Design and Complexity Optimization of a New Digital IF for Software Radio Receivers With Prescribed Output Accuracy

S. C. Chan, *Member, IEEE*, K. M. Tsui, K. S. Yeung, and T. I. Yuk

***Abstract*—This paper studies the design, signal round-off noise, and complexity optimization of a new digital intermediate frequency (IF) architecture for a software radio receiver (SRR). The IF under study consists of digital filters with fixed coefficients, except for a limited number of multipliers required in the Farrow-based sampling rate converter (SRC). The fixed-coefficient filters can be implemented efficiently using sum-of-power-of-two (SOPOT) coefficients and the multiplier- block technique, which gives minimum adder realization. Apart from the multipliers required in the SRC, the digital IF can be implemented without any multiplications. While most multiplier- less filter design and realization methods address only the coefficient round-off problem by minimizing the number of SOPOT terms used, the proposed design methodology aims to minimize more realistic hardware complexity measure, such as adder cells and registers, of the digital IF subject to a given spectral and accuracy specifications. The motivation is that the complexity is closely related to the target output accuracy, which is specified statistically by its total output noise power generated by rounding the intermediate data. Two novel algorithms for optimizing the internal wordlengths of linear time-invariant systems are proposed. The first one relaxes the solution to real valued and formulates the design problem as a constrained optimization. A closed-form solution can be determined by the Lagrange multiplier method. The second one is based on a discrete optimization method called the Marginal Analysis method, and it yields the desired wordlengths in integer values. Both approaches are found to be effective and suitable to large scale systems. A design example and the field programmable gate array (FPGA) realization of a multi-standard receiver are given to demonstrate the proposed method**

***Index Terms*—Design and multiplier-less realization, prescribed output accuracy, sampling rate conversion, software radio receiver (SRR), variable digital filters (VDFs), wordlength determination.**

I. Introduction

SOFTWARE radio is a general hardware/software platform for supporting inter-communication between different wireless communication systems [1], [2]. The basic idea of an ideal software radio receiver (SRR) is to digitize the received signal using high-speed analog-to-digital converters (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 (DSPs). Due to various limitations of current digital technology such as the speed and power dissipation of device technology used to fabricate the ADCs [3], most software radio architectures digitize the down-converted signal at the intermediate frequency (IF). Fig. 1(a) shows a commonly used digital IF for SRR. The analog IF signal is first digitized at a bandwidth, say 20 to 70 MHz. A programmable digital decimator and a sampling rate converter (SRC) are then employed to isolate the desired user's channel from the signal spectrum and convert it to an appropriate sampling rate for further processing by the digital signal processor (DSP) [1].

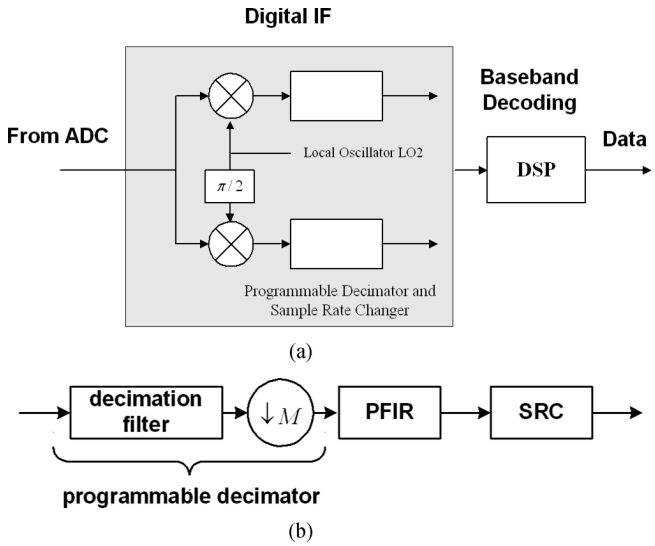


Fig. 1. (a) Digital IF for SRR. (b) Conventional receiver.

Conventional receivers usually consist of a programmable digital decimator, which is normally realized using multistage decimators, a programmable FIR filter (PFIR) and a SRC as shown in Fig. 1(b). The reason for such arrangement is that the required sampling rate of the baseband signal might be considerably lower than that at the IF. By passing the IF sampled signal through the decimators, which consists of the bandlimiting low-pass filters (LPFs) and a downsampler, the unwanted signals can be suppressed and the sampling rate can be lowered. By choosing appropriately the number of decimation stage, the bandwidth of the input signal can be decimated by an integer factor, which is just sufficient to cover the baseband signal. After the integer decimation, a PFIR is usually needed to remove the

Manuscript received June 9, 2005; revised April 18, 2006. This work was presented in part at the XI European Signal Processing Conference (EUSIPCO), Toulouse, France, September 3–6, 2002, at the 14th International Conference on Digital Signal Processing, Hellas, Greec, July 1–3, 2002, and the International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, May 23–26, 2005. This paper was recommended by Associate Editor W. Namgoong.

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: scchan@eee.hku.hk; kmtsui@eee.hku.hk; ksyeung@eee.hku.hk; tiyuk@eee.hku.hk).

Digital Object Identifier 10.1109/TCSI.2006.886003

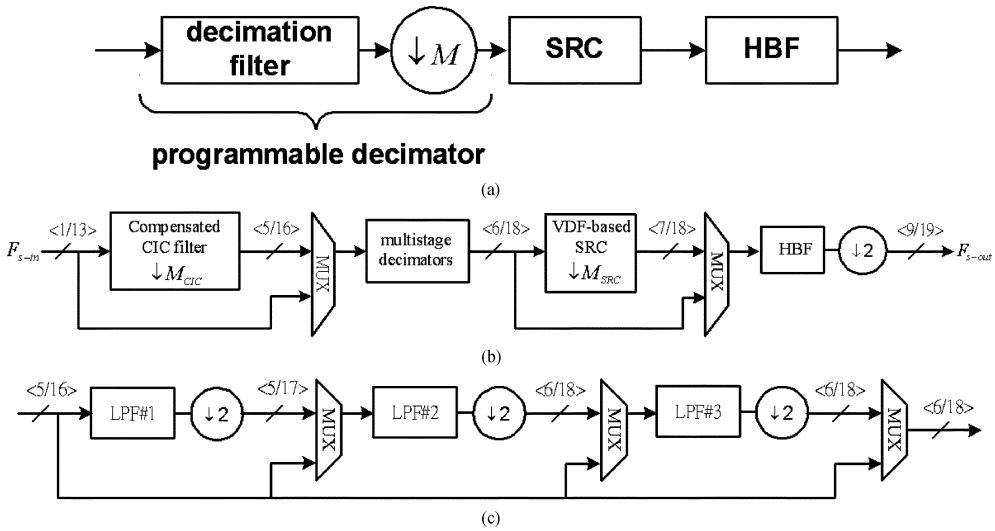


Fig. 2. (a) Proposed digital IF for software ratio receiver. (b) Architecture of the proposed SRR (MUX: multiplexer). (c) Architecture of the multistage decimators.

residual interference from adjacent channels because the sampling rate is usually not an integer multiples of the channel spacing. Finally, an SRC is used to provide the necessary arbitrary rate-change factor so that the sampling rate of its output is suitable for the baseband processor, which usually operates on a multiple of the sampling rate of the baseband signal. Hence, it is now possible to accommodate signals with a wide variety of bandwidths using this architecture [1], [4]–[6]. A drawback of this structure is that the output of the multistage decimators, which is obtained by downsampling the high-rate IF signal from the ADC, has to be upsampled again (say by an L-band interpolated filter) in order to carry out the arbitrary sample rate conversion. Another important problem is the high complexity of the PFIR, since a considerable number of high-speed general-purpose multipliers are required for its implementation especially for wideband signals.

Recently, the authors have proposed a new digital IF architecture for SRRs as shown in Fig. 2 [7], [8]. Unlike conventional SRRs, the SRC, which is based on a Farrow-based variable digital filter (VDF) [9], is performed immediately after the multistage decimators so that the PFIR can be replaced by a half-band filter (HBF) with fixed coefficients, if the arbitrary downsampling ratio is properly chosen. This modification eliminates the need of the PFIR, which is usually the bottleneck for wideband signals. Moreover, instead of upsampling again the output of the multistage decimators as in the conventional receivers to achieve arbitrary dowsampling ratios [1], [4]–[6], the VDF is able to perform the same function since it can be designed to provide a variable fractional delay and additional attenuation in its passband and stopband, respectively. The performance of this SRR can be further improved by using a new second-order compensator, which compensates for the passband droop of the basic cascaded integrator-comb (CIC) filter. In [10], allpass-based low-pass anti-aliasing filters are employed to realize the multistage decimator in order to achieve a lower system delay and implementation complexity.

In this paper, the design, signal round-off analysis and complexity optimization of the proposed digital IF for SRR (or SRR for simplicity) are studied. Since the proposed SRR consists of digital filters with fixed coefficients except for the multipliers required to implement the Farrow-based SRC, it can be implemented using the canonical signed digit or sum-of-power-of-two (SOPOT) coefficients or [7], [11]–[13]. In addition, the redundancy in realizing the multiplications of these SOPOT coefficients can be significantly reduced by using the multiplier-block (MB) technique [14], which gives rise to minimum adder realization. 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. While most multiplier-less filter design and realization methods address the coefficient round-off problem by minimizing the number of SOPOT terms used, the proposed design methodology aims to minimize more realistic hardware complexity measure of the digital IF subject to a given spectral and accuracy specifications. The motivation behind this approach is that the complexity is closely related to the target output accuracy. A lower output accuracy means that a shorter internal wordlength, and hence complexity, can be employed. To this end, the multiplier-less digital filter with no signal round-off noise is first designed. The output accuracy of this digital filter is then specified statistically by its output noise power, which is generated by the rounding operations performed to the intermediate data. Using this signal round-off model, the internal wordlengths of all the intermediate data are determined. The hardware complexity measure to be minimized is the exact wordlengths being used for each intermediate data, which is related to the number of adder cells and/or registers used. Other more sophisticated models such as power consumption may be used instead. For simplicity, only the former measure is considered in this paper. In contrast to conventional approaches that minimize only the total number of SOPOT terms, the new criterion is more realistic and general for hardware implementation. While the random search algorithm in [8] and the mixed integer linear programming approach in [15] are very flexible methods, their design times are expected to increase considerably when large number of variables is involved. In principle, it is possible to optimize the SOPOT coefficients and the internal wordlengths simultaneously. However, the

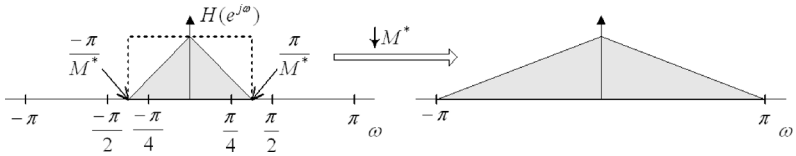


Fig. 3. Graphical illustration of arbitrary sampling rate conversion. $H(e^{j\omega})$: overall frequency response of the proposed SRR for $2 < M^* < 4$.

design complexity increases significantly. Because of these reasons, we propose two other novel and simple algorithms to address the wordlength determination problem and solve the two problems separately. It is shown that if the wordlengths are relaxed from integer- to real-valued quantities, then the problem can be solved using the Lagrange multiplier method [16] and a close-form solution can be obtained. A similar work, which was concerned with the real-time wordlength adaptation in adaptive FIR filters, can be found in [17]. An important limitation of these approaches is that the solution so obtained is usually not integer valued and hence it has to be rounded to the next larger integer. Fortunately, this can be used as an initial guess to the random search algorithm and the searching time is greatly reduced. By recognizing the close similarity between the wordlength determination problem and the bit allocation algorithm for data compression [18]–[20], we further propose an algorithm based on the marginal analysis method [19], [20]. The basic idea of the proposed algorithm is to increase the wordlength of one of the intermediate output points successively in order to lower the output round-off noise power as much as possible, until the given bit accuracy or bit budget (total wordlength or complexity) is met. Design results show that the proposed algorithm works well with large number of variables. Furthermore, when coupling the optimal solution obtained from the previous method with the bit-allocation algorithm, a near optimal solution can be obtained within several seconds in a Pentium 4 personal computer. It should be noted that the proposed algorithms are also applicable to the realization of other linear-time-invariant (LTI) systems, including the low-delay FIR and digital allpass filters-based SRR in [10]. The rest of this paper is organized as follows. Section II is devoted to the principle and design of the proposed SRR and its generalization to multiple receiving channels. Section III is devoted to the multiplier-less realization of the proposed SRR. Section IV presents the signal round-off and overflow analysis. Various algorithms for minimizing the internal wordlengths of the proposed SRR while satisfying the given specifications are described in Section V. This is then followed by a detailed design example and the field programmable gate array realization of a multi-standard receiver in Section VI. Finally, conclusions are drawn in Section VII.

