

ON THE DESIGN AND MULTIPLIER-LESS REALIZATION OF DIGITAL IF FOR SOFTWARE RADIO RECEIVERS WITH PRESCRIBED OUTPUT ACCURACY

S. C. Chan and K. S. Yeung

Department of Electrical and Electronic Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong.
e-mail: scchan@eee.hku.hk, ksyeung@eee.hku.hk

ABSTRACT

This paper studies the design and multiplier-less realization of the digital IF in software radio receivers. The new architecture consists of a compensator for compensating the passband droop of the conventional cascaded integrator and comb (CIC) filter. The passband droop is improved by a factor of four and it can be implemented with four additions using the sum-of-powers-of-two (SOPOT) coefficients. The decimation factor of the multistage decimator is also reduced so that its output can be fed directly to the Farrow structure for sample rate conversion (SRC), eliminating the need for another L -band filter for upsampling. By so doing, the programmable FIR filter can be replaced by a half-band filter placed immediately after the Farrow structure. As the coefficients of this half-band filter, the multistage decimators and the subfilters in the Farrow structure are constants, they can be implemented without multiplication using SOPOT coefficients. As a result, apart from the limited number of multipliers required in the Farrow structure, the entire digital IF can be implemented without any multiplications. A random search algorithm is employed to minimize the hardware complexities of the proposed IF subject to a given specification in the frequency domain and prescribed output accuracy, taking into account signal overflow and round-off noise. Design results are given to demonstrate the effectiveness of the proposed method.

I. INTRODUCTION

Software radio is a general hardware/software platform for supporting inter-communication between different wireless communication systems [10][11]. The basic idea of an ideal software radio receiver is to digitize the received signal using high-speed ADCs and to process it by a sophisticated programmable system, probably consisting of a combination of hardware that is re-configurable or programmable, and digital signal processors (DSP). Due to various limitations of current digital technology and signal converters, most software radio architectures considered digitalize the down-converted signal at the intermediate frequency (IF). It is envisioned that, with the availability of low-cost and high-speed signal converters with reasonable accuracy, software radio employing digital signal processing techniques is a cost effective means to offer more flexibility and less sensitivity to analog components than traditional receiver employing analog IF. Fig. 1 shows a commonly used IF architecture for software radio receivers. The IF-signal is digitized at a bandwidth of 20 to 40 MHz. A programmable digital decimator and a sample rate changer are employed to isolate the desired user's channel from the signal spectrum and convert it to an appropriate sampling rate for further processing in the DSP [10]. The digital decimator will normally consist of multiple stages of decimators. As the sampling rate of the baseband signal is much lower than that at the IF, the output of each stage in the decimator will consist of a bandlimiting (anti-aliasing) digital filter and a downsampler (decimator) to filter out the unwanted signals and lower the sampling rate. By selecting an appropriate number of stages, different integer decimation ratios can be implemented. A programmable FIR is usually needed to remove the residual interference from adjacent channels. It is because the sampling rate is usually not an integer multiples of the channel spacing. Hence, the multiple stages of decimation filters, which implement an integer decimation factor, are unable to remove this residual interference from adjacent channels. Together with the sample rate changer (SRC), which provides the necessary rational or even irrational rate-change factor, it is possible to accommodate signals with a wide variety of bandwidths required by different communication standards. There are several important contributions to the efficient hardware implementation of the programmable receiver and most of them are based on the CIC filter and its variants [2,3,10]. In addition, it is usually assumed that the programmable FIR and the SRC immediately after it are fast enough to handle the decimated input signal. One drawback of this conventional structure is that the output of the multistage decimator, which is obtained by downsampling the high-rate IF signal from the ADC, has to be upsampled again by the L -band filter in order to carry out the

