MHz, centered



(a) (b)

Fig. 2. Example of five bands of an GSM signal(a) Signal in FI. (b) Signal after undersampling

at an IF of 70 MHz and separated into five frequency bands (Figure 2(a)). Our interest here, is to repeat the band B and D, which imply in filtering out the bands A, C and E. The filter design specifications is shown in Table I.

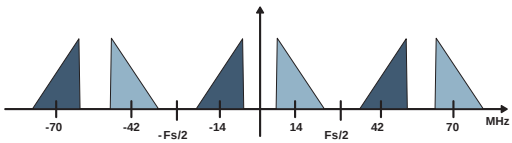


Fig. 3. Signal spectrum after sampling at 56 MHz.

The sampling frequency was obtained using the Eq. 1. The valid values for the sampling frequency considering a IF signal centered at 70 MHz with bandwidth of 25 MHz, (for $m = 0, 1, 2$, respectively) are:

1) $fs \geq 165$ MHz without spectral inversion
2) 82.5MHz $\geq fs \geq$ 115 MHz with spectral inversion
3) 55 MHz $\geq fs \geq$ 57.5 MHz without spectral inversion

Among the valid sampling frequencies, we chose to use a value of 56 MHz, because it has no spectral inversion and it offers a 3 MHz of space between the images, which facilitates the process of filtering. The resulting sampled signal is shown in Figure 3. We are interested in the band of the signal centered at 14 MHz, which extends from 1.5 MHz to 26.5 MHz. The new frequency bands are shown in Figure 2(b).

Because of the proximity of the bands to be filtered, the transitions bands of the filter were chosen to be as small as possible.

TABLE I
DESIGN REQUIREMENT OF FIR FILTER

Sampling Frequency (Fs)	56 M
Transition Band Width	1.5 M
Passband ripple	1 dB
Stopband attenuation	60 dB
Response Type	Multiband

The filter design and its implementation on FPGA were performed using Altera's DSP Builder tool. The DSP Builder tool is a system development that allows the designer to implement DSP functions without deep knowledge of HDL. DSP Builder offers a direct interface between MATLAB's Simulink and the FPGA hardware, allowing the designer to simulate and validate the design in a hardware level. DSP builder also offers many IP Megacores functions to implement DSP systems, such as FIR Compiler Megafunction [17]. The FIR Compiler Megafunction, is a fully integrated FIR filter development environment optimized for use with Altera FPGA devices.

The FIR Compiler Tool supports many types of filters architecture, including the fully parallel distributed arithmetic. More especifically, the FIR compiler Megafunction provides two options to obtaining the filter coefficients:

1) Via the FIR compiler built coefficient generator, which can design low, high, bandpass and other types of FIR filters by the windowing method.
2) By loading the coefficients from a .txt file created previously, for example, by MATLAB and then imported into the FIR compiler.

The coefficient generator of the FIR compiler does not offer a method for designing multiband filters. For that reason, the coefficients were obtained using the MATLAB Filter Design and Analysis Tool (FDATool) [18]. Table II shows the parameters used in FDATool to calculate the filter coefficients.

TABLE II
FDA TOOLS PARAMETERS FOR FIR FILTER DESIGN

Response Type	Multiband
Design Method	FIR: Generalized Equiripple
Density factor	20
Phase	Linear
FIR Type	Type 1: even order, symmetric

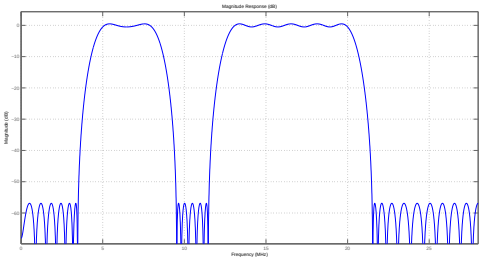


Fig. 4. Multiband FIR filter frequency response

Figure 4 shows the filter frequency response. The calculated coefficients of the implemented filter were saved to a .txt file. The file was then imported into the FIR compiler. Table III shows the specifications of the coefficients , the FPGA family and the structure used to implement the filter in the FIR Compiler Tool.

TABLE III
FIR COMPILER TOOLS PARAMETERS FOR FIR FILTER IMPLEMENTATION.

Input/Output Number System	14 bits Signed Binary
Fpga Device Family	Cyclone III
Structure	Distributed Arithmetic: Fully Parallel Filter

Considering a signal at 70 Mhz with bandwidth of 25 MHz and undersampled at a rate of 56 MHz, the image of interest is located at 14 MHz. To convert the signal to the IF again, upsampling is performed via the insertion of zeros among the samples and filtering. Then, the signal centered at 14 MHz

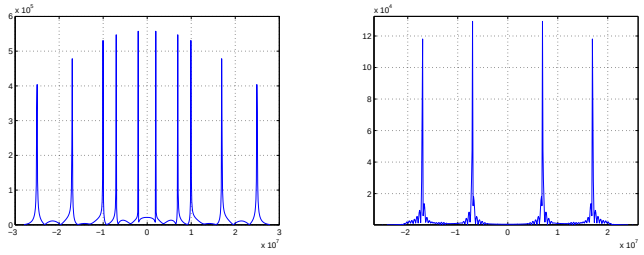
will be resampled in four times its current sampling rate, i.e., $4 \times fs \Rightarrow 4\times$ 56MHz, in order to generate a new signal with fs equal to 224 MHz. Thus, the desired image signal, around 70 MHz, will be located in the valid range of signal (from $-fs/2$ to $fs/2$). Lastly, the resampled signal is sent directly to the D/A converter and, subsequently, it must pass through of a analog filter, e.g. SAW filter, centered at 70 MHz located after the digital module, this filter will be responsible for the signal reconstruction and deleting undesired images.

