# On the Relationship Between the Overlapping Rounding Transform and Lifting Frameworks for Reversible Subband Transforms

Michael D. Adams and Faouzi Kossentini

**Abstract**

Recently, a new framework for reversible subband transforms based on the overlapping rounding transform (ORT) has been proposed as an alternative to the lifting framework. In this correspondence, we show that the ORT framework is, in fact, a special case of the lifting framework with only trivial extensions.

**Keywords**

Reversible integer-to-integer subband transforms, lifting, overlapped rounding transform.

## I. Introduction

In order to efficiently handle lossless coding in subband coding systems, we require transforms that are invertible in finite-precision arithmetic. Such transforms are said to be reversible. In [1], Calderbank et al. showed that the lifting scheme [2] forms an effective framework for constructing reversible transforms. Transforms utilizing this framework have since found application in numerous coding systems including that of the emerging JPEG-2000 standard [3], [4]. More recently, Jung and Prost [5] have proposed an alternative method for constructing reversible transforms based on the overlapping rounding transform (ORT). Although the ORT and lifting frameworks appear quite different at first glance, they are, in fact, intimately related. In what follows, we will show that the ORT framework is, in fact, a special case of the lifting framework with only trivial extensions.

## II. Notation and Other Preliminaries

Before proceeding further, a short digression concerning the notation used in this paper is appropriate. The symbols $\mathbb{Z}$ and $\mathbb{R}$ denote the sets of integer and real numbers, respectively. Matrix and vector quantities are indicated using bold type. For $x \in \mathbb{R}$, the notation $\lfloor x \rfloor$ denotes the largest integer not more than $x$ (i.e., the floor function), and the notation $\lceil x \rceil$ denotes the smallest integer not less than $x$ (i.e., the ceiling function). The $z$-transform of a sequence $x[n]$ is denoted $X(z)$. For convenience, we also define the quantities

$$\lfloor X(z) \rfloor_z \triangleq \sum_{n \in \mathbb{Z}} \lfloor x[n] \rfloor z^{-n} \quad \text{and}$$
$$\lceil X(z) \rceil_z \triangleq \sum_{n \in \mathbb{Z}} \lceil x[n] \rceil z^{-n}$$

Manuscript received October 6, 1998; revised March 22, 1999. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was Dr. Shubha Kadambe.

The authors are with the Signal Processing and Multimedia Group, Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, B.C., Canada V6T 1Z4 (e-mail: mdadams@ece.ubc.ca).

Publisher Item Identifier XXX.

(i.e., $\lfloor x[n] \rfloor \leftrightarrow \lfloor X(z) \rfloor_z$ and $\lceil x[n] \rceil \leftrightarrow \lceil X(z) \rceil_z$). Lastly, we note some simple yet important properties of floor and ceiling operations. For $x \in \mathbb{Z}$ and $\gamma \in \mathbb{R}$, the following identities hold:

$$\lfloor x + \gamma \rfloor = x + \lfloor \gamma \rfloor, \tag{1}$$

$$\lceil x + \gamma \rceil = x + \lceil \gamma \rceil, \text{ and} \tag{2}$$

$$\lfloor -\gamma \rfloor = - \lceil \gamma \rceil. \tag{3}$$

Also, the floor and ceiling operators for sequences are shift invariant. That is, we have that

$$\begin{aligned} \left\lfloor z^K X(z) \right\rfloor_z = z^K \lfloor X(z) \rfloor_z \quad \text{and} \\ \left\lceil z^K X(z) \right\rceil_z = z^K \lceil X(z) \rceil_z \end{aligned} \tag{4}$$

where $K \in \mathbb{Z}$.

## III. Reversible Subband Transforms

In the context of this paper, we consider the class of subband transforms that can be computed using a 2-channel uniformly maximally-decimated filter bank (UMDFB). Although only the 2-channel case is explicitly considered, the ideas discussed herein are also directly applicable to UMDFBs with more than two channels.

