

Design and FPGA Implementation of a Reconfigurable 1024-Channel Channelization Architecture for SDR Application

Xue Liu, Ze-Ke Wang, and Qing-Xu Deng

Abstract—In this paper, we present a novel channelization architecture, which can simultaneously process two channels of complex input data and provide up to 1024 independent channels of complex output data. The proposed architecture is highly modular and generic, so that parameters of each output channel can be dynamically changed even at runtime in terms of the bandwidth, center frequency, output sampling rate, and so on. It consists of one tunable pipelined frequency transform (TPFT)-based coarse channelization block, one tuning unit, and one resampling filter. Based on the analysis of the data dependence between the subbands, a novel channel splitting scheme is proposed to enable multiple subbands to share the proposed TPFT block. The multiplier block (MB) and subexpression sharing techniques are used to reduce the number of arithmetic units of the TPFT block. Moreover, the proposed Farrow-based resampling filter does not require division operation and dual-port RAMs resulting in significant area saving. Finally, we implement the proposed channelization architecture in a single field-programmable gate array. The experiment results indicate that our design provides the flexibility associated with the existing works, but with greater resource efficiency.

Index Terms—Channelization, field-programmable gate array (FPGA), software-defined radio (SDR), tunable pipelined frequency transform (TPFT), variable fractional delay filter.

I. INTRODUCTION

SOFTWARE-DEFINED radio (SDR) has been widely applied to many fields, such as communication, electronic warfare, and instrumentation. The core idea of the SDR is to move the analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) as close as possible to the antenna, and to process the digitized data by a software technique. The development of ADCs, DACs, and digital circuits has greatly promoted the progress of SDR.

A key component in SDR systems is a real-time configurable digital channelization, which is essential for receiving and resampling the radio signal correctly, and is also critical

to monitor the whole spectrum to determine the inactive bands. Compatibility with different communication standards requires that channelization be dynamically reconfigurable. A resource efficient design is another key requirement for the implementation of channelization. For digital channelization, it is difficult to obtain a perfect solution that can balance all the targets, such as performance and resource. The design and the implementation of channelization are affected by many parameters, such as output sampling rate, bandwidth, frequency resolution, and dynamic configuration. Exploring the cost/performance tradeoffs is an important issue for digital channelization.

Digital channelization is generally implemented using the following methods: 1) multichannel digital downconverter (DDC); 2) fast Fourier transform (FFT); 3) polyphase discrete Fourier transform (DFT) filter bank; 4) Goertzel filter bank; 5) tree-structured filter bank; 6) nonuniform filter bank; and 7) analysis/synthesis filter bank.

DDC is generally composed of a numerical control oscillator (NCO) and a sampling rate conversion (SRC) filter. It is mainly used for single-channel channelization, but it is unsuited to multichannel channelization because of its low resource utilization. FFT-based channelization has a simple structure and high resource utilization, but its filtering performance is poor, which can be improved using the windowing function method [1], [2]. A polyphase DFT filter bank has high resource utilization and better filtering performance, but it only realizes a uniform channel division [3]. A Goertzel filter bank can provide a solution to a fixed center frequency problem associated with a polyphase DFT filter bank, but it cannot extract channels with nonuniform bandwidths [3].

According to the tree-structured filter bank approach, RFEL Ltd. has developed two methods for digital channelization, called pipelined frequency transform (PFT) and tunable PFT (TPFT) [4]. The problem of power-of-two channel stacking associated with PFT can be solved by the TPFT technique [4]. TPFT can accurately locate the radio signals, thereby ensuring the correct reception. It consists of a coarse channelization in the PFT stage and a fine channelization using a combination of complex DDC and digital upconversion. However, its implementation complexity is much more than that of PFT.

The frequency response masking and coefficient decimation techniques are used to implement digital channelization [5]–[8]. The advantages of these two techniques are low

Manuscript received July 19, 2015; revised October 27, 2015 and December 8, 2015; accepted December 9, 2015. Date of publication January 6, 2016; date of current version June 23, 2016. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant N140403003, in part by the Doctoral Scientific Research Foundation of Liaoning Province under Grant 20141015, in part by the National Basic Research Program of China (973) under Grant 2014CB360509, and in part by the National Natural Science Foundation of China under Grant 61472072 and Grant 61501103.

X. Liu and Q.-X. Deng are with the Department of Information Science and Engineering, Northeastern University, Shenyang 110004, China (e-mail: liuxue0512@gmail.com; dengqx@mail.neu.edu.cn).

Z.-K. Wang is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: wangzeke638@gmail.com).

Digital Object Identifier 10.1109/TVLSI.2015.2508038

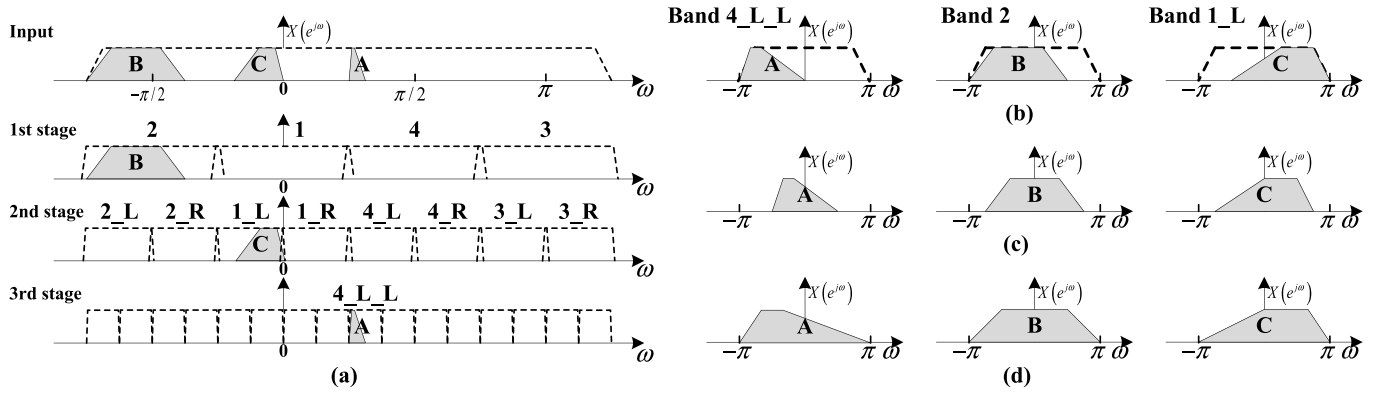


Fig. 1. Channel division of wideband complex input signal. (a) Input signal. (b) Coarse channelization. (c) Tuning. (d) Resampling filter.

computational complexity and nonuniform channel division. For an analysis/synthesis filter bank approach, it uses the analysis filter bank to implement uniform channel division, and then uses the synthesis filter bank to merge the adjacent channels for realizing nonuniform channel division [9]. However, the channel bandwidths generated by the synthesis filter bank are integer multiples of those generated by the analysis filter bank. Such constraint limits the flexibility of the analysis/synthesis filter bank. The reason is that the channel bandwidths of different communication standards are not related by an integer factor.

Channelization typically works at the ADC sampling clock frequency, while the subsequent baseband processors often operate at the symbol rate. Different communication standards typically have different symbol rates. Thus, SRC is crucial to multistandard channelization. The following filters are often used for SRC. They are a cascaded-integrated comb (CIC) filter, half-band filter, variable digital filter (VDF), and polyphase filter. Among them, a VDF can be used for arbitrary SRC [10], [11]. The passband droop of a CIC filter is very obvious. This problem can be solved using a sharpened filter [12] or a second-order compensator [13]. For the filters with fixed coefficients, the implementation complexity can be reduced using the sum-of-power-of-two (SOPOT) coefficients [14] and the multiplier block (MB) technique [15], which significantly reduce the number of additions and multiplications [16], [17].

A representative channelization system is the multiband, multichannel digital channelizing receiver developed by HYPRES [18]. It is composed of two stages. DDC and digital decimation filter lie in the first stage, and they are used for coarse channel division. Field-programmable gate array (FPGA) processing circuits lie in the second stage, and they are used for finer channel division.

In this paper, we present a novel channelization architecture, which can potentially support up to 1024 channels. Each channel with independent parameters can be configurable even at runtime. The corresponding FPGA implementation can be used as a replacement for standard DDC ASIC devices, providing the flexibility associated with the currently available works, but with less hardware resource. The proposed channelization architecture consists of one TPFT-based coarse channelization block, one tuning unit,

and one resampling filter. Its features are summarized as follows.

- 1) A novel channel splitting scheme is proposed to enable multiple subbands to time-multiplex the proposed TPFT block. Compared with the conventional TPFT approach [4], the proposed TPFT block uses the MB and subexpression sharing techniques to reduce the number of arithmetic units, and it removes the channelization blind zones by widening the prototype filter passband.
- 2) Based on the proposed SRC factorization scheme and the gain scaling technique, a division-free VDF transfer function is presented for performing arbitrary SRC. According to our address mapping method, the proposed multichannel Farrow-based VDF does not require dual-port RAMs resulting in significant memory saving.

The experiment results show that our design is multiplication reduction of 21.7% over the nonuniform filter bank [8], 68.7% over the analysis/synthesis filter bank [9], 37.5% over the DFT filter bank [3], and 66.3% over the Goertzel technique [3]. The design example is implemented on Xilinx Virtex 4 FPGA with the 16-bit precision. The synthesis results show that our design is slice reduction of 5.9% over the nonuniform filter bank [8] and 19.4% over the DFT filter bank [3]. Moreover, our channelization solution provides dynamic independent configuration for each channel, such as center frequency, bandwidth, and output sampling rate.

The rest of this paper is organized as follows. The proposed channelization architecture is given in Section II. The complete design details are given in Section III. This is then followed by two specific FPGA implementations in Section IV. Finally, the conclusions are drawn in Section V.

