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

Robert Paško, Luc Rijnders, Patrick R. Schaumont, *Member, IEEE,* Serge A. Vernalde, *Member, IEEE,* and Daniela Ďuračková

***Abstract*—In this paper, the design of an all-digital quadrature up and down converter with high accuracy and flexible intermediate frequency (IF) settings is presented. The signal up or down conversion is achieved by interpolation and decimation combined with a programmable anti-alias filter to preserve the selected frequency band during the sample rate conversion. This way a high-speed solution with low power consumption is obtained. A novel technique, based on the use of canonic signed digit (CSD) code, was utilized to implement the programmable anti-alias filter structure. The resulting chip fabricated in a 0.5-μm CMOS process is capable of handling sample rates up to 160 megasamples per second (MSPS) and is suitable for coaxial access network modem applications.**

***Index Terms*—Multiplierless FIR filters, multirate architecture, quadrature amplitude modulation, signal sampling, tunable digital filters.**

I. Introduction

In RECENT years, there is an obvious trend in communication technologies to shift from analog toward digital techniques. *Software radio* [1] can serve as a term encapsulating these efforts. The goal is to replace as much as possible the analog parts by digital ones to improve the performance. Unfortunately, the digital processing techniques have not been able to match the speed of analog intermediate frequency (IF), not to mention radio frequency (RF), implementations in the past. Thus, the software radio concept has been mostly limited to baseband processing.

However, thanks to the technology progress, digital techniques are today capable of handling IF, or even low RF tasks. This makes the extension of digital processing from baseband to the IF domain possible. Consequently, the boundary between the digital and the analog parts is becoming a design parameter itself. For example, analog 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 provided sufficient accuracy can be used. Another strong argument for digital technology is the possibility to scale down the supply voltage for low-power applications. Finally, the programmability offered by digital techniques provides flexibility which is especially important in the context of rapidly evolving standards. Unfortunately, the use of digital hardware also has its limitations, like the need for fast analog-to-digital converters (ADCs). Because of that, the advantages offered by the digital techniques should be carefully considered during the design process, rather than simply replacing analog circuitry by its digital equivalents. In any case, it is definitely worth to investigate the digital circuitry, implementing IF functionality, with either all-digital or mixed analog–digital solutions in mind.

In this paper, we propose a full-digital implementation of a high-speed quadrature up and down converter. It extends programmable digital processing to the IF range while at the same time keeping an eye on power consumption and implementation complexity issues. Current work in this field is either limiting the IF flexibility or else providing it at the cost of significant amount of hardware working at high frequencies. This usually restricts the sample rate and adds to the power consumption. We will show that the tradeoff 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 good performance and power consumption figures.

The paper is structured as follows: in Section II, a brief analysis of available digital downconversion implementations will be given and the potential advantages and weaknesses of those approaches will be analyzed. Afterwards, the system requirements and overall functionality are described in Section III, followed by an overview of important design issues in Section IV. Next, simulation and measurement results are given in Section V. Section VI proposes the future work and the paper is concluded in Section VII.

Finally, it should be emphasized that the designed chip serves as up and down converter. For conciseness reasons, however, only the down conversion is discussed whenever the same conclusions would apply for up conversion as well.

II. Digital Down Conversion and Related Work

Digital down conversion can be realized in two different ways. Choosing one of them has important consequences on the parameters of the final implementation, such as speed, power consumption, and IF settings flexibility.

One downconversion implementation is centered around a flexible frequency shifting element which moves the band of interest from IF to the baseband, as shown in Fig. 1. There are various options possible to implement this kind of functionality.

Manuscript received July 17, 2000; revised October 19, 2000. This work was supported by COPERNICUS project SISPAS CP94-0223. The work of R. Paško was supported by an international scholarship from the Katholieke Universiteit Leuven, Belgium.

R. Paško, L. Rijnders, P. R. Schaumont, and S. A. Vernalde are with the Inter-University Microelectronics Centre (IMEC), B3001 Leuven, Belgium.

D. Ďuračková is with the Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, 81219 Bratislava, Slovakia.

Publisher Item Identifier S 0018-9200(01)01436-6.

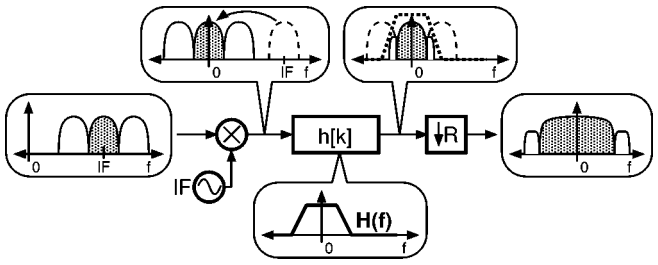


Fig. 1. Down conversion using mixer to shift the frequency band and decimator to reduce the sampling rate.

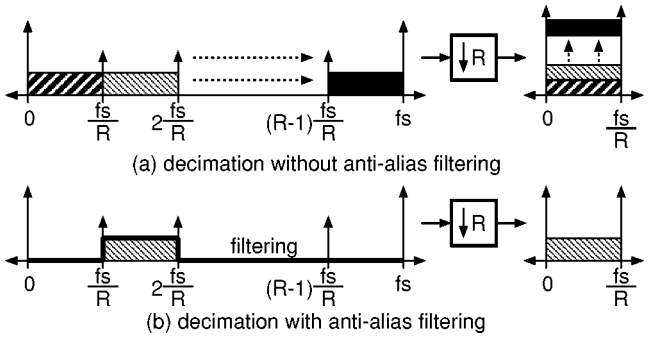


Fig. 2. Effect of decimation, with and without anti-alias filtering, on a frequency spectrum.

One can use a normal mixer, i.e., multiplier, as in [2], [3] and [4], or resort to an alternative use of a coordinate rotation digital computer (CORDIC) algorithm [5]. Also, the implementation of the sine-wave generator offers various options such as a ROM-based look-up table [4] or a digital frequency synthesizer [6]. These approaches offer a high degree of flexibility in IF settings, since the IF reprogrammability can be achieved without having direct impact on the implementation of the rest of the system. Disadvantageous is the high amount of hardware working at the highest frequency, which either limits the sample rate of the application or adds significantly to the power consumption. Also, often nonstandard and full-custom design techniques are required for the high-frequency (HF) circuitry.