The filter was implemented in an Altera DSP Development Kit, Cyclone III Edition. To test the filter, we used the Altera Megacore NCO to generate five sine waves of frequencies, 2, 7, 10, 17 and 25 MHz. The five signals were then summed and filtered. Using Altera Modelsim, the filter delay was estimated as 1.21 μs (taking into account the delay of the block NCO) and, the filter outputs one sample at every clock cycle (fully parallel filter).

To simulate the filter functionality on FPGA (a real world test), the generated signal was sent to one of the D/A converters of the kit and then, the resulted analog signal, was sent back to one of the A/D converters of the kit. The signal was then passed through the filter, the resulted signal was then resampled ($fs = 224$) and sent to another D/A converter. The resulted signal and its FFT were monitored on an oscilloscope (Figure 5). Due to upsampling, the resampled signal was shifted four times in its original frequency and presents the mentioned images (Figure 6(a)) that need to be filtered out after the DA conversion. Table IV shows the FPGA resources used to implement the filter.

TABLE IV
FPGA RESOURCES USAGE.

Logic Cells	7375
Memory Bits	7168
LUT-Only LCs	586
Register-Only LCs	1270
LUT/Register LCs	5519

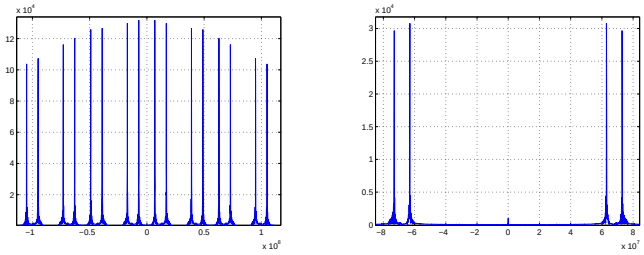


(a) (b)

Fig. 5. (a) FFT of FIR filter input signal. (b) FFT of filtered signal.

5. CONCLUSIONS

Software radio and related technologies are motivating the research on digital signal processing at the IF. This work investigated an architecture based on *undersampling* that uses FPGA to filter signals originally located at an IF above the Nyquist frequency. The obtained results indicate that very flexible and powerful filters can be implemented on a FPGA such as Altera's Cyclone III.



(a) (b)

Fig. 6. (a) FFT of upsampled signal. (b) FFT of the recovered signal.

ACKNOWLEDGMENTS

This work was partially supported by Brasilsat and CNPq.

REFERENCES

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13–18, 1999.

[2] Richard G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall PTR, 1997.

[3] Walt Kester (Editor), *Mixed-signal and DSP Design Techniques (Analog Devices)*, Newnes, 1 edition, 2003.

[4] J. Barry, E. Lee, and D. Messerschmitt, *Digital Communication*, Kluwer, 3rd edition, 2004.

[5] A. Oppenheim and R. Schafer, *Discrete-time signal processing*, Prentice-Hall, 1989.

[6] Berkeley Design Technology., "Choosing a DSP Processor," Tech. Rep., BDTI, 1996-2000.

[7] Altera Corp., "FPGAs Provide Reconfigurable DSP Solutions," Tech. Rep., Altera Corporation, 2001.

[8] Berkeley Design Technology., "The Evolving Role of FPGAs in DSP Applications," Tech. Rep., BDTI, 2007.

[9] Satish Mohanakrishnan Chi-Jui Chou and Joseph Evans, "FPGA Implementation of Digital Filters," *International Conference on Signal Processing Applications and Technology*, pp. 251–255, 1993.

[10] Alejandro Grosso Monica Arroyuelo, Jorge Arroyuelo, "FPGA-Based Digital Filters Using Bit-Serial Arithmetic," Tech. Rep., Departamento de Informatica Universidad Nacional de San Luis Republica Argentina, 2007.

[11] Javier Valls Marcos, Marcos Martnez Peir, Trini Sansaloni, and Eduardo Boemo, "Design and FPGA implementation of Digit-Serial FIR filters," *IEEE International Conference on Electronics, Circuits and Systems*, pp. 191–194, 1998.

[12] Joseph B. Evans, "An efficient fir filter architecture," *IEEE International Symposion on Circuits and Systems*, pp. 627–630, 1993.

[13] Ryan Kastner Shahnam Mirzaei, Anup Hosangadi, "FPGA implementation of high speed FIR filters using add and shift method," *International Conference of Computed Design (ICCD '06)*, pp. 1–4, 2006.

[14] Yong Ching Lim, Joseph B. Evans, and Bede Liu, "An Efficient Bit-Serial FIR Filter Architecture," Tech. Rep., 1993.

[15] D.Sc. Colin K. Campbell Ph.D., "Understanding Surface Acoustic Wave (SAW) Devices for Mobile and Wireless Applications and Design Technique," 2004.

[16] Fujitsu, "SAW Filter," Tech. Rep., Fujitsu Laboratories Ltd., 2008.

[17] Altera Corp, "FIR Compiler MegaCore Function," 2010, www.altera.com.

[18] The Mathworks, " Filter Design and Analysis Tool," 2010, http://www.mathworks.com.