II. PROPOSED ARCHITECTURE

In the current design, the maximum output sampling rate and the antialiasing output bandwidth of each channel are limited to $F_s/8$ and $F_s/20$ ($=12.8$ MHz for $F_s = 256$ MS/s), where F_s is the input sampling rate, and the spurious free dynamic range (SFDR) is required to exceed 70 dB.

A. Overall Architecture

We use an example to illustrate the channelization process of the proposed architecture (see Fig. 1). We plan to extract

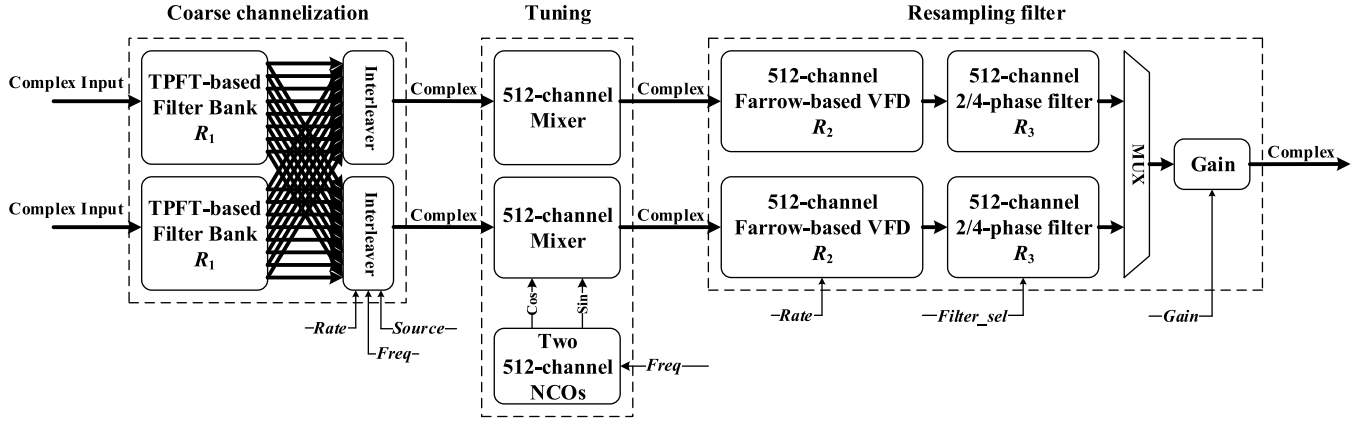


Fig. 2. Proposed channelization architecture.

three narrow-band signals A, B, and C from the input signal.

Their parameters, such as center frequency, output sampling rate, and bandwidth, are different. Fig. 2 shows the high-level architecture of the proposed channelization solution. The channel extraction process is described as follows.

- 1) TPFT is used to implement the coarse channelization block. The spectrum of the input signal is separated into multiple bands [see Fig. 1(a)]. The bands of each TPFT output stage are half as wide as those of the previous stage. The target signals A, B, and C are contained in band 4_L_L of the third stage, band 2 of the first stage, and band 1_L of the second stage [see Fig. 1(b)].
- 2) The tuning unit moves the target signals from their original frequency locations to the zero frequency location [see Fig. 1(c)]. Complex multiplications and multichannel NCOs are used to construct the tuning unit.
- 3) According to the channel parameters, the target signals are further processed by the resampling filter, which is composed of multichannel Farrow-based variable fractional delay (VFD) filters, multichannel 2/4-phase filters, and multichannel gain. The operations performed in the resampling filter include SRC, filtering, and gain control [see Fig. 1(d)].

As shown in Section III, the hardware sharing technique is used throughout the whole design. The runtime configurable feature of the tuning unit, the coarse channelization block, and the resampling filter ensures that the proposed architecture has enough flexibility.

B. SRC Factorization

Referring to the proposed channelization architecture, an SRC factorization scheme is proposed to assign SRC factors to the coarse channelization block and the resampling filter. In the current design, the SRC factor R is defined as the following equation:

$$R = \frac{F_s}{F_{out}} = R_1 \times R_2 \times R_3 \quad (1)$$

where F_{out} is the output sampling rate, R_1 is the decimation factor of the TPFT-based filter bank, R_2 is the decimation

TABLE I
RESULTS OF SRC FACTORIZATION SCHEME

$R=F_s/F_{out}=2^{32}/Rate$	AAOBD*	Rate	R_1	R_2	R_3
[8,16)	$F_s/20$	$(2^{28}, 2^{29}]$	↓4	↓ $2^{29}/Rate$	↓2
[16,32)	$F_s/40$	$(2^{27}, 2^{28}]$	↓8	↓ $2^{28}/Rate$	↓2
[32,64)	$F_s/80$	$(2^{26}, 2^{27}]$	↓16	↓ $2^{27}/Rate$	↓2
[64,128)	$F_s/160$	$(2^{25}, 2^{26}]$	↓32	↓ $2^{26}/Rate$	↓2
[128,256)	$F_s/320$	$(2^{24}, 2^{25}]$	↓64	↓ $2^{25}/Rate$	↓2
[256,512)	$F_s/640$	$(2^{23}, 2^{24}]$	↓128	↓ $2^{24}/Rate$	↓2
[512,1024)	$F_s/1280$	$(2^{22}, 2^{23}]$	↓256	↓ $2^{23}/Rate$	↓2
[1024,2048)	$F_s/2560$	$(2^{21}, 2^{22}]$	↓512	↓ $2^{22}/Rate$	↓2
[16,32)	$F_{out} \times 0.8$	$(2^{27}, 2^{28}]$	↓4	↓ $2^{28}/Rate$	↓4
[32,64)	$F_{out} \times 0.8$	$(2^{26}, 2^{27}]$	↓8	↓ $2^{27}/Rate$	↓4
[64,128)	$F_{out} \times 0.8$	$(2^{25}, 2^{26}]$	↓16	↓ $2^{26}/Rate$	↓4
[128,256)	$F_{out} \times 0.8$	$(2^{24}, 2^{25}]$	↓32	↓ $2^{25}/Rate$	↓4
[256,512)	$F_{out} \times 0.8$	$(2^{23}, 2^{24}]$	↓64	↓ $2^{24}/Rate$	↓4
[512,1024)	$F_{out} \times 0.8$	$(2^{22}, 2^{23}]$	↓128	↓ $2^{23}/Rate$	↓4
[1024,2048)	$F_{out} \times 0.8$	$(2^{21}, 2^{22}]$	↓256	↓ $2^{22}/Rate$	↓4
[2048,4096)	$F_{out} \times 0.8$	$(2^{20}, 2^{21}]$	↓512	↓ $2^{21}/Rate$	↓4

*AAOBD stands for anti-aliasing output bandwidth.

factor of the Farrow-based VFD filter and it is in the range [1, 2), and R_3 is the decimation factor of the 2/4-phase filter and it is 2 or 4.

Let Rate be the sampling rate control word. Rate is calculated as follows:

$$Rate = \text{Round} \left(\frac{2^{32} \times F_{out}}{F_s} \right) = \text{Round} \left(\frac{2^{32}}{R_1 \times R_2 \times R_3} \right) \quad (2)$$

where Round(x) rounds the element x to the nearest integer. According to (2), the resolution of the output sampling rate is $F_s/(2^{32})$ (≈ 0.0596 Hz for $F_s = 256$ MS/s). When R is in the range [8, 4096), Rate satisfies the following equation:

$$2^{20} < Rate \leq 2^{29}. \quad (3)$$

Table I lists the SRC factorization results. From Table I, we find that the SRC factor of each component is codetermined by the sampling rate control word Rate and the antialiasing output bandwidth.

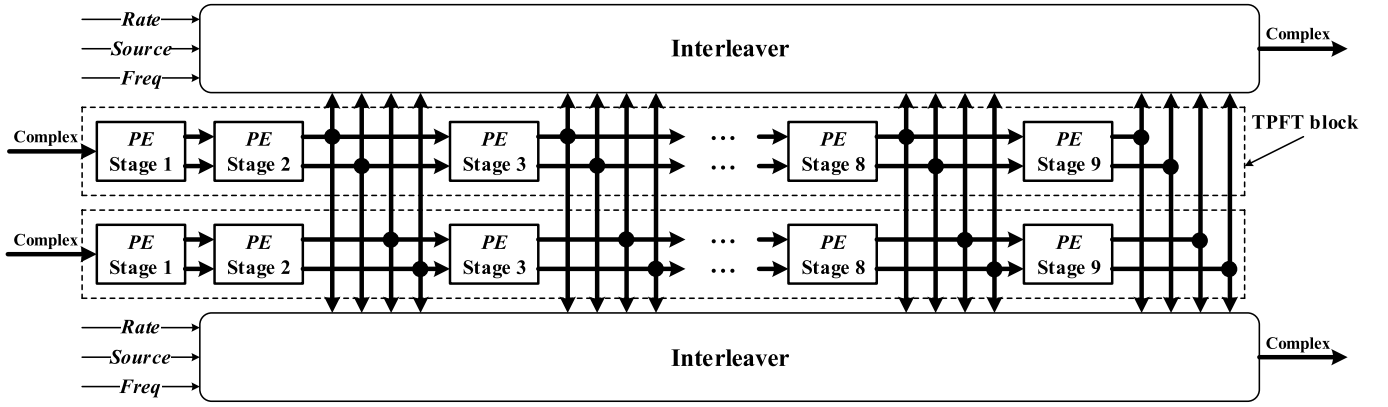


Fig. 3. Structure of the TPFT-based channelization block.