An alternative downconversion technique is based on a well-known decimation property demonstrated in Fig. 2. It states that after decimation, multiple frequency bands are mirrored into the same frequency region, as shown in Fig. 2(a). By removing the unwanted bands through an anti-alias filter, it is possible to achieve an effect equivalent to the down conversion [see Fig. 2(b)]. It is also obvious that the frequencies which are an integral multiple of f_s/R are decimated directly into baseband. This can be formally stated as follows.

Observation 1 (Down Conversion): Provided IF is an integral multiple of f_s/R, where R denotes the decimation ratio, the down conversion to baseband can be performed purely by decimation with a proper anti-alias filter. ■

A similar conclusion is given for up conversion in [7]:

Observation 2 (Up Conversion): An up conversion to a frequency that is an integral multiple of the input sampling rate avoids the need of additional frequency shifts of the input sequence. ■

This reasoning leads to an alternative downconversion scheme where, instead of down converting the signal to baseband before decimation, the anti-alias filter is up converted at the signal position. This has the same final effect, as shown in Fig. 3. The technique provides an interesting choice from the speed [8] as well as the power consumption [7], [9] point of view. However, since the actual IF to down convert is transformed into the modified filter coefficients, any change of IF is inherently more difficult to implement.

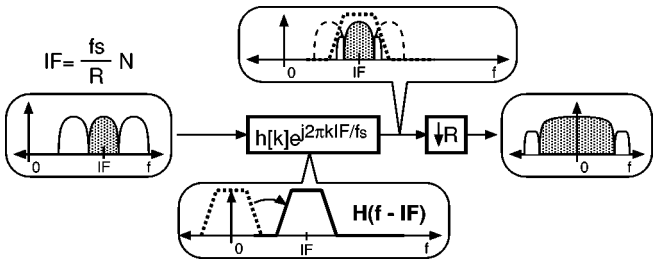


Fig. 3. Down conversion using upconverted anti-alias filter protecting the required frequency band during decimation.

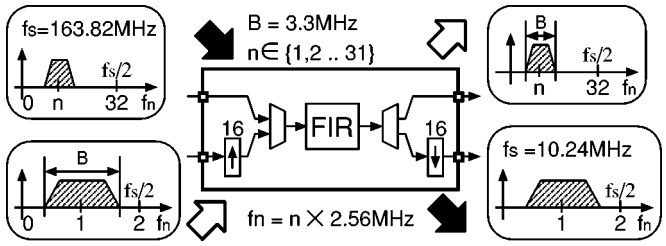


Fig. 4. Aimed overall system functionality.

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

III. System Description and Architecture

The design specs originate from the Multimedia Cable Network System (MCNS) modem specifications [10]. 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. 4. During the down-link path, a 3.3-MHz band, placed at one out of 31 discrete positions, is down converted into low IF. It is simultaneously decimated by 16 and placed at $\mathrm{IF} = 2.56$ MHz. The up conversion operates in the reverse way, i.e., a low IF signal is interpolated by 16 and up converted at one of 31 available IF positions. For these IFs, there is no additional frequency shifting required, as was shown in *Observation 1* and *2*. Thus, having a discrete set of IFs rather than an arbitrary IF is a direct result of the implemented decimation-based downconversion method. Our solution for introducing IF flexibility is to integrate multiple shapes into the anti-alias filters. This is accomplished in such a way that there exists a path to baseband protected by some anti-alias filter shape for every channel. Thus the actual system complexity also reflects the tradeoffs between IF flexibility and the implementation cost of the multishape filters.

The basic system architecture is shown in Fig. 5. In order to relax the anti-alias filter requirements, the total down conver-

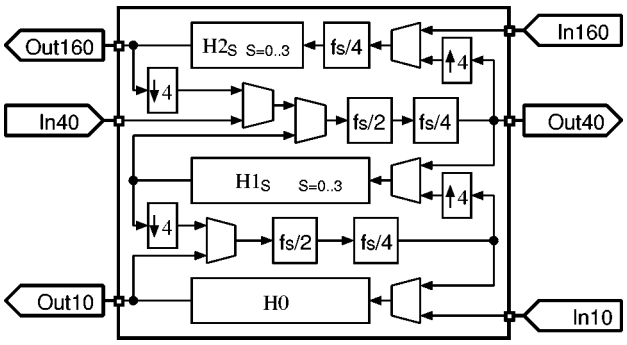


Fig. 5. Basic system architecture.

sion by 16 was split into two successive down conversions by 4. The H2 and H1 are the anti-alias filters during decimation/interpolation, and both are integrating multiple shapes which are selected to protect the chosen channel. The H0 is a single rate filter preventing aliasing between I and Q components. All three filters are complex, processing I as the *real* and Q as the *imaginary* component. The frequency shifters by $f_s/2$ and $f_s/4$ are performing a shifting of the complete signal spectrum by the given factors. This allows to restrict all the processing only on $\langle 0, f_s/4\rangle$ part of the frequency spectrum, since any frequency from the $\langle f_s/4, f_s\rangle$ interval can be shifted into $\langle 0, f_s/4\rangle$ using some combination of $f_s/2$ and $f_s/4$ shifts. As a consequence, the actual number of shapes could be reduced to four in both H2 and H1 filters. Finally, the multiplexers are necessary in the up/down conversion mode selection, since the chip can work in either mode, but not in both simultaneously. The splitting of the anti-alias filters also allows to extend the system functionality by introducing the additional inputs and outputs at the 40-MHz sample rate, as shown in Fig. 5.

The signal path through the filters is shown in Fig. 6. The H2$_s$ filter contains four shapes, each preserving a group of four channels. In case a channel with $n > 16$ is selected, it is placed into the proper position for filtering by means of the first $f_s/4$ frequency shift, as shown in Fig. 6(a). After the anti-alias filtering by one of the H2$_s$ shapes [Fig. 6(b)], the first decimation by four takes place [Fig. 6(c)]. Again, if the channel position $n > 3$, it is shifted into the first quadrant by combination of $f_s/2$ and $f_s/4$ shifts [Fig. 6(d)]. Afterwards, the proper H1$_s$ shape is selected [Fig. 6(e)] to prevent the aliasing during the second decimation by four [Fig. 6(f)]. Finally, the processed channel is brought into $f_s/4$ position again by frequency shifting [Fig. 6(g)] and potential remnants of other channels at $3f_s/4$ are filtered out by the H0 filter, as shown in Fig. 6(h). This will leave the downconverted channel at $\mathrm{IF} = f_s/4$ ready for further baseband processing.