To construct a reversible transform, the filter bank is realized in its polyphase form. More specifically, the analysis polyphase matrix $\boldsymbol{E}(z)$ is factored as

$$\boldsymbol{E}(z) = \boldsymbol{E}_{N-1}(z) \cdots \boldsymbol{E}_1(z) \boldsymbol{E}_0(z)$$

where the $\boldsymbol{E}_i(z)$'s are chosen such that they correspond to filtering networks that can be made reversible with the appropriate introduction of rounding. For integer-to-integer mappings, the rounding is done in such a way as to ensure that each filtering block yields integer output sequences for arbitrary integer input sequences. The synthesis polyphase filtering is then simply forced to be the stepwise inverse of the analysis polyphase filtering. That is, the synthesis polyphase matrix $\boldsymbol{R}(z)$ is realized in terms of the decomposition

$$\boldsymbol{R}(z) = \boldsymbol{E}^{-1}(z) = \boldsymbol{E}_0^{-1}(z) \boldsymbol{E}_1^{-1}(z) \cdots \boldsymbol{E}_{N-1}^{-1}(z). \tag{5}$$

Since each filtering block is reversible and maps integers to integers, the resulting transform computed by the filter bank also has these properties.

## IV. ORT Framework

In the case of the ORT framework, the analysis polyphase matrix $\boldsymbol{E}(z)$ is decomposed into five types of factors. These factors are referred to as type A, B, C, D, and E, and have the respective forms

$$\boldsymbol{\mathcal{A}}(s, H(z)) \triangleq \begin{bmatrix} H(z) & 1 - H(z) \\ s & -s \end{bmatrix},$$

$$\boldsymbol{\mathcal{B}}(k, H(z)) \triangleq \begin{bmatrix} 1 & H(z) \\ 1 & H(z) + (-1)^k \end{bmatrix},$$

$$\mathcal{C}(s, H(z)) \triangleq \begin{bmatrix} 1 & H(z) \\ 0 & s \end{bmatrix},$$

$$\mathcal{D}(k, l) \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^k \begin{bmatrix} z^l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^k, \quad \text{and}$$

$$\mathcal{E} \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

where $k \in \{0, 1\}$, $l \in \mathbb{Z}$, $s \in \mathbb{Z}$ and $s \neq 0$. These types of factors and their inverses are associated with the reversible filtering networks shown in Figure 1. In the decomposition of the analysis polyphase matrix $\boldsymbol{E}(z)$, the five types of factors may appear in any order. That is, the $\boldsymbol{E}_i(z)$'s are chosen such that $\boldsymbol{E}_i(z) \in \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}\}$ for $i = 0, 1, \ldots, N-1$.

## V. Lifting Framework

With the lifting framework, the analysis polyphase matrix $\boldsymbol{E}(z)$ is decomposed using two types of factors. These factors are referred to as type L and S, and have the respective forms

$$\mathcal{L}(k, H(z)) \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^k \begin{bmatrix} 1 & H(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^k, \quad \text{and}$$

$$\mathcal{S}(k, \alpha) \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^k \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^k,$$

where $k \in \{0, 1\}$, $\alpha \in \mathbb{Z}$ and $\alpha \neq 0$. These types of matrices and their inverses are associated with the reversible filtering networks shown in Figure 2. In this figure, the symbol $Q$ is used to denote an arbitrary quantization operator. Although the lifting framework allows $Q$ to be chosen arbitrarily, for the purposes of the discussion which follows, we specifically choose $Q$ as $Q(x) = \lfloor x \rfloor$ (i.e., the floor function). With lifting, the decomposition of $\boldsymbol{E}(z)$ is constrained such that L-type operations are always applied first, followed by S-type operations. That is, the $\boldsymbol{E}_i(z)$'s are chosen such that

$$\boldsymbol{E}_i(z) \in \begin{cases} \mathcal{L} & \text{for } i = 0, 1, \ldots, \lambda - 1 \\ \mathcal{S} & \text{for } i = \lambda, \lambda + 1, \ldots, N - 1. \end{cases}$$

## VI. Relationship Between the ORT and Lifting Frameworks

As stated earlier, the ORT and lifting frameworks are very closely related. Here, we examine this relationship in detail. To begin, we show that the ORT networks in Figures 1(a), 1(c), 1(e), and 1(i) are equivalent to the lifting networks shown in Figures 3(a), 3(c), 3(e), and 3(g), respectively.