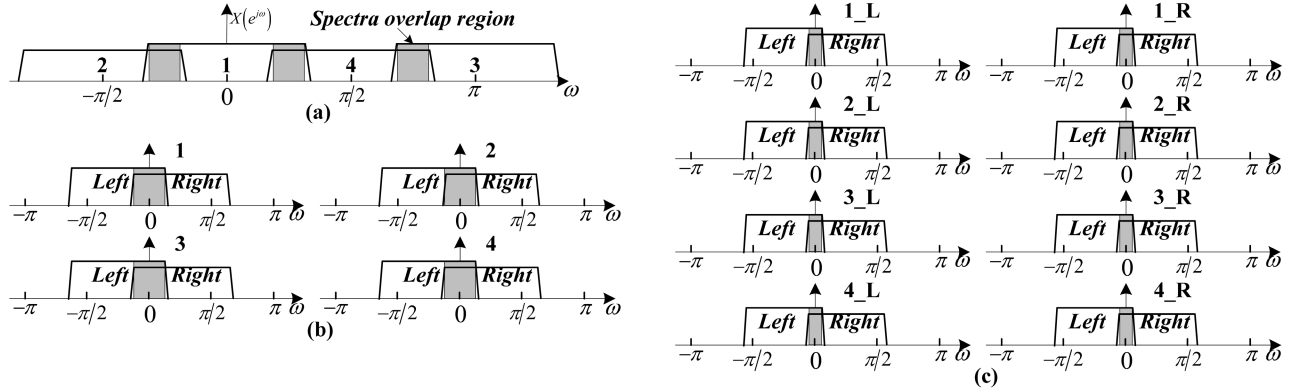


Fig. 4. (a) Channel division of PE 1 (PE in stage 1). (b) Channel division of PE 2 (PE in stage 2). (c) Channel division of PE 3 (PE in stage 3).

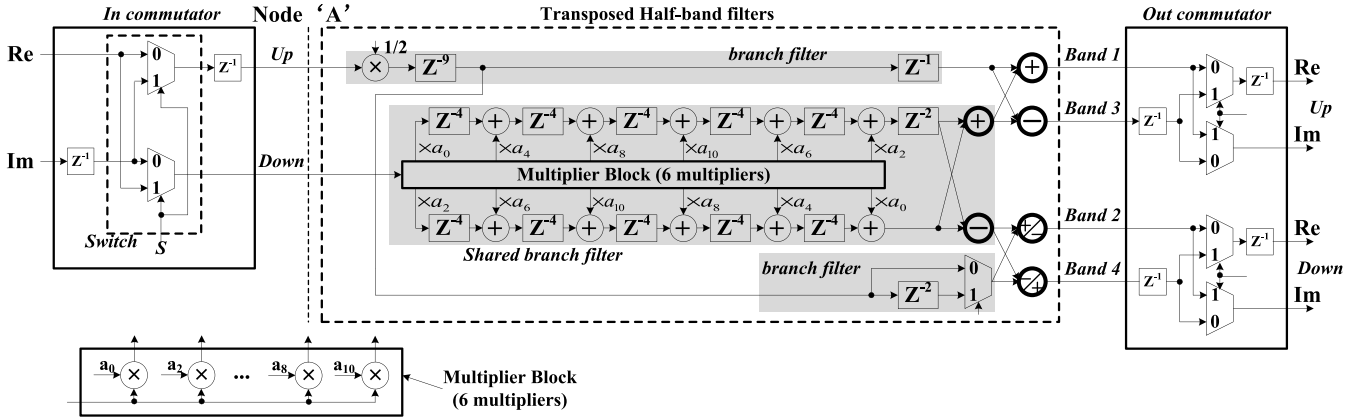


Fig. 5. Structure of PE 1.

III. DESIGN DETAILS

This section describes the designs of the TPFT-based coarse channelization block, the tuning unit, and the resampling filter in more detail.

A. TPFT-Based Coarse Channelization

The TPFT-based coarse channelization block consists of two parallel TPFT blocks and two parallel interleavers (see Fig. 3). For all successive stages in the TPFT block, the output signal is decimated by two. The maximum decimation factor of the TPFT block is 512 (see Table I). Thus, each TPFT block is composed of nine cascaded processing elements (PEs), and is used to process one channel of complex

input data. The TPFT-based channelization block is scalable in terms of the number of input signals, making it easily adapted for multiple applications.

B. TPFT Block

Fig. 4 shows the proposed channel splitting scheme, which is different from that in [4]. As shown in Fig. 4(a), the complex input signal is divided into four bands, called bands 1, 2, 3, and 4. Their center frequencies are 0, $-\pi/2$, π , and $\pi/2$. They are first moved to the zero frequency and, then, filtered by half-band filters in PE 1. After low-pass filtering, the output signals are decimated by two.

PE 1 consists of two data commutators and two transposed half-band filters with one shared branch filter (see Fig. 5).

TABLE II
DATA OUTPUT ORDER IN NODE A

Cycle	Type	Band 1 (Input is $x(n)$)		Band 2 (Input is $x(n) \times e^{j\pi n/2} \times e^{-j\pi n/2}$)		Band 3 (Input is $x(n) \times e^{-j\pi n} \times e^{-j\pi n}$)		Band 4 (Input is $x(n) \times e^{-j\pi n/2} \times e^{j\pi n/2}$)	
		Down $x_{B1_D}(n)$	Up	Down $x_{B2_D}(n)$	Up	Down $x_{B3_D}(n)$	Up	Down $x_{B4_D}(n)$	Up
0	Real	$x_r(1)$	$x_r(0)$	$x_r(1)$	$x_r(0)$	$x_r(1)$	$-x_r(0)$	$x_r(1)$	$-x_r(0)$
1	Imaginary	$x_i(1)$	$x_i(0)$	$x_i(1)$	$-x_i(0)$	$x_i(1)$	$-x_i(0)$	$x_i(1)$	$x_i(0)$
2	Real	$x_r(3)$	$x_r(2)$	$-x_r(3)$	$-x_r(2)$	$x_r(3)$	$-x_r(2)$	$-x_r(3)$	$x_i(2)$
3	Imaginary	$x_i(3)$	$x_i(2)$	$-x_i(3)$	$x_i(2)$	$x_i(3)$	$-x_i(2)$	$-x_i(3)$	$-x_i(2)$
4	Real	$x_r(5)$	$x_r(4)$	$x_r(5)$	$x_r(4)$	$x_r(5)$	$-x_r(4)$	$x_r(5)$	$-x_r(4)$
5	Imaginary	$x_i(5)$	$x_i(4)$	$x_i(5)$	$-x_i(4)$	$x_i(5)$	$-x_i(4)$	$x_i(5)$	$x_i(4)$
6	Real	$x_r(7)$	$x_r(6)$	$-x_r(7)$	$-x_r(6)$	$x_r(7)$	$-x_r(6)$	$-x_r(7)$	$x_i(6)$
7	Imaginary	$x_i(7)$	$x_i(6)$	$-x_i(7)$	$x_i(6)$	$x_i(7)$	$-x_i(6)$	$-x_i(7)$	$-x_i(6)$
8	Real	$x_r(9)$	$x_r(8)$	$x_r(9)$	$x_r(8)$	$x_r(9)$	$-x_r(8)$	$x_r(9)$	$-x_r(8)$
9	Imaginary	$x_i(9)$	$x_i(8)$	$x_i(9)$	$-x_i(8)$	$x_i(9)$	$-x_i(8)$	$x_i(9)$	$x_i(8)$

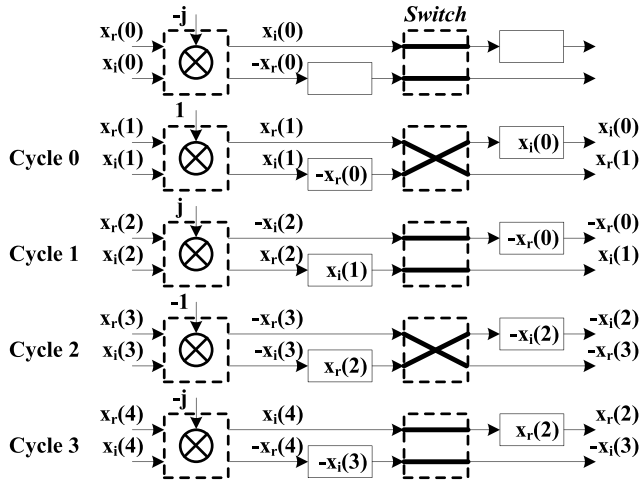


Fig. 6. Frequency shifting and data shuffling process of band 2.

Let $x_r(n)$ and $x_i(n)$ be the real and imaginary parts of the complex input data in cycle n . The input data are first shifted with different frequencies, and then, they are shuffled by *In commutator* to generate two sequences that are composed of the odd- and even-indexed data. After frequency shifting and data shuffling, the output orders of bands 1, 2, 3, and 4 are given in Table II.

The *In commutator* consists of two registers and one switch. The signal S controls the behavior of switch: swap or through. A snapshot of the frequency shifting and data shuffling process of band 2 is captured in Fig. 6. In cycles 1 and 3, the up (down) input of a switch connects the up (down) output. In cycles 0 and 2, the up (down) input of a switch connects the down (up) output.

Half-band filters in PE 1 are implemented with a two-phase structure because of decimation by two. Let $x_{B2_r}(n)$ and $x_{B2_i}(n)$ be the real and imaginary parts of band 2. From Fig. 6, we find that $x_{B2_r}(0)$ and $x_{B2_r}(1)$ are first exported from *In commutator*, and then, they are followed by $x_{B2_i}(0)$ and $x_{B2_i}(1)$. Thus, the two branches of a half-band filter are time-multiplexed by the real and imaginary parts of the input data. Let $h_{HB}(n)$ be the coefficients of a half-band filter. The Down output data of *In commutator* are convolved

with the even-indexed coefficients of $h_{HB}(n)$, while the Up output data do not need to perform the convolution computation. The reason is that the odd-indexed coefficients of $h_{HB}(n)$ are 1/2 or 0.