II. Principle and Design of Proposed Digital IF Architecture

As mentioned earlier, the conventional receiver, which is shown in Fig. 1(b), uses multistage decimators, followed by a PFIR and a SRC. In [7], [8], and [10], a new digital IF architecture as shown in Fig. 2 is proposed. Its purpose is to extract a desired user channel with a given bandwidth and decimate it to a lower sampling rate for further processing by the DSP. Depending on the required downsampling ratio, the digitized IF-signal from the high-speed ADC will optionally be passed through a compensated CIC filter with a decimation factor of $M_{\rm CIC}$, which is a positive power-of-two integer, and an appropriate number of multistage decimators each with a decimation factor of two. Without loss of generality, we assume that our receiver consists of the compensated CIC filter with a maximum decimation factor of 16 and three-stage decimator so that they can support the signal bandwidths ranging from GSM to Hiperlan/2 standards (i.e., a downsampling ratio ranges from 4 to 295.3849, assuming that the digital IF signal is sampled at 80 M samples per second (sps), and the maximum downsampling ratio of the SRR is 512. A detailed example will be discussed in Section VI). The maximum downsampling ratio can be adjusted by increasing or decreasing the decimation factor of the CIC filter and the number of anti-aliasing filters. As an illustration, Fig. 2(c) shows a three-stage decimator consisting of three general LPFs denoted by LPF#1, LPF#2, and LPF#3. Then, the output of the multistage decimators is fed to a VDF-based SRC to provide the required fractional sample rate conversion. Finally, the output of the SRC is fed to an HBF, which is merely sufficient to reduce the residual interference while keeping the system delay and complexity as low as possible. Consequently, the overall downsampling ratio M^* of the proposed SRR is given by

$$M^* = M_{\rm CIC} \cdot M_{\rm SRC} \cdot 2^k \tag{2-1}$$

where $M_{\rm CIC}$ is the downsampling ratio of the compensated CIC filter; $M_{\rm SRC}$ is the arbitrary downsampling ratio of the SRC; and $k \in \{1, 2, 3, 4\}$ is an integer representing the number of the remaining 2-to-1 decimators to be selected. Fig. 3 shows an example of the operation of the proposed SRR for $2 < M^* < 4$. From (2-1), the downsampling ratio M^* can be achieved by choosing $M_{\rm CIC} = 1$, $k = 1$ and $1 < M_{\rm SRC} < 2$. In general, the VDF-based SRC is more complicated to design and realize than the other digital filters in the SRR. Therefore, it is preferable to perform the arbitrary sample rate conversion by the VDF-based SRC after the compensated CIC filter and the multistage decimators so that the sampling rate and hence the power consumption can be lowered.

A. Generalization for Receiving More Channels

The proposed architecture in Fig. 2 can be generalized to receive a set of adjacent, instead of one, user's channels. The basic idea is to employ an $M_{\rm DFT}$-channel over-sampled discrete Fourier transform (DFT) filter bank (FB) after the HBF, as shown in Fig. 4(a). The downsampling ratio of the FB is $M_{\rm DFT}/2$. More precisely, we treat $M_{\rm DFT}$ consecutive channels as a single channel and choose an appropriate sampling factor to remove the interference from other channels. After the HBF, the sampling rate will be $2M_{\rm DFT}f_s$, where f_s is the channel

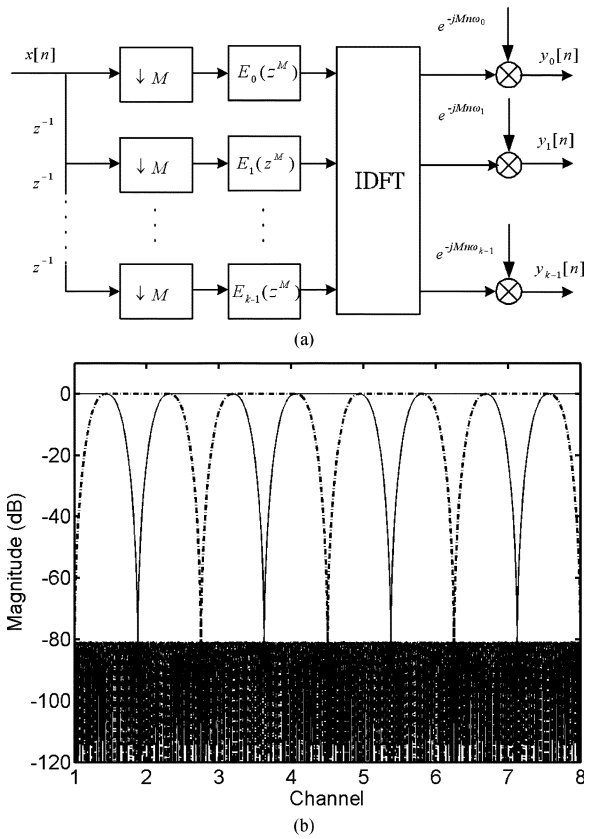


Fig. 4. (a) DFT filter bank channelizer with polyphase decomposition. (b) Frequency response of an 8-channel DFT filter bank.

spacing. By decimating the output by a factor of 2 and using an M_{DFT}-channel over-sampled DFT FB with a downsampling ratio of $M_{\mathrm{DFT}}/2$, the M_{DFT} users' channels can be isolated, each has a sampling rate of $2f_s$. In this paper, we shall only focus on its polyphase structure shown in Fig. 4(a). This can be viewed as the generalization of the software radio base-station first proposed in [21] for the digital advanced mobile phone system in that the programmable decimator and arbitrary SRC proposed above are employed to select the desired channels and convert it to an appropriate sampling rate for the M_{DFT}-channel DFT FB. In DFT FBs, each subband filter, $h_m[n]$, $m = 0, \ldots, M_{\mathrm{DFT}} - 1$, is obtained by modulating a low-pass prototype filter using the inverse DFT (IDFT). In order to avoid aliasing, the passband edge ω_p and stopband edge ω_s of the prototype filter should satisfy [22] $\omega_p = \pi/M_{\mathrm{DFT}}$ and $\omega_s \leq 2\pi/M_{\mathrm{DFT}}$, and the center frequency for the mth channel is given by $\omega_m = 2\pi m/M_{\mathrm{DFT}}$. As an example, Fig. 4(b) shows the frequency response of an 8 channel oversampled DFT FB with downsampling ratio of 4. It is modulated by a FIR linear-phase (LP) prototype with $\omega_p = \pi/8$ and $\omega_s = \pi/4$, which has a passband ripple of 0.00173 and stopband ripple of 0.0001. By cascading this 8-channel DFT FB with the SRR, up to eight consecutive channels can be extracted by the digital IF simultaneously. In what follows, the design of the various components of the proposed SRR will be briefly outlined. Their multiplier-less realization and finite wordlength effects will be described later in Sections III–V.

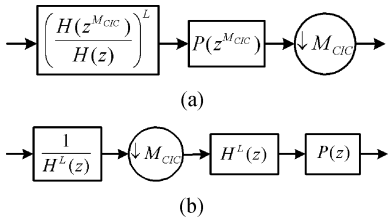


Fig. 5. Compensated CIC filter: (a) before and (b) after the application of the noble identity. (Note: the scaling of $(1/M_{\mathrm{CIC}})^L$ is not shown.)

B. Design of the Second-Order Cascaded Integrator-Comb (CIC) Compensator

The basic CIC filter [4] is commonly employed when a large downsampling ratio is required, because of its reasonable performance and low hardware complexity. However, the passband droop of the CIC filter significantly limits the quality of anti-aliasing filters, if the decimation ratio is small [23]. In general, the transfer function of the CIC filter is given by

$$H_{\mathrm{CIC}}(z) = \left[\frac{H_C(z^{M_{\mathrm{CIC}}})}{M_{\mathrm{CIC}} \cdot H_C(z)}\right]^L \tag{2-2}$$

where $H_C(z) = 1 - z^{-1}$; and L is the number of CIC stages. The sharpened CIC (SCIC) filter [6] and the interpolated second-order polynomial (ISOP) [5] were proposed to improve the passband droop of the basic CIC filters. Although the performance of the simple ISOP is inferior to that of the SCIC, it is usually sufficient for most applications. On the other hand, its implementation complexity is significantly lower than SCIC due to its simple structure. A drawback of the ISOP is the rather high dynamic range of the filter coefficients and its long delay chain. In [7], [8], and [10], the following second-order CIC compensator is proposed:

$$P(z) = a + bz^{-1} + az^{-2} \tag{2-3}$$

where a and b are real-valued constants to be determined. As shown in Fig. 5(a), it is placed after the CIC filter. Note that $P(z)$ is chosen to be linear-phase so as to avoid any phase distortion and reduce the implementation complexity. Given the frequency response of the CIC filter in (2-2), the coefficients a and b can be readily determined using the Parks-McClellan algorithm. It was shown that the compensated CIC filter has lower dynamic range of filter coefficients and shorter delay chain than the ISOP because of the increased flexibility provided by a general linear-phase filter. On the other hand, the hardware complexity is still very low thanks to the use of SOPOT coefficients and MB technique, which will be discussed later in Section III. Table I summarizes the SOPOT coefficients of the CIC compensator. Note that only two adders are required to implement the multiplications with a and b. Furthermore, using the noble identity [24], the compensated CIC filter in Fig. 5(a) can be implemented more efficiently as shown in Fig. 5(b).