First, let us consider the A-type network shown in Figure 1(a). By observing that $x_i[n] \in \mathbb{Z}$ and using (1), this network can be shown to be equivalent to that in Figure 3(a). In the case of the network in Figure 1(a), the relationship between the inputs and outputs is given by

$$Y_0(z) = \lfloor H(z)X_0(z) + (1 - H(z))X_1(z) \rfloor_z \tag{6a}$$
$$= X_1(z) + \lfloor H(z)(X_0(z) - X_1(z)) \rfloor_z \text{, and}$$
$$Y_1(z) = s(X_0(z) - X_1(z)). \tag{6b}$$

Similarly, for the network in Figure 3(a), the input-output behavior is given by

$$Y_0(z) = X_0(z) + \lfloor (1 - H(z))(X_1(z) - X_0(z)) \rfloor_z \tag{7a}$$
$$= X_0(z)$$
$$+ \lfloor X_1(z) - X_0(z) + H(z)(X_0(z) - X_1(z)) \rfloor_z$$
$$= X_1(z) + \lfloor H(z)(X_0(z) - X_1(z)) \rfloor_z \text{, and}$$
$$Y_1(z) = -s(X_1(z) - X_0(z)) \tag{7b}$$
$$= s(X_0(z) - X_1(z)).$$

By comparing (6) and (7), it is clear that both networks have the same input-output characteristics. In other words, we have shown that the structures in Figures 1(a) and 3(a) are equivalent. Thus, an A-type network can be realized using only L- and S-type networks.

Next, we consider the B-type network shown in Figure 1(c). Using the same approach as above, this structure can be shown to be equivalent to that in Figure 3(c). In the case of the network in Figure 1(c), the relationship between the inputs and outputs is given by

$$Y_0(z) = \lfloor X_0(z) + H(z)X_1(z) \rfloor_z \tag{8a}$$
$$= X_0(z) + \lfloor H(z)X_1(z) \rfloor_z \text{, and}$$
$$Y_1(z) = \lfloor X_0(z) + (H(z) \pm 1)X_1(z) \rfloor_z \tag{8b}$$
$$= \lfloor X_0(z) \pm X_1(z) + H(z)X_1(z) \rfloor_z$$
$$= X_0(z) \pm X_1(z) + \lfloor H(z)X_1(z) \rfloor_z .$$

Similarly, for the network in Figure 3(c), the input-output behavior is characterized by

$$Y_0(z) = X_0(z) + \lfloor H(z)X_1(z) \rfloor_z \text{, and} \tag{9a}$$
$$Y_1(z) = \pm(X_1(z) \pm Y_0(z)) \tag{9b}$$
$$= \pm(X_1(z) \pm (X_0(z) + \lfloor H(z)X_1(z) \rfloor_z))$$
$$= \pm X_1(z) + X_0(z) + \lfloor H(z)X_1(z) \rfloor_z$$
$$= X_0(z) \pm X_1(z) + \lfloor H(z)X_1(z) \rfloor_z .$$

Comparing (8) and (9), we see that both networks have the same input-output characteristics. In other words, the filtering structures in Figures 1(c) and 3(c) are equivalent. Thus, a B-type network can also be realized using only L- and S-type networks.

Next, let us consider the C-type network shown in Figure 1(e). This is equivalent to the network shown in Figure 3(e). To see this, we simply invoke identity (1). Hence, a C-type network can be realized using only L- and S-type networks.

Lastly, consider the E-type network shown in Figure 1(i). This linear system has the transfer matrix

$$\boldsymbol{T} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{10}$$

The network shown in Figure 3(g) has the transfer matrix

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{11}$$

Comparing (10) and (11), we see that both networks have the same input-output behavior. In other words, the structures in Figures 1(i) and 3(g) are equivalent. Thus, an E-type network can be realized using only L- and S-type networks.

