High-Performance Flexible All-Digital Quadrature Up and Down Converter Chip

R. Paško, L. Rijnders, P. Schaumont, S. Vernalde[1] and D. Ďuračková[2]

[1] IMEC v.z.w. Leuven, Belgium
[2] Faculty of Electrical Engineering and Information Technology, STU Bratislava, Slovakia

ABSTRACT

In this paper, the design of an all-digital quadrature up and down converter with high accuracy and flexible IF settings is presented. The signal up/downconversion is achieved by interpolation/decimation combined with a programmable anti-alias filter preserving the selected frequency band during the sample rate conversion. This way a high-speed solution with low-power consumption is achieved. We used a novel technique to implement flexible IF settings. The resulting structure is capable of handling signals up to 160 MSPS and is suitable for coaxial access network modem applications.

I. INTRODUCTION

Thanks to the technology progress, digital techniques are today capable of handling also IF/RF tasks. Thus the boundary between digital and analogue parts in the current communication applications is becoming a design parameter. The traditional approach of baseband functionality handled by a digital core and analogue IF/RF front-end is not necessarily also the most efficient one. However, this does not imply, that the digital techniques have to replace the analogue circuitry. Rather the advantages, which are offered by digital technologies, should be carefully considered during the design process. For example, analogue hardware is typically sensitive to all kinds of nonidealities and nonlinearities. This might have a serious impact especially if a complex modulation scheme is required. The use of digital techniques eliminates most of these undesired effects. On the other hand, the use of digital hardware has also its limitations (e.g. availability of fast AD converters), so it can not be considered a general solution.

In this paper, we discuss a full-digital implementation of a high-speed quadrature up and down converter. This area was already investigated by several authors [1-3]. However, all these solutions sacrificed flexibility in order to achieve the desired performance, i.e. they all consider a single IF band. We will show, that the trade-off between the programmability and the performance does not have to be so abrupt and that it is possible to keep significant flexibility while simultaneously achieving the required performance figures.

The paper is structured as follows: in the next section, a brief analysis of the potential digital downconversion methods will be given and potential the advantages and weaknesses of those approaches will be outlined. Afterwards, the system requirements will be described in section 3, followed by the chip architecture overview in section 4. The important implementation issues will be discussed in sections 5 to 7, an overview of chip parameters and simulation results in section 8 and finally conclusions will be drawn in section 9.

II. DIGITAL DOWNCONVERSION

The digital downconversion can be realised in a number of ways. The selection of any of those has a profound impact on the design parameters of the final implementation.

On one hand, programmable implementations (see [4]), centered around a digital mixer and sine generator, as shown in the top part of Fig 1, offer high degree of flexibility in IF settings. The frequency to downconvert can be easily modified by reprogramming the sine generator without having direct impact on the implementation of the rest of the system. The disadvantage is the high amount of hardware working at the highest frequency, which severely limits the working frequency of the application (e.g. the down converter described in [4] is capable to process 75 MSPS and the upconverter 52 MSPS).

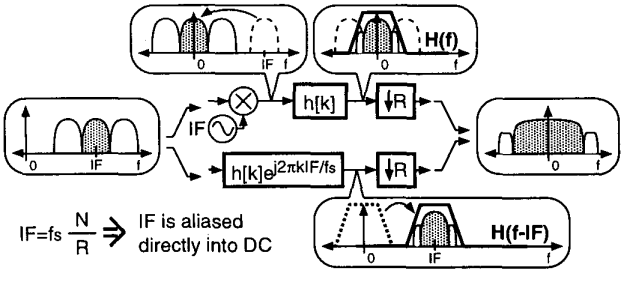


Fig. 1. Different ways of obtaining downconverted signal by using mixing or proper anti-alias filtering.