C. Design of Low-Pass Anti-Aliasing Filters

In this subsection, we shall consider the design of the low-pass anti-aliasing filters in the multistage decimators shown in

TABLE I
SOPOT COEFFICIENTS AND ADDER COMPLEXITY OF SECOND-ORDER CIC COMPENSATOR

Coefficients of CIC compensator	SOPOT Representation
a	$-2^{-4}-2^{-5}$
b	$2^{0}+2^{-3}+2^{-4}$
Adders used before MB	3
Adders used after MB	2

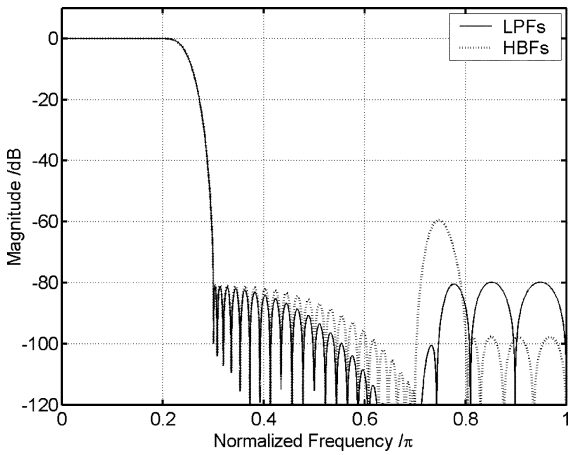


Fig. 6. Comparison between multistage decimators using LPFs (solid line) and HBFs (dotted line) for a decimation ratio of 4.

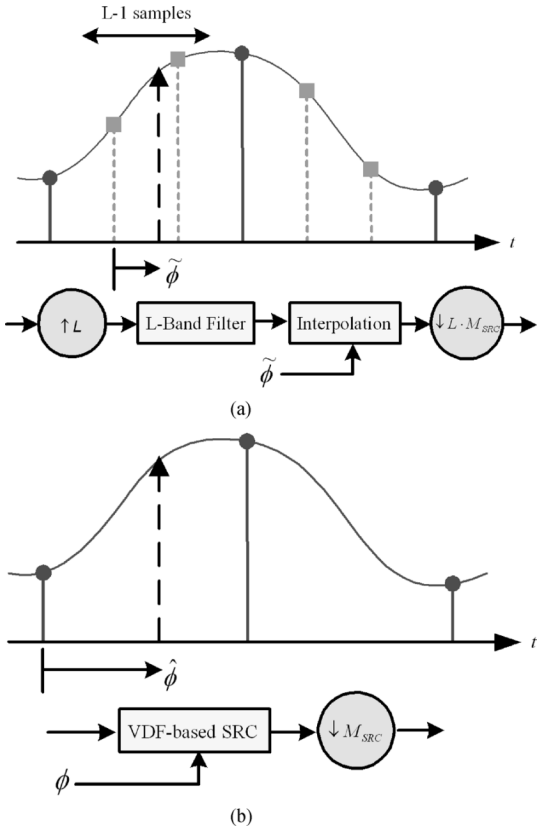


Fig. 7. SRC (a) using L-band filter followed by a simple interpolation (Lagrange or cubic spline) (b) using an VDF.

Fig. 2(c). For the sake of presentation, a multistage decimator with a decimation factor of four is designed as an example. The overall passband and stopband edges of the decimation filter are chosen as 0.2π and 0.3π respectively, and the desired stopband attenuation is 80 dB. The multistage decimator is divided into two sub-stages and each sub-stage decimates the input signal by a factor of two. Although the given specification can be implemented by a single FIR filter, the multistage implementation usually requires much lower complexity [24]. In [6], all the filters in the multistage decimator are chosen as LP HBFs. The passband and stopband edges of the first and second HBFs are $(\omega_p = 0.2\pi, \omega_s = 0.8\pi)$ and $(\omega_p = 0.4\pi, \omega_s = 0.6\pi)$, respectively. Note, due to the structural constraints of the HBF, ω_s has to be equal to $\pi - \omega_p$. This will limit the performance of the decimation, as we can see from the dotted line in Fig. 6. The stopband attenuation of the multistage decimator designed using these HBFs has a peak of about 60 dB between 0.7π and 0.8π because the transition band of the first HBF coincides with one of the transition bands of the interpolated frequency response of the second HBF. Therefore, it cannot attenuate the transition band of the second HBF to the desired stopband attenuation. To avoid this problem, one may slightly modify the specification of the first HBF to $\omega_p = 0.3\pi$ and $\omega_s = 0.7\pi$ since the transition band of the first aliased folding in the second HBF starts at 0.7π. This will however increase the system delay and complexity of the multistage decimators. Another possibility is to employ general linear-phase LPFs. To satisfy the given stopband attenuation, their stopband edges should start at the transition band of the first aliased folding of the previous filters. As a result, the passband and stopband edges of the first LPF should be 0.2π and 0.7π, respectively. The frequency response of the multistage decimator designed using this LPF is shown as the solid line in Fig. 6. It can be seen that the stopband attenuation of 80 dB is now achieved. In general, let $\omega_p^{(i)}$ and $\omega_s^{(i)}$ be, respectively, the passband and stopband edges of the ith anti-aliasing filter, relative to its input sampling rate $F_{s-in}^{(i)} = 2\pi$, and $M^{(i)}$ be the downsampling ratio of the ith decimator. Then the ith anti-aliasing filter satisfies the following inequalities:

$$\omega_p^{(i)} > \frac{\widehat{\omega}_p^{(i-1)}}{M^{(i)}} \quad \text{and} \quad \omega_s^{(i)} < \frac{\left(2\pi - \widehat{\omega}_s^{(i-1)}\right)}{M^{(i)}} \tag{2-4}$$

where $\widehat{\omega}_p^{(i-1)}$ and $\widehat{\omega}_s^{(i-1)}$ are the overall passband and stopband edges of previous $i-1$ digital filters. In this work, general LPFs will be employed in the multistage decimators to improve the flexibility and hence the system performance. Moreover, to attenuate the aliasing component around π, even-length filters are used so that a zero is imposed at $\omega = \pi$ for all the anti-aliasing filters. The filter coefficients can be readily determined using the Parks-McClellan algorithm. Their multiplier-less realization will be discussed later in Section III. Next, we shall consider the design and implementation of the SRC.

D. Design of SRC

The design of programmable SRCs with arbitrary conversion factors was studied in detail by Ramstad [24]. In general, there are two approaches to implement a SRC with different tradeoff between the sampling rate and the hardware complexity. One is to employ the structure in Fig. 7(a) where the input signal is

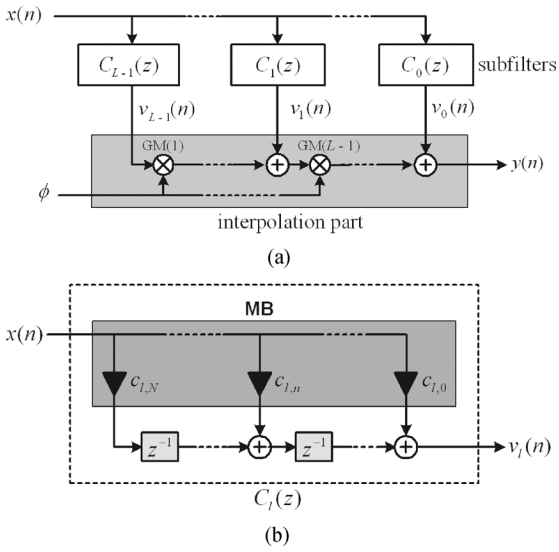


Fig. 8. Implementation of VDF. (a) Farrow structure based and (b) transposed form of FIR subfilters.

first up-sampled by a factor of L by inserting $L-1$ zeros between successive time samples. This creates $L-1$ images in the frequency domain, which are then removed by an L-band interpolated filter with spectral support from $-\pi/L$ to π/L. If L is sufficiently large, further interpolation with an irrational downsampling ratio can be achieved simply by a low-order interpolation such as Lagrange interpolation [26] or cubic spline [27]. One drawback of employing this structure is that the output of the multistage decimators, which is obtained by downsampling the high-rate IF signal from the ADC, has to be upsampled again by the L-band filter. Alternatively, the functions of the L-band filter and the interpolator can be implemented using a VDF [28], [29] with a control parameter ϕ as in Fig. 7(b). For modest downsampling ratios, the VDF-based SRC is more efficient than the structure in Fig. 7(a) because its coefficients can be jointly optimized to fulfill the given spectral and fractional-delay specifications. In the proposed SRR, M_{SRC} is chosen to lie between 1 and 2. This leads to a better performance without having to increase the sampling rate as in the L-band filter approach. As a result, the operating rate of the multistage decimators can be significantly lower to reduce the power consumption. With this choice of M_{SRC}, the overall downsampling factor M^* of the SRR is greater than 2 because the output of the SRC must be fed to the HBF. The proposed VDF-based SRC has the following ideal frequency response:

$$H_d(e^{j\omega},\phi)=\begin{cases} e^{-j\omega\tau(\phi)}, & 0\le|\omega|\le\omega_p \\ 0, & \omega_s\le|\omega|\le\pi \end{cases} \tag{2-5}$$

where $\tau(\phi)=D+\phi$, $\phi\in[-0.5,0.5]$, ω_p and ω_s are the group delay, the passband and stopband edges of the VDF, respectively. One advantage of employing VDF is that it can be implemented efficiently using the Farrow's structure [9] as shown in Fig. 8. It consists of a set of subfilters $C_l(z)$ followed by the multiplications with the appropriate powers of the parameter ϕ.

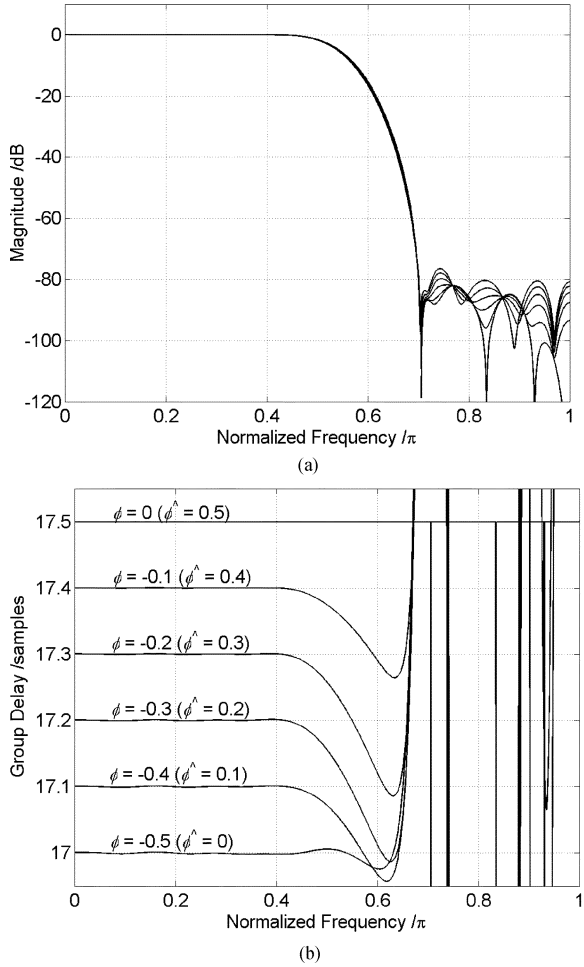


Fig. 9. (a) Frequency responses and (b) the corresponding group delays of the VDF-based SRC with $\phi=\{-0.5,-0.4,-0.3,-0.2,-0.1,0\}$.

More precisely, the transfer function of a VDF can be expressed as follows:

$$H(z,\phi)=\sum_{l=0}^{L-1}\left[\sum_{n=0}^{N-1}c_{l,n}z^{-n}\right]\phi^l=\sum_{l=0}^{L-1}C_l(z)\phi^l. \tag{2-6}$$

This allows us to compute the required samples at fractional sampling intervals by tuning a single parameter ϕ, which in turn provides the required arbitrary sampling rate conversion. For the design of VDFs, interested readers are referred to [28] and [29]. As an example, Fig. 9 shows the frequency responses of the VDF-based SRC with the passband and stopband cutoff frequencies respectively given by 0.4π and 0.7π. This Farrow-based VDF can be efficiently implemented using the method in [12]. More precisely, all the subfilters are redrawn in their transposed forms so that the input will be multiplied directly with all the constant filter coefficients. Consequently, by making use of SOPOT coefficients and the MB technique [14], the total number of additions can be kept to be minimal by reusing the immediate results generated. Finally, it was found that the symmetric or anti-symmetric impulse responses of the subfilters significantly decrease the hardware complexity.