Using techniques similar to those employed above, the networks in Figures 1(b), 1(d), 1(f), and 1(j) can also be shown to be equivalent to those in Figures 3(b), 3(d), 3(f), and 3(h), respectively. The only additional tools needed are identities (2) and (3). For example, consider the networks of Figures 1(b) and 3(b). We proceed by using identities (1), (2), and (3), and the fact that $s^{-1}y_1[n] \in \mathbb{Z}$. For the network in Figure 1(b), the relationship between the inputs and outputs is given by

$$\begin{aligned} X_0(z) &= \left\lceil Y_0(z) + s^{-1}((1 - H(z))Y_1(z) \right\rceil_z & \text{(12a)} \\ &= Y_0(z) + \left\lceil s^{-1}Y_1(z) - s^{-1}H(z)Y_1(z) \right\rceil_z \\ &= Y_0(z) + s^{-1}Y_1(z) + \left\lceil -s^{-1}H(z)Y_1(z) \right\rceil_z \\ &= Y_0(z) + s^{-1}Y_1(z) - \left\lfloor s^{-1}H(z)Y_1(z) \right\rfloor_z, \text{ and} \\ X_1(z) &= \left\lceil Y_0(z) - s^{-1}H(z)Y_1(z) \right\rceil_z & \text{(12b)} \\ &= Y_0(z) + \left\lceil -s^{-1}H(z)Y_1(z) \right\rceil_z \\ &= Y_0(z) - \left\lfloor s^{-1}H(z)Y_1(z) \right\rfloor_z. \end{aligned}$$

In the case of the network in Figure 3(b), the inputs and outputs are related by

$$\begin{aligned} X_0(z) &= Y_0(z) - \left\lfloor -s^{-1}(1 - H(z))Y_1(z) \right\rfloor_z & \text{(13a)} \\ &= Y_0(z) - \left\lfloor -s^{-1}Y_1(z) + s^{-1}H(z)Y_1(z) \right\rfloor_z \\ &= Y_0(z) + s^{-1}Y_1(z) - \left\lfloor s^{-1}H(z)Y_1(z) \right\rfloor_z, \text{ and} \\ X_1(z) &= X_0(z) - s^{-1}Y_1(z) & \text{(13b)} \\ &= Y_0(z) - \left\lfloor s^{-1}H(z)Y_1(z) \right\rfloor_z. \end{aligned}$$

Comparing (12) and (13), we see that the two networks have the same input-output behavior. In other words, the networks in Figures 1(b) and 3(b) are equivalent. This result should not be surprising, however. In fact, it must be so. If we have two invertible transformations with identical forward transforms, the inverse transforms must also be identical. This argument can be used to show that the remaining inverse networks are also equivalent.

To summarize our results so far, we have shown that 1) the A-, B-, C-, and E-type networks of the ORT framework can be realized using only the L- and S-type networks of lifting, and 2) a similar relationship also holds for the inverses of these networks. Using these facts, we now compare the ORT and lifting frameworks more closely. Due to the relationship between the analysis and synthesis polyphase filtering networks given by (5), our examination need only consider one side of the filter bank. In our following treatment, we choose to consider the analysis side.

In the ORT framework, the analysis polyphase filtering is performed by a cascade of A-, B-, C-, D-, and E-type networks in any number and any order. Since the A-, B-, C-, and E-type networks can be realized using only L- and S-type networks, it follows that such an analysis polyphase filtering network can be replaced by an equivalent one consisting of only L-, S-, and D-type operations in any number and any order. This new filtering structure, however, is exactly that employed by the lifting framework with two simple extensions: 1) the L- and S-type operations are allowed to be intermixed, and 2) the use of D-type operations is also permitted. One might wonder what additional degrees of freedom are offered by these extensions. As we shall demonstrate, these extensions are, in fact, of very little practical value.

Initially, let us consider the second extension above. Suppose we allow D-type operations to be used. One can easily show that, given a network where D-type operations are intermixed with other operations, it is always possible to find an equivalent network with the D-type operations applied last. Using (4) and straightforward manipulation, one can easily confirm that the identities shown in Figure 4 hold. By repeated application of these identities, one can always obtain a new network with the D-type operations applied last. From this, we can see that there is little flexibility afforded by the second extension listed above. This extension only allows the introduction of an additional shift in the subband signals. Clearly, this does not change the underlying transform in any fundamental way.