arbitrary sample rate conversion. Another important problem, which limits the throughput of the system for wideband signal, is the high processing requirement of the programmable FIR filter. In a previous work [1], we propose to reduce the decimation factor of the multistage decimator so that its output can be fed directly to the Farrow structure for sample rate conversion, eliminating the need for another L -band filter for upsampling. Furthermore, it was found out that the programmable FIR filter can be replaced by a half-band filter placed immediately after the Farrow structure, i.e. after sample rate conversion. This new structure is shown in Fig.5. This significantly reduces the implementation complexity of the proposed software radio receiver because this half-band filter, which consists of fixed filter coefficients, can be implemented efficiently without multiplication using sum-of-powers-of-two (SOPOT) coefficients. As the coefficients of the multistage decimators and the subfilters in the Farrow structure are also fixed, they can also be implemented efficiently using SOPOT coefficients. As a result, apart from the limited number of multipliers required in the Farrow structure, the entire digital IF can be implemented without any multiplications. In this paper, the hardware complexities of the proposed programmable decimator and the SRC are minimized subject to a given specification in the frequency domain and prescribed output accuracy. The hardware complexity could be the number of adder cells and/or registers used, which is related to the exact wordlength being used for each intermediate data. The output accuracy of the digital filters is specified statistically by its output noise power due to the rounding operations performed, which is modeled by the popular uncorrelated white noise model. The wordlengths and the scaling options of the intermediate data are then determined by a random search algorithm [9] in order to avoid signal overflows, and achieve the objectives mentioned earlier. In contrast to the conventional approaches that minimizes only the total number of SOPOT terms, the new criteria is more realistic and general for hardware implementation. In addition, we propose a new second-order compensator to compensate for the passband droop of the conventional CIC filter. Design results showed that the passband droop is improved by a factor of four and it can be implemented with four additions using the SOPOT coefficients. The paper is organized as follows: Section II is devoted to the design and implementation of the second-order CIC compensator. Section III presents the signal round-off and overflow analyses. Section IV describes the random search algorithm for determining the internal wordlengths of the programmable decimator while satisfying the given specification. This is then followed by a design example in Section V. Finally, conclusion is drawn in Section VI.

II. SECOND-ORDER CIC COMPENSATOR

In this section, the design and multiplier-less realization of a second-order CIC compensator is presented. In the design of the programmable decimator, the CIC filter [4] is commonly employed to reduce the hardware complexity. However, the passband droop of the CIC filter will significantly affect the quality of the anti-aliasing filter, if the decimation ratio is small [12]. The transfer function of the CIC filter is given by

$$H_{CIC}(z) = [H(z^{M_{CIC}})/H(z)]^L, \quad (1)$$

where $H(z) = 1 - z^{-1}$, M_{CIC} is the down-sampling ratio of the CIC filter, and L is the number of CIC stages. To compensate for the passband droop of the CIC filter, the following second-order CIC compensator

$$P(z) = a + bz^{-1} + az^{-2}, \quad (2)$$

where a and b are real-valued constants to be determined, is employed as shown in Fig. 3(a). Note that $P(z)$ is linear-phase. This avoids phase distortion and reduces the implementation complexity. The coefficients a and b are chosen to equalize the passband droop of the CIC filters and they are readily determined using the Parks-McClellan algorithm. For multiplier-less realization, the constants a and b can be expressed as SOPOT representation and they can be determined using a random search algorithm similar to [9][14]. The frequency responses of the CIC filter, the compensator and the compensated CIC filter for $M_{CIC} = 4$ and $L = 3$ are shown in Fig. 2. The worst-case passband deviation and aliasing attenuation of the compensated CIC (CIC) filter for $M_{CIC} \geq 2$ and $L = 3$ is 0.00605-dB and 84.19-dB, respectively (0.02508-dB and 84.21-dB). It can be seen that the second-order CIC compensator improves the passband droop by a factor of four. Using the noble identity, the compensated CIC filter in Fig. 3(a) can be implemented more efficiently as shown in Fig. 3(b). The overall architecture of the programmable decimator is shown in Fig. 5. To further reduce the hardware complexity, all the filters are implemented using the multiplier-block (MB) [8] technique, where all the coefficient multiplications can be realized with minimum number of adders. For example, only two adders are required to implement the multiplication with the coefficients a and b . It is noted that the sharpened CIC (SCIC) filter [3] and the interpolated second-order polynomial (ISOP) [2] can also be used to improve the passband droop of the basic CIC filters. The proposed compensated CIC filter is similar in concept to the ISOP. However, a general linear-phase filter is used to offer more flexibility. The hardware complexity, on the other hand, is still very low, thanks to the use of MB. Next, we shall consider the signal round-off and overflow analyses of the proposed system.