TABLE II
DESIGN RESULTS OF LPFS, VDF AND HBF

	LPF#1	LPF#2	LPF#3	VDF	HBF
Passband edge $\omega_p^{(i)}$	$0.05\,\pi$	$0.1\,\pi$	$0.2\,\pi$	$0.4\,\pi$	$0.4\,\pi$
Stopband edge $\omega_s^{(i)}$	$0.925\,\pi$	$0.85\,\pi$	$0.7\,\pi$	$0.7\,\pi$	$0.6\,\pi$
Filter Length (N)	8	12	18	36	48
Number of subfilters (L)	N/A	N/A	N/A	4	N/A
Maximum wordlength in fractional part (bits)	14	14	16	16	16
Passband deviation (in dB)	0.00113	0.00117	0.00248	0.00898	0.0023
Stopband attenuation (in dB)	88.68	90.13	77	76.48	81.95
Fractional delay error (in dB)	N/A	N/A	N/A	54.33	N/A
Average SOPOT terms per coefficient	4.5	3.67	3.67	3.29	3.29
Adders used for SOPOT coefficients before MB	14	16	22	145	52
Adders used for SOPOT coefficients after MB	9	9	14	54	23

III. MULTIPLIER-LESS REALIZATION OF SRR

In this section, the multiplier-less realization of the proposed SRR will be described. As mentioned earlier, the constant coefficients in the CIC compensator, LPFs, HBF, and the subfilters of the VDF can be efficiently implemented as limited number of shifts and additions by employing the SOPOT representations [11]–[13] as follows:

$$\hat{h}(n) = \sum_{r=1}^{R} u_r 2^{a_r} \tag{3-1}$$

where $u_r \in \{-1, 1\}$ and $a_r \in \{-l_b, \ldots, -1, 0, 1, \ldots, u_b\}$; l_b and u_b are positive integers and their values determine the dynamic range of the coefficients; R is the number of terms used in the coefficient approximation. To further reduce the implementation complexity, the MB technique proposed in [14] is also employed. The basic idea of MB is to reduce the redundancies in implementing all the SOPOT coefficients by removing any possible common sub-expressions in their representations. We now briefly describe how the SOPOT coefficients of the proposed SRR can be determined. For example, given the real-valued coefficients $c_{l,n}$ in the subfilters of the VDF-based SRC, the corresponding SOPOT coefficients can be obtained by a number of methods [7], [10]–[13], [30]. Here, we shall employ the random search algorithm reported in [7] because different types of constraints can be easily incorporated. The objective function to be minimized can be written as follows:

$$\min \langle T_{\mathrm{SOPOT}} \rangle \text{ s.t. } \begin{cases} \delta_p < \delta_{p-\max} \\ \delta_s < \delta_{s-\max} \\ \delta_d < \delta_{d-\max} \end{cases}, \phi = [-0.5, 0.5] \tag{3-2}$$

where

$$\delta_p = \max_{0<|\omega|<\omega_p} \left\langle \left| \hat{H}(e^{j\omega}, \phi) - H_d(e^{j\omega}, \phi) \right| \right\rangle$$

is the passband peak ripple error;

$$\delta_s = \max_{\omega_s<|\omega|<\pi} \left\langle \left| \hat{H}(e^{j\omega}, \phi) - H_d(e^{j\omega}, \phi) \right| \right\rangle$$

is the stopband peak ripple error;

$$\delta_d = \max_{0<|\omega|<\omega_p} \langle |\tau(\phi) - \hat{\tau}(\phi)| \rangle$$

is the fractional-delay peak ripple error; T_{SOPOT} is the total number of SOPOT terms used to implement all the SOPOT coefficients; $H_d(e^{j\omega}, \phi)$ and $\tau(\phi)$ are, respectively, the desired frequency response and group delay, which are defined in (2-6) ; $\hat{H}(e^{j\omega}, \phi)$ and $\hat{\tau}(\phi)$ are, respectively, the frequency response and group delay of the multiplier-less VDF; $\delta_{p-\max}$, $\delta_{s-\max}$ and $\delta_{d-\max}$ are the maximum tolerance of the passband, stopband, and fractional-delay peak ripple errors, respectively. In the random search algorithm, the given real-valued coefficients $c_{l,n}$ are first obtained by the WLS approach proposed in [28] and [29]. Let $\boldsymbol{X}$ be the vector containing these real-valued coefficients. Then, the algorithm repetitively calculates a candidate SOPOT vector $\boldsymbol{X}_c$ given by

$$\boldsymbol{X_c} = [\boldsymbol{X} + \lambda \boldsymbol{X_p}]_{\mathrm{SOPOT}} \tag{3-3}$$

where $\boldsymbol{X_p}$ is a random vector with elements chosen in the range ± 1. λ is a user-defined variable used to control the size of the neighborhood to be searched, and $[\cdot]_{\mathrm{SOPOT}}$ is the rounding operator that converts every element inside the input vector to its closest SOPOT value bounded between u_b and l_b. The performance measures δ_p, δ_s and δ_d of the new SOPOT coefficients are then calculated. The set that yields the minimum total number of SOPOT terms, while satisfying the given specifications and the wordlength constraints, is declared as the optimum solution. The SOPOT coefficients for the other components, namely HBF, LPFs, and CIC compensator can be determined by the same approach. Table II summarizes

TABLE III
SOPOT COEFFICIENTS OF THE LPF#1 (Filter Length $= 8, h(n) = h(7-n)$)

$h(n)$	SOPOT Representation	$h(n)$	SOPOT Representation
$h(0)$	$-2^{-6}-2^{-8}-2^{-11}-2^{-13}-2^{-14}$	$h(2)$	$2^{-3}+2^{-7}+2^{-8}+2^{-10}+2^{-12}$
$h(1)$	$-2^{-5}+2^{-8}-2^{-13}$	$h(3)$	$2^{-2}+2^{-3}+2^{-5}+2^{-8}-2^{-11}$

TABLE IV
SOPOT COEFFICIENTS OF THE LPF#2 (Filter Length $= 12, h(n) = h(11-n)$)

$h(n)$	SOPOT Representation	$h(n)$	SOPOT Representation	$h(n)$	SOPOT Representation
$h(0)$	$2^{-8}+2^{-11}+2^{-13}$	$h(2)$	$-2^{-5}-2^{-9}-2^{-14}$	$h(4)$	$2^{-3}+2^{-6}+2^{-8}+2^{-9}+2^{-13}$
$h(1)$	$2^{-8}+2^{-12}$	$h(3)$	$-2^{-5}-2^{-7}-2^{-11}-2^{-13}$	$h(5)$	$2^{-1}-2^{-4}-2^{-6}-2^{-8}-2^{-12}$

TABLE V
SOPOT COEFFICIENTS OF THE LPF#3 (Filter Length $= 18, h(n) = h(17-n)$)

$h(n)$	SOPOT Representation	$h(n)$	SOPOT Representation	$h(n)$	SOPOT Representation
$h(0)$	$-2^{-10}-2^{-12}-2^{-13}$	$h(3)$	$2^{-6}+2^{-10}+2^{-11}-2^{-14}$	$h(6)$	$-2^{-5}+2^{-8}-2^{-15}$
$h(1)$	$-2^{-9}-2^{-11}-2^{-12}$	$h(4)$	$-2^{-8}-2^{-10}-2^{-13}$	$h(7)$	$2^{-3}+2^{-5}+2^{-6}+2^{-9}+2^{-14}$
$h(2)$	$2^{-8}+2^{-10}+2^{-12}+2^{-14}+2^{-16}$	$h(5)$	$-2^{-4}+2^{-8}+2^{-11}$	$h(8)$	$2^{-2}+2^{-3}+2^{-6}+2^{-7}$

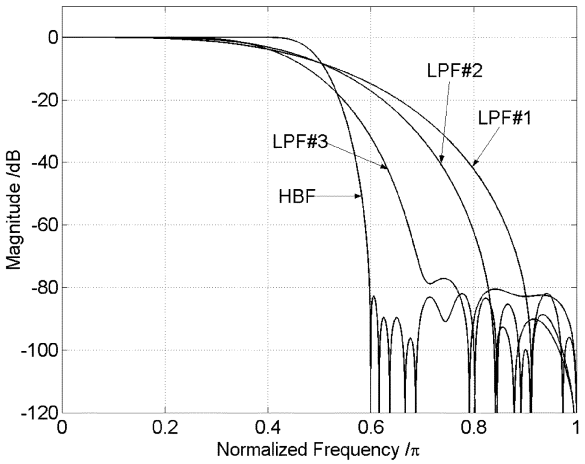


Fig. 10. Frequency responses of LPF#1, LPF#2, LPF#3, and HBF.

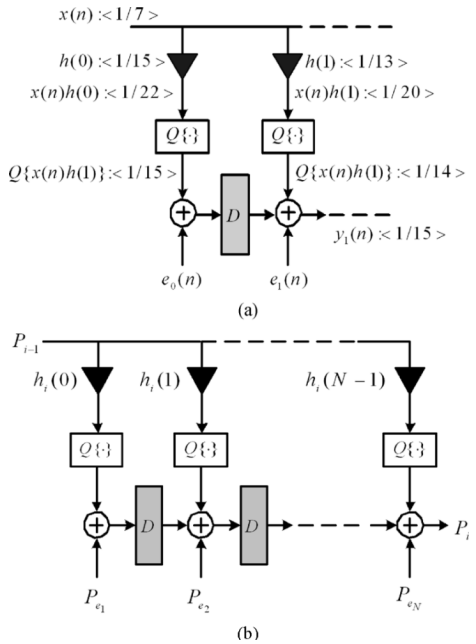


Fig. 11. Transposed form implementation of a typical FIR digital filter with (a) round-off noise model due to the finite wordlength effect, (b) noise power being modeled as uncorrelated white noise sources. D: register; $Q\{\cdot\}$: rounding operator; $e_k(n)$: rounding noise; P_{e_k}: rounding noise power; P_i: total output noise power at ith stage.

the specifications and performances before and after SOPOT optimization for the components of the proposed SRR. The corresponding SOPOT coefficients are listed in Tables III–V except those for the VDF-based SRC and the HBF because of page limitation. Figs. 9 and 10 show the corresponding frequency responses. It can be seen that they possess good frequency characteristics while achieving a low implementation complexity. In fact, there is a tradeoff between the filter performance and the hardware complexity. If the lower bound l_b is sufficiently large, the performance of the multiplier-less filters will be close to their real-valued counterpart, at the expense of increased hardware complexity.

To implement this multiplier-less SRR using the MB technique, all FIR filters or subfilters are implemented in their transposed form as shown in Fig. 8(b). Instead of passing the input signal $x(n)$ through the delay chain as in the direct form implementation, it is now multiplied with a large number of constant coefficients in SOPOT form before adding the products together. Therefore, the redundant additions in these SOPOT products can be removed by a MB, which greatly reduces the arithmetic complexity. In principle, it is possible to remove all the redundancy found in the SOPOT coefficients leading to a minimum adder realization. This can drastically reduce the number of adders required for realizing the SRR. Interested readers are referred to [14] for more details on the generation of the MB. Similarly, the above multiplier-less realization approach can be applied to the DFT-FB-based channelizer discussed in Section II using the technique proposed in [31] and [32]. However, details are omitted due to page limitation. We now present the signal round-off model for the SRR. The problem of wordlength determination using this model will be given later in Section V.