Let us now consider the first extension above. For lossless signal coding applications, we would like a transform that yields coefficients having small magnitudes. Such coefficients are desirable as they can be more efficiently encoded. This being the case, it is undesirable to employ S-type operations having scaling factors other than $\pm 1$. The use of such operations would unnecessarily increase the dynamic range of the subband signals and, hence, the resulting transform coefficients as well. Suppose now that we restrict the scaling factors to be $\pm 1$. Since scaling by one has no effect, we need only consider the case of scaling by $-1$. In this scenario, one can easily show that given a network of L- and S-type operations where the two types of operations are intermixed, it is always possible to find an equivalent network with the S-type operations applied last. Using (3) and straightforward manipulation, we can show that the identities in Figure 5 hold. By repeated application of these identities, one can always obtain a new network with the S-type operations applied last. Although the L-type operations in the resulting network may employ either the floor or ceiling function for quantization, the network is still a lifting network. As noted previously, the lifting framework allows any quantization operation to be used in the L-type networks. As we can see from the above analysis, allowing a scaling factor of $-1$ only allows one to change the sign of subband signals. Thus, there is little to be gained by employing S-type operations in the first place.

As we have shown above, the additional degrees of freedom offered by the two lifting framework extensions are of little practical value. Thus, we assert that there are no practically useful transforms that can be generated with the ORT framework that cannot be generated with the lifting framework. As a final note, it is important to emphasize the generality of lifting. For the purposes of showing the relationship between the ORT and lifting frameworks, it was necessary to use the floor and ceiling functions for the quantization operation in the lifting case. In general, however, the lifting framework allows the use of an arbitrary quantization function (e.g., floor, ceiling, truncation, rounding to the nearest integer, etc.). In fact, with the appropriate choice of rounding function, the lifting framework can even be used to generate reversible real-to-real mappings [3]. This, however, is not generally possible with the ORT framework due to its use of the A- and B-type networks. The lifting framework has another important advantage over the one based on the ORT. Lifting always facilitates in-place calculation of a transform. This, however, is not generally true for the ORT framework. This is, again, due to its use of the A- and B-type networks. Therefore, lifting is arguably a more powerful and practical framework than that based on the ORT.

## VII. Conclusions

The ORT framework for reversible transforms has been shown to be equivalent to a special case of the lifting framework with only trivial extensions. Moreover, we have demonstrated that any practically useful reversible integer-to-integer mapping that can be realized using the ORT framework can also be realized using the lifting one. This development is significant in that it demonstrates the generality of the lifting framework for the construction of reversible transforms.

## Acknowledgment

The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

## References

[1] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.

[2] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proceedings of SPIE: Wavelet Applications in Signal and Image Processing III*, San Diego, CA, USA, Sept. 1995, vol. 2569, pp. 68–79.

[3] M. D. Adams and F. Kossentini, "Performance evaluation of several reversible integer-to-integer wavelet transforms in the JPEG-2000 baseline system (VM release 0.0)," ISO/IEC JTC 1/SC 29/WG 1 N 866, June 1998, Available from `http://www.ece.ubc.ca/~mdadams`.

[4] M. D. Adams and F. Kossentini, "Evaluation of reversible integer-to-integer wavelet transforms for image compression," in *Proc. of IEEE International Conference on Image Processing*, Oct. 1999, To appear.

[5] H.-Y. Jung and R. Prost, "Lossless subband coding system based on rounding transform," *IEEE Trans. on Signal Processing*, vol. 46, no. 9, pp. 2535–2540, Sept. 1998.