III. SIGNAL ROUND-OFF AND OVERFLOW ANALYSES

1. Analysis of Signal Round-off Noise

Signal round-off errors occur because of signal overflows and rounding of intermediate data after multiplications with the filter coefficients. Due to the difficulty in analyzing exactly the rounding errors, they are usually modeled as uncorrelated white noises. That is, the quantization noise [6] will have zero mean and a variance $\sigma^2 = \Delta^2 / 12$, where Δ is the quantization step-size, which is determined by the number of fractional bits that is retained after each multiplication.

The finite wordlength implementation of the integrator in the CIC filter deserves careful consideration to avoid excessively rounding and overflow errors. Since each integrator has a DC gain of M_{CIC} , additional $\lceil \log_2(M_{CIC}) \rceil$ bits has to be allocated to the fractional part of each integrator to prevent signal round-off and overflow error, where $\lceil \cdot \rceil$ denotes the nearest lower integer. The comb section, the CIC compensator and all the anti-aliasing filters (LPFs) are implemented in their

transposed forms, similar to the one shown in Fig. 4. The filter coefficients are represented as SOPOT coefficients and they are simultaneously realized as a multiplier-block (MB) [8]. Once the SOPOT coefficients of the filters are determined, say using the random search algorithm [9], the wordlength for the products $x[n] \cdot h_i[n]$ are available. To minimize the hardware complexity, these products might be rounded using the signal round-off operator $Q(\cdot)$. In fixed-point representation, each intermediate signal is represented in the form $\langle n/m \rangle$, where n is the number of integer bits including the sign bit, and m is the number of fractional bits. In general, if m bits are rounded to B bits with $B < m$, then the round-off noise power P_e is given by

$$P_e = \Delta^2 / 12, \quad (3)$$

where $\Delta = 2^{-(B-1)}$. If there are N such rounding processes at the i^{th} stage of the programmable decimator, then the total noise power $P^{(i)}$ due to these rounding sources is simply given by

$$P^{(i)} = \sum_{k=1}^N P_{e_k}. \quad (4)$$

The total output noise power at the i^{th} stage, P_i , taking into account the noise sources at previous stages is

$$P_i = P_{i-1} \cdot \sum_{n=0}^{N-1} |h_i[n]|^2 + P^{(i)}, \quad (5)$$

where $h_i[n]$ is the impulse response of a digital filter in the current stage, which is assumed to have a filter length of N . The output accuracy A_i at the i^{th} stage, in terms of the number of fractional bits, is therefore approximately given by

$$A_i = \lceil 10 \cdot \log_{10}(P_i) / 6 \rceil \text{ bits}. \quad (6)$$

It should be noted that the larger the number of noise sources, the lower will be the accuracy. The noise power can however be reduced by increasing the internal wordlengths for the fractional bits at different stages of the digital filters, at the expense of increased hardware complexity.

2. Overflow Handling

Signal overflows occur when the allocated wordlength of the integer bits is insufficient to accommodate the growth in integer wordlength of the signal after additions. In order to avoid overflow, more bits must be allocated to the integer part of the adder output and the register holding it. There is, however, an option to retain or decrease the number of bits in the fractional part, depending on the required output accuracy. To determine whether signal overflow will occur at a particular adder, the conservative L1 scaling measure is used in this paper. More precisely, the input signal $x[n]$ is assumed to take on its maximum value denoted by x_{\max} . Assuming that the FIR filter is implemented in its transposed form with a transfer function of

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n},$$

the k^{th} impulse response coefficient of the digital filter is bounded by

$$y_{\max,k} = x_{\max} \sum_{n=0}^k |h[n]|, \quad k = 0, 1, \dots, N-1. \quad (7)$$

Using (7), it is possible to determine the worst-case integer wordlength of each adder and hence the size of its output register to avoid signal overflow. It should be noted that there are other methods such as L2 scaling to handle signal flows. However, there is still a small probability that overflows will occur.