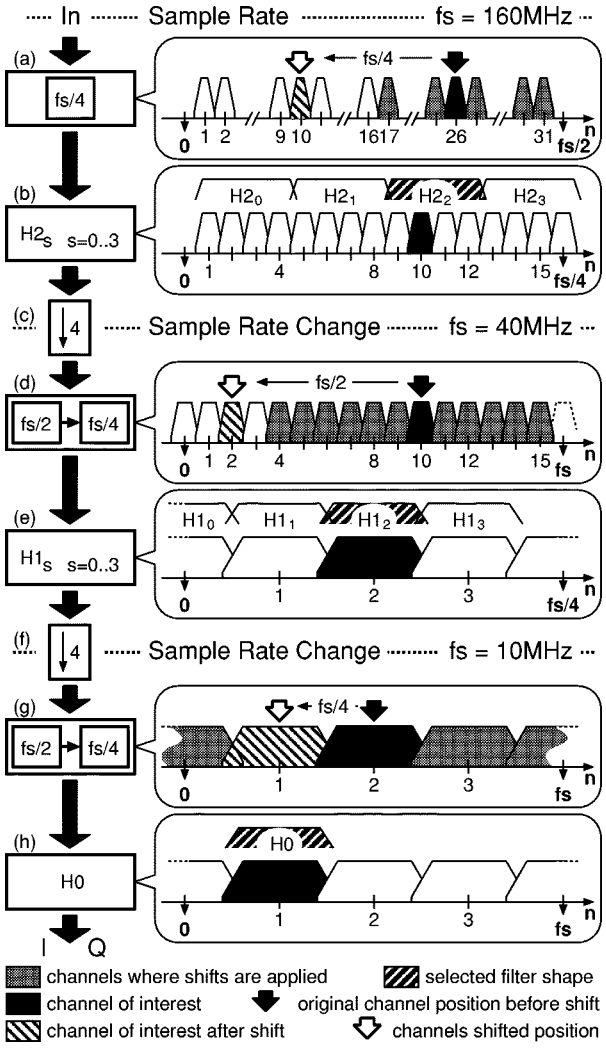


Fig. 6. Down conversion of channel $n = 26$.

C. *Filter Optimization*
D. *Frequency Spectrum Manipulation*
E. *Polyphase Expansion Elements*
F. *Clocking Scheme, Control and Reset*
G. *Chip Design Summary*

IV. Chip Design Issues

In order to obtain an efficient solution, some novel and some well-known techniques have been combined together with different design optimizations. The following design issues will be discussed in more detail in Section IV-A–Section IV-G.

A. *Programmable Anti-Alias Filter Design*
B. *Multishape FIR Filter Implementation*

A. Programmable Anti-Alias Filter Design

The whole filter system is very similar to a filter bank. The basic difference is that in our case only one frequency band is processed at any time. As a consequence, it is not necessary to have a separate bank for every frequency band, but the hardware can be shared. In order to lower the working frequency of the anti-alias filters, we have used the polyphase decomposition technique well known from filter bank theory. It takes advantage of the fact that some samples are either discarded during decimation or equal to zero during interpolation, so it is not necessary to calculate them in any case. This allows to split the anti-alias filters into subfilters working on a lower frequency as shown in Fig. 7(a). Since the filter requirements for down conversion and up conversion are equivalent, the same filter structure can be shared for both. The merging of up and down conversion structures is shown in Fig. 7(b).

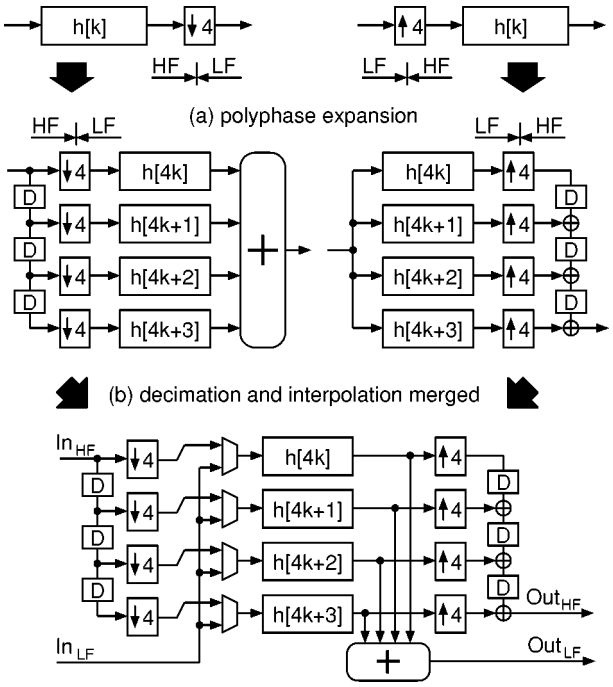


Fig. 7. Polyphase decomposition and sharing of decimation and interpolation hardware.

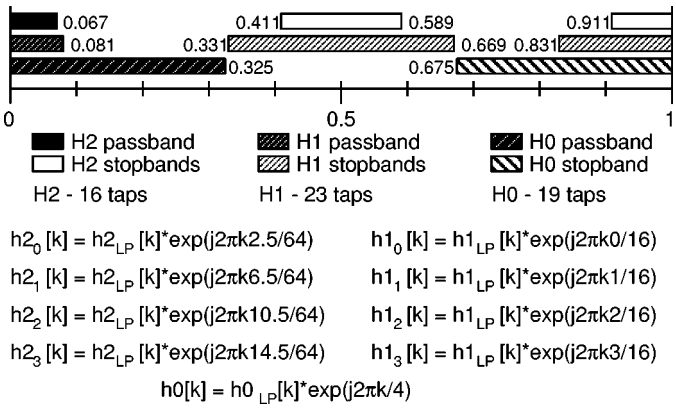


Fig. 8. Parameters of the low-pass filter prototypes ($f_s = 2$) and the resulting rotated characteristics.