## List of Figures

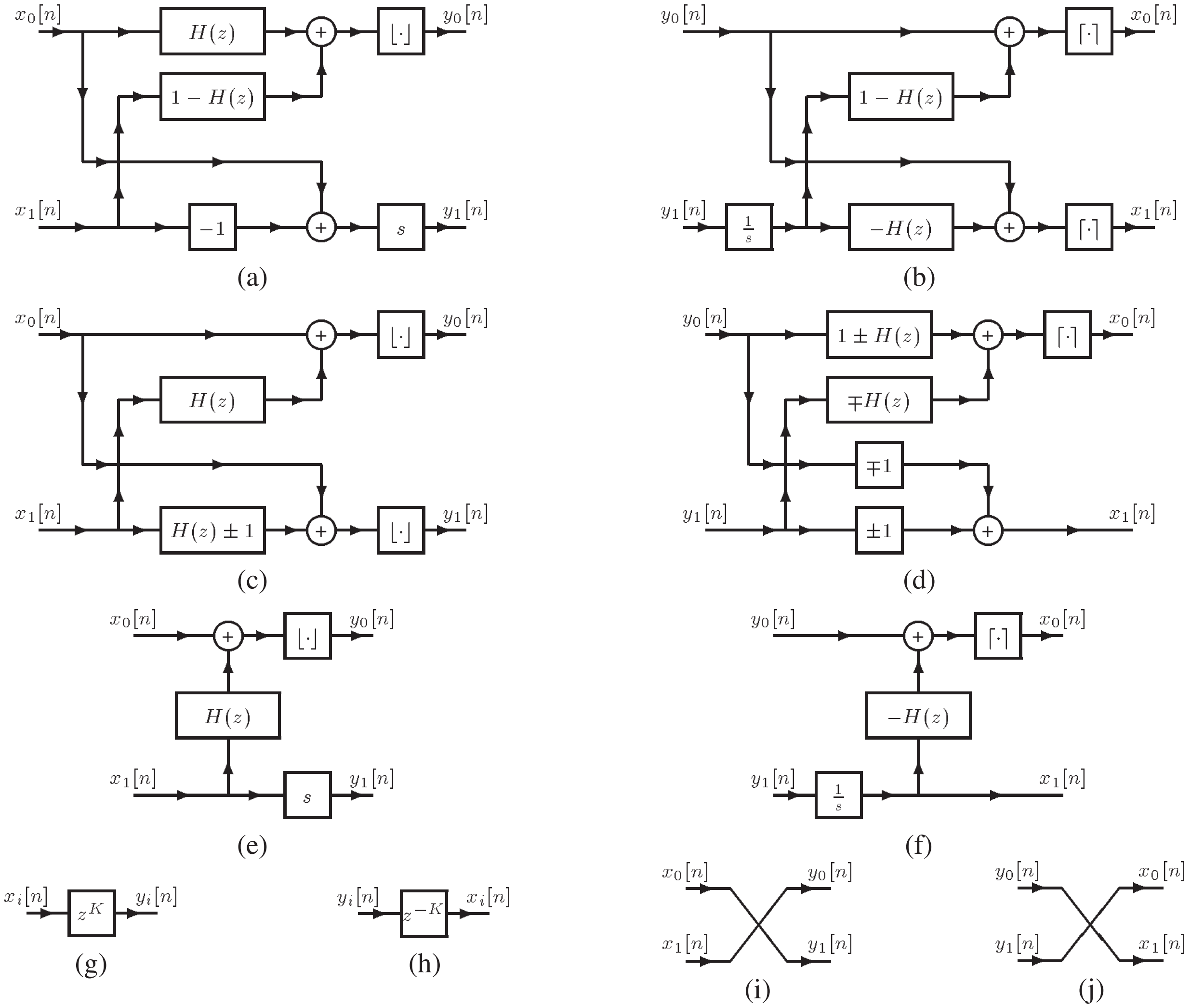


Fig. 1. Networks used for polyphase filtering in the ORT framework. Networks for (a) A-type, (b) $A^{-1}$-type, (c) B-type, (d) $B^{-1}$-type, (e) C-type, (f) $C^{-1}$-type, (g) D-type, (h) $D^{-1}$-type, (i) E-type, and (j) $E^{-1}$-type operations.

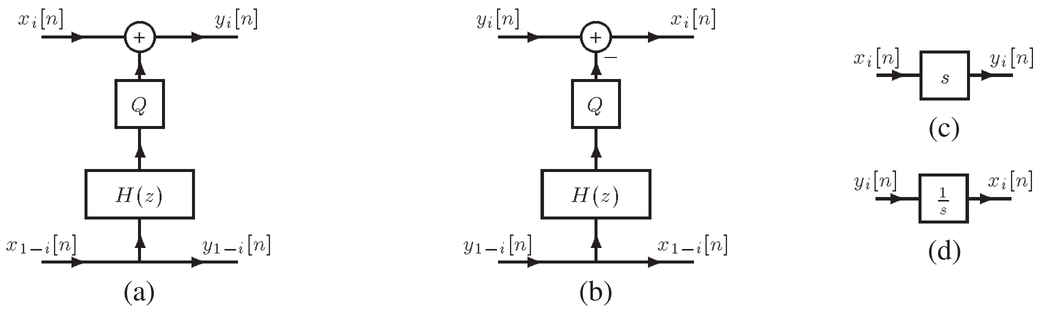


Fig. 2. Networks used for polyphase filtering in the lifting framework. Networks for (a) L-type, (b) $L^{-1}$-type, (c) S-type, and (d) $S^{-1}$-type operations.