Let $h_E(n)$ be the even-indexed coefficients of $h_{HB}(n)$. Let $x_{B1_D}(n)$, $x_{B2_D}(n)$, $x_{B3_D}(n)$, and $x_{B4_D}(n)$ be the Down output data of bands 1, 2, 3, and 4 in node A. The real convolution result $y_{B1_D_r}(n)$ between $x_{B1_D}(n)$ and $h_E(n)$ is calculated as follows:

$$\begin{aligned}
 y_{B1_D_r}(n) &= x_{B1_D}(2n) * h_E(n) = \sum_{k=-\infty}^{+\infty} x_{B1_D}(2k) h_E(n-k) \\
 &= \sum_{m=-\infty}^{+\infty} x_{B1_D}(4m) h_E(n-2m) \\
 &\quad + \sum_{m=-\infty}^{+\infty} x_{B1_D}(4m-2) h_E(n-(2m-1)).
 \end{aligned} \tag{4}$$

The relationships between four Down output data are $x_{B1_D}(4m) = x_{B2_D}(4m) = x_{B3_D}(4m) = x_{B4_D}(4m)$ and $x_{B1_D}(4m+2) = -x_{B2_D}(4m+2) = x_{B3_D}(4m+2) = -x_{B4_D}(4m+2)$, so $y_{B3_D_r}(n)$ is equal to $y_{B1_D_r}(n)$, and the real convolution results $y_{B2_D_r}(n)$ and $y_{B4_D_r}(n)$ are calculated as follows:

$$\begin{aligned}
 y_{B2_D_r}(2q) &= y_{B4_D_r}(2q) \\
 &= \sum_{m=-\infty}^{+\infty} x_{B1_D}(4m) h_E(2q-2m) \\
 &\quad - \sum_{m=-\infty}^{+\infty} x_{B1_D}(4m-2) h_E(2q-(2m-1))
 \end{aligned} \tag{5}$$

$$\begin{aligned}
 y_{B2_D_r}(2q+1) &= y_{B4_D_r}(2q+1) \\
 &= \sum_{m=-\infty}^{+\infty} x_{B1_D}(4m) h_E(2q-(2m-1)) \\
 &\quad - \sum_{m=-\infty}^{+\infty} x_{B1_D}(4m+2) h_E(2q-2m).
 \end{aligned} \tag{6}$$

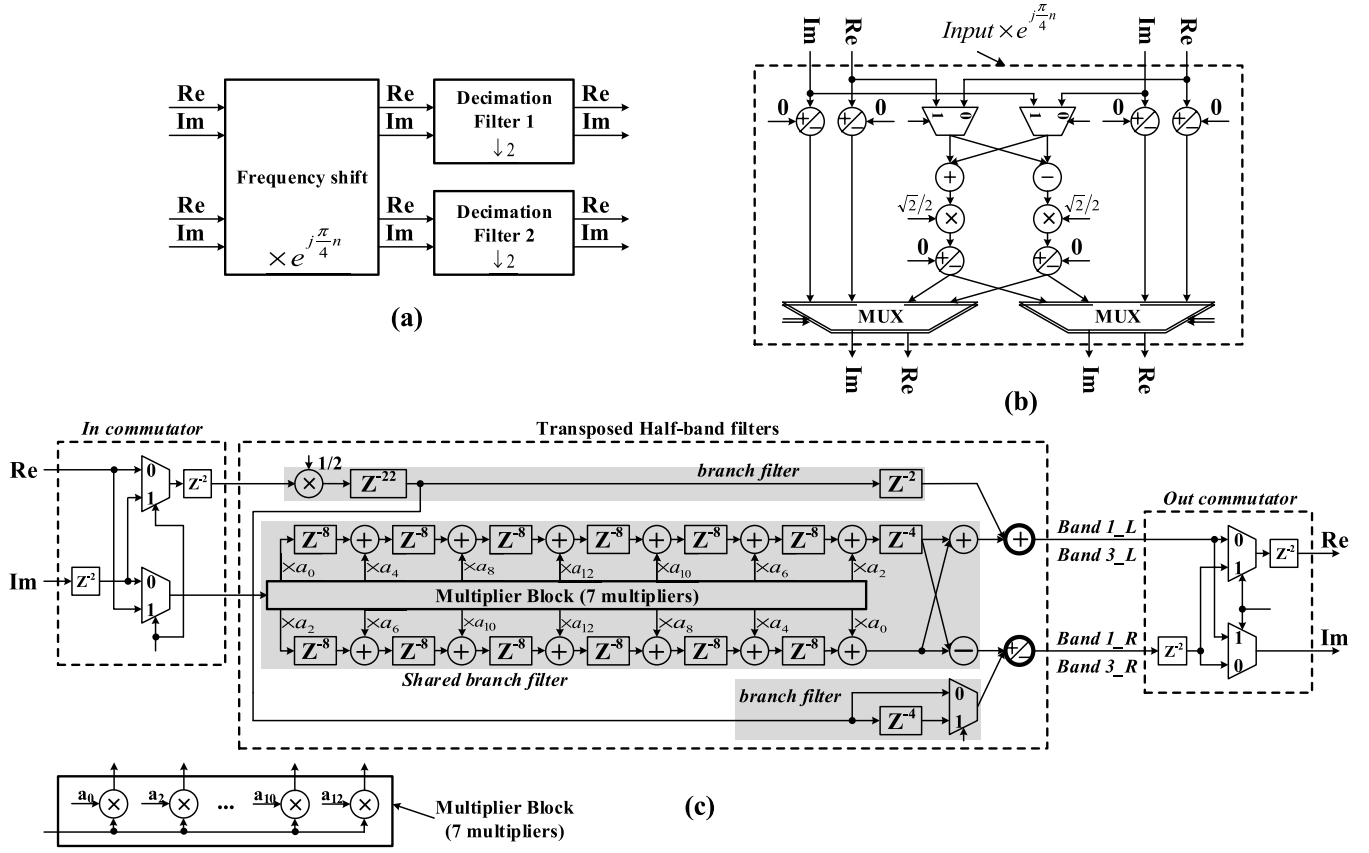


Fig. 7. (a) Structure of PE 2. (b) Structure of frequency shift. (c) Structure of decimation filter.

Here, we divide $y_{B2_D_r}(n)$ and $y_{B4_D_r}(n)$ into two groups by parity.

According to (4)–(6), the convolution computations can share the same branch filter (see Fig. 5). Considering the symmetry properties of $h_E(n)$, the number of multiplications can be reduced by half. The implementation complexity can be further reduced using the MB technique [15]. By reversing the sign of the Up output data of band 1, we can get the Up output data of band 3. In the same way, we can get the Up output data of band 4 from the Up output data of band 2. By replacing the final adder with subtractor, we can get the filtered data of band 3 from the output results of two branch filters of band 1. In the same way, the filtered data of band 4 can be obtained from the output results of two branch filters of band 2. Finally, the filtered data of bands 1, 2, 3, and 4 shuffled by *Out commutator* are exported in the complex format. The data shuffling process of *Out commutator* is similar to that of *In commutator* (see Fig. 6). The Up output sequence of *Out commutator* is $\{(x_{B1_r}(0), x_{B1_i}(0)), (x_{B3_r}(0), x_{B3_i}(0)) \dots (x_{B1_r}(n), x_{B1_i}(n)), (x_{B3_r}(n), x_{B3_i}(n)) \dots\}$, where $x_{B1_r}(n)$ and $x_{B3_r}(n)$ are the real parts of bands 1 and 3, and $x_{B1_i}(n)$ and $x_{B3_i}(n)$ are the imaginary parts of bands 1 and 3. Similarly, the Down output sequence of *Out commutator* is the data of band 2 followed by the data of band 4.

Band 1 is divided into two bands, called bands 1_L and 1_R [see Fig. 4(b)], and so are bands 2, 3, and 4. The center frequencies of bands 1_L and 1_R are $(-\pi/4)$ and $\pi/4$.

PE 2 consists of one frequency shift block and two decimation filters [see Fig. 7(a)]. After frequency shifting, the data corresponding to bands 1_L and 1_R are $[x_{B1_r}(n) + jx_{B1_i}(n)] \times e^{j\pi n/4}$ and $[x_{B1_r}(n) + jx_{B1_i}(n)] \times e^{j\pi n/4} \times e^{-j\pi n/2}$, respectively. The computation process of bands 1_L and 1_R is similar to that of bands 1 and 4 in Table II. It means that we can get the filtered data of band 1_R from the output results of two branch filters of band 1_L. Thus, the frequency shift block only needs to achieve $\pi/4$ frequency shift for bands 1, 2, 3, and 4. The utilization of complex multiplier for $\pi/4$ frequency shift is only 50%, so it can be designed by time-multiplexing the shared multiplications [see Fig. 7(b)]. Referring to the structure of PE 1, the structure of a decimation filter is shown in Fig. 7(c), and each filter is time-multiplexed by two bands. As shown in Fig. 4(c), band 1_L is further divided into two bands, called bands 1_L_L and 1_L_R, and so are bands 1_R, 2_L, 2_R, 3_L, 3_R, 4_L, and 4_R. Referring to the structure of PE 2, we can get the structures of PEs 3, 4, 5, 6, 7, 8, and 9. The difference between PE 2 and PE s ($s = 3, 4 \dots 9$) is the size of storage units of the decimation filters, which depends on the number of output bands. In PE s ($s = 2, 3 \dots 9$), the number of output bands is 2^{s+1} .