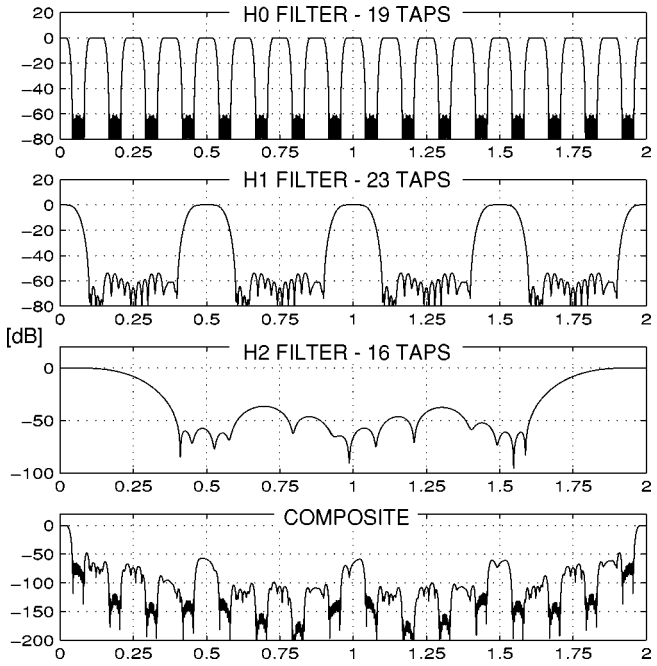


Fig. 9. Individual FIR filter characteristics ($f_s = 2$) and the composite one for interpolation case.

Despite of the polyphase implementation, the anti-alias banks 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 anti-alias filters, an approach called *interpolated finite impulse filter* [9] can be used. It takes advantage of the fact that during processing of multiple stages, it is not necessary to filter out the parts of the spectrum which have been already removed by the previous stage, or which are not mirrored into the band of interest. This way, the order of the anti-alias filters could be significantly reduced. The number of taps was determined by the requirement to have a stopband attenuation $\mathrm{SB} < -45$ dB and passband ripples $\mathrm{PR} < 0.1$ dB for the composite filter characteristic. The filter parameters and the resulting numbers of taps are given in Fig. 8. According to these parameters, filter coefficients of the low-pass filter prototypes were obtained by the Parks–McClellan algorithm [11]. The introduced *don't care* regions are the result of the interpolated finite impulse filter technique. The actual filter shapes have been obtained from the low-pass prototypes through a complex rotation which corresponds to a frequency up-shift of the filter passband to the desired frequency. The shift factors are given in Fig. 8 as well. As an example, the individual characteristics of filters $\mathrm{H2}_0$, $\mathrm{H1}_0$, and H0, as well as the resulting composite characteristic, all shifted back into baseband, are shown in Fig. 9.

B. Multishape FIR Filter Implementation

The crucial problem of the individual finite-impulse-response (FIR) filter implementations was an effective realization of multiple coefficients corresponding to the different filter shapes. The traditional implementation method for normal FIR filters is the canonic signed digit (CSD) technique. The CSD encoding [12] is a redundant number format, using -1, often denoted as $\underline{1}$, in addition to 0 and 1, where no two nonzero digits are adjacent. This results in a representation with a minimal number of nonzero digits. The basic idea of the CSD method is to replace the multipliers by a series of add-shift structures, as depicted in Fig. 10(a). Thanks to the above-mentioned property of CSD encoding, the number of adder or subtracter cells is guaranteed to be minimal. The use of CSD coefficients in FIR filter design is discussed in detail in [13].

To tackle the implementation of multishape FIR filters, we have introduced a modified CSD technique. The multiplications are decomposed into a set of add-shift operations as with the normal CSD method. Since the different coefficients, corresponding to the different shapes, result basically in various shift factors, it is possible to implement multiple coefficients by making the shifts programmable instead of hardwired, as shown in Fig. 10(b). This way appropriate shift factors can be chosen during operation. If the sign has changed as well, an

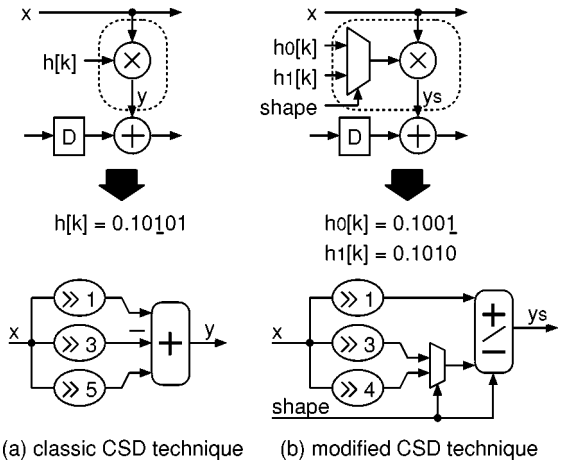


Fig. 10. Comparison of normal and modified CSD technique in single and multishape FIR filter implementation.

adder or subtracter cell must replaced by an adder/subtracter element, as shown in Fig. 10(b).

C. Filter Optimization

We used two different optimization strategies during the filter design. The first one was aimed at improving the filter characteristics (e.g., in terms of stopband attenuation) by scaling. The second one was an optimization of filter hardware, which was possible thanks to some specific architecture-related issues.

Due to the nonlinear distribution of CSD numbers, scaling of the CSD coefficients by a constant might result in better filter characteristics, as was demonstrated in [13]. The best scaling ratio can be determined by an exhaustive search. This means to gradually multiply the filter coefficients by one octave of scaling factors k and select the scaling ratio minimizing the cost function given in our case by

$$J(k) = \max[\delta_p(k)/W, \delta_s(k)]/b(k) \quad k = 0.5 \ldots 1 \qquad (1)$$

where $\delta_p(k)$ and $\delta_s(k)$ are the passband and stopband ripple amplitudes of the characteristics scaled by k, $b(k)$ is the average passband gain, and W is the ripple weighting factor, i.e., the same cost function as proposed in [13]. This strategy was used to optimize the implemented filters. Since the H2 and H1 filters integrate four different shapes, resulting in four CSD sets per filter, the optimization was run for each CSD set separately. Afterwards, the scaling ratio which gave the best average improvement for all four shapes was selected.

Thanks to some architecture-related considerations, additional optimizations of filter hardware could be performed. Since these optimizations are filter specific, they will be discussed separately.