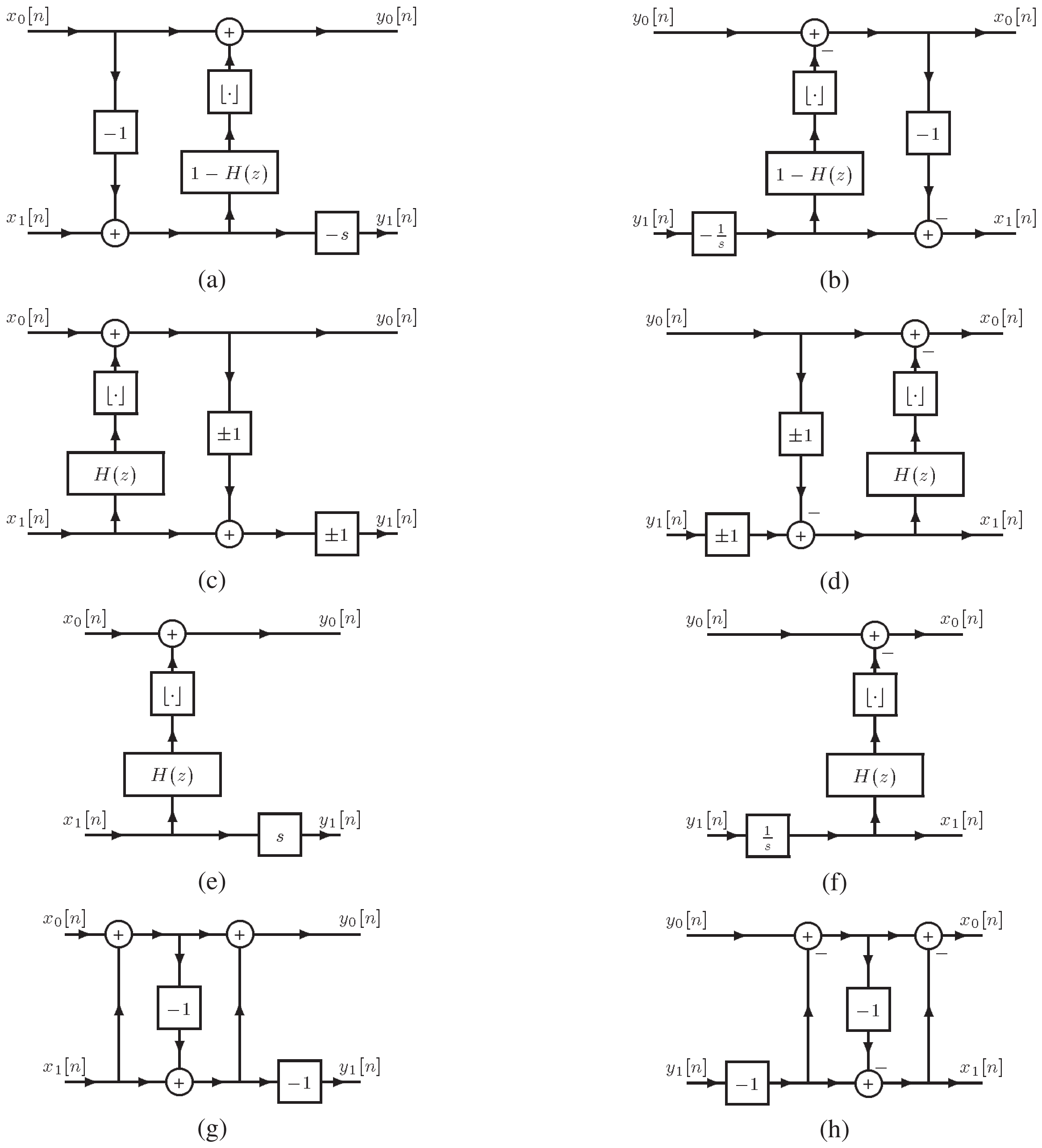


Fig. 3. Equivalent lifting networks for (a) A-type, (b) $A^{-1}$-type, (c) B-type, (d) $B^{-1}$-type, (e) C-type, (f) $C^{-1}$-type, (g) E-type, and (h) $E^{-1}$-type operations.