IV. RANDOM SEARCH ALGORITHM

In this section, we introduce a random search algorithm to minimize the hardware complexity of the programmable

decimator and the SRC to satisfy a given specification in the frequency domain and prescribed output accuracy. The hardware complexity could be the number of adder cells and/or registers used, which is related to the exact wordlength being used for each intermediate data. The output accuracy of the programmable decimator is specified statistically by its output noise power. It is assumed to be generated from the rounding operations performed, which are modeled using the uncorrelated white noise model. The internal wordlengths and scaling options of each intermediate data are the variables to be optimized. First of all, the real-valued coefficients of the various filters are designed using the Park-McClellan algorithm, except for the FDDF, which is designed using the method in [14]. They are then converted into SOPOT coefficients using the random search algorithm [9] and are implemented using MB. After that, the maximum wordlengths of the intermediate data $x[n] \cdot h_i[n]$ as shown in Fig. 4 are available. The wordlength formats of all internal registers and the structures of all adders for avoiding any overflow can then be determined using the method described in Section III. We can either retain the fractional parts for those scaled outputs or reduce its value by one or an appropriate integer. This option is stored in another vector \vec{a}_f which will be optimized together with the parameter vector \vec{a} storing all the intermediate signal formats. The noise power at the filter output is readily computed accordingly to the analysis in Section III. Our goal is to lower the internal wordlengths of each intermediate data as specified in \vec{a} and \vec{a}_f so that a measure of hardware cost C , say the total number of adder cells, is minimized subject to the given specification. More precisely, the design problem is

$$\min_{(\vec{a}, \vec{a}_f)} C(\vec{a}, \vec{a}_f) \text{ subject to } P_{total} \leq P_{spec}, \quad (8)$$

where P_{total} is the total output noise power and P_{spec} is the specified output accuracy. Using the random search algorithm [9], the vectors (\vec{a}, \vec{a}_f) are searched in the neighborhood of their full precision values, i.e. the values without rounding. The one with the minimum hardware cost C is declared as the solution of this problem. There are several advantages of this algorithm. First of all, with the high computational power of nowadays personal computer (PC), the time to obtain a high quality solution is still manageable, especially when an initial solution is available by some means. Secondly, it is applicable to problems with general objective functions and very complicated inequality constraints, as illustrated in this work. It is possible to combine this searching process with the MB generation for better performance. But the computational time will be greatly increased. We now present a design example.

V. DESIGN EXAMPLE

The specifications of the proposed programmable decimator, are shown as follows:

$$\delta_{p-\max} = 0.00173 \text{ (0.015-dB in passband deviation)}$$

$$\delta_{s-\max} = 0.0001 \text{ (80-dB in stopband attenuation)}$$

$$\delta_{d-\max} = 0.00316 \text{ (50-dB in fractional delay error)}$$

$$P_{spec} = 2.512 \times 10^{-10} \text{ (96-dB in output accuracy)},$$

where $\delta_{p-\max}$ is the maximum passband ripple error, $\delta_{s-\max}$ is the maximum stopband ripple error, $\delta_{d-\max}$ is the maximum fractional delay ripple error and P_{spec} is the maximum output noise power. These values are chosen to support GSM and W-CDMA standards as we shall see later in this section. The output accuracy, in terms of the number of fractional bits, can be calculated from (6) to be $A=16$. In Fig. 5, the programmable decimator consists of a compensated CIC filter, three lowpass filters (LPFs), a fractional-delay digital filter

(FDDF) for sample rate changing [5], and a halfband filter (HBF). The multiplier-less realization of the FDDF follows the method described in [14] while the LPFs and HBF are realized using the method described in Section IV [1]. The worst-case passband deviation and the stopband attenuation of the programmable decimator for $L=3$ and $2 \leq M_{cic} \leq 10$ are 0.01372-dB and 81.98-dB, respectively. The architecture of the CIC filter is shown in Fig. 6. There are L stages ($L=3$ in our case) of integrator and comb section. Each integrator section consists of an adder, a register and a programmable shifter S_k . The maximum down-sampling ratio M_{cic} of the CIC filter employed is 10. The programmable shifter is then required to shift from 0 up to $\lceil \log_2(M_{cic}) \rceil$ bits, i.e. 0 to 3 bits in our example. For each additional integrator, an $\lceil \log_2(M_{cic}) \rceil$ bits increase in the fractional part of the wordlength is required to prevent excessively signal round-off error. On the other hand, each comb section consists of an adder and a register. The fractional part of the comb section is equal to that of the previous integrator stage to avoid rounding operation. After the comb sections, a programmable SOPOT scalar is used to implement the remaining scaling due to the DC gain of the CIC filter. This constant r is given by