The H2 filter has a unique property that in upconversion mode, only the *real* part has to be produced, while during down conversion only the *real* part is being read. Provided 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. 11. Thanks to the introduced multiplexers, the first $f_s/4$ shifter could be merged easily into the filter structure. This eliminated virtually

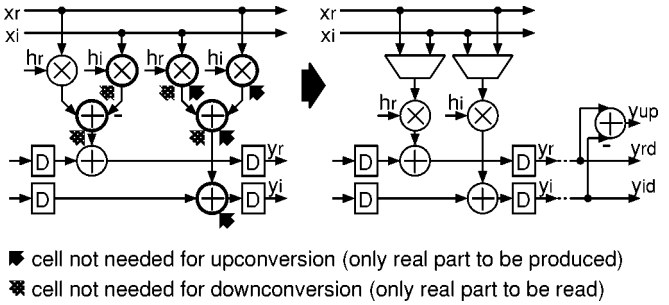


Fig. 11. Filter H2 optimization (one tap).

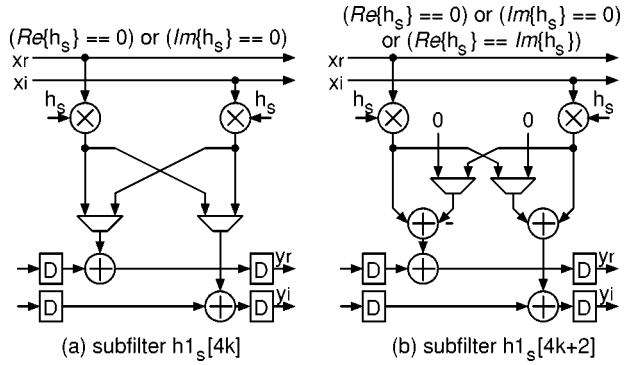


Fig. 12. Subfilters $\mathrm{h1}_s[4k]$ *and* $\mathrm{h1}_s[4k+2]$ optimizations (one tap).

all hardware working at 160 MHz, apart from the polyphase structures.

In filter H1, the coefficient rotation factors are multiples of $2\pi/16$. This has several important consequences.

1) Subfilter $\mathrm{h1}_s[4k]$ remains *real* or *imaginary only*, since the rotation factors are always $k\pi/2$, so it can be implemented as a *real only* filter [see Fig. 12(a)].
2) Subfilters $\mathrm{h1}_s[4k+1]$ and $\mathrm{h1}_s[4k+3]$ remain symmetrical provided the original 23-tap filter was symmetrical as well, so only half of the taps need to be implemented.
3) Filter $\mathrm{h1}_s[4k+2]$ is not *real only*, but the rotated coefficients are either *real only* or *imaginary only* or $h_r = h_i$. Thus, provided some additional multiplexing is introduced, only two instead of four multipliers are necessary per tap, as shown in Fig. 12(b).

D. Frequency Spectrum Manipulation

The reduction of the necessary filter shapes by the introduction of the frequency shifters was advantageous thanks to the fact that certain operations on the signal spectrum can be performed at a little hardware cost. The frequency shifts by $f_s/4$ and $f_s/2$ only require a multiplication by 1, -1, and 0, i.e., the $\sin/\cos$ values of $k\pi/2$, and some additional multiplexing of *real* and *imaginary* parts, as depicted in Fig. 13.

E. Polyphase Decomposition

The polyphase expansion elements remain the only hardware which has to work at the maximal frequency. We implemented both down conversion, shown in Fig. 14(a), and up conversion, depicted in Fig. 14(b), using multiplexers. These simple structures fulfilled the hard speed requirements.

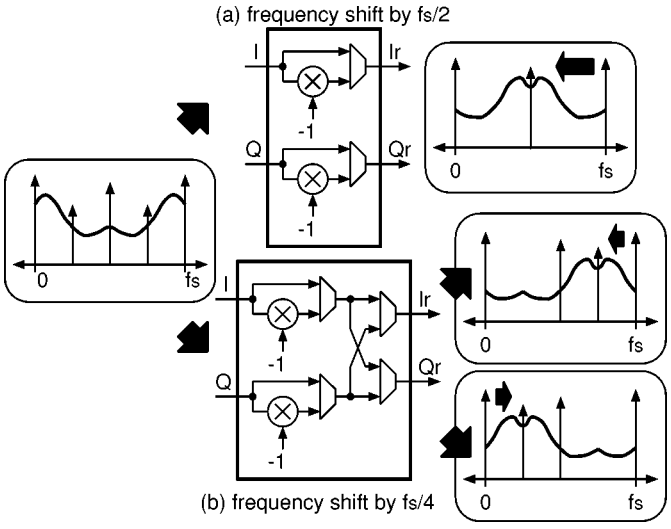


Fig. 13. Frequency spectrum manipulation by $f_s/2$ and $f_s/4$ shifts.

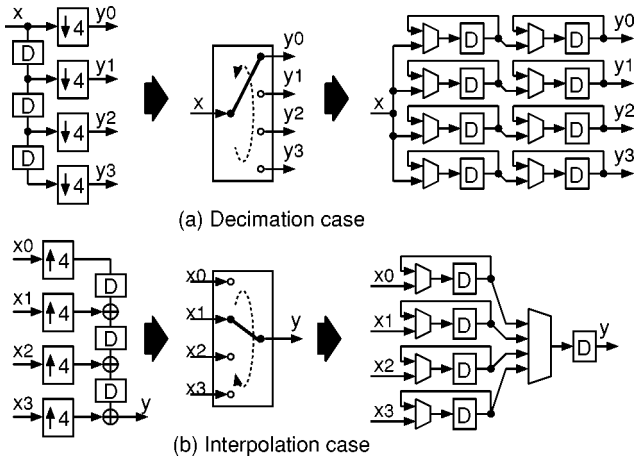


Fig. 14. Polyphase expansion elements implementation.

F. Clocking Scheme, Control, and Reset

The clocking scheme is of importance since some hardware works at a high frequency and multiple clocks are used through the chip as well. There are three basic clock regions in the design: 160, 40, and 10 MHz. However, as many as six different clocks are actually present. The 40-MHz clock is internally split into clk40_up, clk40_down and clk40_iphase, as shown in Fig. 15. This splitting of the 40-MHz clock region was motivated by power-saving issues. Since the H2 filter bank is not needed during the 40-MHz mode, power can be saved by turning it off. This is achieved by gating of the clk40_up clock. The clk40_down is clocking the 40-MHz hardware necessary also in this mode. The clk40_iphase is an inverse phase 40-MHz clock used for the registers at the boundary of the clk40_up and clk40_down regions to avoid clock skew. The mode is selected by the at40_CNT control signal. A more detailed description of our methodology for deriving and verifying the multiple clocks at high-level is given in [14].