IV. SIGNAL ROUND-OFF AND OVERFLOW ANALYSES

A. Analysis of Signal Round-Off Noise

Signal round-off errors occur due to rounding of the intermediate signal after multiplications. Since the exact round-off errors are difficult to analyze, they are usually treated as uncorrelated white noises [33]. For rounding operations, the quantization noise will have a zero mean with a variance σ equal to $\Delta^2/12$, where Δ is the quantization step-size. In other words, the variance is determined by the number of fractional bits that

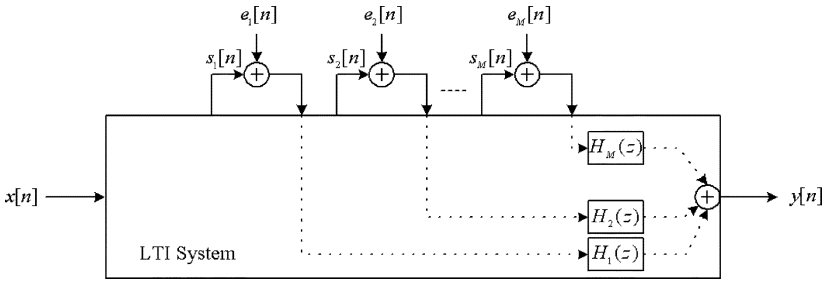


Fig. 12. Variation in dc offset at $V_{C1} - V_{C2}$ at both left and right cochleae. This offset is a major source of offset in spike rates.

is retained after multiplication. Let's consider an example in Fig. 11(a), where a digital FIR filter with impulse response $h(n)$ is implemented in the transposed form. Assume that the filter coefficients are represented as SOPOT coefficients and are simultaneously realized using the MB technique. Hence, the maximum wordlengths required for the products $x(n) \cdot h(n)$ can be determined. To minimize the hardware complexity, these products may be rounded using the signal round-off operator $Q\{\cdot\}$. In fixed-point arithmetic, each intermediate signal can be represented in the form of $\langle I/F \rangle$, where I is the number of integer bits including the sign bit and F is the number of fractional bits. In general, if F bits are rounded to B bits, where $B < F$, then the noise variance P is given by

$$P = \frac{\Delta^2}{12} = \frac{2^{-2B}}{3} \tag{4-1}$$

where $\Delta = 2^{-(B-1)}$. More generally, consider the round-off noise model of the LTI system in Fig. 12, where the signals to be quantized are $s_m[n]$ for $m = 1, \ldots, M$; M is the total number of rounding sources. From (4-1), if $s_m[n]$ is rounded to b_m bits, then the variance of the quantization error, $e_m[n]$, is given by $2^{-2b_m}/3$. Let the transfer function from $s_m[n]$ to the output $y[n]$ be $G_m(\omega)$, $m = 1, \ldots, M$. Furthermore, we assume that the noise sources are uncorrelated. Hence, the variance of the output noise at $y[n]$ can be expressed as follows:

$$\sigma_e^2 = \sum_{m=1}^{M} c_m \sigma_m^2 = \frac{1}{3} \sum_{m=1}^{M} c_m 2^{-2b_m} \tag{4-2}$$

where $c_m = (1/2\pi) \int_{-\pi}^{\pi} |G_m(\omega)|^2 d\omega = \sum_k g_m^2[k]$; $G_m(\omega)$ is the transfer function from $s_m[n]$ to $y(n)$; and $g_m[k]$ is the impulse response corresponding to $G_m(\omega)$. Returning to the proposed SRR, Fig. 11(b) shows the noise power model of the ith stage of the SRR. If there are N_i such rounding processes at the ith stage, then the total noise power $P^{(i)}$ due to these rounding sources is simply given by

$$P^{(i)} = \sum_{k=1}^{N_i} P_{e_k}. \tag{4-3}$$

The total output noise power at the ith stage, P_i, taking into account noise sources at previous stages is

$$P_i = P_{i-1} \cdot \sum_{n=0}^{N_i - 1} |h_i(n)|^2 + P^{(i)} \tag{4-4}$$

where $h_i(n)$ is the impulse response of the digital filter in the current stage, which is assumed to have a filter length of N_i. The output accuracy A_i at the ith stage, in terms of the number of fractional bits, is therefore approximately given by [33]

$$A_i \approx \left| \frac{10 \cdot \log_{10}(P_i)}{6} \right| \text{ bits.} \tag{4-5}$$

It should be noted that the larger the number of noise sources, the lower will be the output accuracy. The noise power can however be reduced by increasing the internal wordlengths for the fractional bits at different stages of the SRR, at the expense of increased hardware complexity. Next, we shall consider the signal overflow effects.

B. 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. In FIR filters, it is possible to determine whether signal overflow will occur at a particular adder using the L1 scaling measure. More precisely, the input signal $x[n]$ is assumed to take on its maximum value denoted by $x_{\max}$. Then, the maximum value after implementing the kth impulse response coefficient of the target system is bounded by

$$y_{\max,k} = x_{\max} \sum_{n=0}^{k} |h[n]|. \tag{4-6}$$

Using (4-6), 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 overflows. However, there is still a small probability that overflows will occur. To determine this option, we can imagine that a noise is generated by the rounding option and the minimum acceptable wordlength is then determined as if it was a rounding source due to multiplication. If the minimum wordlength obtained is larger than the existing wordlength, then the wordlength has to be increased. Otherwise, rounding can be performed if the additional noise generated does not violate the prescribed accuracy. For IIR filters, scaling is usually performed at certain stages of

the system to avoid overflow [33]. Since scaling is a multiplication operation, it can be treated similarly by our model.

V. Wordlength Determination

In this section, the problem of minimizing the hardware complexity of the SRR subject to a prescribed output accuracy is studied. Since the number of adder cells and/or registers is usually the major hardware resources, they are employed as the measure of the hardware complexity. Other measures can also be used with slight modification of these algorithms. From Section IV, we know that the number of adder cells and registers is related to the exact wordlengths being used for each intermediate data. Therefore, the internal wordlengths of each intermediate data are the variables to be optimized. In general, the determination of the internal wordlength can be done in three steps. First of all, the real-valued coefficients of the SRR are designed as detailed in Section II. They are then converted into SOPOT coefficients using the random search algorithm and are implemented using the MB technique as mentioned in Section III. After that, the output accuracy of the SRR is specified statistically by its output noise power. It is assumed to be generated from the rounding operations performed, which depend on the formats of the internal wordlengths as described in Section IV-A. In what fallows, we shall propose three algorithms to determine these internal wordlengths. In our proposed IF architecture, all the filters are FIR filters except for the CIC filter, which is implemented as an integrator. The wordlengths of the basic CIC filter without any round-off and overflow errors will be treated separately in Section V-D.

A. Analytic Solution

The problem of determining the wordlengths for a given output noise power $\sigma_e^2 = P_{\text{spec}}$ can be formulated as the following constrained optimization problem:

$$\min_{\boldsymbol{b}} \sum_{m=1}^{M} w_m b_m = \boldsymbol{w}^T \boldsymbol{b} \ \text{s.t.} \ \frac{1}{3}\sum_{m=1}^{M} c_m 2^{-2b_m} = P_{\text{spec}} \quad (5\text{-}1)$$

where $\boldsymbol{w}$ is a constant weight vector, $\boldsymbol{b}$ is the variable vector representing the fractional part of the internal wordlengths to be determined. In most cases, w_m are chosen as one for all m. If we allow $\boldsymbol{b}$ to take on real values, instead of integer values, then the minimization problem in (5-1) can be solved analytically using the method of Lagrange multiplier [16]. Define the following Lagrangean function:

$$L(\boldsymbol{b}, \lambda) = \boldsymbol{w}^T \boldsymbol{b} + \lambda \left(\frac{1}{3}\sum_{m=1}^{M} c_m 2^{-2b_m} - P_{\text{spec}} \right) \quad (5\text{-}2)$$

where λ is the Lagrange multiplier associated with $\boldsymbol{b}$. Taking the partial derivatives and setting them to zero yields

$$\frac{\partial L(\boldsymbol{b}, \lambda)}{\partial b_m} = w_m - \frac{2\lambda c_m}{3} \cdot 2^{-2b_m} \cdot \ln 2 = 0. \quad (5\text{-}3)$$

From which, we obtain λ as follows:

$$\lambda = \frac{3w_m}{2c_m \ln 2} 2^{2b_m}, \qquad m = 1, \ldots, M. \quad (5\text{-}4)$$

Equating the left hand side of (5-4) for $m = 1$, one gets

$$\frac{3w_1}{2c_1 \ln 2} 2^{2b_1} = \frac{3w_m}{2c_m \ln 2} 2^{2b_m} \Leftrightarrow \left(\frac{c_m}{c_1}\right)\left(\frac{w_1}{w_m}\right) = 2^{2(b_m - b_1)}$$

and after slight manipulation, it gives

$$b_m = b_1 + \frac{1}{2}\log\left(\frac{w_1 c_m}{w_m c_1}\right), \qquad m = 1, \ldots, M. \quad (5\text{-}5)$$

Substituting (5-5) into the equality $(1/3)\sum_{m=1}^{M} c_m 2^{-2b_m} = P_{\text{spec}}$ enables us to solve for the optimal value of b_1

$$P_{\text{spec}} = \frac{1}{3}\sum_{m=1}^{M} c_m 2^{-2b_1} \frac{w_m c_1}{w_1 c_m} = \frac{c_1 \cdot 2^{-2b_1}}{3w_1} \sum_{m=1}^{M} w_m$$
$$b_1^{\text{opt}} = \frac{1}{2}\log_2\left(\frac{c_1}{3w_1 P_{\text{spec}}} \sum_{k=1}^{M} w_k\right). \quad (5\text{-}6)$$

Finally, we have

$$b_m^{\text{opt}} = \frac{1}{2}\log_2\left(\frac{c_{1,M}}{3w_1 P_{\text{spec}}} \sum_{k=1}^{M} w_k\right) + \frac{1}{2}\log_2\left(\frac{w_1 c_m}{w_m c_1}\right) \quad (5\text{-}7)$$

$m = 1, \ldots, M$, and after slight manipulation the desired result

$$b_m^{\text{opt}} = \frac{1}{2}\log_2\left(\frac{c_m}{3w_m P_{\text{spec}}} \sum_{k=1}^{M} w_k\right), \qquad m = 1, \ldots, M. \quad (5\text{-}8)$$

Alternatively, we can minimize σ_e^2 subject to a given bit budget: $\sum_{m=1}^{M} w_m b_m = B_{\text{spec}}$. The design problem becomes

$$\min_{\boldsymbol{b}} \frac{1}{3}\sum_{m=1}^{M} c_m 2^{-2b_m} \ \text{s.t.} \ \sum_{m=1}^{M} w_m b_m = B_{\text{spec}}. \quad (5\text{-}9)$$

Using again the Lagrange multiplier method, the optimal solution of b_m is found to be

$$b_m^{\text{opt}} = \frac{1}{2}\log_2\left(\frac{c_m}{w_m}\right) + \frac{B_{\text{spec}} - \sum_{k=1}^{M} \left(\frac{w_k}{2}\right)\log_2\left(\frac{c_k}{w_k}\right)}{\sum_{k=1}^{M} w_k}, \qquad m = 1, \ldots, M. \quad (5\text{-}10)$$

A possible problem with the analytical formula above for the wordlength is that b_m are real valued. To obtain an integer solution, they need to be rounded to the next largest integers. Moreover, for extremely low bit budget or large target variance, b_m can even become negative. On the other hand, for high bit budget

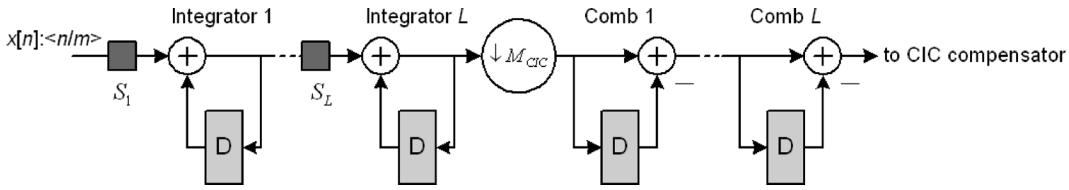


Fig. 13. Architecture of an L-stage cascaded integrator-comb (CIC) decimation filter. S_k: programmable shifter.