$$r = 2^{L \lceil \log_2(M_{cic}) \rceil} / (M_{cic})^L, \quad (9)$$

which should be equal to or less than one. The wordlength of its output is also optimized using the method described in Section III. Due to page length limitation, detailed internal wordlengths of the programmable decimator are not shown here. Interested readers are referred to [1] for more details. Here, we only summarize the major results for the proposed programmable decimator. The input signal $x[n]$ of the CIC filter is assumed to have a format of $\langle 1/13 \rangle$, i.e. 14-bits with $x_{\max} = 0.99988$. The wordlengths of the integrator and comb sections are shown in Table 1. The final output of the programmable decimator has a wordlength format of $\langle 9/20 \rangle$. The whole programmable decimator uses 8149 adder cells and the output noise power P_{total} is 2.3×10^{-10} , i.e. 96.38-dB or 16.0085 fractional bits. If the programmable decimator employs fixed wordlengths of 24 bits, it requires 8664 adder cells (with MB. Otherwise it will be much higher) but the prescribed output accuracy of 16-bit cannot be met. Next, we demonstrate the application of this programmable decimator to a multi-standard software radio receiver for supporting the GSM and W-CDMA standards. It is assumed that the IF signal is sampled at 80M samples per second. Table 2 shows the values of the various parameters in order to down-convert and isolate the GSM and W-CDMA signals to 800ksps and 3.84Msps, respectively. It should be noted that the sample rate change factor M^* is given by

$$M^* = M_{cic} \cdot 2^k \cdot M_f, \quad (10)$$

where M_f is the rational down-sampling ratio of the FDDF while k is the number of the remaining decimators to be realized by the compensated CIC and LPFs.

VI. CONCLUSION

The design and multiplier-less realization of a new digital IF architecture for software radio receivers are presented. The new architecture consists of a compensator for compensating the passband droop of the conventional CIC filter and eliminates the need for a programmable FIR filter. Apart from the limited number of multipliers required in the Farrow structure, the entire digital IF can be implemented without any multiplications. The hardware complexities of the proposed IF are minimized using a random search algorithm subject to a given specification in the frequency domain and prescribed output accuracy. Design results are given to demonstrate the effectiveness of the proposed method.

REFERENCES

- [1] S. C. Chan and K. S. Yeung, "On the design and multiplier-less realization of digital IF for software radio receivers," *Internal Report, The University of Hong Kong*, Dec. 2001.
- [2] H. J. Oh, S. Kim, G. Choi and Y. H. Lee, "On the use of interpolated second-order polynomials for efficient filter design in programmable downconversion," *IEEE J. Select. Areas Commun.*, April 1999, pp. 551-560.
- [3] A. Y. Kwentus, Z. Jiang and A. N. Willson, "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Processing*, vol. 45, pp. 457-467, Feb 1997.
- [4] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, Singapore, McGraw-Hill, 1998.
- [5] C. W. Farrow, "A continuously variable digital delay element," *IEEE Int'l. Conf. Circuits and Sys. 1988*, pp. 2641-2645.
- [6] A. V. Oppenheim and R. W. Schaffer, *Discrete-time signal processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [7] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete power-of-two coefficient space," *IEEE Trans. ASSP-31*, pp. 583-591, April 1983.
- [8] A. G. Dempster and M. D. MacLeod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II*, pp. 569-577, Sept. 1995.
- [9] C. K. S. Pun, S. C. Chan and K. L. Ho, "Efficient design of a class of multiplier-less perfect reconstruction two-channel filter banks and wavelets with prescribed output accuracy," *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pp. 599-602, 2001.
- [10] T. Hentschel and G. Fettweis, "Sample rate conversion for software radio," *IEEE Commun. Mag.*, pp. 142-150, Aug 2000.
- [11] C. Y. Fung and S. C. Chan, "A multistage filterbank-based channelizer for software radio base stations," *Accepted by IEEE ISCAS'2002*.
- [12] K. Y. Khoo, Z. Yu and A. N. Willson, "Efficient high-speed CIC decimation filter," *Proceedings of the 11th Annual IEEE International*, pp. 251-254, 1998.
- [13] A. Kwentus, O. Lee and A. N. Willson, "A 250 Msample/sec programmable cascaded integrator-comb decimation filter," *VLSI Signal Processing*, IX, pp. 231-240, 1996.
- [14] C. K. S. Pun, Y. C. Wu, S. C. Chan and K. L. Ho, "An efficient design of fractional-delay digital FIR filter using Farrow structure," *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pp. 595-598, 2001.