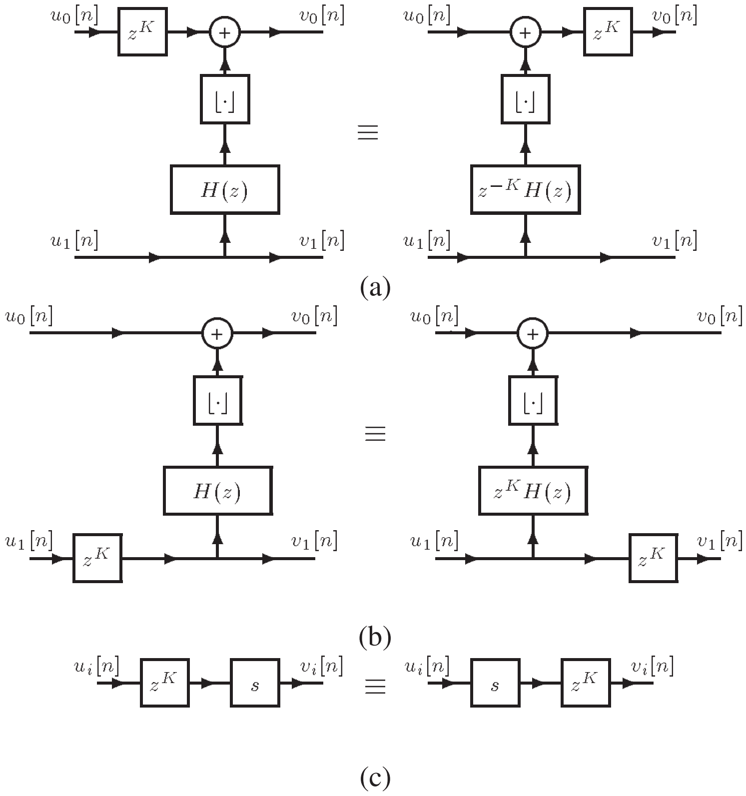


Fig. 4. Identities for interchanging (a) D-type and L-type operations (first case), (b) D-type and L-type operations (second case), and (c) D-type and S-type operations.

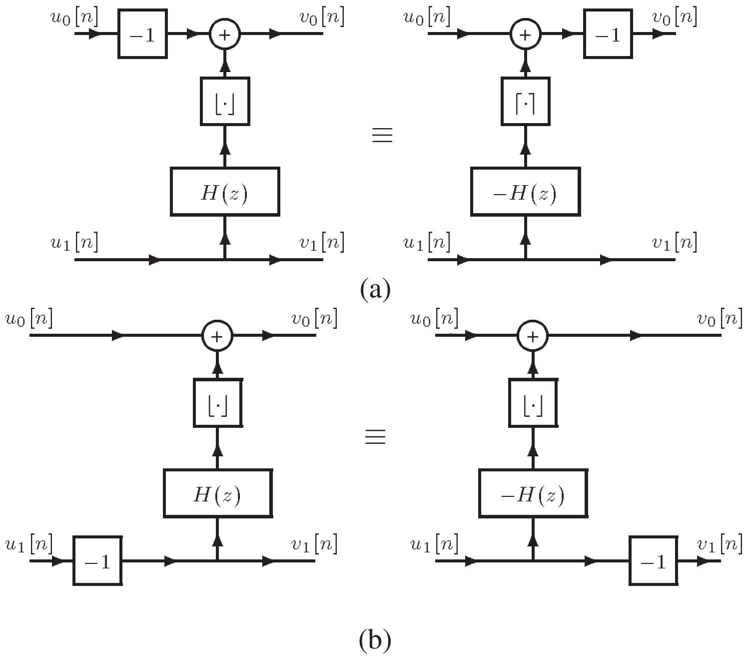


Fig. 5. Identities for interchanging L-type and S-type operations (in the scaling by $-1$ case).