If the center frequency of the target signal lies on boundary between the output bands of the TPFT block, it cannot be channelized. In order to eliminate these channelization blind zones, the passband of the prototype filter is extended by 50% of the antialiasing bandwidth, while maintaining

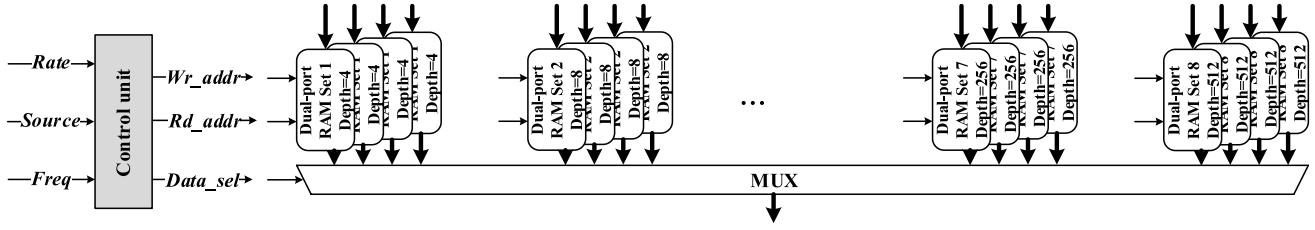


Fig. 8. Structure of interleaver.

the same stopband attenuation. This leads to spectral overlap regions (the shaded regions in Fig. 4). In the current design, the input sampling rate and the antialiasing bandwidth of PE 1 are limited to F_s and $F_s/20$ ($=12.8$ MHz for $F_s = 256$ MS/s), so the passband cutoff frequency of half-band filters in PE 1 is set to $(\pi/4 + \pi/20)$, where $\pi/20$ is 50% of the antialiasing bandwidth normalized with respect to the input sampling rate. The passband ripple and the stopband attenuation are set to 0.1 and 70 dB. According to MATLAB FDATool software, the required filter order is 23. Considering the symmetry of the nonzero coefficients, the number of multiplications is 6. In PE s ($s = 2, 3, \dots, 9$), the input sampling rate and the antialiasing bandwidth of each band are $F_s/2^{s-1}$ and $F_s/(20 \times 2^{s-2})$, respectively. Thus, the passband cutoff frequency of a half-band filter is $(\pi/4 + \pi/10)$. The passband ripple and the stopband attenuation are set to 0.1 and 70 dB, so the corresponding filter order is 27. According to the symmetry properties, the number of multiplications is 7.

C. Interleaver

The output results of all the PEs are available because of the cascaded architecture. Hence, it is possible to extract frequency bands with different bandwidths and center frequency. Let Freq be the center frequency control word. Freq is calculated as follows:

$$\text{Freq} = \text{Round} \left(2^{32} \times \frac{F_c}{F_s} \right) \quad (7)$$

where F_c is the center frequency of the target signal. When F_s is 256 MS/s, the center frequency resolution is 0.0596 Hz. The coarse channelization block includes two interleavers, each of which can generate up to 512 channels of complex data. The structure of interleaver is shown in Fig. 8. It is composed of 32 dual-port RAMs, 1 multiplexer, and 1 control unit. The output results of PEs 2, 3, ..., and 9 are written into dual-port RAM Sets 1, 2, ..., and 8, respectively. The depth of each dual-port RAM is equal to one half of the number of the output bands of a PE stage. For example, PE 2 produces eight bands, i.e., bands 1_L, 1_R, 2_L, 2_R, 3_L, 3_R, 4_L, and 4_R. They are written into two dual-port RAMs, and the depth of dual-port RAM is 4. According to the control word Rate, Source, and Freq of each channel, the control unit generates the corresponding read addresses for the 32 dual-port RAMs and performs a selection among them.

We use an example to explain the working principle of the control unit. Assuming that the interleaver exports six channels

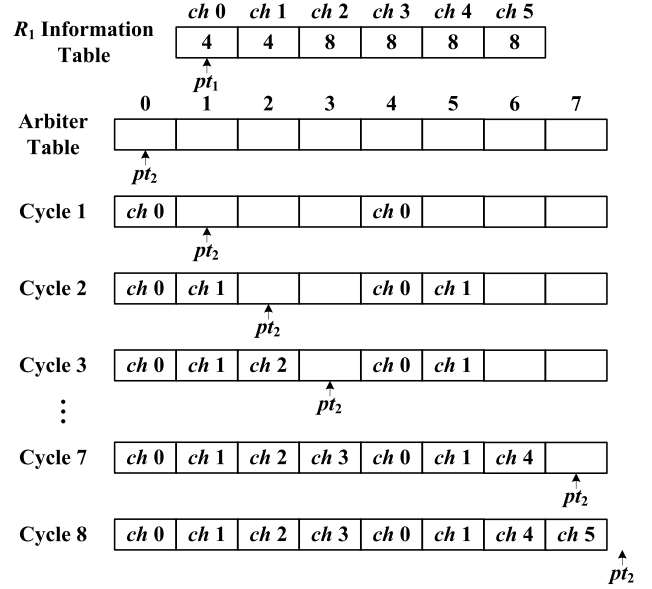


Fig. 9. Principle of the control unit of interleaver.

of complex data, called channels 0, 1, 2, 3, 4, and 5, and their decimation factors R_1 are 4, 4, 8, 8, 8, and 8, respectively. The goal of the control unit is to generate an arbitrator table, which assigns time slots of different lengths to the six channels. The size of the arbitrator table is equal to the maximum value of the decimation factors R_1 of all the channels. In this example, it is 8. Let pt_1 and pt_2 be the pointers, and their initial values are 0, which point to the heads of the R_1 information table and arbitrator table, respectively. The control unit repeatedly performs the following operations until the arbitrator table is full or all of the channels are arranged (see Fig. 9).

- 1) Read the decimation factors R_1 from address pt_1 .
- 2) If address pt_2 is not occupied, the control unit writes pt_1 into addresses $pt_2 + k \times R_1$ ($k = 0, 1, 2, \dots$). It then executes the update operations for pt_1 and pt_2 , i.e., $pt_1 = pt_1 + 1$ and $pt_2 = pt_2 + 1$.
- 3) If address pt_2 is occupied, the control unit only executes the update operation for pt_2 , i.e., $pt_2 = pt_2 + 1$.

D. Tuning Unit

The tuning unit is composed of two parallel 512-channel mixers (see Fig. 2). Each mixer consists of one 512-channel NCO and one complex multiplication (see Fig. 10). For the design of NCO, interested readers are referred to [19]. Note that the design in [19] can only support one channel. The frequency control word Freq is used by 512-channel NCO

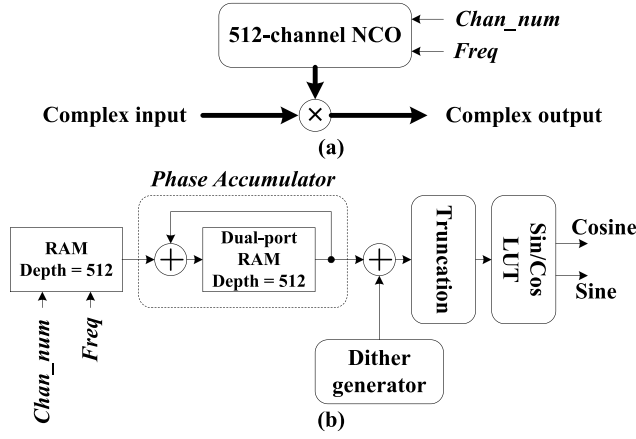


Fig. 10. (a) Structure of mixer. (b) Structure of 512-channel phase dithered NCO.

to generate up to 512 independent tuning signals. Tuning signals must be generated based on the arbiter table of the interleaver. It must be consistent with the output order of a TPFT-based coarse channelization block. The complex multiplication utilizes the tuning signals to move the target signals from their original frequency location to the zero frequency location.

E. VFD Filter

The first stage of the resampling filter consists of two parallel Farrow-based VFD filters (see Fig. 2). A VFD filter is used to implement arbitrary SRC. The SRC factor R_2 is in the range $[1, 2)$. The maximum input sampling rate of each VFD filter is $F_s/4$ (decimation by R_1 in the coarse channelization, $R_1 \geq 4$), and the corresponding antialiasing output bandwidth is $F_s/20$. Thus, the frequency response of a VFD filter is as follows:

$$H_{\text{VFD}}(e^{j\omega}) = \begin{cases} e^{-j\omega(D+\phi)}, & 0 \leq |\omega| \leq 0.2\pi \\ 0, & 0.8\pi \leq |\omega| \leq \pi \end{cases} \quad (8)$$

where $D + \phi$ ($\phi \in [0, 1]$), 0.2π , and 0.8π are the group delay, the passband and the stopband edges of the VFD filter, respectively. The transfer function of the VFD filter can be expressed by the following equation:

$$H_{\text{VFD}}(z, \phi) = \sum_{l=0}^{L-1} \left[\sum_{n=0}^{N-1} c_l(n) z^{-n} \right] \phi^l \quad (9)$$

where L is the order of a VFD filter, N is the length of a subfilter, and $c_l(n)$ is the coefficient of a subfilter. According to the method in [20] and [21], the order of a VFD filter is 3, and the length of a subfilter is 16.