On the other hand, narrowly focused implementations [1-3] tuned to downconversion of a single IF can be mentioned. These techniques are based on a well-known decimation property, which says that it is not possible to determine the original position of a signal in frequency spectrum after decimation took place (e.g. after decimation by 2, it is not possible to conclude, whether the signal was centered around 0 or $f_s/2$). So

0-7803-5809-0/00/$10.00 © 2000 IEEE IEEE 2000 CUSTOM INTEGRATED CIRCUITS CONFERENCE

instead of downconverting the signal to the baseband before decimation, the anti-alias filter is upconverted at the signal position which has the same final effect as shown in the bottom part of Fig. 1. This makes it an interesting choice from speed (250 MSPS QAM modulator/demodulator in [1]), as well as from a power consumption (see [2-3]), point of view. However, since the actual IF to downconvert is transformed into the modified filter coefficients, any change of IF is inherently more difficult to implement.

Our design shows, that it is possible to overcome these problems and achieve a substantial amount of flexibility in decimation based downconversion while retaining high speed. This is done by combining a novel decimation based approach with a smart implementation strategy.

III. System Description

The design specs originate from Multimedia Cable Network System (MCNS) modem specifications (see [5]) and the intention was to cover the complete signal spectrum for the upstream communication path. The overall functionality of the complete system is shown in Fig. 2. During signal down-

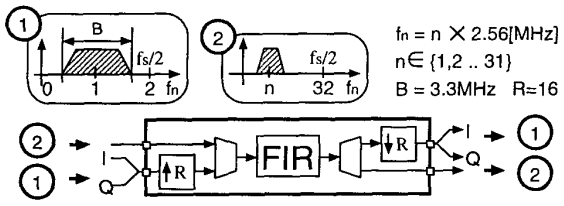


Fig. 2. Basic system functionality

conversion, a 3.3 MHz band, placed at one out of 31 possible positions, is downconverted into low IF. It is simultaneously decimated by 16 and placed at $IF = 2.56$ MHz. The upconversion works in exactly reverse way, i.e. a low IF 3.3 MHz signal is interpolated by factor 16 and upconverted at one of 31 available IF positions. Having the discrete set of IFs, rather than an arbitrary IF, is a direct result of the implemented decimation based downconversion method.

In order to relax the anti-alias filter requirements, the total downconversion by 16 was split into two successive downconversions by 4. The signal path through the filters is shown in Fig. 3. The $H2_s$ filter contains four shapes each preserving a group of four IF bands. In case, a channel starting from $n = 17$ is selected, it is placed into the proper position before filtering by means of a frequency shift. The same applies for the $H1_s$ filter, which is also integrating four shapes each preserving one IF band and using frequency shifts when necessary. Finally, $H0$ filter prevents aliasing between I and Q components. For example, let us consider a channel with $n = 25$. In the first step, it is shifted by $\pi/2$ shift at the position of channel 9, so the shape $H2_2$ is selected during the first decimation by 4. In the second stage, the signal is shifted by π shift from $n = 9$ into $n = 1$ position so the shape $H1_1$ is

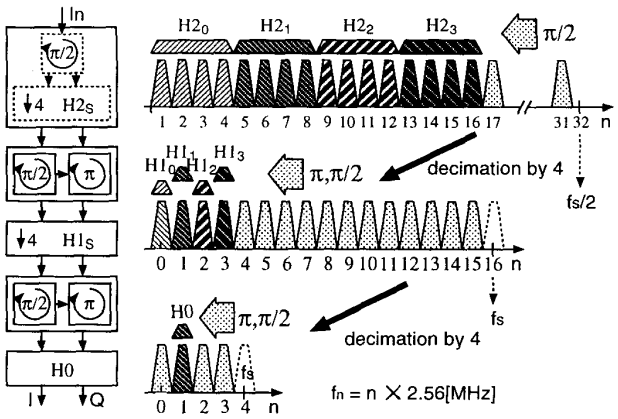


Fig. 3. Signal path through anti-alias filters

selected dering the second desimation by 4. This will leave the downconverted channel exactly at $IF = f_s/4$ ready for decoding after processign by the last filter *H0*.

The actual system complexity reflects the trade-off between flexibility (i.e. the number of possible IFs) and the implementation cost of the multishape filters. The chip can work in either of both modes (upconversion and downconversion), but not in both simultaneously. The mode is selected by the *up/down* control signal.