TABLE VI
MINIMIZATION OF $\boldsymbol{w}^T\boldsymbol{b}$ SUBJECT TO PRESCRIBED NOISE POWER P_{spec}

$b_m = 0$; (or $b_m = floor(b_m^{opt})$; // from (5-8))
while ($\frac{1}{3}\sum_{m=1}^{M} c_m 2^{-2b_m} > P_{spec}$) {
compute $k = \arg\max_m \xi_m$, where $\xi_m = \left.\frac{\Delta D}{\Delta B}\right|_m = \left|\frac{c_m(2^{-2(b_m+1)} - 2^{-2b_m})}{w_m}\right|$;
$b_k \leftarrow b_k + 1$;}

TABLE VII
MINIMIZATION OF σ_e^2 SUBJECT TO PRESCRIBED BIT BUDGET B_{spec}

$b_m = 0$; (or $b_m = floor(b_m^{opt})$; // from (5-10))
while ($\sum_{m=1}^{M} w_m b_m < B_{spec}$) {
compute $k = \arg\max_m \xi_m$, where $\xi_m = \left.\frac{\Delta D}{\Delta B}\right|_m = \left|\frac{c_m(2^{-2(b_m+1)} - 2^{-2b_m})}{w_m}\right|$;
$b_k \leftarrow b_k + 1$;}

or small target variance, the problem is less serious and the solution so obtained is more accurate.

B. Random Search Algorithm

Similar to the random search algorithm proposed in [8], the minimization problem in (5-1) can be formulated as follows:

$$\min_{(\boldsymbol{b},\boldsymbol{b_f})} C(\boldsymbol{b},\boldsymbol{b_f}) \text{ s.t. } \sigma_e^2 \le P_{\text{spec}} \tag{5-11}$$

where $C(\boldsymbol{b},\boldsymbol{b_f})$ is a measure of hardware cost, say the total number of adder cells and $\boldsymbol{b_f}$ is a reference vector, which stores the maximum wordlengths of the intermediate data [e.g., $x(n) \cdot h_i(n)$ in Fig. 11(a)]. The latter provides a rounding option to either retain the fractional part for each scaled output or reduce it by an appropriate value depending on the required output accuracy. More precisely, the basic idea is to search for the vectors $(\boldsymbol{b},\boldsymbol{b_f})$ in the neighborhood of their full precision values, i.e., the values without rounding. Our goal is to minimize the internal wordlengths of each intermediate data as specified by $\boldsymbol{b}$ and $\boldsymbol{b_f}$ so that $C(\boldsymbol{b},\boldsymbol{b_f})$ is minimized subject to the given specifications. The one with the minimum $C(\boldsymbol{b},\boldsymbol{b_f})$ is declared as the solution of this problem. Similarly, the above formulation can be modified to handle the problem of maximizing the output bit accuracy with a given bit budget. There are several advantages of this algorithm: 1) it is applicable to problems with very complicated inequality constraints, as illustrated in this work; 2) the time to obtain a high quality solution is manageable in nowadays computers, especially when an initial solution is available by some means. For instance, one may use the solution obtained in Section V-A to speed up the searching process; 3) It is possible to combine this searching process with the SOPOT determination, but the computational time will be greatly increased. For simplicity, the two processes are performed separately in this work. Finally, the solution is integer valued.

C. Bit Allocation Algorithm

It is interesting to note that the above analysis in Section V-A is similar to a classical problem in signal compression, known as bit allocation problem. There are in general two different approaches to solve this problem, namely the discrete Lagrange multiplier method [18] and the Marginal Analysis method [19], [20]. Next, we shall extend the latter to solve the wordlength determination problem. The first problem we address below is to minimize $\boldsymbol{w}^T\boldsymbol{b}$ subject to a given noise power P_{spec}. The variable b_m is first initialized to zero. Then the algorithm allocates one bit to one of b_m 's until the target noise power is met. In each step, the one with the largest reduction in output noise power is selected and its wordlength will be increased by one bit. The pseudo code of this algorithm is summarized in Table VI. Note that b_m 's are both non-negative and integer valued. A similar algorithm for minimizing σ_e^2 subject to a given bit budget B_{spec} can be derived as in Table VII. Again, b_m are both non-negative and integer valued. For multiplier-less realization using SOPOT coefficients, one can easily compute the wordlength required to achieve a given output error variance. Once it is determined, the exact rounding operation at each node can be determined and hence the complexity of the adders and registers can be determined exactly. The overflow prevention can also be determined according to Section IV-B if the maximum input format is known. Finally, the algorithms described in this section can be combined to shorten the search time, as we shall illustrate by an example in Section IV.

D. CIC Filters

In this subsection, the wordlength determination of the basic CIC filter will be briefly studied. Without loss of generality, the CIC filter is assumed to have L stages as shown in Fig. 13. First of all, consider the integrator section at the left-hand side of the downsampler. Here, we propose to scale down the input signal by a factor of $1/M_{\text{CIC}}$, i.e., shifted by $\log_2(M_{\text{CIC}})$ bits to the

TABLE VIII
WORDLENGTHS OF THE CIC FILTER

Stage L	Integrator	Comb
1	*<1/17>*	*<2/25>*
2	*<1/21>*	*<3/25>*
3	*<1/25>*	*<4/25>*

right at each integrator to avoid excessive round-off and overflow errors, at the expense of slightly increased hardware complexity. Each integrator has a programmable shifter S_k, $k = 1, 2, \ldots, L$, which is designed to shift the incoming signal from 0 up to $\log_2(M_{\rm CIC}^{\max})$ bits, where $M_{\rm CIC}^{\max}$ is the maximum downsampling ratio of the CIC filter and is a positive power-of-two integer. In general, to implement an arbitrary right shift up to $2^N - 1$ bits, an N stage programmable shifter is required. Next, consider the comb section at the right-hand side of the downsampler shown in Fig. 13. Each comb filter can be viewed as the transposed form of a particular FIR filter with two coefficients of 1 and -1 such that the L1 scaling measure as mentioned in Section IV-B can also be applied. Therefore, both integrator and comb sections will be free of round-off and overflow errors if the integer and fractional bits are appropriately allocated. As an example, the number of stages L and the maximum downsampling ratio $M_{\rm CIC}^{\max}$ of the CIC filter are chosen to be 3 and 16, respectively in order for the proposed SRR to support an overall downsampling ratio M^* between 2 to 512. If a larger $M_{\rm CIC}^{\max}$ is chosen, the SRR can support a larger range of M^* by slightly modifying the programmable shifters of the CIC filter. In this work, the input signal $x(n)$ to the CIC filter is assumed to have a wordlength format of $\langle 1/13 \rangle$, i.e., 14-bits with $x_{\max} = 0.99988$. The wordlengths of the basic CIC filter so obtained are shown in Table VIII. The output signal at the CIC filter has a wordlength format of $\langle 4/25 \rangle$. It should be noted that the proposed wordlength determination algorithms can also be used to further minimize the wordlengths of the CIC filter since its internal transfer functions are known. For simplicity, we employ the wordlength formats in Table VIII and use the proposed algorithms to determine the wordlengths of the remaining components of the SRR, as we shall illustrate in the next section. Now, we present a detailed design example of the proposed SRR for a multi-standard receiver.

VI. DESIGN EXAMPLE

In this section, we demonstrate the application of the proposed SRR for a multi-standard receiver to support the GSM, W-CDMA, CDMA2000 and Hiperlan/2 standards. The hardware complexity and the performance of the SRR using the real valued and SOPOT coefficients are then examined. Finally, a comparison between the proposed and conventional SRR is presented. First of all, let us assume that the digitized IF signal is sampled at 80 M sps. Table IX summarizes some of the useful parameters of the GSM, W-CDMA, CDMA2000 and Hiperlan/2 standards [30], [34], [35]. It also includes the configurations and the computational complexities for both the real valued and SOPOT coefficient realizations of the proposed SRR. The target specifications of the proposed SRR are as follows: $\delta_{p-\max} = 0.00173$ (0.015 dB in passband deviation), $\delta_{s-\max} = 0.0001$ (80 dB in stopband attenuation), $\delta_{d-\max} = 0.00316$ (50 dB in fractional-delay error), and $P_{\rm spec} = 2.512 \times 10^{-10}$ (96 dB in output accuracy). The output accuracy, in terms of the number of fractional bits, can be calculated from (4-4) to be $A_{\rm out} = 16$. Table X shows the passband deviations and the stopband attenuations of the proposed SRR using the real valued and the SOPOT coefficients for different operating ranges of M^*, i.e., cascading different components. In particular, the frequency responses of the SRR with $4 \leq M^* \leq 8$, i.e., cascading the LPF#3, HBF and the VDF-based SRC with $M_{\rm SRC} \in (1, 2)$, using the real valued and SOPOT coefficients are shown in Fig. 14. It can be seen that the stopband attenuations of the SRR using the real-valued coefficients are slightly better than that using the SOPOT coefficients. The opposite is true for the passband deviations. This is because the random search algorithm succeeded in finding a set of SOPOT coefficients which meet the stopband specification at 80 dB (though inferior to the real-valued coefficients), and at the same time slightly improve the given specification of passband deviation of 0.015 dB. This substantiates the usefulness of the proposed multiplier-less realization method as described in Section III. It should be noted that the total

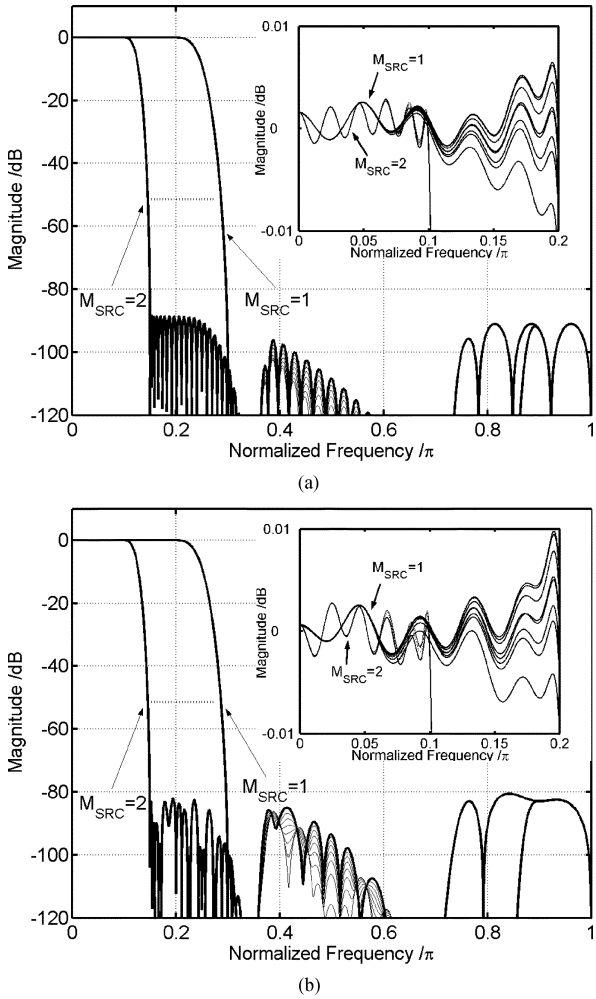


Fig. 14. Frequency responses of the proposed SRR with $4 \leq M^* \leq 8$, i.e., cascading the LPF#3, HBF and the VDF with $M_{\rm SRC} \in (1, 2)$, using (a) real-valued and (b) SOPOT coefficients. (Note: The corresponding passband ripples are shown in the top right corner of the figure.)