The frequency response of the VFD filter is shown in Fig. 11, and the corresponding subfilter coefficients are given in Table III. The antialiasing stopband attenuation of the VFD filter can reach to 80 dB. The VFD filter can be implemented using the Farrow structure [22]. All the subfilters are implemented in transposed forms, so the input data are directly multiplied with all the subfilter coefficients [17]. The SOPOT coefficients [14] and the MB technique [15]

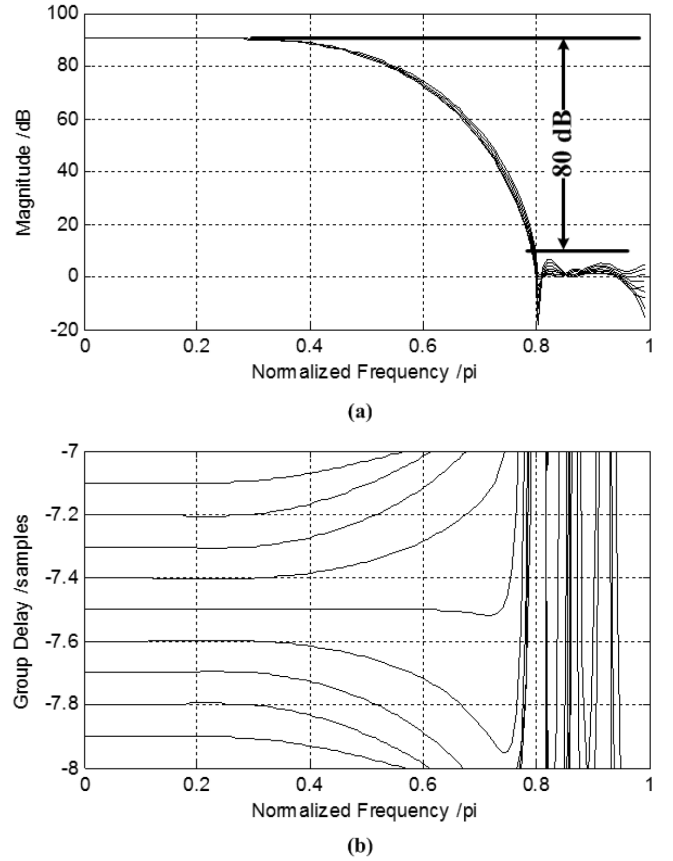


Fig. 11. (a) Frequency response and (b) group delays of a VFD filter with $\phi = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

TABLE III
SUBFILTER COEFFICIENTS OF VFD FILTER

n	$c_0(n)$	$c_1(n)$	$c_2(n)$
0	-78	27	53
1	1	-161	82
2	553	-231	-334
3	-5	1188	-639
4	-2245	1280	1013
5	-12	-5321	3154
6	10014	-10986	887
7	16448	-2378	-4218
8	9852	10813	-4218
9	-85	9211	887
10	-2179	-987	3154
11	48	-3306	1013
12	544	89	-639
13	-12	898	-334
14	-77	-4	82
15	2	-133	53

can be used to minimize the total number of additions and multiplications. As a result, the number of additions used for the SOPOT coefficients can be reduced to 41.

The architecture of a VFD filter is shown in Fig. 12(a). The real or imaginary parts of 512-channel complex input data can share the same VFD filter, so the storage units of subfilters are implemented by RAMs with depths of 512. In the current

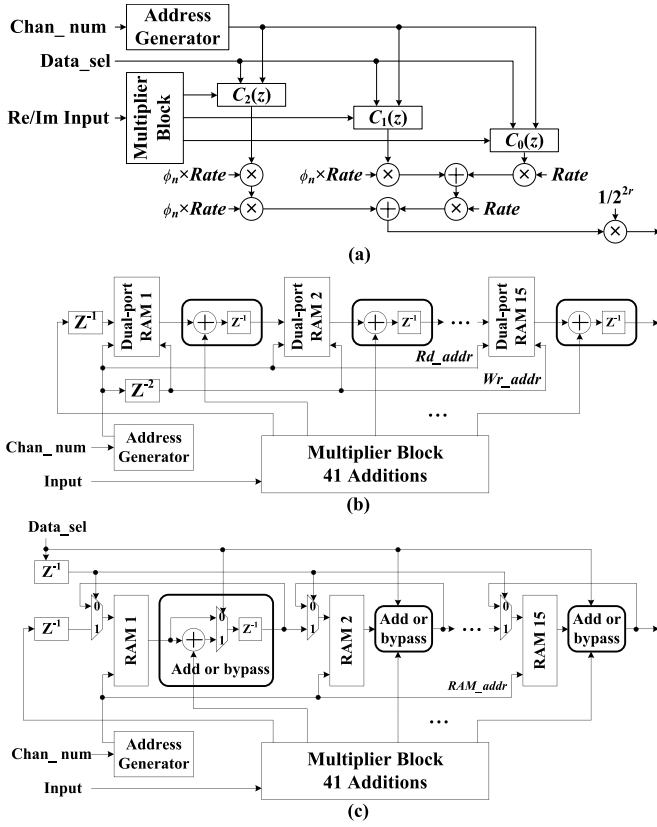


Fig. 12. (a) Structure of VFD filter. (b) First structure of subfilter $C_2(z)$. (c) Second structure of subfilter $C_2(z)$.

design, the fractional delay ϕ is expressed as follows:

$$\phi_{n+1}' = \begin{cases} \phi_n - 1, & \phi_n \geq 1 \\ \phi_n + (R_2 - 1) = \phi_n - 1 + (2^r / \text{Rate}), & \phi_n < 1 \end{cases} \quad (10)$$

where Rate is in the range $(2^{r-1}, 2^r]$ (see Table I) and ϕ_0 is 0. When $\phi_n \geq 1$, the filtered result of a VFD filter is discarded.

The parameter Rate of each channel is configurable, so the division operation in (10) is unavoidable. Fortunately, it can be removed by adjusting the gain of the transfer function $H_{\text{VFD}}(z, \phi)$. Note that this gain variation can be compensated by the gain module in the resampling filter. The new transfer function is expressed as follows:

$$H_{\text{New_VFD}}(z, \phi) = \frac{\text{Rate} \times \text{Rate}}{2^r \times 2^r} H_{\text{VFD}}(z, \phi). \quad (11)$$

The subfilters of VFD filters can be implemented by two architectures [see Fig. 12(b) and (c)], respectively. The first one is operated as follows.

- 1) When a valid data $x_m(n)$ of channel m comes, the MB generates all the results, which are the products of $x_m(n)$ multiplied with all the subfilter coefficients $c_0(0), c_0(1), \dots, c_0(15), c_1(0), c_1(1), \dots, c_1(15), c_2(0), c_2(1), \dots, c_2(15)$.
- 2) The product $x_m(n) \times c_0(i)$ ($0 \leq i \leq 15$) is first added with the intermediate result read from the address m of the buffer i , and then, the sum is written into the same

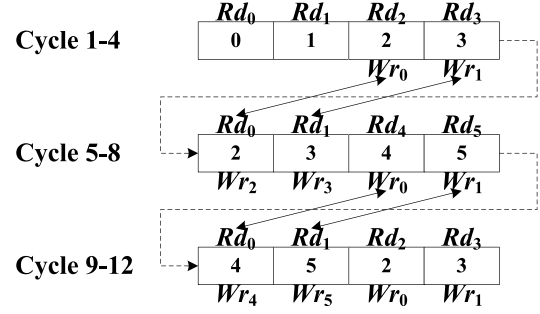


Fig. 13. Map between RAM address and channel number.

address m of buffer $i + 1$. Considering the latencies caused by the buffer read operation and the addition, the buffer should be implemented using dual-port RAM.

Compared with the first architecture, the buffer in the second one is implemented using single-port RAM. Besides the buffer, the computation unit called add or bypass is the other major difference. The subfilter is shared by 512 input channels. When the time slice is switched to channel m , the computation process of the second architecture is described as follows.

- 1) If $x_m(n)$ is a valid data, the MB generates all the products of $x_m(n)$ multiplied with all the subfilter coefficients $c_0(0), c_0(1), \dots, c_0(15), c_1(0), c_1(1), \dots, c_1(15), c_2(0), c_2(1), \dots, c_2(15)$. The product $x_m(n) \times c_0(i)$ ($0 \leq i \leq 15$) is first added with the intermediate result read from the address j of the buffer i . Considering the latencies of the buffer read operation and the addition, the sum is then written into the address $j + 2$ of buffer $i + 1$.
- 2) If $x_m(n)$ is an invalid data, the intermediate result is still read from the address j of the buffer i . Considering the latencies, it is then written into the address $j + 2$ of the original buffer i .

In the first architecture, the data of channel m is kept in the address m . However, the address in the second architecture, corresponding to channel m , is variable. Assuming that channels 0, 1, 2, 3, 4, and 5 enter the VFD filter in the order shown in Fig. 9, and the initial RAM address of channel m is set to m . The map relationship between the RAM address and the channel number m is shown in Fig. 13. Rd_m indicates the read address of the data of channel m , and Wr_m indicates the write address of the data of channel m .

F. 2/4-Phase Filter

The second stage of the resampling filter consists of two parallel 2/4-phase filters. After decimation by $(R_1 \times R_2)$, the minimum input sampling rate of each 2/4-phase filter is $F_s/2R_1$ ($1 \leq R_2 < 2$), and the maximum antialiasing output bandwidth is limited to 80% of the output sampling rate. The decimation factor R_3 is 2 or 4. When R_3 is 2, the passband and stopband edges are set to 0.4π and 0.6π , and the passband ripple and the stopband attenuation are set to 0.1 and 70 dB, respectively. According to FDATool software, the required filter order is 32, so each phase includes 16 filter coefficients. When R_3 is 4, the passband and stopband edges are set to 0.2π

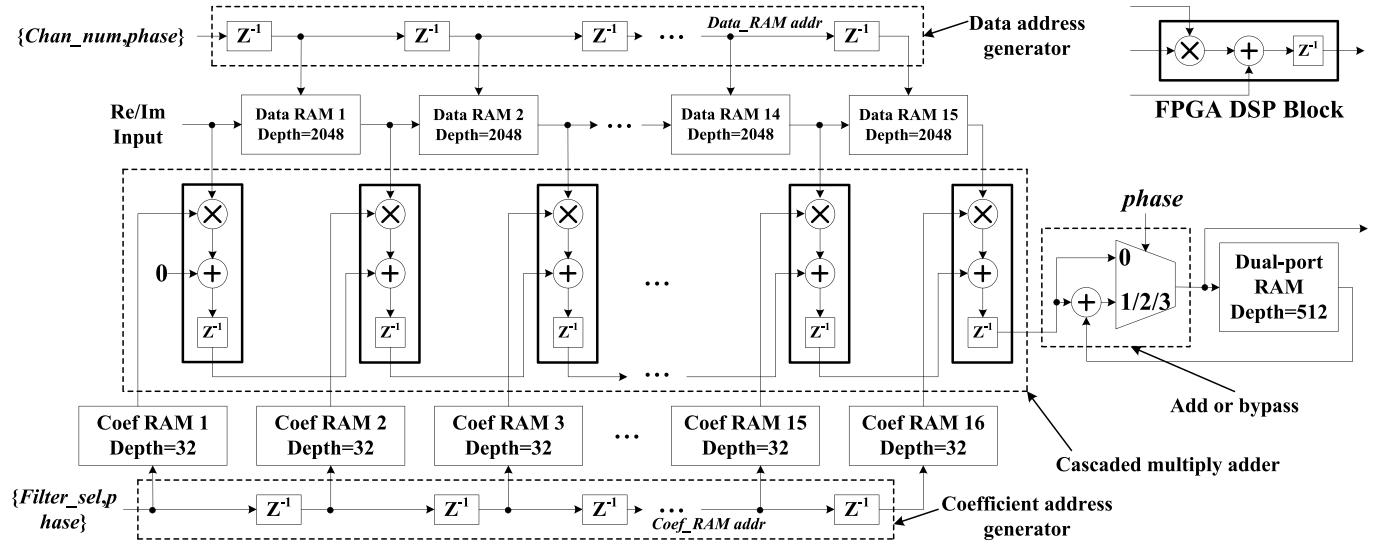


Fig. 14. Structure of 2/4-phase filter.