IV. System Architecture

The complete system architecture is shown in Fig. 4. The filters *H2* and *H1* are decimating/interpolating filters (depending on the mode), the *H0* is a single rate filter. The filters *H2* and *H1* are implemented using polyphase decomposition to lower the working frequency of the filters (*H2* is working at 40 MHz and *H1* at 10 MHz). The rotors by π and $\pi/2$ are used to perform frequency shifts of the signal spectrum in order to simplify the anti-alias filter implementation as mentioned in the previous section. Finally, the muxes are necessary in the *up/down* mode selection. In order to obtain an efficient solution different design optimisations have to be carried out. Now some of these will be descussed more in detail.

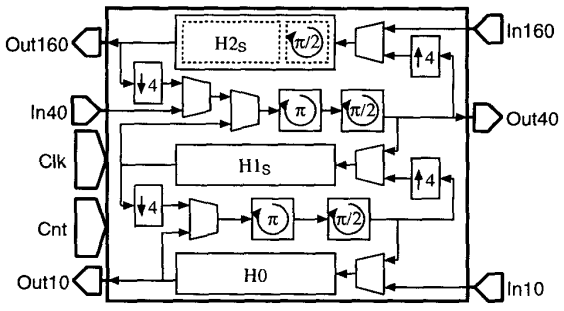


Fig. 4. The complete system architecture

V. FIR Filter Design

The filters are critical components of the whole design in terms of area as well as power consumption. In order to relax the requirements imposed on the filters, an approach called *Interpolated Finite Impulse Filter* is used [3]. The idea is, that in multiple stages, it is not necessary to filter out a signal, which has been already removed by the previous stage. This way, orders of the anti-alias filters were significantly reduced. The different filter shapes have been obtained from a low-pass prototype through a complex rotation what corresponds to a frequency shift or upconversion of the filter passband to IF.

In order to reduce the working frequency of anti-alias filters, polyphase decomposition was used in case of filters *H2* and *H1* as shown in Fig. 5. The polyphase implementation of

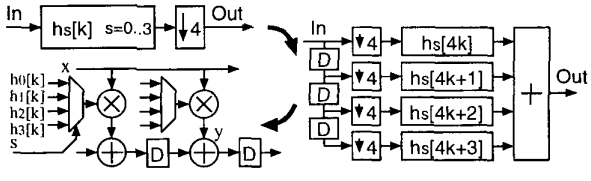


Fig. 5. Implementation of multishape anti-alias FIR filter *H2* and *H1*

FIR filters is described in detail in [1] and [3] so it is sufficient to note here that these filters were split up in four parallel subfilters working at four times lower frequency. The crucial part of FIR filters implementation was an effective realization of multiple coefficients corresponding to the different filter shapes. We have introduced a modified Canonic Signed Digit (CSD) technique to tackle this problem (for more details about standard CSD technique see [6]). The multiplications are decomposed into a set of add-shift operations as it is with the CSD coefficients. Since the different coefficients basically result in different shift factors, it is possible to implement multiple coefficients by making the shifts programmable instead of hardwired. This way appropriate shift factors can be chosen during operation. When necessary, adders or subtractors are replaced by add/sub cells (see Fig. 6).

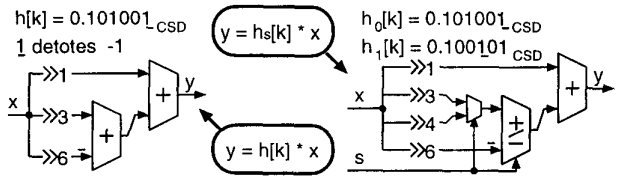


Fig. 6. Comparision of normal and modified CSD technique

Due to the nonlinear distribution of the CSD numbers, scaling of the CSD coefficients by a constant might result in a smaller number of total non-zero bits (see [6]). The proper scaling ratio is determined by an exhaustive search. This strategy was used also in our design. Since the *H2* and *H1* filters integrate four different shapes, resulting in four CSD sets for each filter, the optimisation was run for each CSD set separately. Afterwards, the scaling ratio, which gave the best average improvement for all four shapes, was selected.