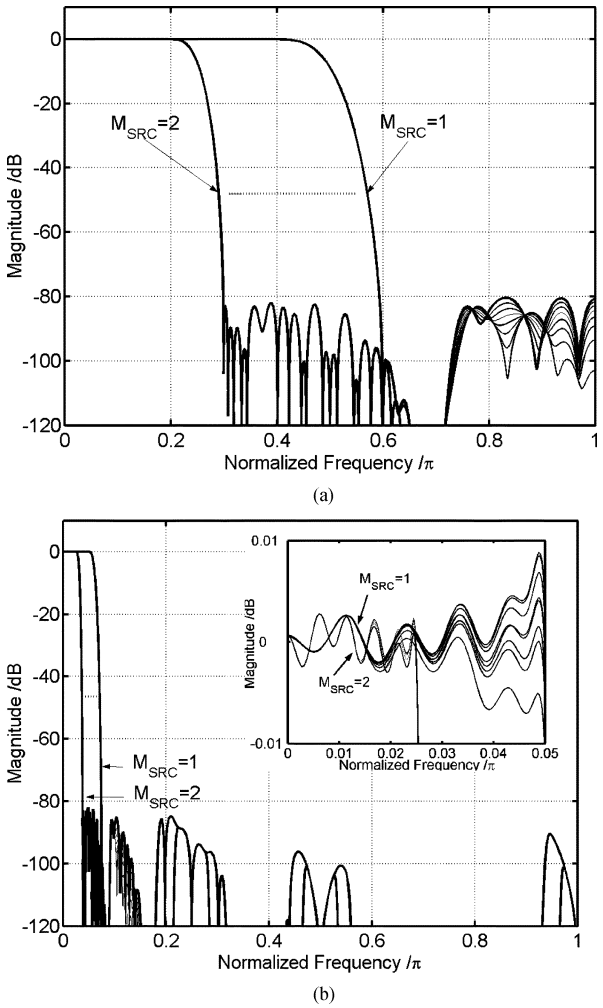


Fig. 15. Frequency responses of proposed SRR with (a) $2 \leq M^* \leq 4$, i.e., cascading the HBF and the VDF with $M_{\rm SRC} \in (1, 2)$ (b) $16 \leq M^* \leq 32$, i.e., cascading LPF#1, LPF#2, LPF#3, HBF and VDF with $M_{\rm SRC} \in (1, 2)$.

number of multipliers required to implement all the real-valued coefficients of the SRR is 117. On the other hand, the multiplier-less realization using SOPOT coefficients requires only 252 adders. After using the MB technique, the number of adders is further reduced to 111, which is about 44% of the hardware resources required for directly implementing all the SOPOT coefficients. The frequency responses of the proposed SRR, using the SOPOT coefficients, with the operating ranges: a) $2 \leq M^* \leq 4$, i.e., cascading the HBF and the VDF-based SRC with $M_{\rm SRC} \in (1, 2)$, and b) $16 \leq M^* \leq 32$, i.e., cascading the LPF#1, LPF#2, LPF#3, HBF and the VDF with $M_{\rm SRC} \in (1, 2)$, are shown in Fig. 15(a) and (b), respectively.

Once the SOPOT coefficients are determined, the internal transfer functions of these filters are known and there are totally $M = 236$ rounding sources in the receiver except the basic CIC filter. Using (5-8), the optimal wordlength format for each intermediate signal is obtained. The weight vector $\boldsymbol{w}$ has all its entries equal to one. The optimal value of $\boldsymbol{w}^T\boldsymbol{b}$ is found to be 4145.2. As mentioned earlier, the entries of the vector $\boldsymbol{b}$ are not integer valued. Therefore, for practical implementation, they are rounded to the closest integer just larger than them such that the 16-bit accuracy is still met. The corresponding value of $\boldsymbol{w}^T\boldsymbol{b}$ becomes 4276 and the total noise power σ_e^2 is decreased to 1.21×10^{-10} (or 16.527 bit accuracy). The results obtained using random search algorithm are summarized as follows: $\boldsymbol{w}^T\boldsymbol{b} = 4196$; $\sigma_e^2 = 2.02 \times 10^{-10}$ (i.e., 16.157 bit accuracy) and the computational time is about 20 minutes. For the bit allocation method, we obtain the following results in Table XI: $\boldsymbol{w}^T\boldsymbol{b} = 4171$; $\sigma_e^2 = 2.50 \times 10^{-10}$ (16.002 bit accuracy) and the computation time is within one minute. This method gives the best solution among the three algorithms studied with a much lower computational time than the random search algorithm. This suggests that the proposed bit allocation algorithm is well-suited even for a large scale system. It should be noted that the computational time of the proposed algorithm can be further reduced to a few seconds if the solution obtained from (5-8) is used as an initial guess. In order to avoid overflow, the worst-case integer bit format of each intermediate signal can then be calculated as described in Section II-B, assuming that the input signal $x[n]$ to the compensated CIC filter has a format of $\langle 1/13 \rangle$, i.e., 14-bits with $x_{\max} = 0.99988$. The final output is found to have a wordlength format of $\langle 9/19 \rangle$. The wordlength formats of each filter output format are also shown in Fig. 2(b) and (c). Table XI summarizes the design results for various wordlength determination algorithms. If a fixed wordlength of 24-bit is used, the following results are obtained: $\boldsymbol{w}^T\boldsymbol{b} = 4600$; $\sigma_e^2 = 6.16 \times 10^{-9}$ (13.684 bit accuracy). This suggests that the proposed variable internal wordlength approaches is more efficient than traditional method in minimizing the hardware complexity of the SRR while achieving a prescribed output bit accuracy.

Since the proposed SRR is considerably different from the traditional programmable SRR, it is very difficult to make an exact comparison. Anyway, to give the reader an idea of the potential benefits and hardware savings of the proposed SRR, a comparison with the programmable receiver proposed in [5] is considered below. The architecture in [5] consists of a CIC filter with $L = 5$, an ISOP sharpening filter, five modified HBFs (MHBFs) as the multistage decimators, and an PFIR. Since the SRC was not designed in [5], we assume that it is done using the same VDF-based SRC that we have proposed in Section II-C so that they have the same complexity. In addition, the finite wordlength effects were not considered in [5]. Therefore, the comparison will be based on the number of adders and multipliers required. Except for the ISOP sharpening filter, the dynamic range of the filter coefficients in the two structures is comparable. Table XII shows the hardware complexities of the other components for the two SRRs. It can be seen that the major hardware resources of the structure in [5] is the variable multipliers required in the PFIR. Although the multiplications can be time multiplexed using a high-speed multiplier, it will limit the maximum clock speed of the receiver for wideband applications, i.e., small downsampling factors. In the proposed architecture, the PFIR is replaced by a HBF with fixed coefficients, thanks to the novel VDF-based SRC. Therefore, the complexity can be greatly reduced. Note, though the VDF still requires three variable multipliers, it is still much less than that required for the PFIR. Apart from the PFIR, the hardware complexities of the two digital IF architectures are comparable. The

TABLE IX
CONFIGURATIONS AND COMPUTATIONAL COMPLEXITIES OF THE SRR FOR SUPPORTING GSM, W-CDMA, CDMA2000 AND HIPERLAN /2 STANDARDS. MOPS: MILLION OPERATIONS PER SECOND

Parameters	GSM	W-CDMA	CDMA2000	Hiperlan/2
Multiple Access	TDMA/FDMA	CDMA	CDMA	TDMA
Duplex Mode	TDD, FDD	FDD	FDD	TDD
Uplink Freq. Band (MHz)	890-915	1920-1980	Any existing band	5150-5350
Downlink Freq. Band (MHz)	935-960	2110-2170		5470-5725
Channel Bandwidth (MHz)	0.2	5	1.25	20
Data Modulation	GMSK	QPSK	QPSK	OFDM
Input sampling rate (F_{s-in})	80 Msps			
Output Accuracy	96 dB (16 bits in fractional part)			
Output sampling rate (F_{s-out})	270.833 ksps	3.84 Mcps	1.2288 Mcps	20 Msps
M^*	295.3849	20.83	65.1041	4
M_{CIC}	16	N/A	4	N/A
M_{SRC}	1.153847	1.302083	1.01725	N/A
k	4	4	4	2
Proposed SRR using real-valued coefficients				
Multiplications (MOPS)	115.875	1667.36	471.355	1680
Additions (MOPS)	463.33	3124	1147.8798	3240
Proposed SRR using SOPOT coefficients				
Multiplications (MOPS)	1.625	23.04	7.3728	0
Additions (MOPS)	604.54	5200.64	1719.41	5280

TABLE X
PERFORMANCES OF THE PROPOSED SRR (PASSBAND DEVIATION, STOPBAND ATTENUATION) USING THE REAL-VALUED AND SOPOT COEFFICIENTS FOR DIFFERENT OVERALL DOWNSAMPLING RATIOS

Overall downsampling ratio	Real-valued	SOPOT
$2 \le M^* \le 4$	(0.01162, 88.24) dB	(0.00968, 80.37) dB
$4 \le M^* \le 8$	(0.01179, 88.24) dB	(0.00978, 80.55) dB
$8 \le M^* \le 16$	(0.01105, 88.42) dB	(0.00992, 81.98) dB
$16 \le M^* \le 32$	(0.00906, 88.42) dB	(0.00882, 81.98) dB
$M^* \ge 32$	(0.01648, 82.41) dB	(0.01372, 81.98) dB

TABLE XI
SUMMARY OF DESIGN RESULTS USING VARIOUS WORDLENGTH DETERMINATION ALGORITHMS

	$w^T b$	Output noise power	Output bit accuracy
Random search algorithm	4196	2.02×10^{-10}	16.157
Analytic solution	4145.2	2.51×10^{-10}	16
Rounded analytic solution	4276	1.21×10^{-10}	16.527
Bit allocation algorithm	4171	2.50×10^{-10}	16.002

proposed SRR, however, considerably outperforms [5] in passband deviation. Moreover, the results for the allpass-based realization of the proposed SRR [10] are also given in Table XII as a reference. It has slightly higher complexity and larger group delay error, but a considerably lower system delay, than its FIR linear-phase counterpart. Interested readers are referred to [10] for more details.