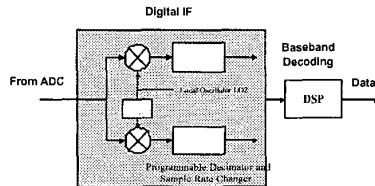


Fig. 1. Digital IF for software radio receiver.

Stage	Integrator	Comb
1	<2/16>	<5/22>
2	<3/19>	<6/22>
3	<4/22>	<7/22>

Table 1. Internal wordlengths of the integrator and comb sections.

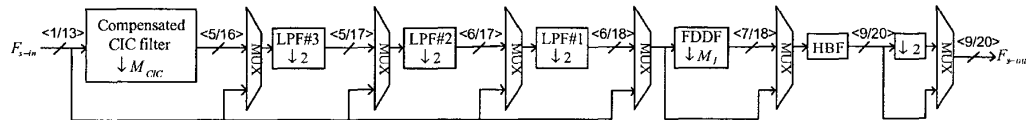


Fig. 5. Proposed programmable decimator structure.

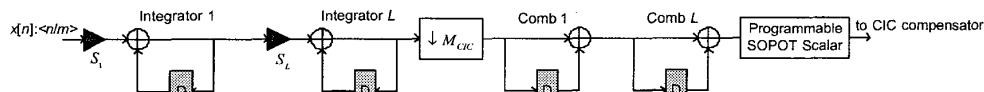


Fig. 6. L-stages cascaded integrator-comb (CIC) decimation filter architecture.

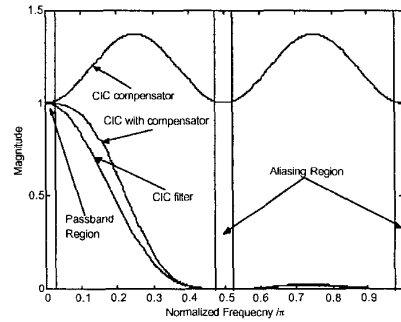


Fig. 2. Frequency responses of the CIC filter, the CIC compensator and the compensated CIC filter ($M_{cic} = 4$ and $L = 3$).

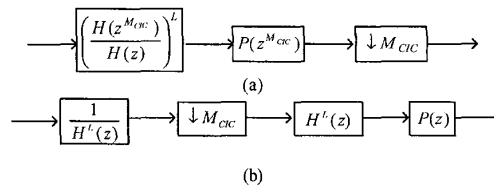


Fig. 3. Block diagrams of the compensated CIC filter: (a) before and (b) after the application of the noble identity.

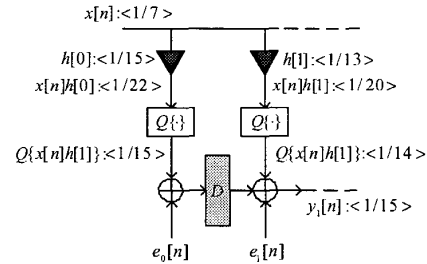


Fig. 4. Transposed form implementation of a typical FIR digital filter with round-off errors being modeled as uncorrelated white noise sources. D denotes a register.

	GSM	W-CDMA
M^*	100	125/6
M_{cic}	4	N/A
M_I	25/16	125/96
k	4	4

Table 2. Down-sampling ratio of the proposed programmable decimator for supporting GSM and W-CDMA standards.