All clocks are internally derived from the 160/40-MHz clock. However, for testing purposes, we retained the option to supply them externally as well, as shown in Fig. 15. Additionally, there

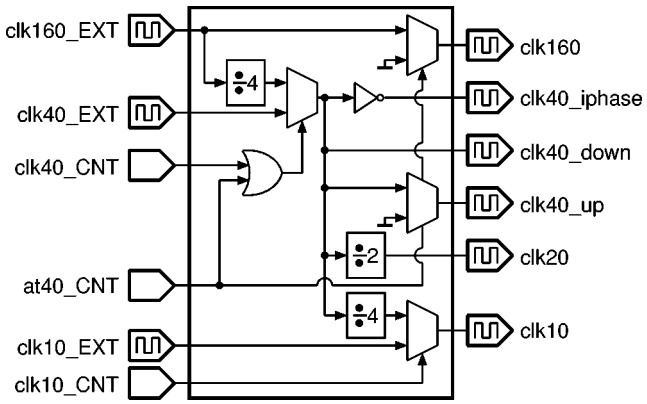


Fig. 15. Internal clock generation circuitry.

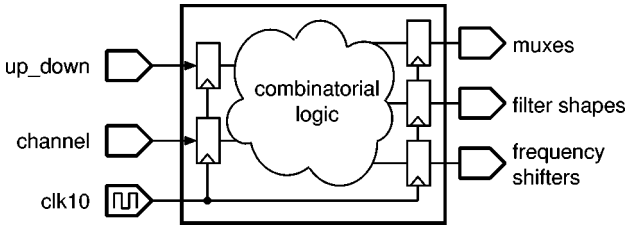


Fig. 16. Control signals generation.

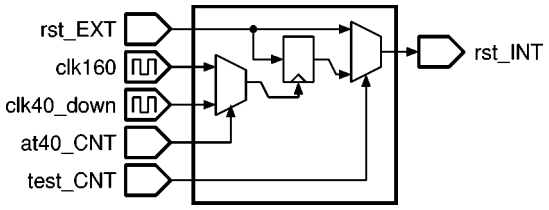


Fig. 17. Internal *reset* synchronization with the proper master clock.

is also a 20-MHz clock generated for the QAM receiver chip intended for processing of the downconverted signal.

The control circuitry basically contains combinatorial logic calculating the control signals for the filter shapes, frequency shifters, and multiplexers, as shown in Fig. 16. The *channel* is a 5-bit input indicating the channel number to select (i.e., from 1 to 31), while up_down selects the up or down conversion mode. The input and output signals of the control unit are registered with the 10-MHz clock.

Finally, since *synchronous reset* is used through the design, it has to be synchronized to the proper master clock, as shown in Fig. 17.

G. Chip Design Summary

The complete detailed architecture with indicated clock regions is shown in Fig. 18. The signal inputs and outputs are 12 bits wide, while the internal word length is 14 bits. The chip parameters summary is given in Table I.

The chip was designed using a multirate version of the OCAPI C++ design environment [15]. The C++ description has 4400 lines which resulted in 12 000 lines of generated RT-level VHDL code. This was synthesised by means of the Synopsys design compiler using Alcatel Microelectronics 0.5-μ CMOS standard cell library. It was successfully taped-out,

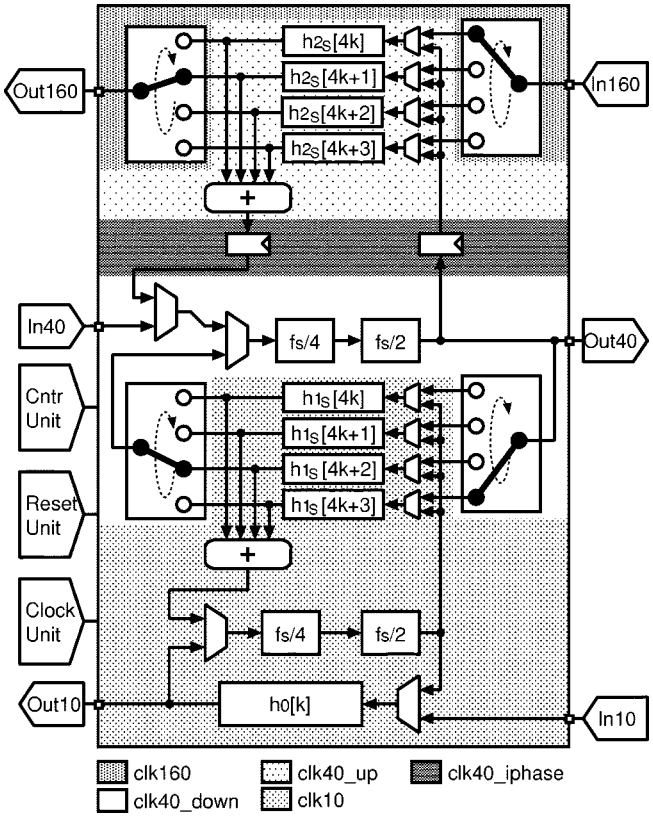


Fig. 18. Detailed chip architecture with indicated clock regions.

manufactured, and functionally tested. The chip layout is shown in Fig. 19. The three distinctive areas at the both sides and at the bottom contain extra analog circuitry for measurements of the substrate noise of different clock regions.

V. Simulation and Measurement

To verify the chip functionality, cosimulations with a high-level model of the QAM transmitter and receiver were run. The evaluated parameter was error vector magnitude (EVM), which gives the variance of the received QAM signal constellation points, normalized to the constellation power [5]. The simulation was run for each channel in both upconversion and downconversion modes and the results are given in Fig. 20. The minimal $\mathrm{EVM}_{\min} = -49.85$ dB value was obtained by processing the QAM data without up or down conversion. The quantization scheme was also tested in this setup. The resulting EVM values from cycle-accurate floating- and fixed-point simulations are given in Fig. 20 as well.