The proposed SRR has also been implemented and tested using the Altera Stratix FPGA and the Quartus II EDA tools from Altera Corporation [36]. Because of the use of SOPOT coefficients and the MB technique, the main hardware resources needed to implement the proposed SRR are shifters and carry save adders. The hardware complexity in terms of logic elements (LEs) and *LC* registers (LRs) required for the proposed SRR are summarized in Table XIII. It can be seen that the implementation of the VDF consumes about half of the total hardware resources and most of them are coming from the four expensive multipliers in the Farrow structure, which are implemented using the DSP blocks in the Stratix FPGA. Since the conventional receiver in [5] requires 55 multipliers, the numbers of LEs and LRs required are about 110000 and 55000 respectively, which are approximately nine times more than those required for the proposed SRR. For ASIC implementation, we expect that similar savings can be obtained if the SOPOT representation is employed, while the multiplication may be realized as optimized multiplier modules. For simplicity, we do not explore these additional implementation issues and options further in this paper. Overall, the design results illustrate the effectiveness of the proposed approach in the reduction of hardware complexity. It should be noted that the DFT-FB-based channelizer can also be implemented without any multipliers using the technique in [31] and [32] so as to reduce hardware complexity. However, details are omitted due to page limitation.

TABLE XII
COMPARISON OF HARDWARE COMPLEXITIES BETWEEN [5] AND THE PROPOSED SRR USING FIR LINEAR-PHASE FILTERS (ALLPASS-BASED FILTERS [10]). NOTE: COMPLEXITY OF VDF-BASED SRC IS NOT INCLUDED

	Architecture in [5]				**Proposed SRR**			
Passband deviation	0.2 dB				0.015 dB			
Stopband attenuation	80 dB				80 dB			
	CIC	**ISOP**	**MHBFs**	**PFIR**	**CIC**	**2nd order**	**LPFs**	**HBF**
Multiplications	0	2	0	50	0	0	0	0
Additions	10	2	68	49	6	4	67 (103)	70 (78)
Total multiplications	52				0			
Total additions	129				147 (181)			

TABLE XIII
USAGE OF THE HARDWARE RESOURCES FOR DIFFERENT SRR COMPONENTS

Hardware Recourse	**Total**	**CIC**	**CIC Compensator**	**LPF#1**	**LPF#2**	**LPF#3**	**VDF**	**HBF**
Logic Elements	12324	400	157	486	636	950	6992	2703
LC Registers	6072	388	100	230	316	463	3266	1309
DSP block 9-bit elements	24	0	0	0	0	0	24	0

VII. CONCLUSION

The signal round-off analysis and complexity optimization of a new digital IF for SRR are presented. An advantage of the SRR is that it can be implemented without any multiplications, apart from the limited number of multipliers required in the Farrow structure. Moreover, two novel algorithms for determining the internal wordlengths of the digital IF subject to a prescribed output accuracy are presented. The first one gives a closed-form analytic solution using Lagrange multiplier method, assuming that the wordlength is a real-valued quantity. The second one is based on the Marginal Analysis method and it gives an integer-valued solution. Design results show that the proposed algorithms work well with large number of variables and they are applicable to the wordlength determination problems for the realization of related digital filtering systems. Because of the short computational time, they may be useful for redesigning and reconfiguring systems for not only the proposed SRR but also other real-time applications. Another interesting direction is the extension of the current IF architecture to utilize IIR filters to reduce the overall system delay. The main challenge will be the more complicated signal overflow and roundoff problems because of the increased dynamic range of the intermediate signals.

REFERENCES

[1] T. Hentschel and G. Fettweis, "Sample rate conversion for software radio," *IEEE Commun. Mag.*, pp. 142–150, Aug. 2000.

[2] C. Y. Fung and S. C. Chan, "A multistage filterbank-based channelizer for software radio base stations," in *Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS)*, 2002, vol. 3, pp. 429–432.

[3] R. H. Walden, "Performance trends for analog-to-digital converters," *IEEE Commun. Mag.*, vol. 37, no. 2, pp. 96–101, Feb. 1999.

[4] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*. Singapore: McGraw-Hill, 1998.

[5] 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.*, vol. 17, no. 4, pp. 551–560, Apr. 1999.

[6] A. Y. Kwentus, Z. Jiang, and A. N. Willson, "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 457–467, Feb. 1997.

[7] K. S. Yeung and S. C. Chan, "On the design and multiplier-less realization of digital IF for software radio receivers," in *Proc. XI Eur. Signal Proces. Conf. (EUSIPCO)*, Toulouse, France, Sep. 3–6, 2002, vol. 1, pp. 695–698.

[8] S. C. Chan and K. S. Yeung, "On the design and multiplier-less realization of digital IF for software radio receivers with prescribed output accuracy," in *Proc. 14th Int. Conf. on Digital Signal Process. (DSP)* , Hellas, Greece, Jul. 1–3, 2002, vol. 1, pp. 277–280.

[9] C. W. Farrow, "A continuously variable digital delay element," in *Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS)*, 1988, pp. 2641–2645.

[10] K. S. Yeung and S. C. Chan, "The design and multiplier-less realization of software radio receivers with reduced system delay," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 1, pp. 2444–2459, Dec. 2004.

[11] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete power-of-two coefficient space," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 31, no. 4, pp. 583–591, Apr. 1983.

[12] 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," in *Proc. 11th IEEE Signal Processing Workshop on Statistical Signal Process.*, 2001, pp. 595–598.

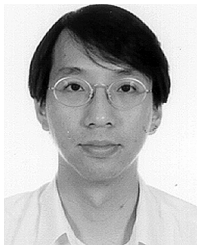
[13] Y. C. Lim, R. Yang, D. Li, and J. Song, "Signed power-of-two term allocation scheme for the design of digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 5, pp. 577–584, May 1999.

[14] A. G. Dempster and M. D. MacLeod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 9, pp. 569–577, Sep. 1995.

[15] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 10, pp. 1432–1442, Oct. 2003.

[16] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester, U.K.: Wiley, 1987.

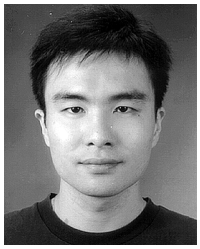
[17] P. D. Fiore and L. Lee, "Closed-form and real-time wordlength adaptation," in *Proc. IEEE ICASSP*, 1999, vol. 5, pp. 1897–1990.
[18] A. Segal, "Bit allocation and encoding of vector resource," *IEEE Trans. Inf. Theory*, vol. 22, no. 4, pp. 162–169, Mar. 1976.
[19] B. Fox, "Discrete optimization via marginal analysis," *Manag. Sci.*, vol. 13, pp. 210–216, Nov. 1966.
[20] S. W. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders," in *Proc. IEEE ICASSP*, 1993, vol. 5, pp. 390–392.
[21] K. C. Zangi and R. Koilpillai, "Software radio issues in cellular base stations," *IEEE J. Select. Areas Commun.*, vol. 17, no. 4, pp. 561–573, Apr. 1999.
[22] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1983.
[23] K. Y. Khoo, Z. Yu, and A. N. Willson, "Efficient high-speed cic decimation filter," in *Proc. 11th Annu. IEEE Int. Conf. ASIC*, 1998, pp. 251–254.
[24] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[25] T. A. Ramstad, "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. ASSP-32, no. 3, pp. 577–591, Jun. 1984.
[26] G. S. Liu and C. H. Wei, "Programmable fractional sample delay filter with lagrange interpolation," *Electron. Lett.*, vol. 26, no. 19, pp. 1608–1610, Sep. 1990.
[27] J. M. de Carvalho and J. V. Hanson, "Efficient sample rate conversion with cubic splines," in *Proc. Telecomm. Symp.*, 1990, pp. 439–442.
[28] W. S. Lu and T. B. Deng, "An improved weighted least-squares design for variable fractional delay FIR filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 8, pp. 1035–1040, Aug. 1999.
[29] C. K. S. Pun, S. C. Chan, K. S. Yeung, and K. L. Ho, "On the design and implementation of FIR and IIR digital filters with variable frequency characteristics," in *Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS)*, 2002, vol. 2, pp. 185–188.
[30] A. Mehrotra, *Cellular Radio: Analog and Digital Systems*. Natick, MA: Artech House, 1994.
[31] S. C. Chan and K. M. Tsui, "Multiplier-less real-valued FFT-like transformation (ML-RFFT) and related real-valued transformations," in *Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS)*, May 25–28, 2003, vol. 4, pp. 257–260.
[32] K. M. Tsui and S. C. Chan, "Error analysis and efficient realization of the multiplier-less FFT-like transformation (ML-FFT) and related sinusoidal transformations," *J. VLSI Signal Process.*, vol. 44, no. 1-2, pp. 97–115, Aug. 2006.
[33] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[34] T. Ojanpera and R. Prasad, *WCDMA: Towards IP Mobility and Mobile Internet*. Natick, MA: Artech House, 2001.
[35] *ETSI HIPERLAN/2 Standard*, [Online]. Available: http://www.etsi.org/technicalactiv/Hiperlan/hiperlan2.htm
[36] Altera Corporation [Online]. Available: http://www.altera.com
[37] S. C. Chan and K. M. Tsui, "Wordlength determination algorithms for hardware implementation of linear time-invariant systems with prescribed output accuracy," in *Proc. IEEE Int. Symp. on Circuits Syst. (ISCAS)*, Kobe, Japan, May 23–26, 2005, pp. 2607–2610.



S. C. Chan (S'87–M'92) received the B.Sc. (Eng) and Ph.D. degrees from the University of Hong Kong, Hong Kong, in 1986 and 1992, respectively.

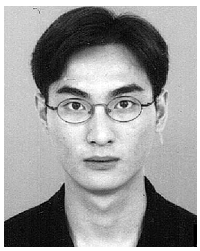
He joined City Polytechnic of Hong Kong in 1990 as an Assistant Lecturer and later as a University Lecturer. Since 1994, he has been with the Department of Electrical and Electronic Engineering, the University of Hong Kong, and is now an associate Professor. He was a Visiting Researcher at Microsoft Corporation, Redmond, CA, and Microsoft, Beijing, China, in 1998 and 1999, respectively. His research interests include fast-transform algorithms, filter design and realization, multirate signal processing, communications signal processing, and image-based rendering.

Dr. Chan is currently a member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He was Chairman of the IEEE Hong Kong Chapter of Signal Processing from 2000 to 2002.



K. M. Tsui received the B.Eng. and M.Phil. degrees in electrical and electronic engineering from The University of Hong Kong, Hong Kong, in 2001 and 2004, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Electronic Engineering, The University of Hong Kong.

His main research interests are in biomedical signal processing, digital signal processing, multirate filter bank and wavelet design, and digital filter design, realization and application.



K. S. Yeung received the B.Eng. and M.Phil. degrees in electrical and electronic engineering from The University of Hong Kong, Hong Kong, in 2001 and 2003, respectively.

His main research interests are in digital signal processing, multirate filter bank and wavelet design, digital filter design, realization and application.

T. I. Yuk received the B.S. degree from Iowa State University, Ames, in 1978, and the M.S. and Ph.D. degrees from Arizona State University, Tempe, in 1980 and 1986, respectively.

Since 1986, he has been teaching at the University of Hong Kong. His current research interests include: study of space-time codes in wireless communication systems, filter bank designs, modeling of wireless communication network.