TABLE IV
HARDWARE RESOURCES FOR THE PROPOSED ARCHITECTURE

Hardware Resource	Coarse Channelization block	Tuning units	Farrow-based VFD filters	2/4-phase filters	Gain	The whole design
Registers	46245	369×2	7668×2	1113×2	273	64818
6-input LUTs	36995	199×2	5780×2	840×2	49	50682
18Kb Block RAMs	170	4×2	48×2	62×2	1	399
DSP48Es	268	3×2	10×2	32×2	2	360

and 0.3π . According to FDATool software, the required filter order is 64, so each phase still includes 16 coefficients. The real or imaginary parts of 512-channel complex input data can share the same 2/4-phase filter. For $R_3 = 4$, the storage units are implemented by RAMs of size 2048. The storage units for filter coefficients are implemented by 16 RAMs of size 4×8 , so the proposed design can store eight predefinable filters. The filter for each channel may be independently selected using the filter selection control word *Filter_sel*, and the eight filters are all programmable at runtime. The architecture of a 2/4-phase filter is shown in Fig. 14. It consists of one data address generator, one coefficient address generator, fifteen data RAMs, sixteen coefficient RAMs, one cascaded multiply adder, one add or bypass unit, and one dual-port RAM. To take use of the DSP block in the FPGA, the convolution is implemented using the cascaded multiply adder. As a result, the addresses of data and coefficient RAMs are implemented using shift registers.

The third stage of the resampling filter consists of one multiplexer and one gain module. The gain for each channel is configurable at runtime.

IV. VALIDATION AND COMPARISON

A. Functionality Test

The development of the proposed solution included three phases: 1) simulation using MATLAB; 2) design and FPGA implementation with development tools; and 3) validation using MATLAB and ModelSim.

We implement the proposed channelization architecture on the Xilinx Virtex-6 FPGA XCV6SX315T-2ff1156, and the corresponding hardware resource is summarized in Table IV. Results are obtained from Xilinx ISE 13.4 after place and route analysis. The input bitwidth is 15 bit, and the bitwidths of the TPFT-based filter banks, the tuning unit, the VFD filters, and the 2/4-phase filters are all 18 bit. The contribution of each component of the proposed architecture is also shown in Table IV. The maximum clock rate for our design is 279 MHz. According to (2) and (7), the center frequency tuning accuracy and the output sampling rate resolution are both within 0.065 Hz.

The functionality of the proposed solution is tested as follows. A wideband signal is generated by MATLAB. It is then fed into the FPGA implementation. After simulation using ModelSim, the output results are fed back to MATLAB for analysis. As shown in Fig. 15, the proposed design can extract all the subbands from the input signal, and perform resampling for these subbands by setting the Freq and Rate parameters. The antialiasing stopband attenuation of all the filters in the datapath is larger than 70 dB. These filters include the transposed half-band filters in the TPFT blocks, the Farrow-based VFD filters, and the 2/4-phase filters in the resampling filter. From Fig. 15, we find that the minimum antialiasing stopband attenuation is 67.8 dB, which is consistent with a theoretical analysis. We impose two constraints on the current design, i.e., the maximum antialiasing output bandwidth and the maximum output sampling rate. Note that the channel parameters mentioned above, including antialiasing

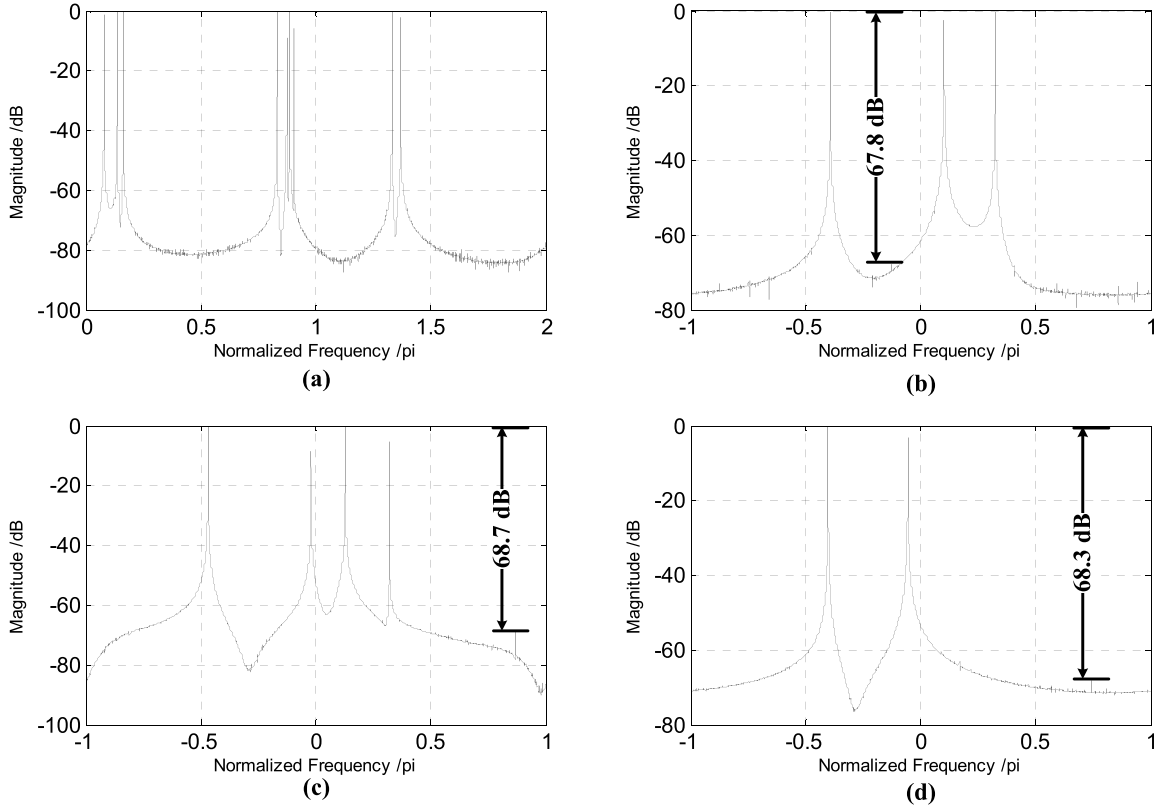


Fig. 15. (a) Input signal. (b) Extracted channel 1 with $R_1 = 4$, $R_2 = 4/3$, and $R_3 = 2$. (c) Extracted channel 2 with $R_1 = 4$, $R_2 = 8/7$, and $R_3 = 2$. (d) Extracted channel 3 with $R_1 = 4$, $R_2 = 16/15$, and $R_3 = 2$.

output bandwidth, the maximum output sampling rate, frequency and sampling rate precision, and SFDR, can be modified for a specific application.

B. Channel Reconfiguration

The parameters of each channel much be programmed. They are organized into register blocks, with 1024 addresses for supporting 1024 channels. The reconfiguration includes two phases: 1) the update of channel parameters and 2) the generation of arbiter table. Each channel takes one clock cycle to update its channel parameters, so the number of clock cycles updating all the channel parameters is equal to the total number of output channels. Once the update of channel parameters is completed, the control unit in the interleaver is responsible to generate an arbiter table. The control unit repeatedly performs the writing operations until the arbiter table is full or all of the channels are arranged (see Fig. 9). The size of the arbiter table is equal to the maximum value of R_1 of all the channels. Therefore, the reconfiguration time is expressed as follows:

$$T = [N_c + \max(R_1)]/F_s \quad (12)$$

where N_c is the total number of output channels and $\max(R_1)$ is the maximum value of R_1 of all the channels, which directly determines the size of the arbiter table.

C. Comparison With Other Existing Solutions

The number of multiplications is usually used as a metric of area complexity. The nonuniform filter bank in [8]

divides the spectrum from 0 to $F_s/2$ into nine channels. From Table II in [8], we find that the minimum and maximum antialiasing output bandwidths are $0.0311 \times F_s/2$ and $0.1638 \times F_s/2$, respectively, and the corresponding stopband attenuation is 40 dB. For a fair comparison with other existing solutions, we simplify the proposed architecture (see Fig. 16), which can process one channel of complex input data and provide up to 32 channels. Its maximum antialiasing output bandwidth is limited to $F_s/10$ ($>0.1638 \times F_s/2$). The passband cutoff frequency of half-band filters in PEs 1, 2, 3, and 4 is set to $(\pi/4 + \pi/10)$, where $\pi/10$ is 50% of the antialiasing bandwidth. The passband ripple and the stopband attenuation are set to 0.1 and 45 dB. According to FDATool software, the required filter order is 15, so the number of multiplications of the shared branch filter is 4. Considering the multiplications in the frequency shift block, the number of multiplications of the coarse channelization block is $4 + (2 + 4 \times 2) \times 3 = 34$. The number of multiplications of the tuning unit is $3 \times 2 = 6$. The passband and stopband edges of a VFD filter are set to 0.2π and 0.8π , and the stopband attenuation is set to 40 dB. According to the method in [20] and [21], the order of a VFD filter is 3, and the length of a subfilter is 8. Considering the symmetry properties of the coefficients, the number of multiplications used for the convolution computation of three subfilters is 20. From Fig. 12(a), we conclude that the number of multiplications of the resampling filter is $4 \times (20 + 5) = 100$. Thus, the total number of multiplications of the simplified channelization architecture is $34 + 6 + 100 = 140$.