The test setup for validation of the chip functionality is given in Fig. 21. To avoid a high-speed ADC, we used two tested chips in series, the first one in upconversion and the second one in downconversion mode. As input, a sine wave at $f_s/4$ was used. Various combinations of channels and modes were tested. An example of measured data is given in Fig. 22. Fig. 22(a) shows the power spectral density (PSD) of the input sine-wave after the ADC. Fig. 22(b) gives a power spectrum of the first chip output after up conversion to channel 18. Fig. 22(c) and (d), respectively, are giving outputs of the second chip in downconversion mode, when the channel was set to 18 [see Fig. 22(c)] and 1 [see Fig. 22(d)]. The dc element was introduced by the sine generator. It should be mentioned that due to parasitic effects on the PCB, we were not able to conduct the measurements up to 160 MHz. Instead, we used a lower master clock of 100 MHz. The 40-MHz mode was measured completely. The power consumption was not specifically measured for the chip alone, but we observed a consumption of approximately 100 mW per 10 MHz for the complete test board.

TABLE I
Chip Parameters Summary

Technology	Alcatel CMOS 0.5μ/3.3V
Master clock	163.84/40.94MHz
Derived clocks	40.94, 20.48 and 10.24MHz
IO wordlength	12 bits
Internal wordlength	14/16 bits
Gate count	86k
Core area	24.22mm^2
Chip area	38.4mm^2
Package	120 pin PGA

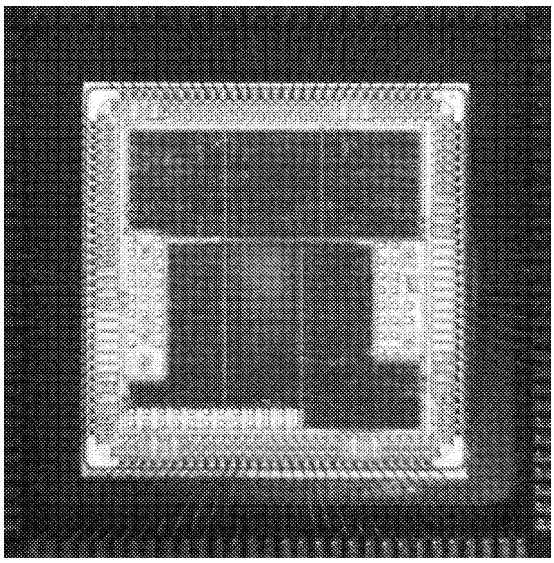


Fig. 19. Die photo of the chip.

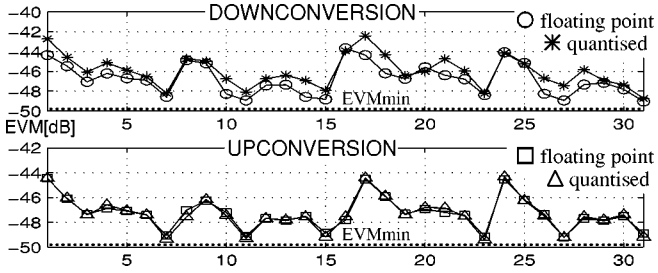


Fig. 20. Simulated EVM values with respect to the selected channel.

VI. Future Work

The major drawback of the presented scheme is the necessity of a preprocessing step when the channel is not placed at one of the distinct positions. This would mean, eventually, that there does not exist an anti-alias filter shape to accommodate the

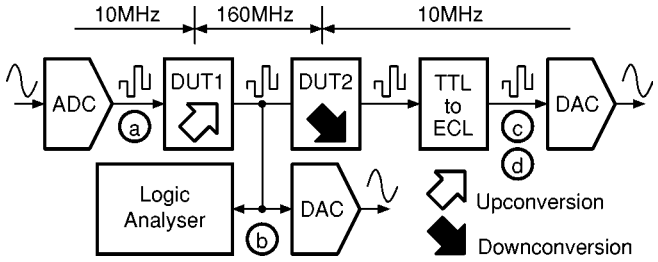


Fig. 21. Setup for chip measurements.

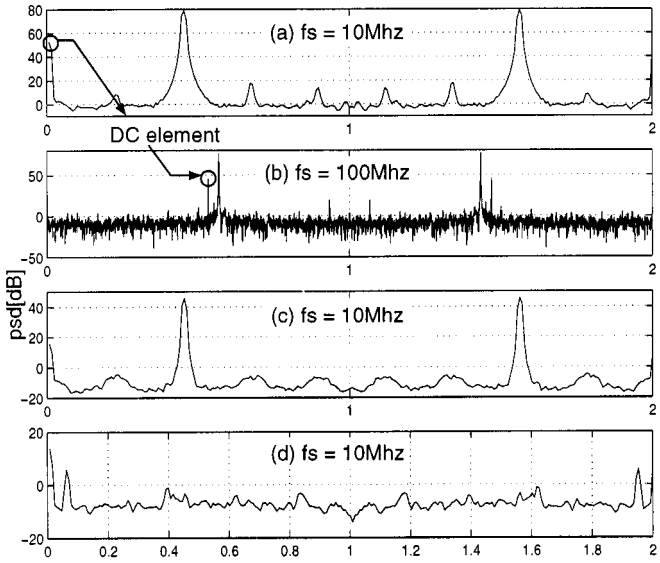


Fig. 22. Chip measurements results ($f_s = 2$). (a) Input signal. (b) Upconverted signal at channel $n = 18$ position. (c) Downconverted signal (b) with channel parameter $n = 18$. (d) Down converted signal (b) with channel parameter $n = 1$.

whole band of interest, as shown in Fig. 23(a). Another problem is that, provided it would be possible to get the complete frequency band into the baseband, it will not be centered around $f_s/4$ [or dc; see Fig. 23(b)].

However, the presented technology can provide answers for these problems as well. The first one can be solved by designing the anti-alias filters sufficiently overlapping. Then, it would always be possible to select an anti-alias filter for a signal of given bandwidth [Fig. 23(c)] and eventually get the signal unscathed into the baseband. The second problem can be solved afterwards by introducing an additional fine frequency tuning element, as shown in Fig. 23(d). This can be done with a great precision and efficiency, since it would be performed at a low frequency.

It would result in a stand-alone all-digital IF processing element which can be very interesting as a part of a frequency recovery loop.