The reduction of the necessary filter shapes was possible thanks to the fact, that certain operations on the signal spectrum can be performed at a little hardware cost. The frequency shifts by π and $\pi/2$ require only the multiplications by 1, -1 and 0, i.e. *sin/cos* of $k\pi/2$, and some additional multiplexing of *real* and *imaginary* parts. This allows to do all the processing only on $< 0, f_s/4 >$ part of frequency spectrum, since any frequency from $< f_s/4, f_s >$ interval can be shifted into $< 0, f_s/4 >$ using some combination of π and $\pi/2$ shifts. Because of that, the actual number of shapes was reduced to one half respecively one fourth in *H2* and *H1* filters.

VI. FIR Filter Architecture Optimisation

The anti-alias filters are the core of the design, so their implementation must be handled with utmost care. Thanks to some design considerations, the filter architecture can be further optimised. Since the optimisations are filter specific, they will be discussed separately.

The *H2* filter has a unique property that in upconversion mode, only real part has to be produced, while during downconversion only real part is being read (see Fig. 7). Provided

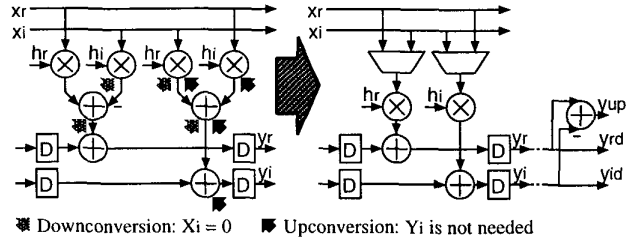


Fig. 7. Filter *H2* optimisation (one tap)

multiplexers are introduced to select the real or imaginary input component for the multipliers, it is possible to significantly reduce the hardware of the *H2* filter as shown in Fig. 7. As a side-effect, the $\pi/2$ rotor can be also easily merged into the filter structure thanks to the introduced multiplexers. This also eliminates virtually all hardware, working at the 160 MHz, apart from the polyphase decomposition structures.

In filter *H1* design, the filter frequency shift factors are multiples of $\pi/16$. This has important consequences.

1. Filter $H1_0$ remains real only since the rotation factors are always $4k\pi/4$.
2. Filters $H1_1$ and $H1_3$ remain symmetrical provided the original 23-tap filter was symmetrical as well (which is the case), so only half of the taps is needed to be implemented.
3. Filter $H1_2$ is not real only, but provided some additional multiplexing is introduced, only two instead of four multipliers are necessary per tap. This reduction is possible because the rotated coefficients are either *real only* or *imaginary only* or $h_r = h_i$.

VII. Polyphase Decomposition

The polyphase expansion elements are the only hardware which has to work on maximal frequency. We implemented both downconversion (Fig. 8) as well as upconversion parts using multiplexers. This simple structure fulfilled the hard speed requirements.

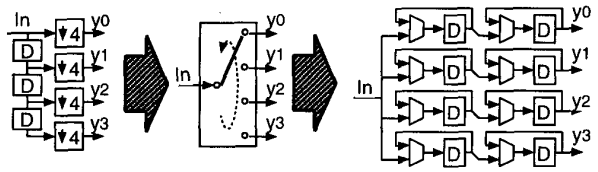
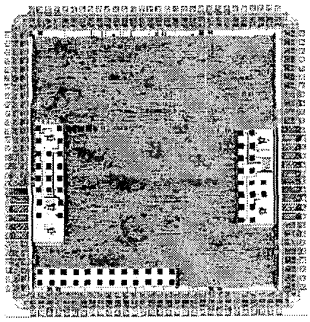


Fig. 8. Polyphase Expansion Elements

VIII. Chip Characteristics Overview

The design was made using multirate version of OCAPI C++ design environment [7]. The C++ description has 4400 lines which results in 12000 lines of generated RT-level VHDL code, which was synthesized using Synopsys design compiler. It was successfully tape-out, manufactured and functionally tested. The chip parameters are given in Fig. 9. The three distinctive areas at the both sides and at the bottom



Technology	Alcatel-Mietec CMOS 0.5m/3.3V
Master Clock	163.84 MHz
Internal Clocks	40.94, 20.48 and 10.24 MHz
IO Word	12 bits
Internal Word	14 bits
Gate Count	86k
Core Area	24.22 mm^2
Chip Area	38.4 mm^2
Package	120 pin PGA

Fig. 9. RoBo4 chip parameters

are containing extra analogue circuitry for measurements of substrate noise of different clock regions.