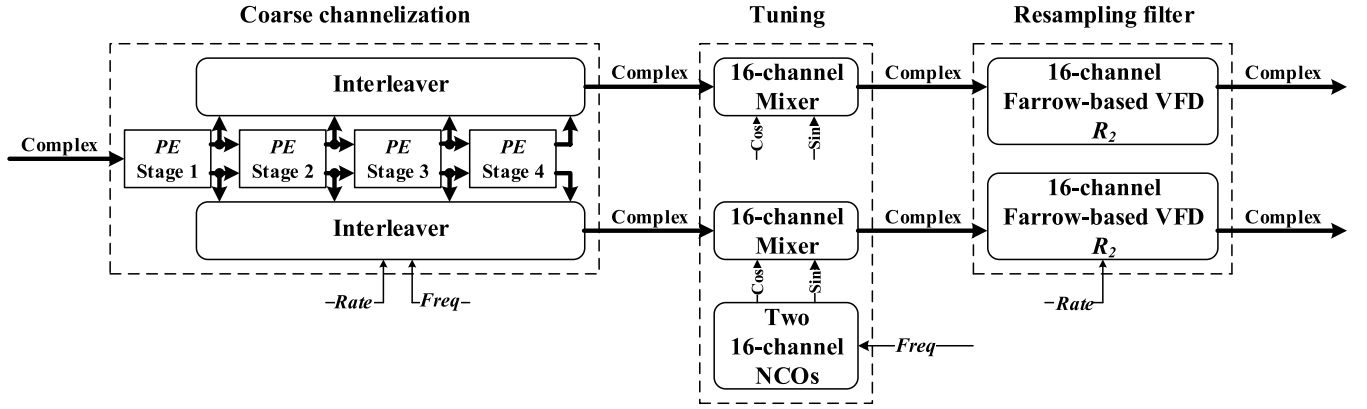


Fig. 16. Simplified architecture for a fair comparison.

TABLE V
NUMBER OF MULTIPLICATIONS FOR DIFFERENT
CHANNELIZATION SOLUTIONS

Channelization solution	No. of multiplications	% saving
Goertzel channelization [3]	416	66.3
DFT filter bank [3]	224	37.5
Analysis/synthesis filter bank [9]	448	68.7
Non-uniform filter bank [8]	179	21.7
Proposed solution in Fig. 16	140	-

TABLE VI
IMPLEMENTATION RESULTS FOR DIFFERENT
CHANNELIZATION SOLUTIONS

Standard	DFT filter bank [3]	Non-uniform filter bank [8]	Proposed solution in Fig. 16
Device	Virtex II	Virtex II	Virtex IV
Slices	11865	10157	9557
Latency(ns)	26.549	30.12	190–525

Note that the spectrum of our design is in the range 0 to F_s , which is twice as width as that of the design in [8]. Table V shows that our design offers the multiplication reduction of 21.7% over the nonuniform filter bank in [8]. We implement the simplified channelization architecture on Xilinx Virtex IV FPGA XC4VSX35-11ff668 with the 16-bit precision. From Table VI, we find that the proposed solution offers slices reduction of 5.9% over the nonuniform filter bank and 19.4% over the DFT filter bank. The minimum and maximum latencies of the proposed solution in Fig. 16 are $(38/F_s)$ and $(105/F_s)$, respectively. When $F_s = 200$ MHz, the minimum and maximum latencies are 190 and 525 ns, which are much higher than that of the design in [8]. The reason for this is the cascaded implementation of the coarse channelization and the time-multiplexing technique used in the whole design. Although our solution has a higher latency than the nonuniform filter bank and DFT filter bank, it is a better choice for digital channelization due to the advantages of dynamic configuration and area saving over other existing solutions.

V. CONCLUSION

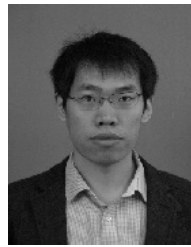
A reconfigurable SDR channelization architecture that can provide up to 1024 channels of complex downconverted output data was presented. Each channel is independently

configurable at runtime and can be programmed to support multistandard digital downconversion. These configurable parameters include input source, center frequency, output sampling rate, filter response, and gain. The proposed solution can filter individual channels of arbitrary center frequency to a required standard. It can replace multiple DDC application-specified integrated circuit devices with a single FPGA, significantly reducing the power, size, weight, and cost. Therefore, the proposed architecture is a better solution for SDR applications, such as instrumentation, electronic warfare, and broad-spectrum surveillance.

REFERENCES

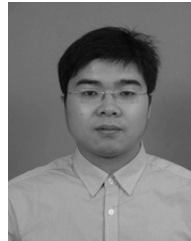
- [1] G. López-Risueño, J. Grajal, and A. Sanz-Osorio, "Digital channelized receiver based on time-frequency analysis for signal interception," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 879–898, Jul. 2005.
- [2] Z. Wang, X. Liu, B. He, and F. Yu, "A combined SDC-SDF architecture for normal I/O pipelined radix-2 FFT," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 5, pp. 973–977, May 2015.
- [3] T. Hentschel, "Channelization for software defined base-stations," *Ann. Telecommun.*, vol. 57, nos. 5–6, pp. 386–420, May/Jun. 2002.
- [4] J. Lillington, "Flexible channelisation architectures for software defined radio front ends using the tuneable pipelined frequency transform," in *Proc. IEEE Colloq. DSP Enabled Radio*, Sep. 2003, pp. 1–13.
- [5] R. Mahesh and A. P. Vinod, "Reconfigurable low area complexity filter bank architecture based on frequency response masking for nonuniform channelization in software radio receivers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1241–1255, Apr. 2011.
- [6] R. Mahesh and A. P. Vinod, "Low complexity flexible filter banks for uniform and non-uniform channelisation in software radios using coefficient decimation," *IET Circuits Devices Syst.*, vol. 5, no. 3, pp. 232–242, May 2011.
- [7] R. Mahesh, A. P. Vinod, E. M.-K. Lai, and A. Omondi, "Filter bank channelizers for multi-standard software defined radio receivers," *J. Signal Process. Syst.*, vol. 62, no. 2, pp. 157–171, Feb. 2011.
- [8] S. J. Darak, A. P. Vinod, and E. M.-K. Lai, "A low complexity reconfigurable non-uniform filter bank for channelization in multi-standard wireless communication receivers," *J. Signal Process. Syst.*, vol. 68, no. 1, pp. 95–111, Jul. 2012.
- [9] W. A. Abu-Al-Saud and G. L. Stuber, "Efficient wideband channelizer for software radio systems using modulated PR filterbanks," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2807–2820, Oct. 2004.
- [10] T. I. Laakso, V. Valimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, Jan. 1996.
- [11] B. H. Tietche, O. Romain, and B. Denby, "A practical FPGA-based architecture for arbitrary-ratio sample rate conversion," *J. Signal Process. Syst.*, vol. 78, no. 2, pp. 147–154, Feb. 2015.
- [12] A. Y. Kwentus, Z. Jiang, and A. N. Willson, Jr., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 457–467, Feb. 1997.

- [13] 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. Sel. Areas Commun.*, vol. 17, no. 4, pp. 551–560, Apr. 1999.
- [14] 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.
- [15] 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.
- [16] S. C. Chan, K. M. Tsui, and T. I. Yuk, "Design and complexity optimization of a new digital IF for software radio receivers with prescribed output accuracy," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 2, pp. 351–366, Feb. 2007.
- [17] C. K. S. Pun, Y. C. Wu, S. C. Chan, and K. L. Ho, "On the design and efficient implementation of the Farrow structure," *IEEE Signal Process. Lett.*, vol. 10, no. 7, pp. 189–192, Jul. 2003.
- [18] D. Gupta *et al.*, "Digital channelizing radio frequency receiver," *IEEE Trans. Appl. Supercond.*, vol. 17, no. 2, pp. 430–437, Jun. 2007.
- [19] N. Lashkarian, E. Hemphill, H. Tarn, H. Parekh, and C. Dick, "Reconfigurable digital front-end hardware for wireless base-station transmitters: Analysis, design and FPGA implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 8, pp. 1666–1677, Aug. 2007.
- [20] 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.
- [21] 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," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 11, pp. 689–703, Nov. 2002.
- [22] C. W. Farrow, "A continuously variable digital delay element," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 1988, pp. 2641–2645.



Xue Liu received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively.

He is currently an Assistant Professor with the Institute of Cyber-Physical System Engineering, Northeastern University, Shenyang, China. His current research interests include field-programmable gate array-based design and software defined radio.



Ze-Ke Wang received the B.Sc. degree from the Harbin University of Science and Technology, Harbin, China, in 2006, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2011.

He is currently a Research Fellow with the Parallel & Distributed Computing Center, Nanyang Technological University, Singapore. His current research interests include high performance computing and database systems.



Qing-Xu Deng received the Ph.D. degree from Northeastern University, Shenyang, China, in 1997.

He is currently a Full Professor with the Institute of Cyber-Physical System Engineering, Northeastern University. His current research interests include multiprocessor real-time scheduling and formal methods in real-time system analysis.