VII. Conclusion

An implementation of a high-performance all-digital quadrature up and down converter was presented. It features a novel mixerless approach introducing flexibility in IF selection without sacrificing performance or cost. The up and down conversion is accomplished by a sample rate change with programmable anti-alias filters implemented using a modified CSD technique. It also shows that digital processing is feasible at IF frequencies.

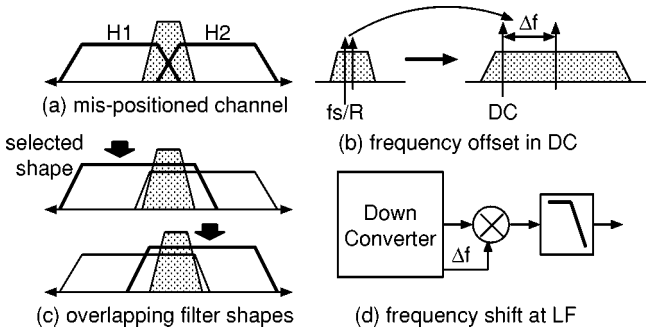


Fig. 23. Drawbacks of presented architecture (a),(b) and proposed solutions (c),(d).

Acknowledgment

The authors wish to thank B. Van Thielen for providing the chip measurement data and M. Van Eylen for making the die photo.

References

[1] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, pp. 26–38, May 1995.
[2] "HSP 50 215 digital up converter," Harris Semiconductor Data Sheet, Jan. 1999.
[3] "HSP 50 016 digital down converter," Harris Semiconductor Data Sheet, Jan. 1999.
[4] K. Cho and H. Samueli, "A 8.75-Mbaud single-chip digital QAM modulator with frequency-agility and beamforming diversity," in *Proc. Custom Integrated Circuits Conf. (CICC)*, Orlando, FL, May 2000, pp. 27–30.
[5] J. Vankka, M. Kosunen, I. Sanchis, and K. Halonen, "A multicarrier QAM modulator," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 1–10, Jan. 2000.
[6] L. Tan and H. Samueli, "A 200-MHz quadrature digital synthesizer/mixer in 0.8-μ CMOS," *IEEE J. Solid State Circuits*, vol. 30, pp. 193–200, Mar. 1995.
[7] P. Schaumont, S. Vernalde, M. Engels, and I. Bolsens, "Low-power digital frequency conversion architectures," *J. VLSI Signal Processing*, vol. 18, pp. 187–197, 1998.
[8] 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 J. Solid-State Circuits*, vol. 26, pp. 1970–1980, Dec. 1991.
[9] S. Jou, S. Wu, and CH. Wang, "Low-power multirate architecture for IF digital frequency down converter," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 1487–1494, Nov. 1998.
[10] MCNS/DOCSIS Specs (2001). [Online]. Available: http://www.cable-modem.com
[11] T. Parks and J. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 189–194, Mar. 1972.
[12] K. Hwang, *Computer Arithmetic, Principles, Architecture and Design*. New York: Wiley, 1979.
[13] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with power-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1044–1047, July 1989.
[14] L. Rijnders, P. Schaumont, S. Vernalde, and I. Bolsens, "High level analysis of clock regions in C++ system description," in *Proc. Workshop Synthesis and Systems Integration of Mixed Technologies (SASIMI)*, Kyoto, Japan, Apr. 2000, pp. 38–42.
[15] P. Schaumont, S. Vernalde, L. Rijnders, M. Engels, and I. Bolsens, "A programming environment for the design of complex high-speed ASICs," in *Proc. Design Automation Conference (DAC)*, San Francisco, CA, June 1998, pp. 315–320.

Robert Paško received the M.S. degree in electronics from the Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Bratislava, Slovakia, in 1994. He is currently working toward the Ph.D. degree at the Katholieke Universiteit Leuven, Belgium.

His main research interests are in the areas of IC/FPGA design for digital signal processing, telecommunications and embedded networking, and the use of high-level programming languages for these tasks.

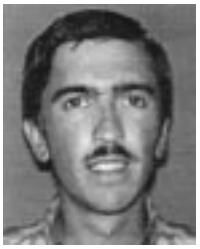
Luc Rijnders received the electrical engineering degree and the Ph.D. degree from the Katholieke Universiteit Leuven, Belgium, in 1982 and 1990, respectively.

In 1986 he joined the Inter-University Microelectronics Centre (IMEC), Leuven, Belgium, where he worked on the development of CAD tools for symbolic and procedural layout design, data path optimization and silicon compilers. He co-designed several ASICs in the telecom area. His research interests are focused on all aspects of improving design productivity for embedded systems. His favorite programming language is LISP.

Patrick R. Schaumont (M'97) received the electronics engineering degree from the Industriele Hogeschool van het Rijk, Gent, Belgium, and the computer science degree from the Rijksuniversiteit, Gent, Belgium.

Currently, he is a Senior Researcher at the Inter-University Microelectronics Centre (IMEC), Leuven, Belgium, guiding the OCAPI project and team toward innovative object-oriented design methods. His previous research activities include code generation in the Cathedral Silicon Compiler and various DSP demonstrator designs.

Mr. Schaumont is a member of the VSIA and received the 1999 Outstanding Contributor Award.



Serge A. Vernalde (S'88–M'90) received the electrical engineering degree in 1990 at the University of Leuven, Belgium.

In 1990 he joined the Inter-University Microelectronics Centre (IMEC) laboratory, where he developed the Cathedral-2/3 datapath compiler for behavioral synthesis of high-speed DSP algorithms on accelerator processors. He was responsible for the design of a 80-Mbit/s Reed–Solomon decoder for HDTV, a motion estimator for H261 videophony, an upstream cable modem, and a cordless telephony transceiver. He is co-author of the book *Accelerator Data-Path Synthesis for High-Throughput Signal Processing Applications* (Norwell, MA: Kluwer, 1997). Currently, he is heading the networked reconfigurable appliances group in IMEC which focuses on embedded information appliances and advanced system design methodologies based on object oriented programming techniques.

Daniela Ďuračková received the M.S. and Ph.D. degrees from the Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Bratislava, Slovakia, in 1974 and 1981, respectively.

Since 1991 she has been an Associate Professor in the Department of Microelectronics at the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology. The main areas of her research and teaching activities are the design of analog and digital circuits and neural networks.