The three FIR filters characteristics, as well as the composite one are shown in Fig. 10. The shapes in Fig. 10 are $H2_0$, $H1_0$ and H0 shifted into DC. The composite characteristic is similar for every channel, i.e. the stopband attinuation $SB < -45dB$ and the passband ripples $PR < 0.1dB$.

The quantization was tested by cosimulation of the chip with QAM transmitter and receiver. We have evaluated the Error Vector Magnitude (EVM) of the received QAM-16 signal after upconversion or downconversion performed by RoBo4. The resulting EVM values from clock-true floating point and fixed point simulations is given in Fig. 11.

IX. Conclusion

An implementation of a high-performance all-digital quadrature up and down converter was presented. It features

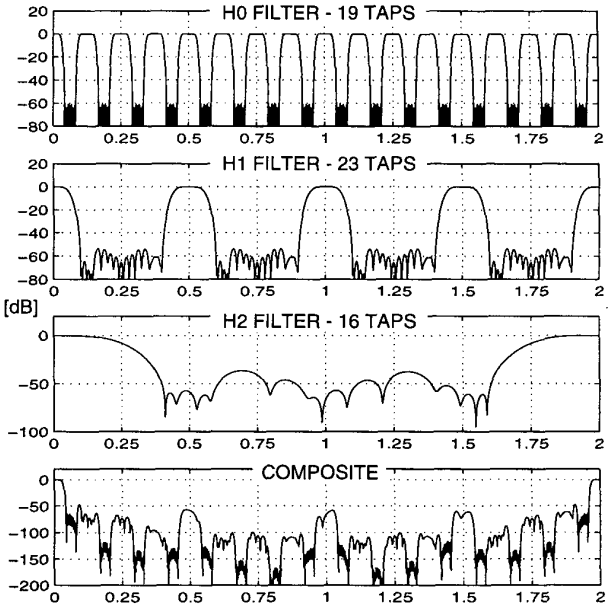


Fig. 10. FIR filter characteristics ($f_s = 2$)

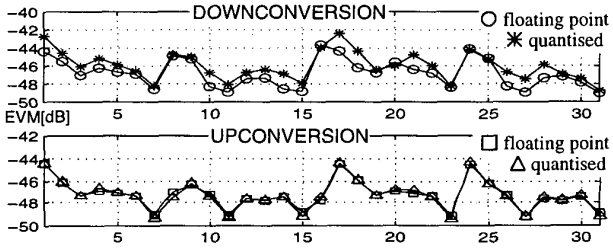


Fig. 11. Simulated EVM values with respect to the selected channel

a novel approach introducing flexibility in IF selection without sacrifying performance or cost. It also shows, that all-digital up and downconversion is feasible into relatively high frequencies and with high performance.

References

[1] B. C. Wong and H. Samueli, *A 200-MHz All-Digital QAM Modulator and Demodulator in 1.2-μm CMOS for Digital Radio Applications*, IEEE Journal of Solid-State Circuits, Vol. 26, No. 12, December 1991

[2] P. Schaumont, S. Vernalde, M. Engels and I. Bolsens, *Low Power Digital Frequency Conversion Architectures*, Kluwer Journal VLSI Signal Processing, Vol. 18, 1998

[3] S. Jou, S. Wu and CH. Wang, *Low-Power Multirate Architecture for IF Digital Frequency Down Converter*, IEEE Transactions on Circuits and Systems II, Vol. 45, No. 11, November 1998

[4] *HSP 50016 Digital Down Converter, HSP 50215 Digital UpConverter*, Harris Semiconductor Data Sheet January 1999

[5] *MCNS/DOCSIS specs*, http://www.cablemodem.com

[6] H. Samueli, *An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Power-of-two Coefficients*, IEEE Transactions on Circuits and Systems, Vol. 36, July 1989

[7] *OCAPI Design Environment*, http://www.imec.be/ocapi